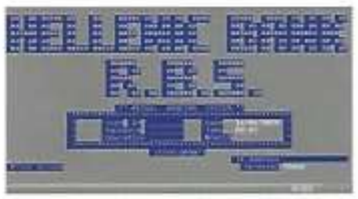
Hellenic Bank Web-Enables Its Legacy NonStop Applications

Chris (Smith) Anderson >> President/CEO >> Crystal Point

Hellenic Bank, headquartered in the Republic of Cyprus, has been a long-time HP NonStop user. Its NonStop banking applications date back to the green-screen days and in the past required complex interactions between multiple screens in order for a teller or other employee to perform a single task. The bank wanted to provide its staff with new banking applications that exhibited a modern look and feel, that were intuitive and easy to use, and that could permit the incorporation of additional banking functionality.

Hellenic accomplished its goal by turning to AppViewXS from Crystal Point. Not only did the bank succeed in replacing its green screens with modern GUI interfaces, but it also web-enabled its legacy applications to support many additional functions. Today, Hellenic Bank's legacy NonStop banking applications can be accessed securely and reliably anywhere and anytime via a standard web browser.

A striking example of web-enabling these applications can be seen in the logon screen displayed in Figure 1:


Before

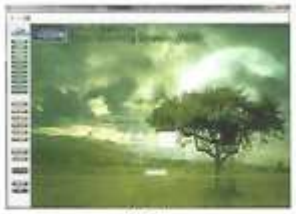

After

Figure 1: Logon Screen

Hellenic Bank

Founded in 1976, Hellenic Bank Public Company Ltd. is the second largest bank in Cyprus. With over 1,400 employees and assets of almost nine billion euros, the bank not only weathered the recession of 2008 but also successfully passed the European Union Bank Stress Test. Satisfying the stress test is indicative of the bank's healthy financial position and reflects its ability to maintain strong capital adequacy even in the worst-case scenarios.

The bank offers a myriad of services to its customers. Its products include consumer and corporate banking, credit-card services, ATMs, mortgage loans, factoring, insurance, wealth management, portfolio management, investment banking, and brokerage services. It is listed on the Cyprus Stock Exchange.

The Bank's Legacy Applications

Hellenic Bank operates two major applications on its NonStop systems – the Financial Banking System (FBS) and the Retail Banking System (RBS). Twelve different banking services are supported by these applications, including personal banking, business services, corporate banking, Internet banking, card/ATM services, and insurance services.

In the past, tellers and other employees of the bank accessed these services via Tandem 6530 terminals – dumb legacy terminals with "green screens." On the average, 800 to 900 staff members were logged on at any one time, with a peak load of 1,200 connections. A staff member often had to access multiple screens and combine the data from these screens to complete a task. The manual processing of requests could be confusing and error-prone.

Thomas Stylianou, Hellenic's head of Information Technologies, recognized that in order for the bank to maintain its competitive standing, change was needed in the area of customer service and via the incorporation of emerging technologies into Hellenic's banking applications. Its legacy banking applications suffered from several challenges:

- The green-screen interface was restrictive, confusing to learn, difficult to use, and error-prone. This affected the level of service offered to the bank's customers.
- Several new technologies, including signature verification systems and check-reading machines, could enhance the bank's processing of customer activities. However, incorporating these technologies into the legacy banking applications was prohibitive both in terms of cost and time.

Under Mr. Stylianou's leadership, the bank's IT Steering Committee embarked on a project to modernize Hellenic's legacy applications.

Web-Enabling Its Legacy Applications

The Search for a Solution

The Steering Committee examined several approaches to modernization:

- *Third-party banking applications* were considered. However, this approach was rejected because Hellenic Bank preferred the specific functionality of its own applications. It was important to the bank to retain control over functionality, ongoing maintenance, and enhancements.
- *A complete rewrite* would be too costly, would take too long, and was too risky.
- *Doing nothing* would not achieve the goals of the bank.
- *Opening the banking applications to the Internet* via web browsers and a web server appeared to be a viable solution. A web server could use the 6530 terminal interface of the NonStop server to access application services. The web server could provide browser-based screens that incorporated data from several green screens into common, intuitive GUI screens for the client PCs. The web server could integrate other systems into the information flow to the browsers.

The IT Steering Committee evaluated several products designed to web-enable NonStop legacy applications. After a thorough evaluation, the Committee selected AppViewXS from Crystal Point. AppViewXS is a web-server application that communicates between NonStop legacy services and browser-based client systems. The AppViewXS selection was based on several factors:

- Hellenic Bank had been a customer of Crystal Point for many years and had used its OutsideView product to deliver 6530 green screens to users.
- Prototyping indicated that AppViewXS could web-enable the NonStop applications with little if any changes to the NonStop legacy code, an extremely important consideration. No change in the core functionality of the banking applications would be required.
- AppViewXS allowed rapid and incremental deployment of new GUI screens, with no need for a "big bang" conversion. The web enablement could be accomplished in a controlled fashion, screen-by-screen, over any period of time.
- AppViewXS is a Java application. Therefore, new Java-based banking tools could be integrated easily into the data stream.
- AppViewXS requires very little programming effort. It is configurable via drag-and-drop controls and is extensible via a built-in scripting language.

The Web-Enabled System

The use of AppViewXS to web-enable Hellenic Bank's NonStop legacy applications is reflected in Figure 2. The bank chose an Apache/Tomcat web server to host AppViewXS. Two such servers are provided for redundancy.

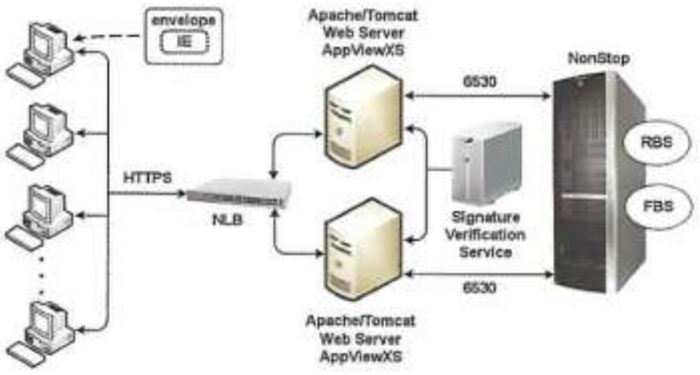


Figure 2: The Web-Enabled NonStop Legacy Banking Applications

AppViewXS communicates with the NonStop server as if the server were a collection of 6530 terminals. Via the 6530 protocol, AppViewXS can access and update any of the services provided by the RBS and FBS banking applications.

Via its configured services, AppViewXS responds to a user request by gathering the required information from one or more 6530 screens, creating the desired GUI screen, and sending the resulting GUI screen to the requesting browser. Likewise, GUI screens returned from the browser are parsed; and the data contained in the screens is used to populate appropriate 6530 response screens that are then sent to the banking applications.

Traffic flowing between the client PC browsers and the AppViewXS web servers uses the HTTPS protocol to ensure maximum security. Messages are compressed to optimize response times. During the logon process, the requesting browser's IP address is captured and placed in the logon screen. This allows the legacy banking applications to control access to the system and to determine the location and department of the user for routing of unsolicited messages requesting user action.

Browser traffic is directed to a Network Load Balancer (NLB), which distributes requests to the two AppViewXS web servers on a round-robin basis. Both web servers are configured to handle all of the browser traffic. Should one of the web servers fail, all requests are simply routed by the NLB to the surviving server. There is no single point of failure.

Since AppViewXS runs on its own web server, there is no additional load on the NonStop system. The only activity of the NonStop applications is to process 6530 screens just as they did before web-enabling.

The Signature Verification Service shown in Figure 2 is described later.

Modifying the Browser Look-and-Feel

The bank wanted to hide the controls of the Internet Explorer browser that it was using in the client PCs to avoid confusion and errors on the part of the users. It therefore developed a C++ application to host the browser. The application acts as an envelope around the browser and removes all unnecessary IE information and buttons. Users see in their GUI screens only the data fields and controls that are important to them.

Unsolicited Messages

A key challenge in the bank's use of web-enabling was that it had to send host-initiated unsolicited messages to users to direct them to take immediate actions such as authorizing a transaction. These messages were to appear in a special window at the top of the GUI screens. Unsolicited messages were not an issue with the legacy 6530 terminals, but such messages generally are not a capability of the Web. Pop-up windows are the closest thing, but they would hide active areas of the screen and would interfere with user activity.

In cooperation with the bank, Crystal Point integrated a custom program into AppViewXS to accept unsolicited messages from the NonStop applications and to insert them into the messaging windows of the GUI screens being returned to the user. A history function also was implemented to allow users to see previously received messages.

Integrating Other Banking Tools

Many new and useful banking tools are now appearing on the market. The bank wanted to take advantage of these tools

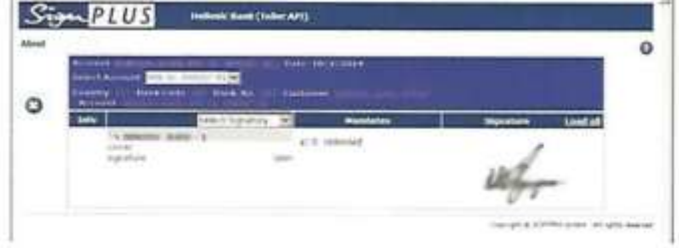
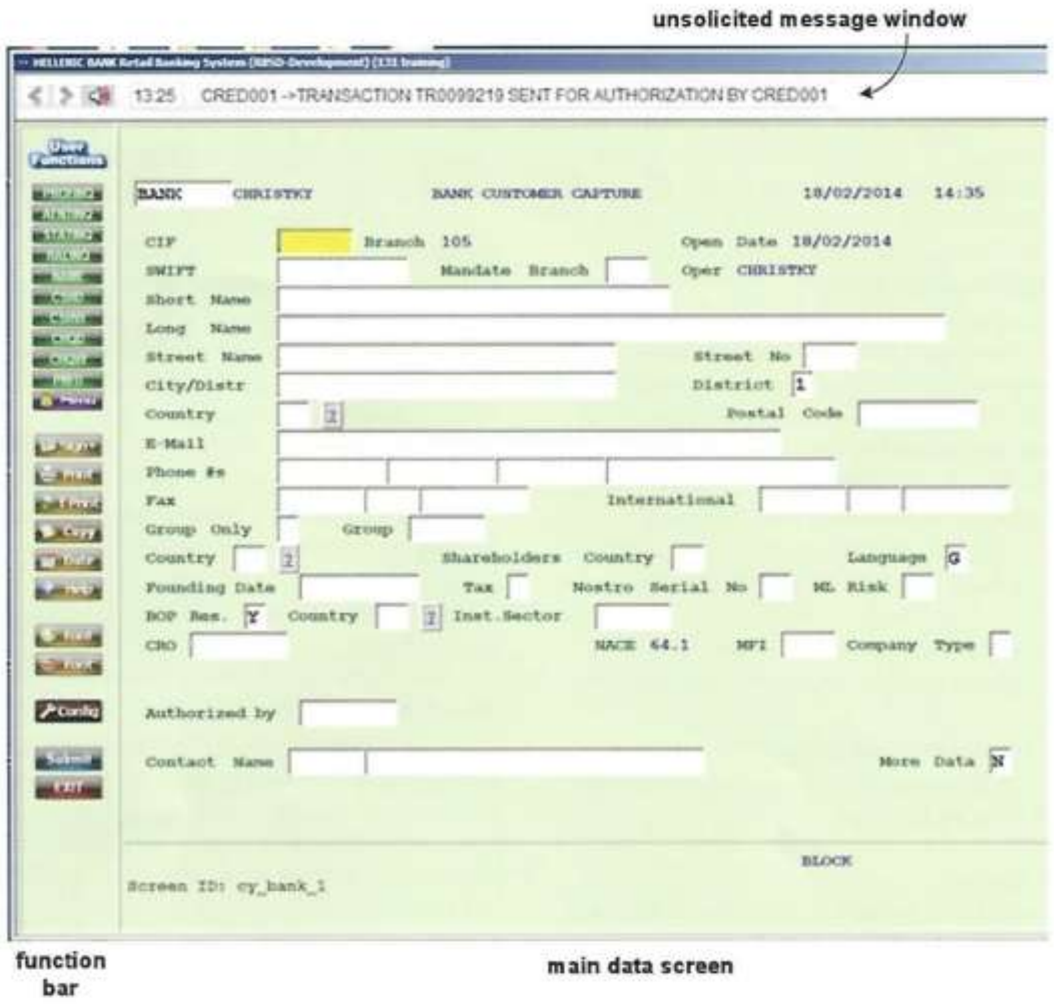


Figure 3: Signature Verification Pop-up



```
13:25   CRED001 ->TRANSACTION TR0099219 SENT FOR AUTHORIZATION BY CRED001

BANK    CHRISTKY            BANK CUSTOMER CAPTURE            18/02/2014   14:35

CIF                Branch  105                     Open Date 18/02/2014
SWIFT                 Mandate Branch               Oper  CHRISTKY
Short  Name
Long   Name
Street  Name                                        Street  No
City/Distr                                          District  1
Country        ?                                      Postal  Code
E-Mail
Phone #s
Fax                                 International
Group  Only       Group
Country      ?          Shareholders  Country                  Language  G
Founding Date           Tax     Nostro  Serial  No      ML  Risk
BOP  Res.  Y   Country     ?  Inst.Sector
CRO                                  NACE  64.1    MFI      Company  Type

Authorized by

Contact  Name                                               More  Data  N

                                                     BLOCK
Screen ID: cy_bank_1
```

Figure 4: An Example of a Hellenic Bank Web-Enabled Data Screen

by integrating them into its banking applications. However, to modify its existing legacy applications to use these tools would require major coding changes. Such changes would be risky, expensive, and time-consuming.

AppViewXS provided a convenient solution to this conundrum. The systems hosting the new technologies can communicate with AppViewXS directly in the web browser. AppViewXS then can be configured to integrate these services into the screens sent by AppViewXS to the client systems.

The first such tool to be integrated was a Signature Verification System, shown in Figure 2. When a user at a browser requests a customer's signature, the signature appears in a pop-up window, displayed here in Figure 3.

Via the same AppViewXS-assisted process, the bank is now considering adding check-reading machines to its banking applications.

The New User Interface

An example of the new user interface is shown in Figure 4. The interface comprises a main data screen and a function bar to the left of the data screen. At the top of the data screen is the window for unsolicited messages. Note the absence of Internet Explorer controls.

The function bar provides the following:

- User functions – The user can select ten favorite Banking System functions for easy access.
- Sign + sends the customer identification to the Signature Verification System, which returns the customer's signature.
- Print generates a document containing the contents of the screen and sends it to a printer connected to the user's computer. This can be used to create official documents such as Letters of Guarantee. The Print function takes advantage of AppViewXS' capability to pass data to a client browser without having to display the data to the user.
- T-Print generates a print-screen function of the current screen and directs it via the NonStop spooler to a designated printer on the Banking System network.
- Copy generates a document containing the screen contents and places the document in the Windows Clipboard for use by other applications such as Microsoft Word.
- Date pops up a calendar from which the user can select a date to be entered into a date field on the screen.
- Help provides a relevant online Help screen for the specific Banking System function being displayed.
- +Font/-Font changes the font size among five choices.
- Config allows the user to select background colors.
- Submit sends the information on the screen to the NonStop host.
- Exit terminates the application.

The screen background color can be changed to draw the attention of the user. Field colors can be changed to make them easier to read.

Right-clicking on the screen (left-clicking for left-handers) provides a windows menu that is context-aware of the application screen.

Benefits Achieved by Web-Enabling

The bank realized several benefits by web-enabling its legacy banking applications:

- It reduced its training costs for new employees. The banking application interfaces are now much more intuitive.
- It reduced its support costs. Many enhancements now can be made in the Java-based AppViewXS application rather than having to modify legacy code in the NonStop system.
- It improved employee perception and satisfaction and increased customer service by providing intuitive, easy-to-use banking applications.
- It positioned itself to integrate future technology such as check-reading machines into the banking applications.

Constantinos Papadamou, Manager of Network, Systems & Technical Support, stated:

"The key advantages of this approach are a friendly, graphical interface for our employees. They have been asking for mouse capability, drop-down lists and the like in our core applications. Now we can make them happier and require less training while experiencing fewer errors and fewer help-desk calls."

AppViewXS

AppViewXS is a Java application that web-enables a legacy system with little if any changes to the legacy code. It employs the legacy terminal interfaces of the system (Tandem 6530, IBM 3270, IBM 5250) to access the system applications, and it converts the interfaces to powerful and flexible browser interfaces for use by client PCs. AppViewXS' conversion between legacy and GUI formats is aided by a flexible drag-and-drop configuration utility and a powerful scripting language.

AppViewXS provides security over the Internet through the use of the HTTPS protocol. It can also encrypt traffic between the web server and the host via SSL or SSH2. AppViewXS can run on a variety of web-server configurations. Supported server platforms include Windows, Solaris/Unix, and Red Hat/Linux systems. AppViewXS can run on NonStop iTP, Apache, and Microsoft IIS web servers under Tomcat or IBM WebSphere. Compatible client browsers include Internet Explorer 5.0 or higher and Netscape Navigator 4.75 or higher.

Crystal Point (www.crystalpoint.com), formed in 1986, is headquartered in Bothell, Washington, U.S.A. The company offers a full range of host-connectivity solutions, from host-access products performing terminal emulation to solutions for rejuvenating, repurposing, and integrating multiple legacy applications.

Summary

Crystal Point's AppViewXS solution allowed Hellenic Bank to web-enable its NonStop server-based FBS and RBS legacy banking applications. This modernized the bank's years-old legacy applications, with measurable improvements in functionality, accessibility, and security. The modernization effort was accomplished with no significant changes to the underlying legacy applications.

The result has been: (1) an enhanced experience for the bank staff via the new GUI screens, (2) the ability to incorporate new banking technologies, and (3) reduced support and training costs. The applications are easier to learn and trigger fewer technical issues. AppViewXS replaces the complex and confusing legacy green-screens with a browser-based modern, efficient, and intuitive GUI interface for the system's users.

Founded in 1986 by Fred Stephens and Chris Smith Anderson, Crystal Point's first dos-based product emulated over 25 terminals. In 1990 Tandem contracted Crystal Point to develop the first GUI-based Tandem emulator for OS/2 and Windows, OutsideView. Our software solutions have now expanded by developing AppViewXS, our legacy modernization software tool.

Reason #4 to Connect:

Access to a global community of over 70k HP users

CONNECT

Your Independent HP Business Technology Community