

Abstract: John Douglas, Host Integration Product Line Manager, discusses Rational Host Access Transformation Services (HATS), what it is, and how it can help you modernize your enterprise. This short session will address both the products green-screen transformation capability and its ability to generate Web Services from mainframe applications. Key features and functions of the latest release, how customers use the product enhancements, and the future of HATS will also be discussed.

Title: Build Modern Interfaces and Services with Rational Host Access Transformation Services (HATS)

Speaker: John Douglas

What is Rational Host Access Transformation Services (HATS)?

HATS is a tool that helps customers to transform their 3270 applications from a “green-screen” to a GUI. With the tool, a developer creates an application through which end users interact with their mainframe application with familiar user interfaces, like radio buttons, drop down menus and hyperlinks...all without altering the original code. The developer can deploy the HATS application with “default” setting with hours of creating a project, and then continue to work iteratively with their end users to further customize and streamline the application. HATS applications can be accessed by Web, Portal, rich client platform, and even mobile device end users, extending valuable business logic and data beyond the terminal emulator.

How does HATS transform 3270 screens?

The HATS technology goes beyond what has traditionally been referred to as “screen scraping”. Out of the box, HATS recognizes information in the datastream generated by the host application (for example, function keys and menus) and transforms it to a more intuitive GUI. HATS also has the capability to combine multiple screens from the same application or from multiple applications into one page. Also, the flow of the original application can be streamlined by using scripts, called macros, to navigate screens on the behalf of the end user. Business logic can be used to integrate other data sources (for example, database information) with the HATS application... And to further simplify the application, data, perhaps from business logic, a macro, the end user, or another terminal application, can be saved in HATS global variables, and used to further simplify the application- for example, to enter data on behalf of the end user or to fill a drop down in the application.

How can HATS help in an SOA environment?

The HATS technology allows for the quick and easy development of Web services from your mainframe applications. The macro capability within HATS is the first step. Once a specific screen flow is recorded as a macro, that macro can be easily converted to, and deployed as a Web service that performs a specific terminal application task- for example, check balance, or get customer information. Any application that consumes the Web service need only provide the input and HATS will be driven through standard Web services interfaces to communicate with the terminal application and deliver the requested output to the Web service consumer.

How are customers using HATS?

HATS applications are being built for a variety of purposes across all industries. Many governments are making their applications “self service” over the Web. Companies with call centers are finding that using HATS not only helps them retain their employees with an easier to use and learn application, but customer service and productivity is markedly improved as they streamline their call center applications. The Web services capability is being used to integrate host applications into other applications, even .NET applications, avoiding the risk a rewriting logic that is already proven and stable.

As an example, a leading European automobile manufacturer sought to enhance the customer experience and streamline the sales process at automotive dealerships by modernizing its Vehicle Management System software. The company needed to move its back-office green screen dealer systems to the showroom as a thin-client, browser-based system that provided common entry points, simplified IT governance, complied with corporate image standards, and retained organizational knowledge.

An IBM Rational Business Partner Oxford International developed a Web-based system solution for global vehicle distribution, inventory, and order management. Using IBM WebSphere Host Access Transformation Services (HATS) and IBM Rational Application Developer, Oxford demonstrated and delivered a comprehensive modernization solution deployed on IBM System i servers running IBM WebSphere Application Server.

Rational Application Developer and HATS enabled Oxford to focus on business needs rather than the underlying technology to deliver a comprehensive solution. The time to achieve deployment readiness was three months, at less than 20% of the cost of building an entirely new application. The new system integrates an elegant interface and substantially improves the customer sales experience with a seamless ordering process.

What are some of the recent enhancements to HATS?

As I mentioned earlier, HATS applications can be accessed from the Web, a portal, a rich client platform, and with the latest release, HATS applications can be accessed from mobile devices. You can now develop HATS Web applications that provide transformations specifically tailored for mobile devices such as cellular phones, data collection terminals, and personal digital assistants (PDAs). This gives your increasingly mobile workforce access to host application data wherever and when ever they need it, which can help you broaden your customer base and improve your level of service.

The Visual Macro Editor, or VME, is a new tool for visually developing HATS macros. The Visual Macro Editor, is a technology preview that combines many of the features of the HATS Macro Editor, Host Terminal, and Advanced Macro Editor into one tool, and provides a graphical representation of the macro, which can help the developer more easily see errors and missing flows. It provides optional, automatic capture of screens as a macro is being recorded in the host terminal. It also allows flows to be copied between macros and provides drag-and-drop support for adding new screens. This tool saves time and helps developers to create more bullet proof macros.

You can now create HATS portlets that comply with the standard Java Portlet Specification (JSR 168) API. HATS JSR 168 portlet support includes integration with property broker so that HATS portlets can send and receive data with other portlets.

The next release of HATS also includes an easier method of updating HATS Web services. If changes are made to HATS Integration Objects, the corresponding Web services support files must also be updated. A new wizard gives you an easy way to determine what Integration Objects are included in an existing HATS Web service, making it easier and less error-prone to update the Web services support files.

What are customers saying about HATS?

Many of our beta customers have been very vocal about the latest release of HATS. One customer stated that, "HATS is a cornerstone of our SOA efforts. It has given us the ability to quickly and inexpensively provide data from our legacy systems to a number of new applications. "

Another said that, "<HATS is> very useful for organizations with Mainframe systems looking for service enabling their legacy assets. "

Customers are very excited about the VME. A few of the statements are, "HATS 7.1 has a great new VME (Visual Macro Editor) and this is a must have feature. " and, "It's quite easy to use from both system i and web developer's point of view. <We appreciate> the rich features of <the> VME. "

And finally, one customer pronounced that, "This is an excellent product that delivers what is promised."

What are some of the future plans for HATS?

HATS is, and will continue to be, a great option for customers that need to quickly deliver green screen applications to the Web as either interactive applications or as services. Because HATS does not require access to or modification of application source code, HATS will continue to be a critical component of the modernization strategy for some customers.

HATS can also be a good first step in a long-term SOA strategy since the service interfaces developed with HATS today can be used long after the green screen application has been replaced. This provides customers with the peace-of-mind to build long term applications on these services and to expose these interfaces to internal organizations or partners.

In order to provide customers with even more freedom in the approach they take to modernization, we're looking for ways to further help customers integrate their HATS applications with the latest IBM technologies, such as EGL. EGL is a modern business-oriented programming language that is easy to learn, allowing developers to write in one language, but compile into multiple languages, such as COBOL. We believe that some customers are interested in evolving to more modern and flexible architectures that may include long term plans to create new applications using EGL, or reengineer parts of their terminal applications using EGL. Today, HATS Web services can be used in an EGL application created with Rational Business Developer, and HATS Web applications can integrate with other Web applications, such as those created with Rational Business Developer. However, we are looking for ways to better help customers to start with HATS, and then evolve to more modern architectures over time when there is business justification to do so. For example, a customer might transform an existing application using HATS, and then decide to apply the 80/20 rule to their terminal application... rewriting the 20% of the application that is used 80% of the time using EGL for flexibility and maintainability, leaving the remaining 80% of the application to HATS to transform. We want to make sure HATS plays well in this scenario.

HATS will also continue to stay current with the latest versions of WebSphere Application Server, WebSphere Portal, and the Rational tooling products, such as Rational Developer for System z. We're also looking to expand our Web 2.0 capabilities by using AJAX to provide a richer and more dynamic user experience.