EM04

Service-Oriented Architecture Enabling Green Screen Applications using IBM Rational HATS

Rod Little
IBM Rational Enterprise Modernisation
rod_little@uk.ibm.com

IBM Rational Software Development Conference 2008

WHERE TEAMS ARE R-HEROES

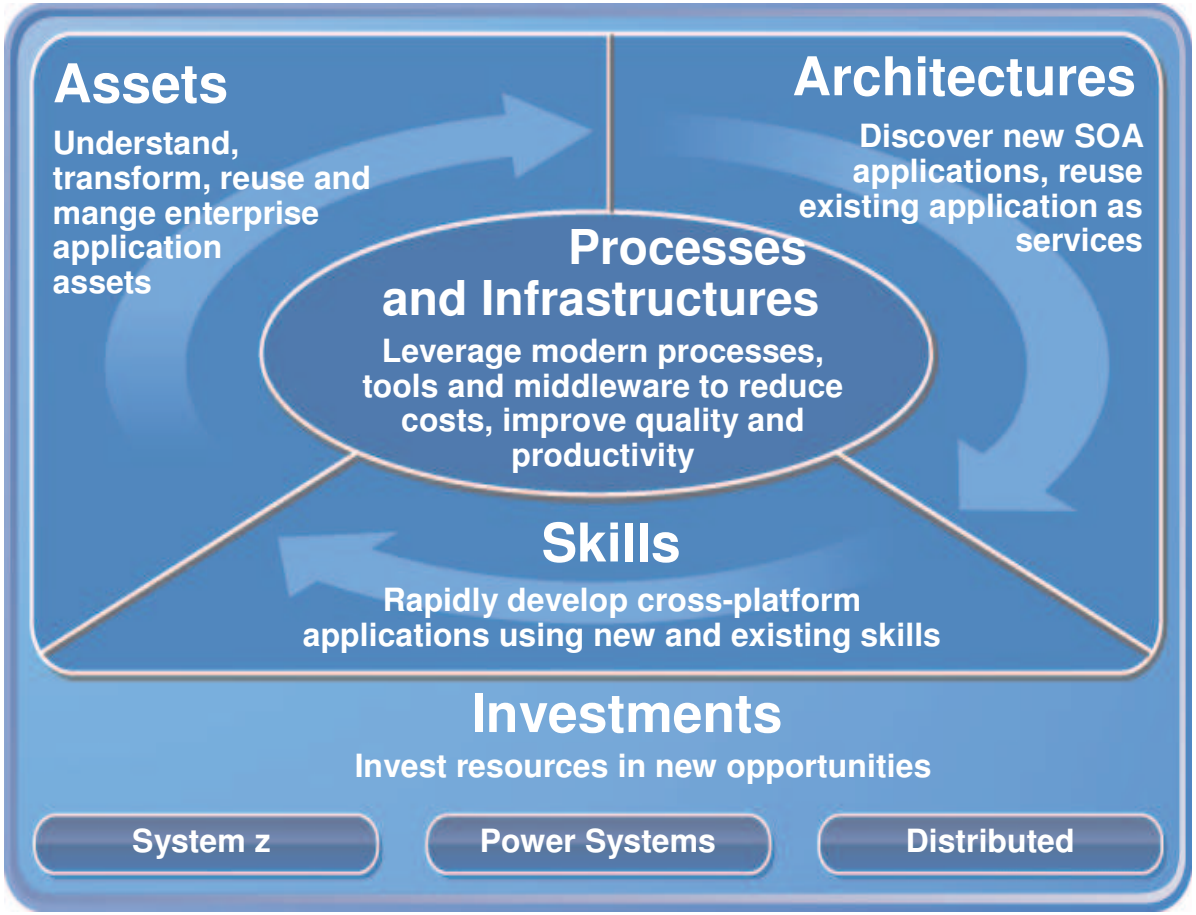
IBM

Rational. software

Agenda

- What is IBM Enterprise Modernisation?
- Overview of IBM Rational Host Access Transformation Services (HATS)
- Introduction to SOA
- Building Web services using HATS

Enterprise Modernisation – Solution Overview

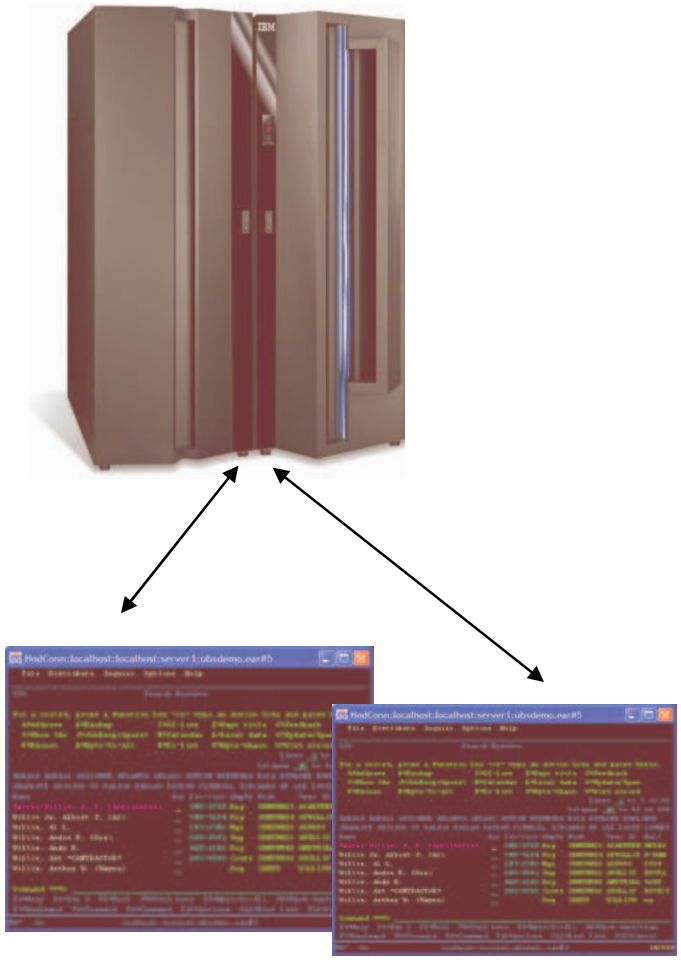


- ✓ *Leverage value in existing assets*
- ✓ *Drive innovation with SOA and web technology advancements*
- ✓ *Leverage existing and new staff on multi-platform projects*
- ✓ *Improve quality and flexibility with a consolidated team infrastructure*
- ✓ *Reduce maintenance costs*

http://www-306.ibm.com/software/info/developer/solutions/em/

Value of Existing Applications

- Existing applications are among the most valuable assets a company owns
  - Fully functional applications that run the business today
  - Significant investments over the years
  - Fine-tuned for dependable optimal performance
  - Run within fine-tuned reliable, robust, scalable IT infrastructure and platforms
- Accessed via character / text based terminals
  - Non-intuitive, difficult to navigate user interfaces
  - Difficult to integrate in modern business processes
- 5X less expensive to reuse existing applications than to write new applications from scratch
- Reusing proven, time-tested applications can result in significantly lower risk and faster time to market

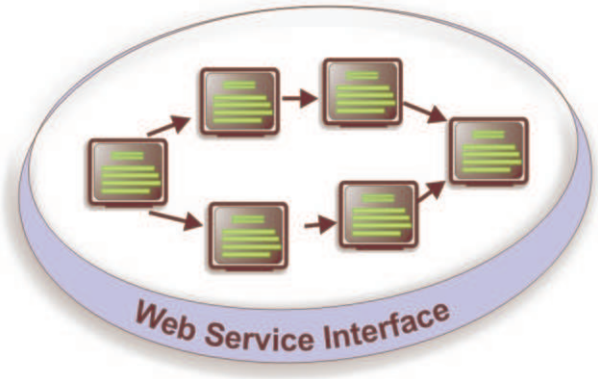


Challenges and Customer Requirements

- Improve the appearance of user interfaces
  - Reduce learning curve for new employees
  - Extend to new departments, new business partners, new end-user customers, who rely on web application ease-of-use
- Improve employee productivity by streamlining application flow
  - Automate where appropriate - eliminate unnecessary keystrokes and screen navigations
  - Customise access for a specific set of users
- Leverage “integration at the glass” / portal technology
  - Access existing applications from a common portal (e.g. Customer Service Portal, or Employee/Human Resources Portal, etc.)
- Reuse existing applications in their new business processes
  - SOA, Web Services based composite applications
  - Require integration with new J2EE, .NET, etc. applications for totally new user interfaces and/or business process automation

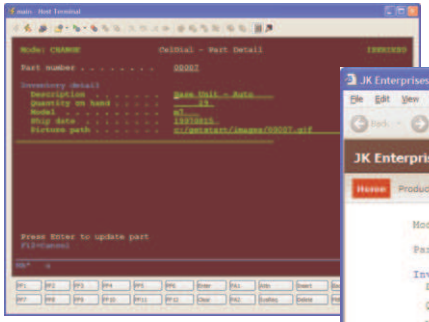
Host Access Transformation Services (HATS)

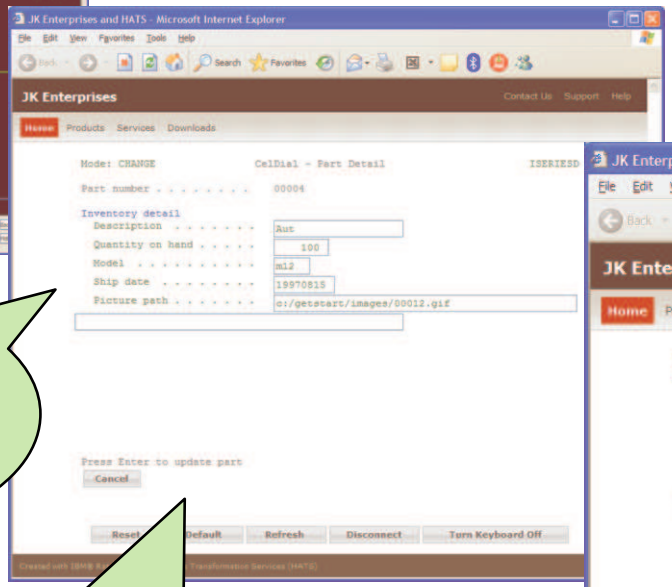
- What can you do with HATS?
  - Quickly and easily create Web, portal or rich client applications that provide an easy-to-use GUI for your green-screen applications
    - Low skills requirement
    - Highly customisable
    - Iterative development process
    - Transformation “on the fly”
  - Extend terminal application tasks as Web services
- Benefits
  - Extend host application to new users
  - Improve the navigation of your host application
  - Reuse your existing assets in a Service Oriented Architecture
  - Avoid rewriting applications (no impact to code)

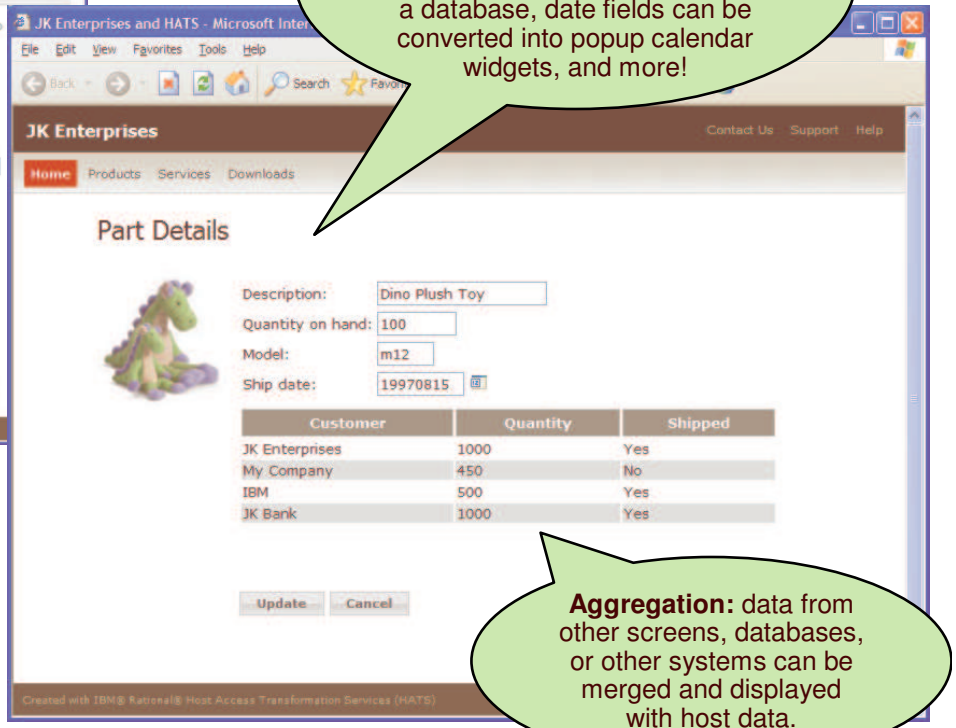


Screen Modernisation At Your Own Pace

Today → Day 1 → Tomorrow







Modern UI: All pages share the same theme and style, which can be based on an existing Web site.

Instant Value: HATS default rendering automatically transforms actions into clickable links and buttons. No changes required to the host application.

Integration: Text can be replaced with images, input fields can be populated from data on other screens or from a database, date fields can be converted into popup calendar widgets, and more!

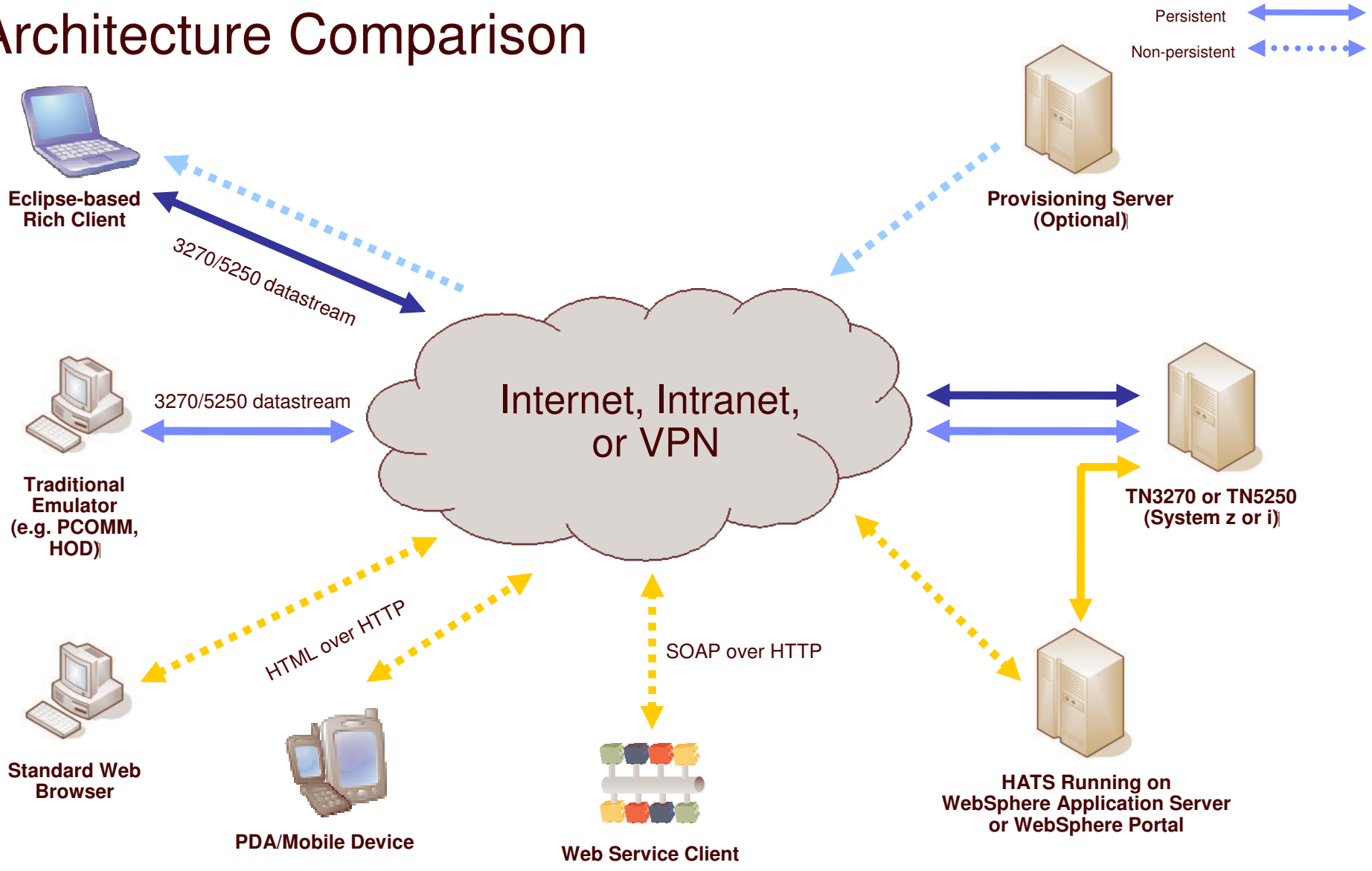
Aggregation: data from other screens, databases, or other systems can be merged and displayed with host data.

Host Access Transformation Services Values

Rich Client	Browser	Portal	Mobile	Web Services
				
• Integration at the desktop with other Eclipse-based applications • Client side processing • Rich set of user interface widgets • Supports Lotus Expeditor deployment	• Zero footprint • Pure HTML • Access through your favorite browser, including Internet Explorer and Firefox.	• Integration at the glass • Cooperative portlet support • JSR 168 compliant	• Access host applications from mobile devices	• Build self-service transactions • Expose host business processes as Web Services • Provide controlled access to vital host applications and host data.

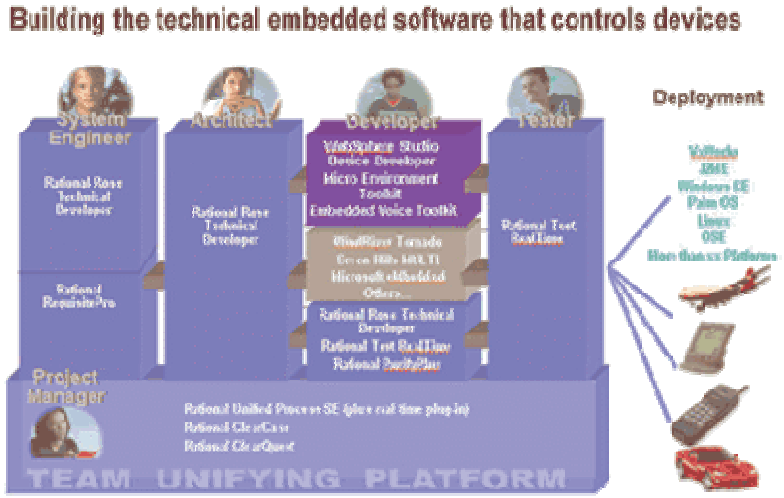
- *Quick ROI : Host applications can be quickly deployed with a new user interface*
- *Low cost : No need to rewrite application*
- *Low risk : Leverage open, proven platforms and technologies*
- *Increase productivity and reduce training costs*
- *Improving work flow from multiple applications*
- *Provides integration of host business processes and data with other Web, portal, and rich client applications*

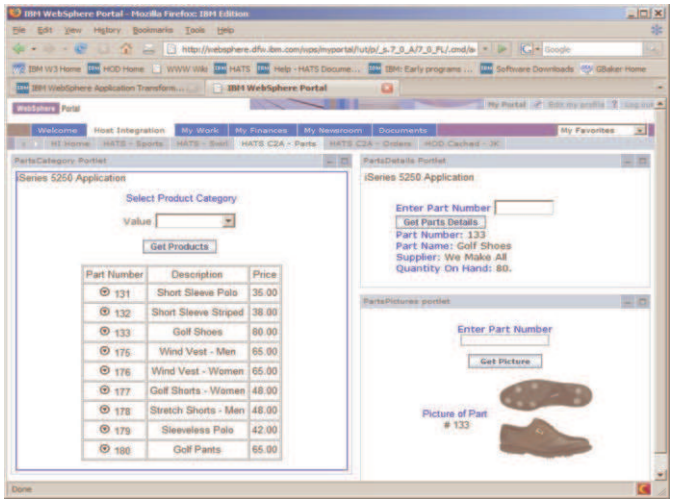
Architecture Comparison



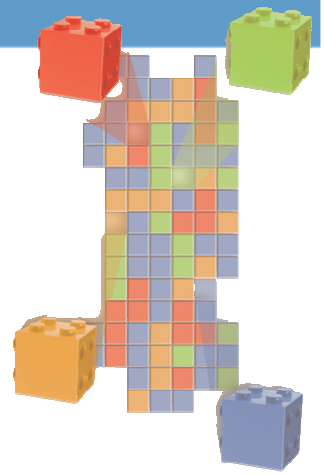
Built on Rational, WebSphere, and Lotus Technologies

- IBM Rational Software Development Platform Tool
  - HATS Toolkit is an Eclipse plug-in to the IBM Rational Software Delivery Platform Tool
  - Wizard-based
  - Creates HATS applications
- WebSphere Application Server, WebSphere Portal Server
  - HATS leverages WebSphere Application Server (WAS) or WebSphere Portal Server at runtime
  - HATS Toolkit creates J2EE .war and .ear files for deployment
  - HATS runtime is contained in the HATS application
- Lotus Expeditor, Eclipse Rich Client
  - Open, standards-based platform
  - HATS runs on the client machine, not on a server





What is Service Oriented Architecture?



- SOA is an approach for building distributed systems that deliver application functionality as services to end-user applications or are used for building other services.
  - Leverages open standards to represent software assets as services
  - Individual software assets become building blocks that can be reused in developing other applications
  - Shifts focus to application assembly rather than implementation details
  - Used internally to create new applications out of existing components
  - Used externally to integrate with applications outside of the enterprise
- SOA is a nearly ideal integration technology to use in an IT environment where software and hardware from multiple vendors is deployed.

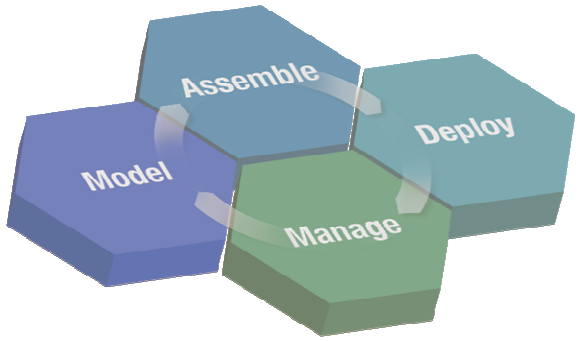
Service Oriented Architecture (SOA) Definitions

What is a service?

A **repeatable business task** – e.g., check customer credit; open new account

What is a Service Oriented Architecture (SOA)?

An IT **architectural style** that supports integrating your business as linked services



- Model your business processes
- Build or re-use existing assets into business-level, loosely-coupled IT components based on open standards
- Assemble your IT services
- Deploy, monitor, and manage your system

The Value of HATS SOA

- HATS is an important part of the IBM SOA strategy!
  - IBM SOA Foundation helps to **extend** the value of the applications and business processes that currently run your business, **not replace** them.
- HATS does not require any application re-engineering or rewriting
  - Rewriting can consume significant amounts of time and resources.
  - Rewriting requires access to and understanding of application source code.
  - Rewriting inherently introduces new bugs that must be found and fixed.
  - Simply put - rewriting is not an option for some customers!
- HATS is unique
  - HATS is the only IBM product that Web Service-enables 3270, including CICS and IMS, 5250, and VT terminal applications – great for customers with a mix of applications (one tool can be used to enable all of them).
- HATS generates standard Web Service interfaces
  - HATS services can be integrated into an existing SOA.

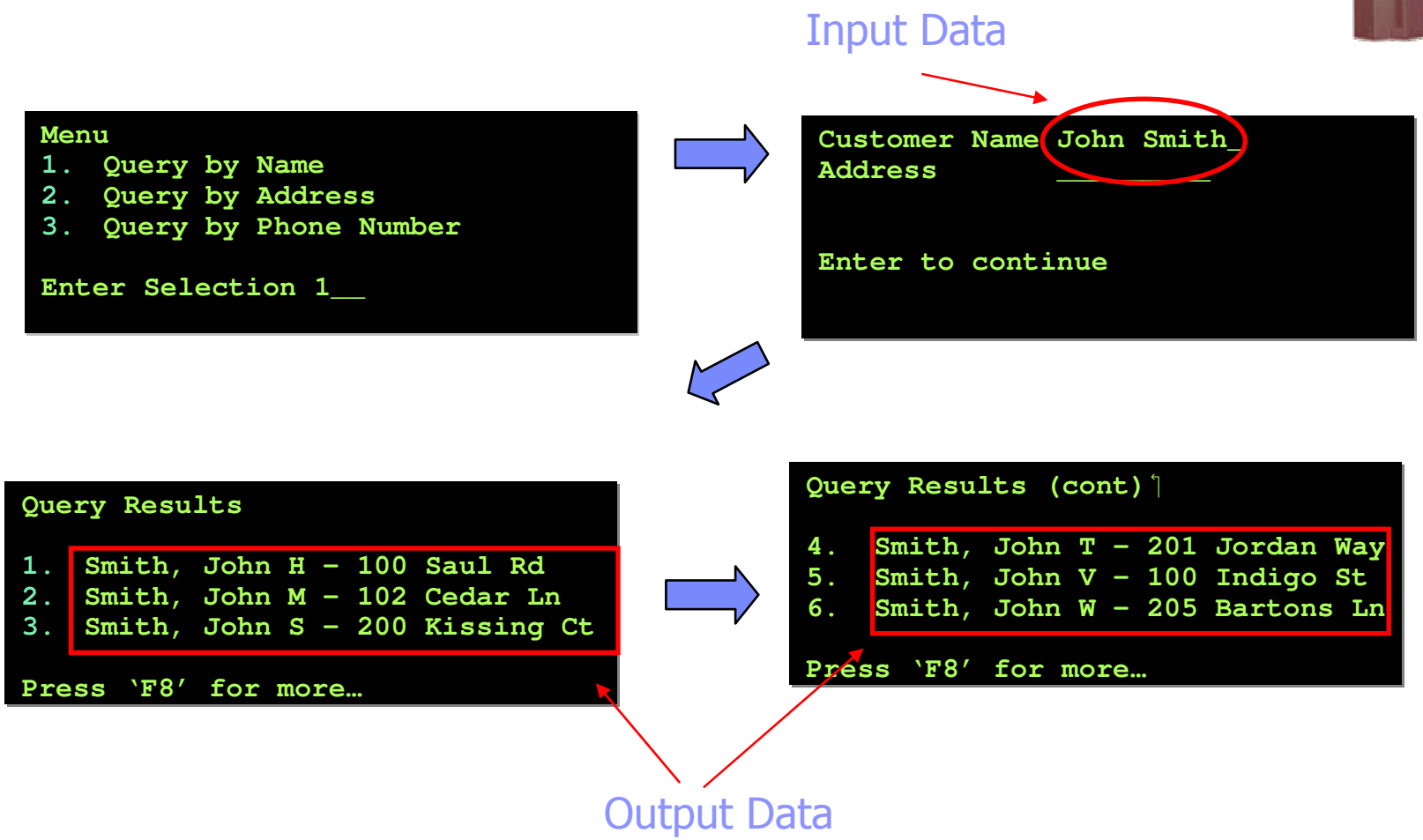
How HATS Fits into an SOA World

- HATS provides the tools to develop industry standard Web services that can drive existing terminal applications.
  - Existing processes can be broken down and exposed as reusable components in a larger SOA environment.

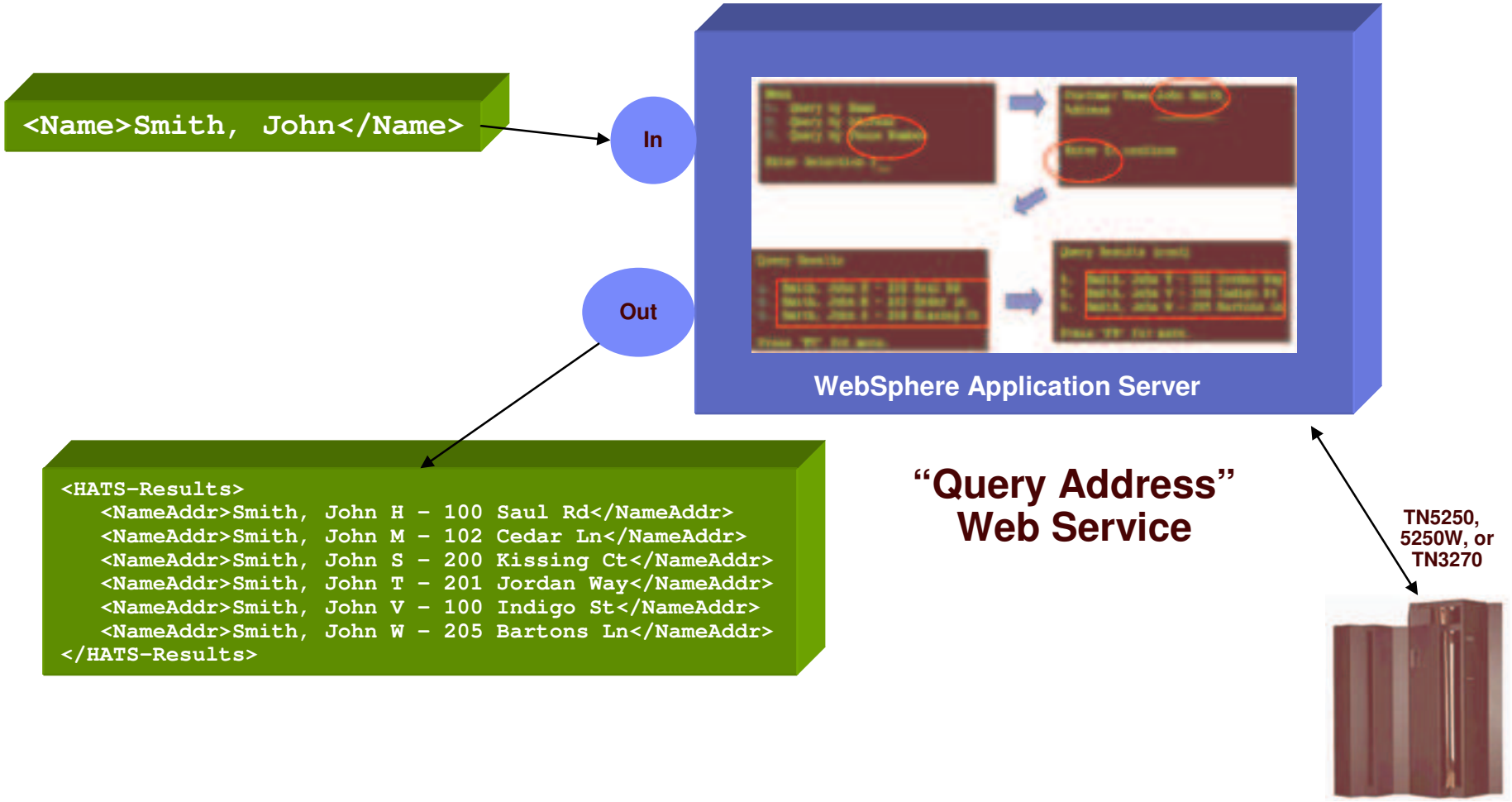
- The process for building Web services using HATS is straight-forward:
  1. Identify the process to expose
  2. Identify the flow of screens
  3. Identify the required inputs
  4. Identify the required outputs

- HATS does not require modification to your existing applications.

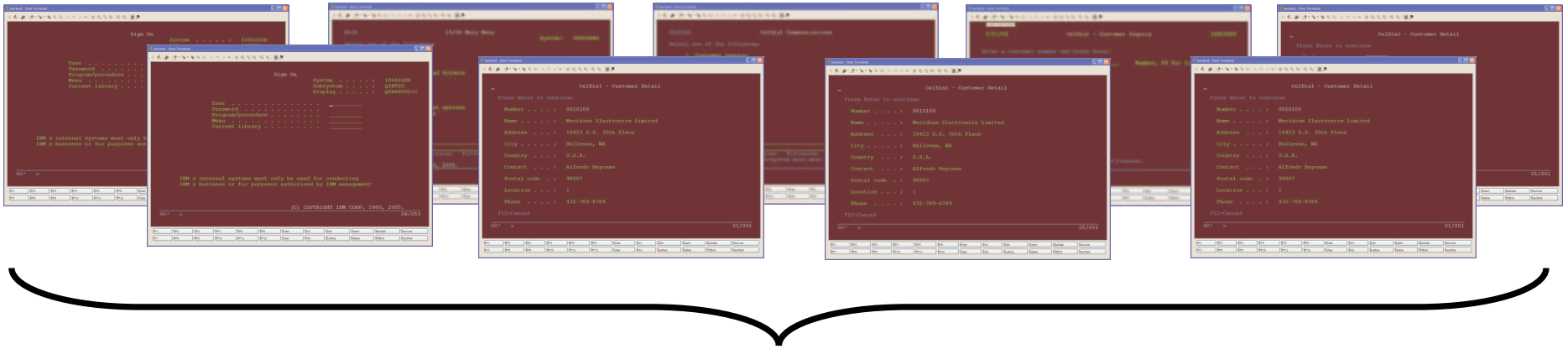
How It Works - A simple example



How It Works - A simple example



Hiding Screen Navigation Complexity

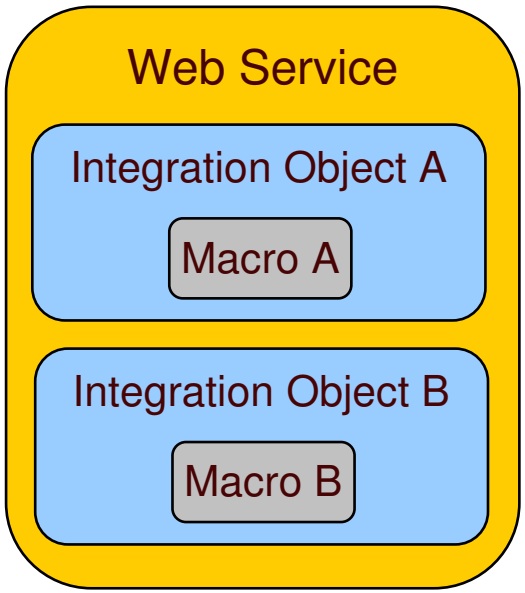


Input: Customer number

Output: Customer name, address, city, state, zip, phone, contact name, past order history, etc.

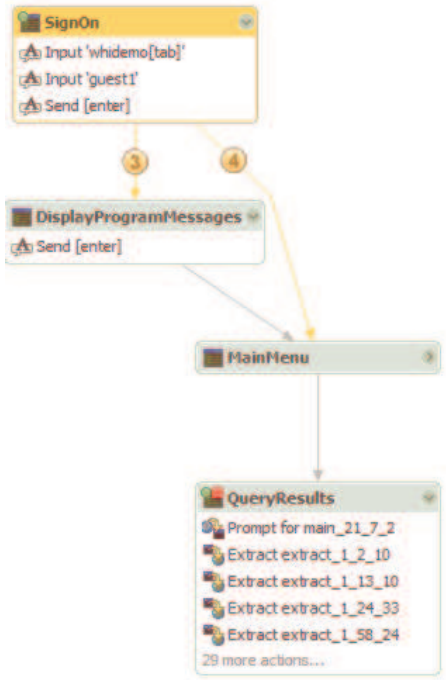
HATS Macros, Integration Objects, and Web Services

- A **macro** is a script that drives through and interacts with the green screen applications.
  - Developer defines “screens” within the macros by identifying screen recognition criteria and actions to perform when the screen is reached.
  - Developer links screens together to define the flow of the macro.
- An **Integration Object (IO)** is a Java class that encapsulates a macro.
  - IOs can be called directly from JavaServer Faces (JSF), Strutus, or standard JSP Web pages.
  - IOs can also be used in Enterprise Java Beans (EJBs) or Web Services.
- A **Web service** is a repeatable operation that can be accessed over a network.
  - A Web service is composed of 1 or more operations.
  - An Integration Object is represented by an operation.



Visual Macro Editor (VME) Technology Preview

- The VME is tool for visually developing HATS macros
  - Macros are the scripts that navigate and interact with 3270, 5250, VT host applications.
  - Macros are the foundation for the Web Service support provided by HATS.
  - Inputs and outputs in a HATS macro are eventually manifested as inputs and outputs in a Web Service (screen interactions are hidden behind the Web Service interface).
- The VME provides value by:
  - Significantly increasing the productivity of HATS macro developers.
  - Allowing for easier handling of alternate / error flows.
  - Helping decrease the number of logic / flow problems (since the flow can be clearly analysed by both developers, expert users, and business analysts).

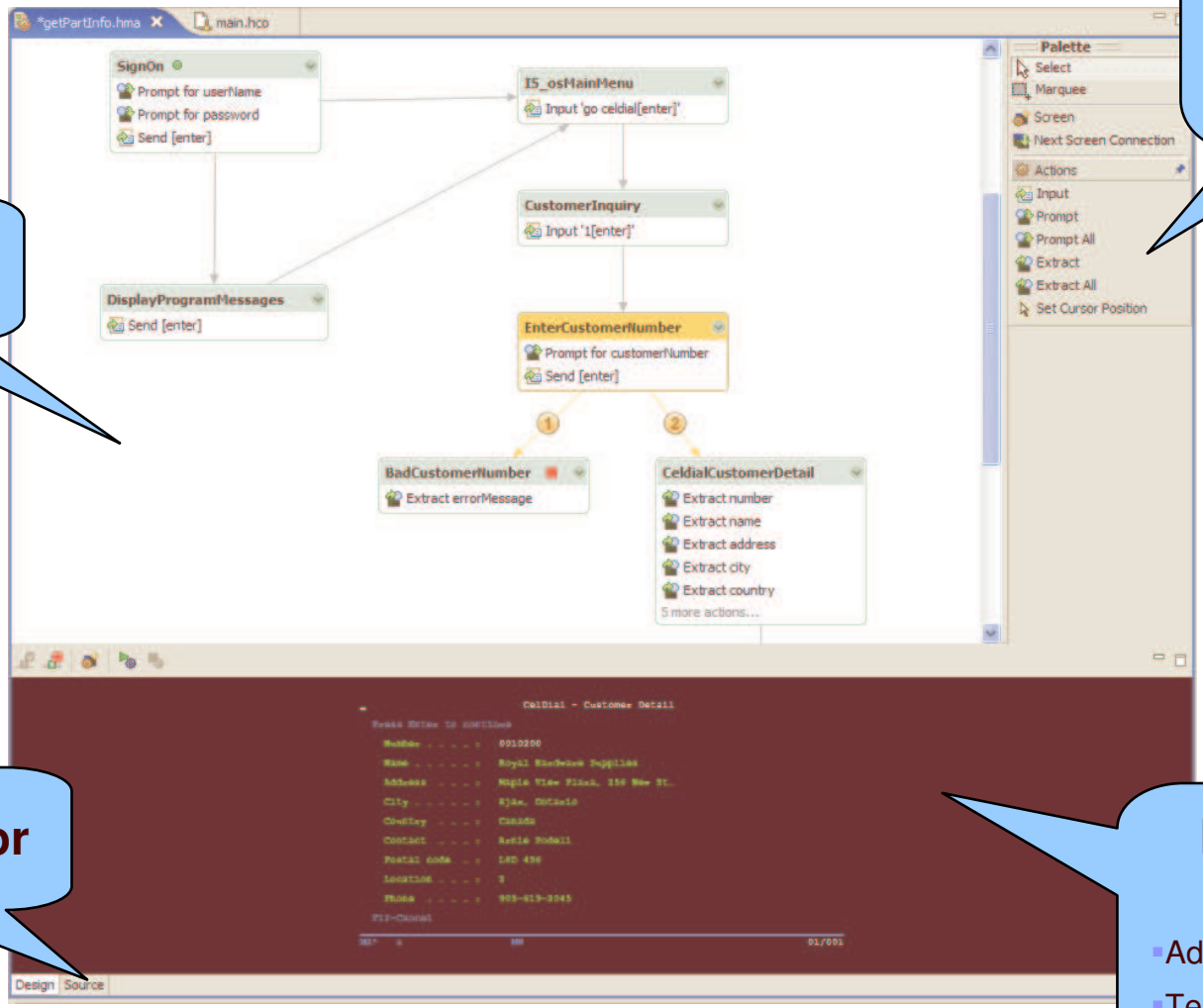


HATS Visual Macro Editor 7.1

Palette

- Add new screens and actions.
- Connect screens together.

Design Canvas



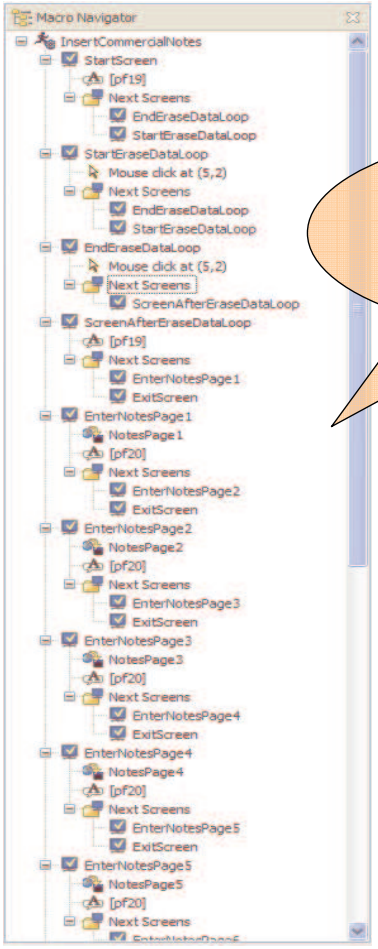
Source Editor

Integrated Terminal

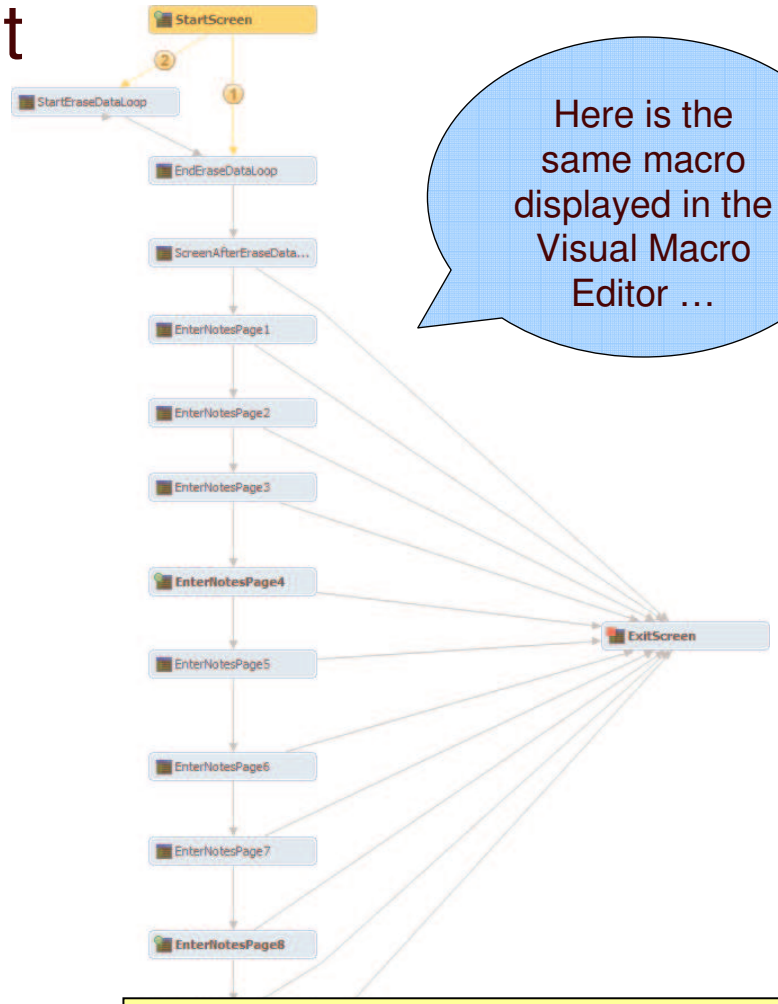
- Add new screens
- Test macro

Why Visualisation is Important

Current HATS macro tooling:



Can you ***easily*** determine the flow of this macro?

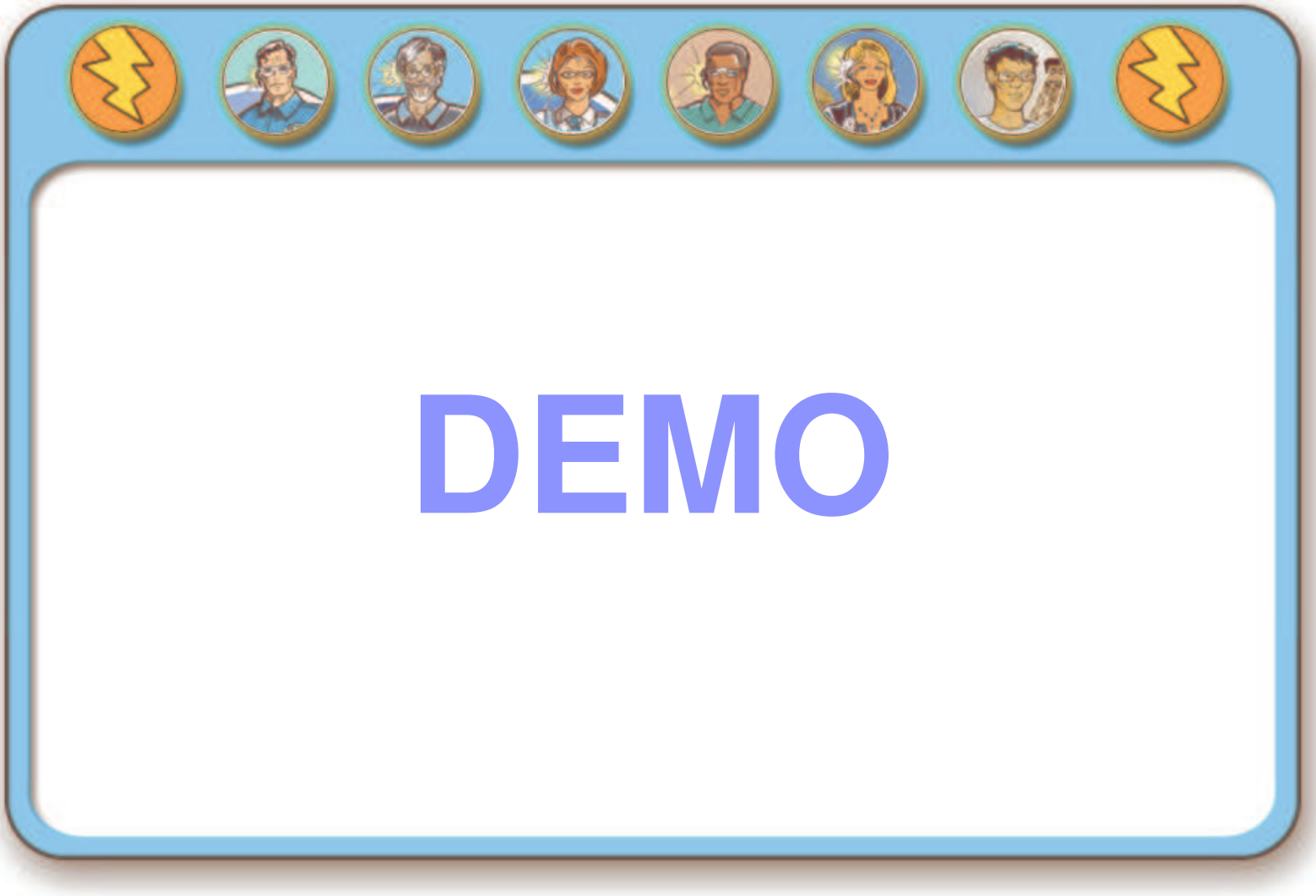


Here is the same macro displayed in the Visual Macro Editor …

It is much easier to understand the logic of a macro and find logic errors and missing flows when you can see the entire macro!

Other HATS SOA Capabilities

- Connection pooling
  - Automatically connects a specified number of connections at application startup. Connections can be reused (this decreases connect/disconnect activity on the system).
- Connect / disconnect macros
  - Automatically signs on sessions at startup and signs them out when no longer needed.
- User lists
  - Signs on sessions using a set of predefined credentials.
  - Starting in HATS 7.1, user list information can be encrypted.
- Stateful or stateless interaction (even for Web services)
  - Enables clients to have stateful interactions (i.e the host session stays active between interactions) with host applications using Web Services.



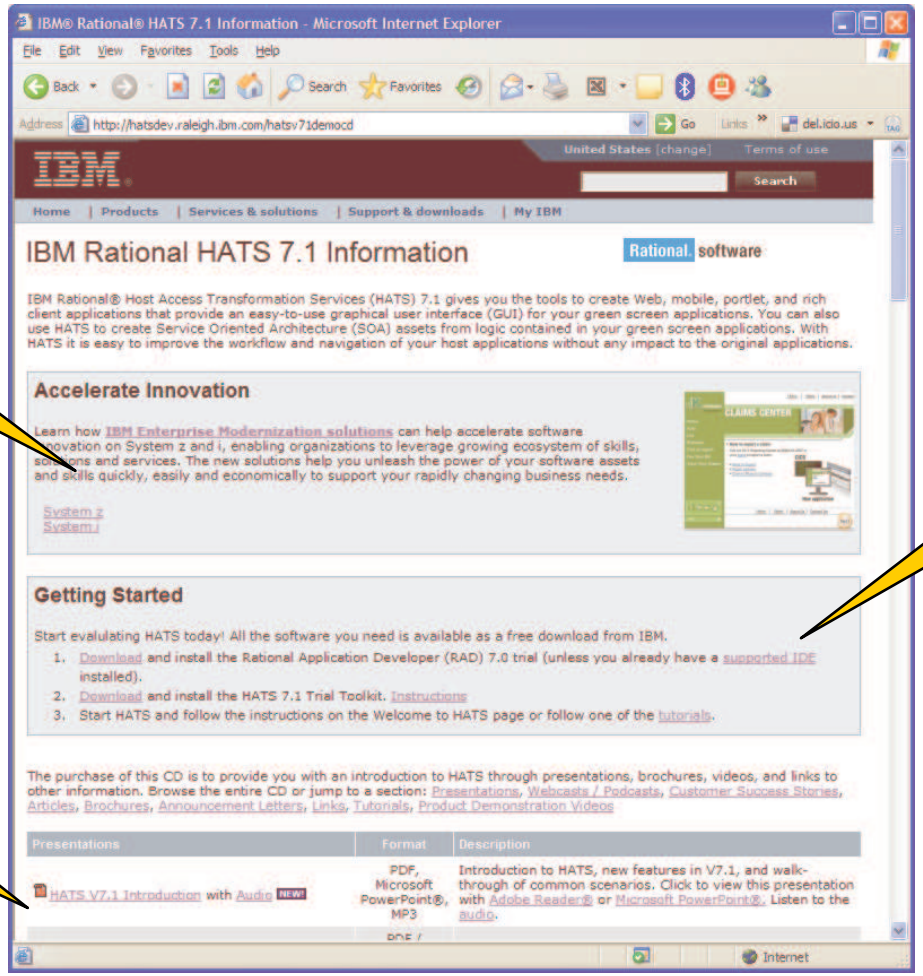
DEMO

HATS and EGL

- HATS and EGL can be used together as part an overall modernisation effort.
- Use HATS to service-enable programs you do not want to touch or impact
  - Reuse proven logic and abstract the details behind a Web service
  - Consume HATS Web services in an EGL Web application
- Why is HATS a good first step?
  - HATS Web services can be built very quickly and without impact to code
  - HATS tooling and EGL tooling run on the same Rational tooling platform
- Videos available on IBM developerWorks
  - http://www.ibm.com/developerworks/offers/lp/demos/summary/eglhats1.html?S_TACT=105AGX01&S_CMP=HP

HATS 7.1 Demo CD – Available Now!

Send an email to hatsdemo@us.ibm.com to request a copy (include mailing address and number of CDs).



Success Story

Business need

- 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.

Solution

- 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.

Benefits

- 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.

- HATS Product Page
  - http://www-.ibm.com/software/awdtools/hats/
- Demos
  - http://rational.dfw.ibm.com/
- Trial Code
  - http://www14.software.ibm.com/webapp/download/product.jsp?s=p&id=GMDL-5BJRPP
- HATS Demo CD
  - hatsdemo@us.ibm.com

Other Online Resources

- IBM WebSphere Host Integration Solution Product Page: http://www-306.ibm.com/software/webservers/hostintegration/
- Demos: http://rational.dfw.ibm.com/whidemo/atdemo_hats.html
- Library, Webcasts, Podcasts, Brochures, Presentations, Redbooks: http://www-306.ibm.com/software/webservers/hats/library.html
- Education: http://www-306.ibm.com/software/webservers/hats/education.html
- Services: http://www-306.ibm.com/software/webservers/hats/services.html
- Trial Code: http://www14.software.ibm.com/webapp/download/search.jsp?go=y&rs=hatst
- Customer Comments: http://www-306.ibm.com/software/webservers/hats/hats_customers.html
- HATS IBM Education Assistant content: http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.hats_v6/plugin_coverpage.html


QUESTIONS

THANK YOU

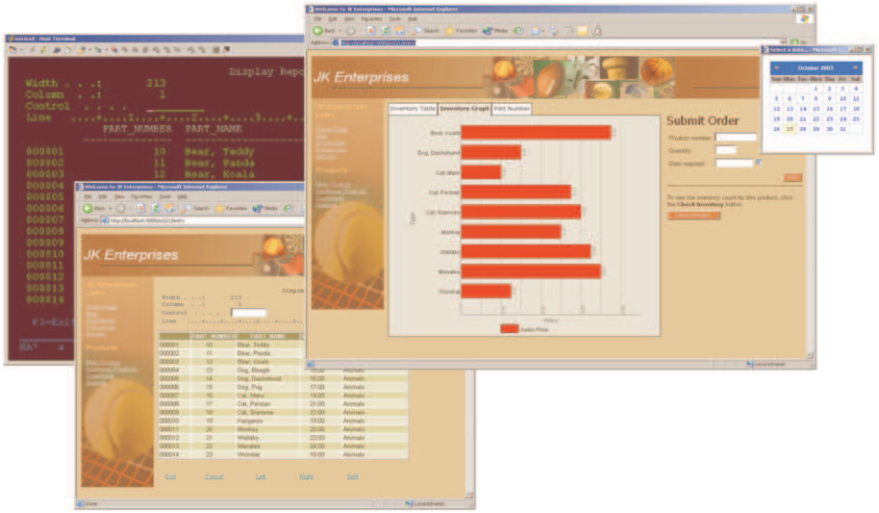
Learn more at:

- IBM Rational software
- IBM Rational Software Delivery Platform
- Process and portfolio management
- Change and release management
- Quality management
- Architecture management
- Rational trial downloads
- Leading Innovation Web site
- developerWorks Rational
- IBM Rational TV
- IBM Rational Business Partners

Backup

Summary of HATS Features and Value Propositions

- Host screens are converted to GUIs on the fly, in real time
- Improved navigation of host application
- Zero footprint on the desktop with Web deployment
- Support deployment to Rich Client
- Access host assets from mobile devices
- Combine screens and data from multiple applications
- Eliminates need to customize every screen
- Generate Web services from host transactions
- Exploits security & scalability of WebSphere Application Server and WebSphere Portal
- Extensible through IBM Rational Software Delivery Platform
- Supports iterative development



HATS 7.1 Release Highlights

- Rebranding
- Mobile device support
- JSR 168 portlet support
- Improved SOA tooling
- Other new features
  - Arrow key navigation support (rich client only)
  - Automatic setting of WAR class loader policy
  - New Remove Global Variable action

- Dates:
  - 3/11/2008 – Announcement
  - 3/28/2008 – eGA
  - 4/10/2008 – GA

HATS 7.1 - Rename/Rebrand/Reprice/Repackage

- **Rename**
  - From WDHT to Host Access Transformation Services for 5250 Applications
- **Rebrand**
  - From WebSphere to Rational
- **Reprice**
  - From Multiple Tier prices to Standard Value Unit measures
- **Repackage**
  - WDSC and WDSC AE will not advance – no future release to include WebFacing Tool
  - WebFacing Tool will be an installable option in HATS Toolkit

Packaging

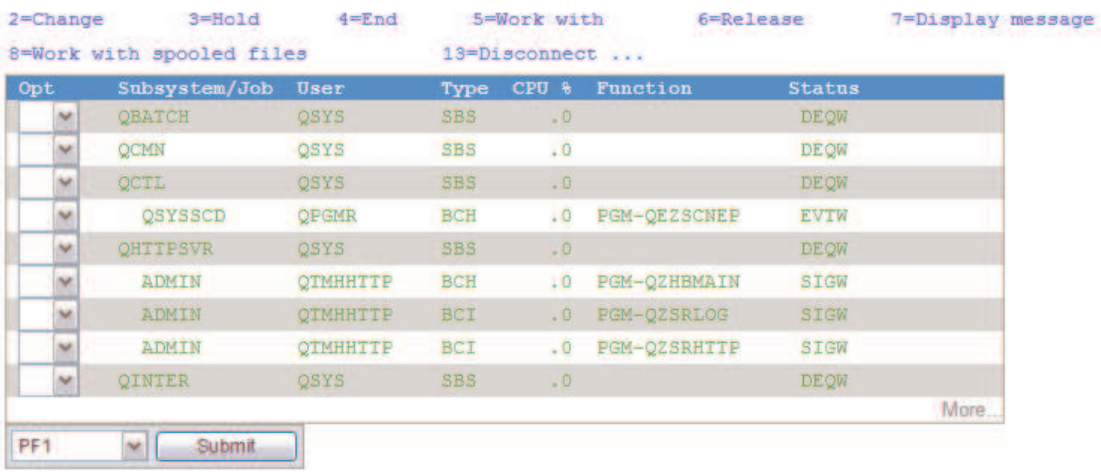
- **HATS for Multiplatforms and HATS for Linux on zSeries**
  - Available for use with 3270 and 5250 applications
  - HATS applications can be deployed for Web, portal, rich client and mobile access.
  - HATS applications can be deployed on any supported platform (refer to server runtime requirements)
  - Ships with a restricted WebSphere Application Server (WAS) Network Deployment license
- **HATS for 5250 applications on Multiplatforms**
  - Available for use with 5250 applications only
  - HATS applications can be deployed for Web, portal, rich client, and mobile access
  - HATS applications can be deployed on any supported platform with the exception of i5/OS (refer to server runtime requirements)
  - Ships with a restricted WebSphere Application Server (WAS) - Express license
- **HATS for 5250 applications on i5/OS**
  - Available for use with 5250 applications only
  - HATS applications can be deployed for Web, portal, rich client, and mobile access
  - HATS applications can be deployed on i5/OS (refer to server runtime requirements)
- **HATS is also included as a component of Host Integration Solution (HIS)**

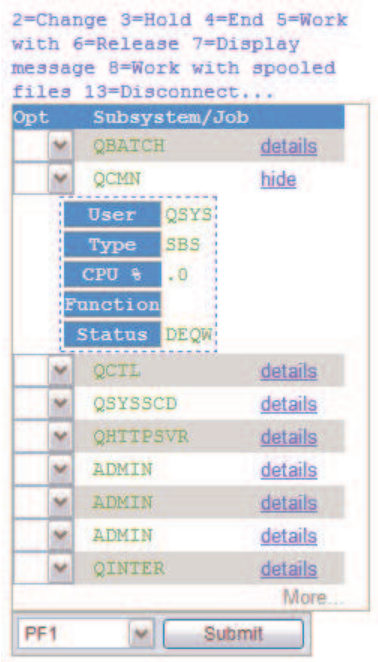
What is HATS mobile device support for v7.1?

- HATS transformation applications can now be accessed by pervasive devices such as cellular phones, data collection terminals, and Personal Digital Assistants
- This allows for the HATS developer to:
  - Customize and transform individual screens
  - Add business logic to screen interactions
  - Enhance the presentation, layout, and flow of the original host systems
- Supported browser: Internet Explorer Mobile 5.0 or later
- Reminder: HATS Integration Object or Web service data can also be used in conjunction with mobile devices

Table and Subfile column arrangement enhancement

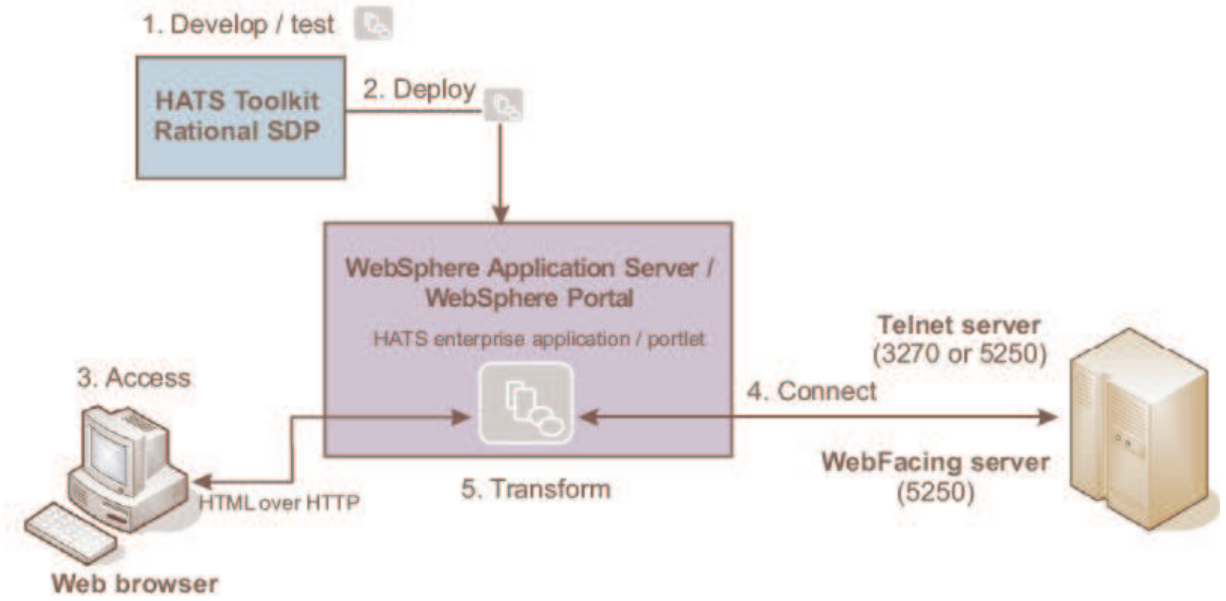
- This enhancement enables tables to be collapsed and expanded so that more data can fit in the table or subfile representation without resulting in horizontal scrolling.
- As an additional option, the table and subfile widgets will create the expandable areas for retrieval by the browser only when requested
  - Uses AJAX technology
  - Reduces the memory and size footprint on the client device
- The feature can also be used in standard HATS Web applications.





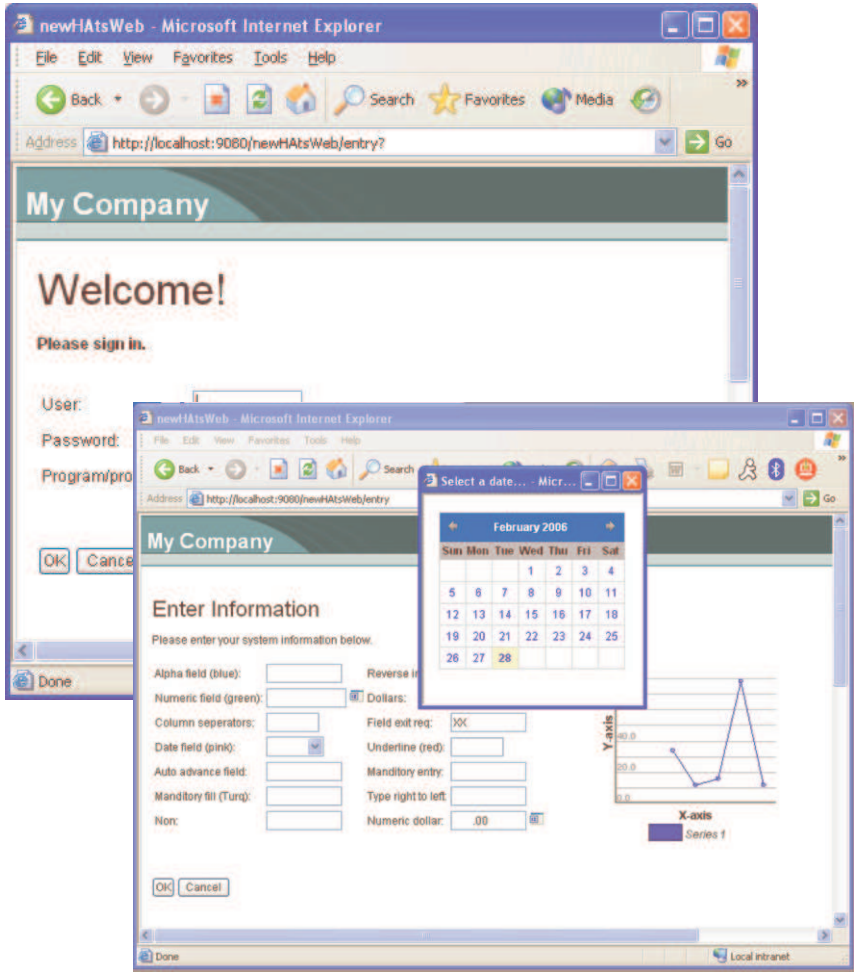
Creating a HATS Web Application

1. The developer uses the HATS Toolkit to develop and test a HATS Web application.
2. When ready, the developer deploys the HATS application by exporting it as a J2EE application and installing and running it on a WebSphere Application Server system.
3. Using a Web browser, the user accesses the HATS application.
4. The HATS runtime connects to the target host system through either a Telnet or a WebFacing server.
5. As the user interacts with the host through the HATS application, the HATS runtime transforms host screens to a GUI.

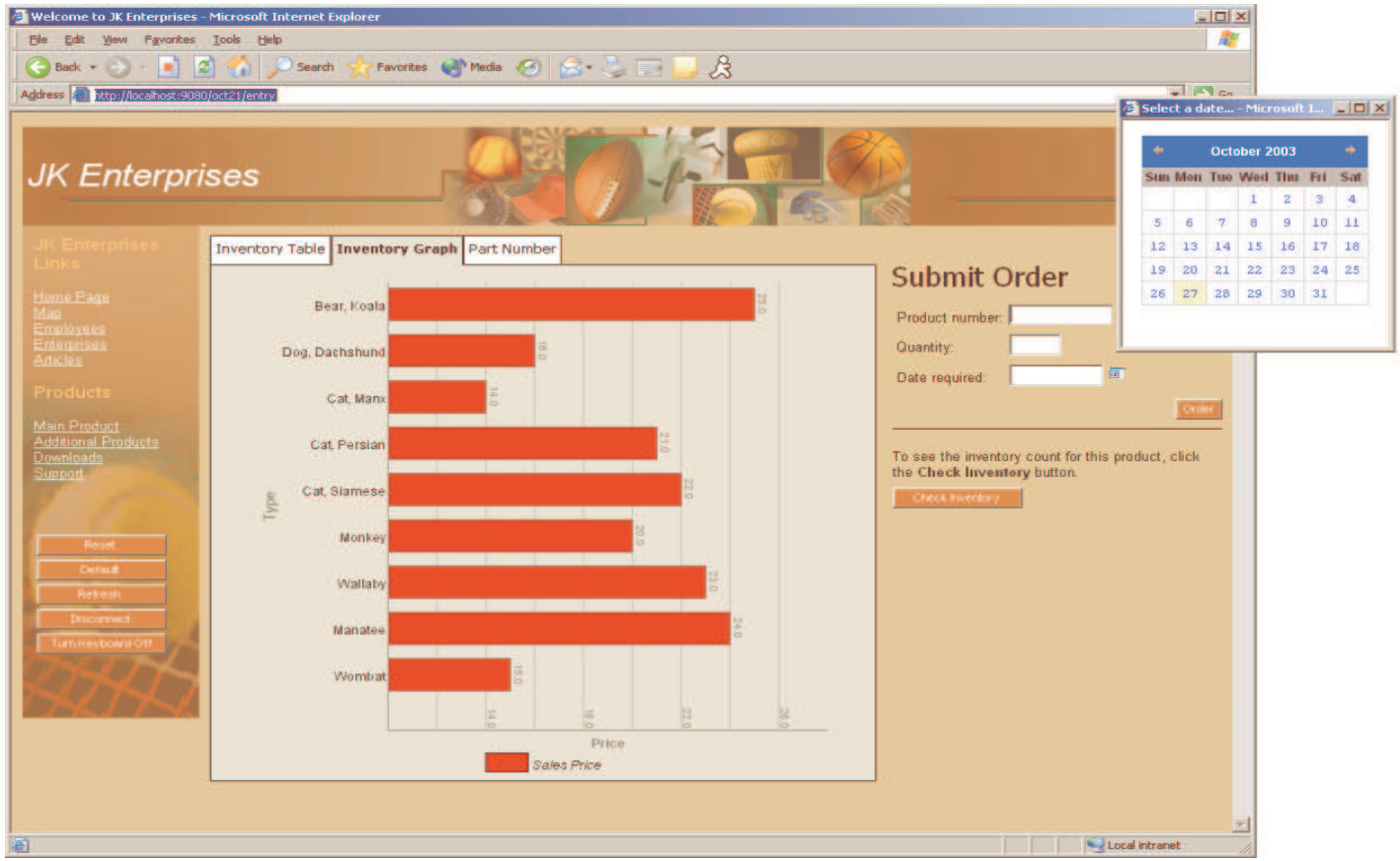


Achieving a Modern Look with HATS

- Customise screens with intuitive web user controls, including:
  - Popup lists, drop-downs, checkboxes, lists, radio buttons, calendar popups, graphs, etc.
- Simplify screens and work flow by
  - Split one screen into multiple screens
  - Organize data in tabs
  - Remove functions / actions no longer needed by removing full screens or partial screens
- One customised Web page can be applied to more than 1 host screen!



HATS Customised Transformation



Now that we have extended the applications as services…

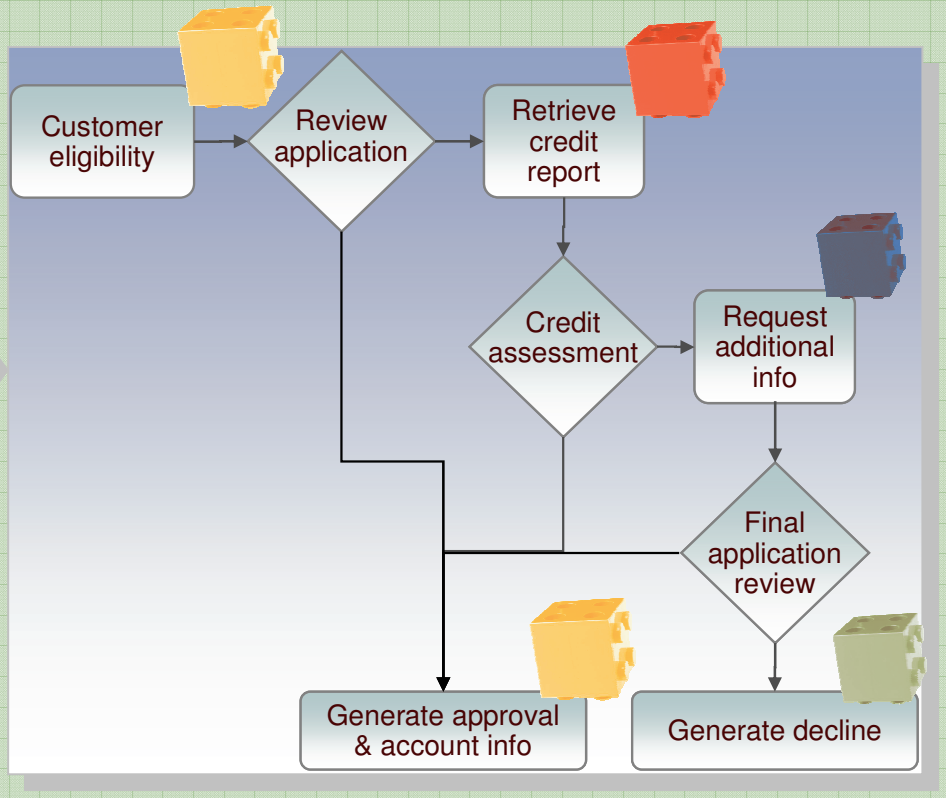
Determine Customer Eligibility

Retrieve Credit Report

Request additional info

Generate decline

Generate approval

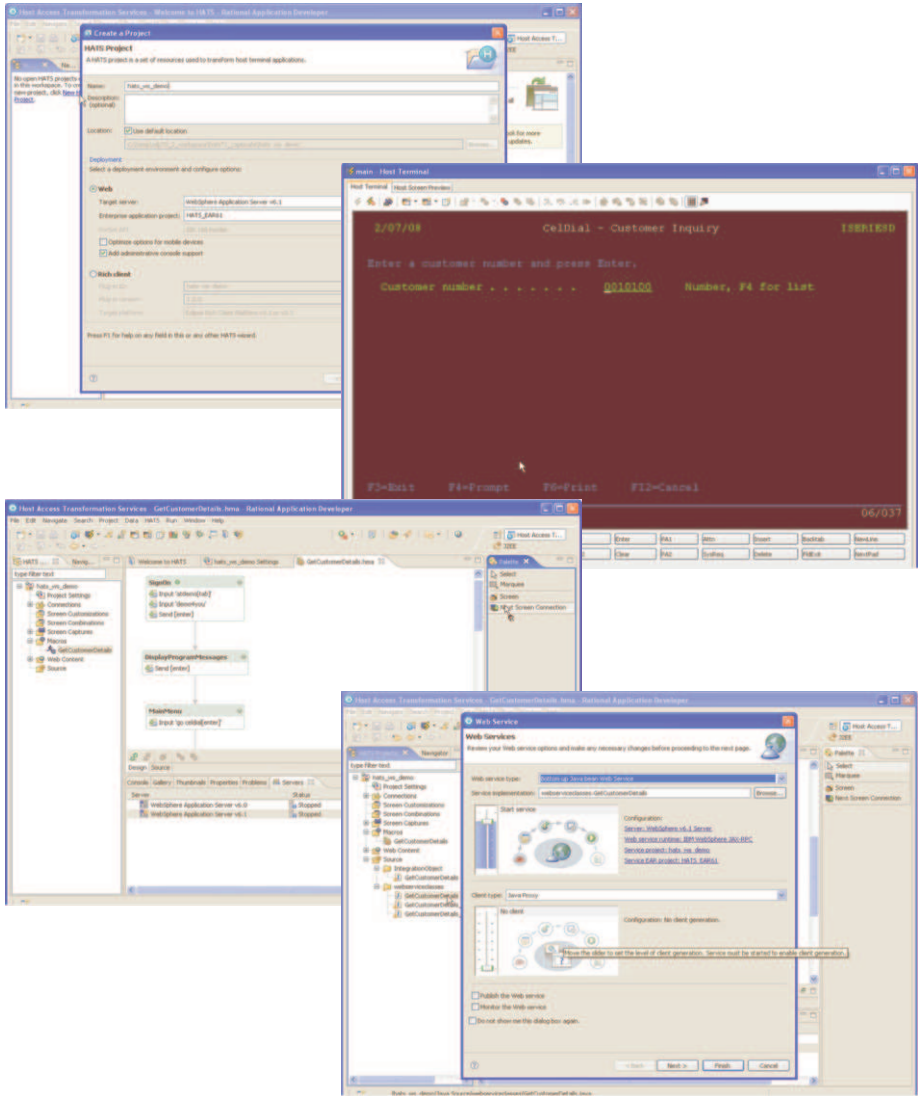


Assemble your services into business processes with WebSphere Integration Developer

Run your business processes with WebSphere Process Server

HATS Developer Tasks

1. Create a HATS Web project using the HATS toolkit.
2. Create a macro to navigate the screens, supply input, and extract output.
3. Use the HATS toolkit to generate a Java bean wrapper for the macro.
4. Use the Rational tooling to generate a Web service (and WSDL file) to drive the Java bean.
5. Deploy to WebSphere Application Server.



EGL Developer Tasks

1. Create a new Dynamic Web project, and add EGL support to it.
2. Import HATS Web services WSDL into EGL project and use the tooling to generate the EGL interface code.
3. Create new JSP page and code the page handler code to call the service.
4. Add new UI components to the page and wire to the page handler.
5. Deploy and run.