Modern Application Development Featuring Web 2.0 for System z

Discover and Transform Application Assets

Rational. software

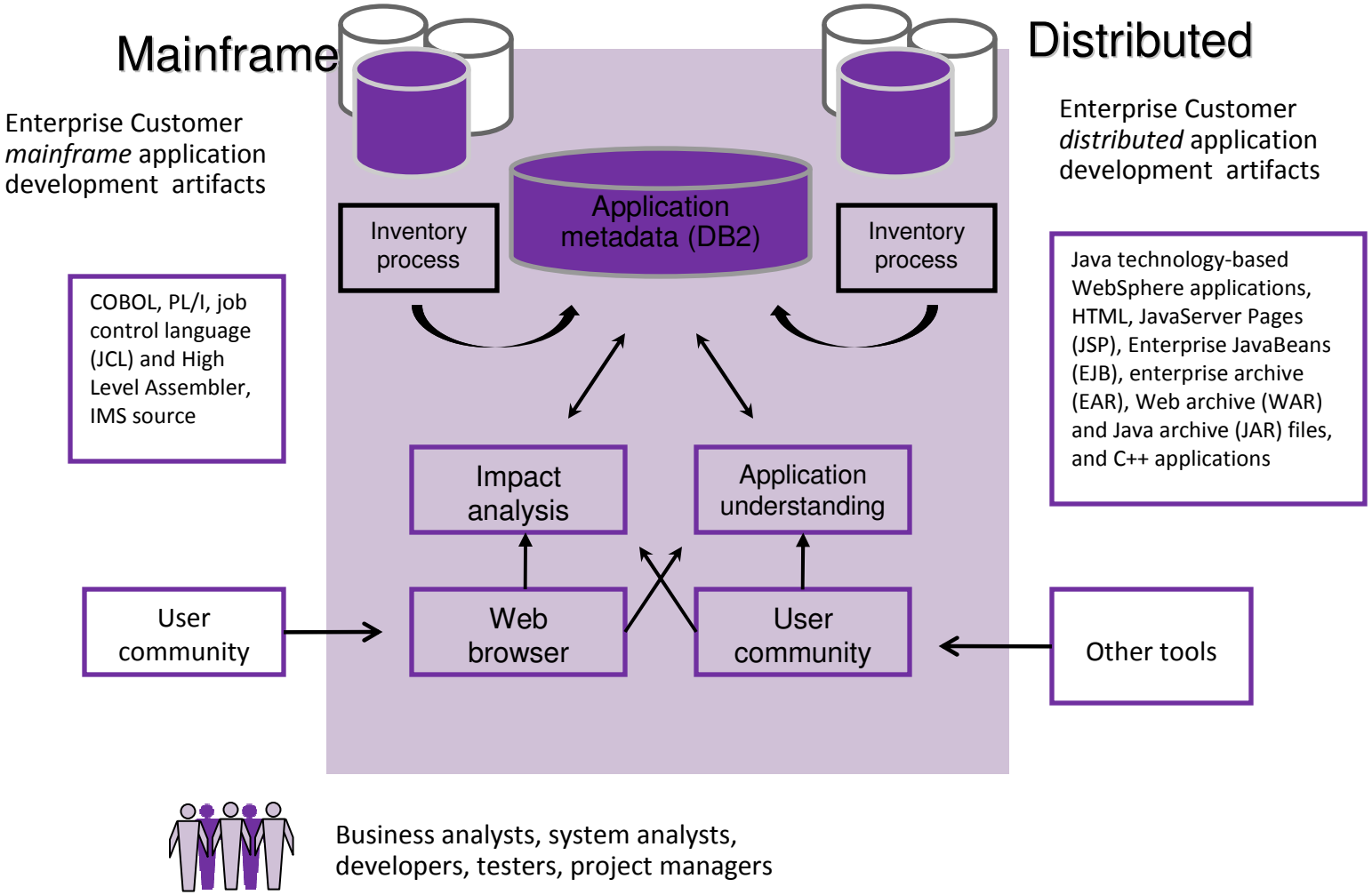
Discover and Transform Application Assets

- The purpose of this section is to show how solutions from IBM Rational can help you discover and reuse existing assets in modern applications.
- **Agenda**
  - WebSphere Studio Asset Analyzer Overview
  - Rational Asset Analyzer Overview
  - Rational HATS Overview and Demo

What is WebSphere Studio Asset Analyzer (WSAA)?

- Provides a browser-based interface to search, explore and report on the gathered information
- Support for System z, enterprise wide software applications
- Scans source “where it lives”
- Assists in the development and test phases to identify impacts across the enterprise and identifying areas to test, thereby reducing risk and improving test coverage
- Identifies relationships among the artifacts
  - Gathers metrics, counts related to the IT artifacts in the enterprise

WebSphere Studio Asset Analyzer
Architectural View



What is Rational Asset Analyzer (RAA)?

A sister product of WSAA

- Based on proven WSAA technology
- Support for the same languages/subsystems as WSAA

WSAA

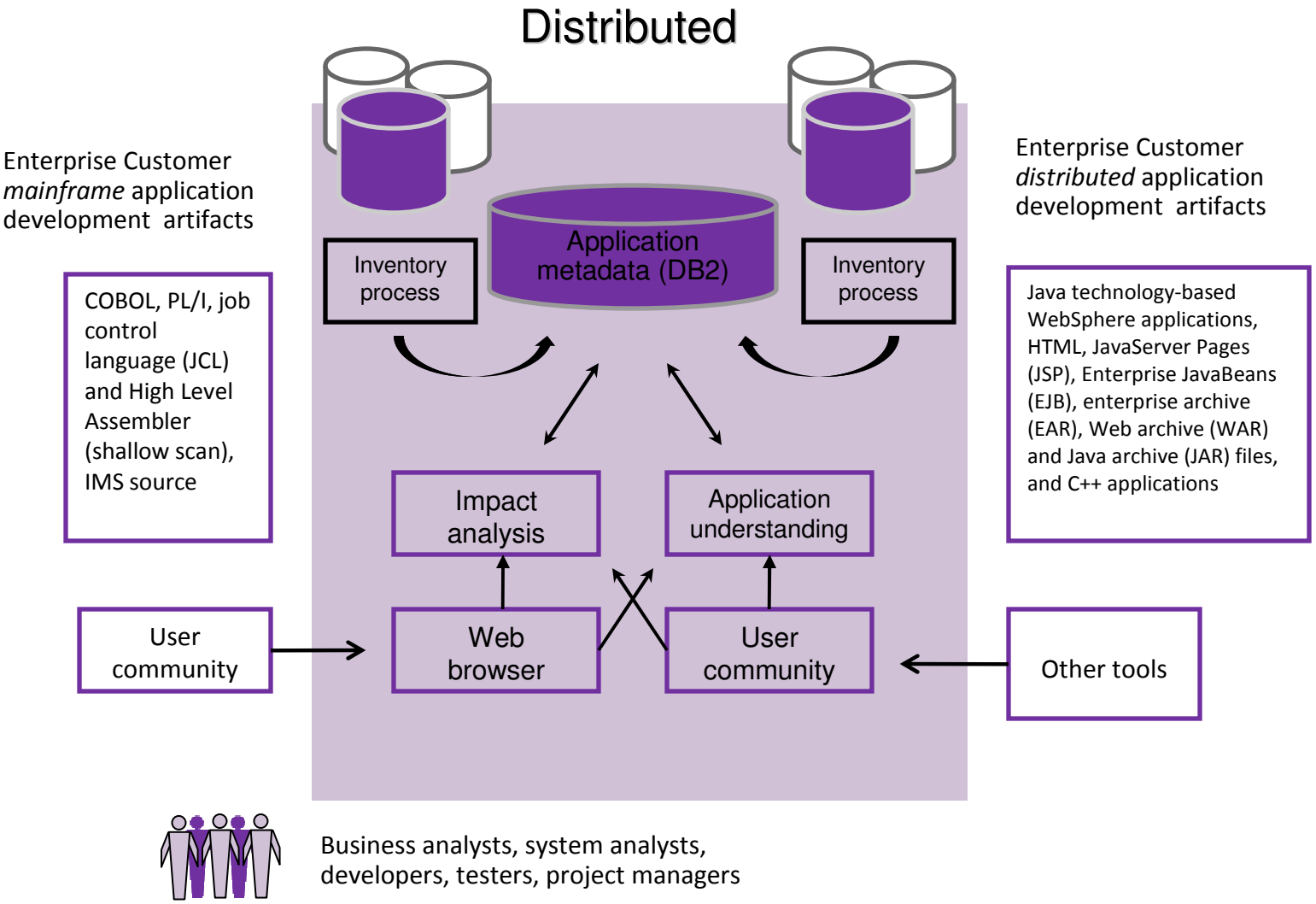
- Installs on z/OS
- Scans source "where it lives" - on z/OS, Windows and AIX
- Typically used for large teams
- Browser-based user interface
- Identifies relationships among the artifacts
- Gathers metrics, counts related to the IT artifacts

RAA

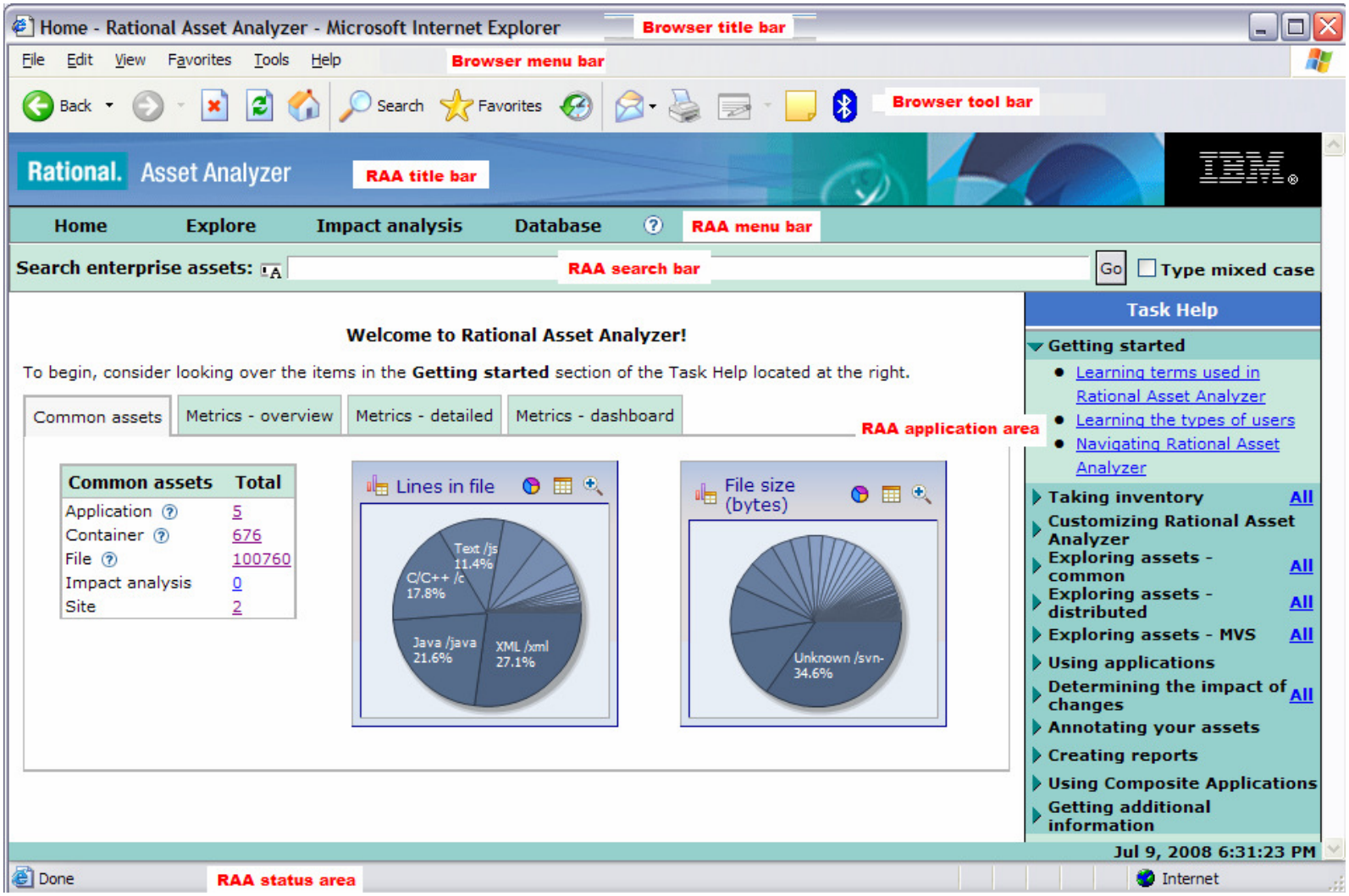
- **Installs on Windows**
- **Scans source on Windows**
- Can be used for large **or small teams**
- Browser-based user interface
- Identifies relationships among the artifacts
- Gathers metrics, counts related to the IT artifacts

- **Dashboards for counts & metrics**
- **Ability to control depth of inventory scanning**
- **Very fast directory scans to gather counts & metrics quickly**
- **Eliminates need for DB2 on System z**
- **RESTful API (documented in External Interface Guide)**

Rational Asset Analyzer
Architectural View



RAA – Home Page



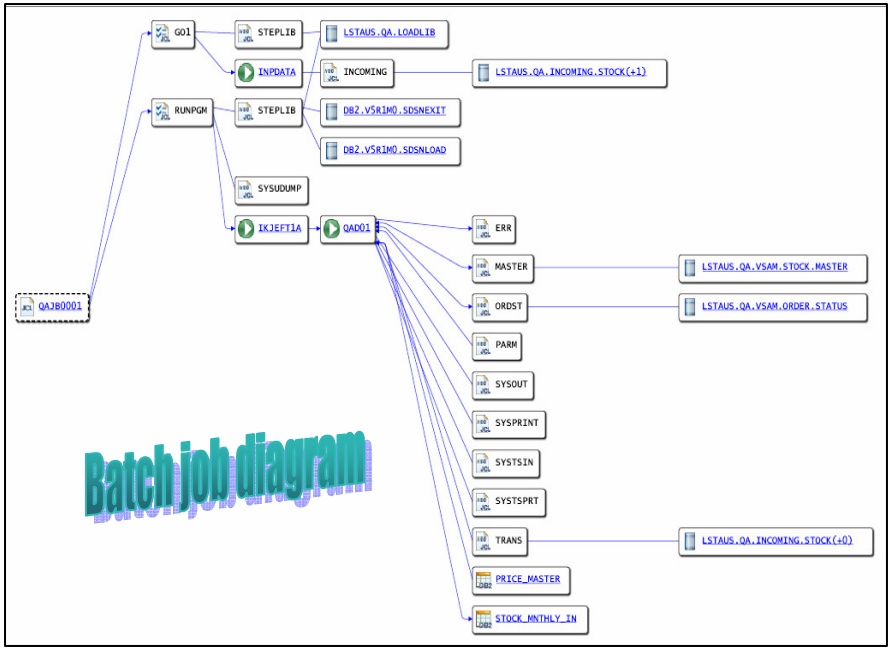
The Value of Asset Analyzer

- **Accelerate project delivery in response to business drivers**
  - Reduce risk
  - Increase productivity
  - Improve quality of application changes
- Gain **intellectual control** of applications
- Gain **transparency** into outsourced development
- **Customize** RAA to organizational processes and IT environments

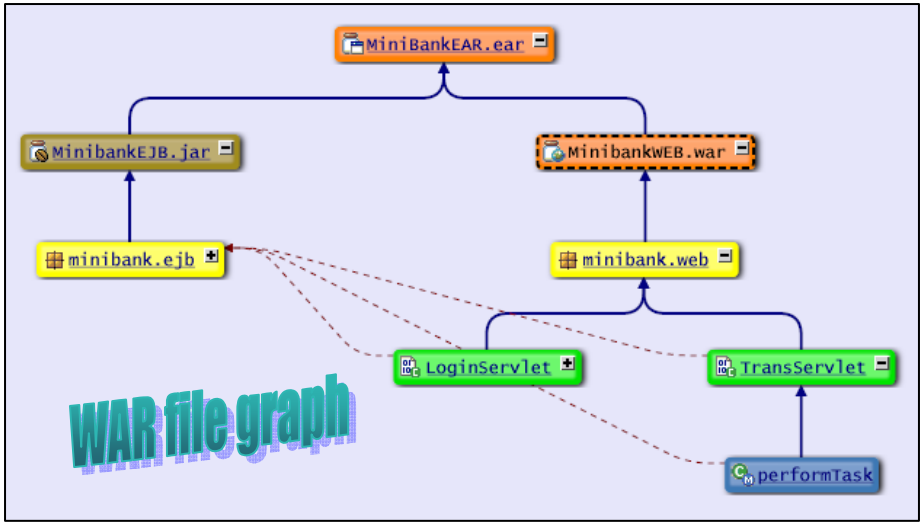
Rational Asset Analyzer – Application Understanding

Quickly understand code with little or no documentation, and relationships across the enterprise

- Group artifacts into **user-defined groups** called Applications to limit scope to area of interest
- Use **various types of diagrams** for understanding how the application "hangs together"



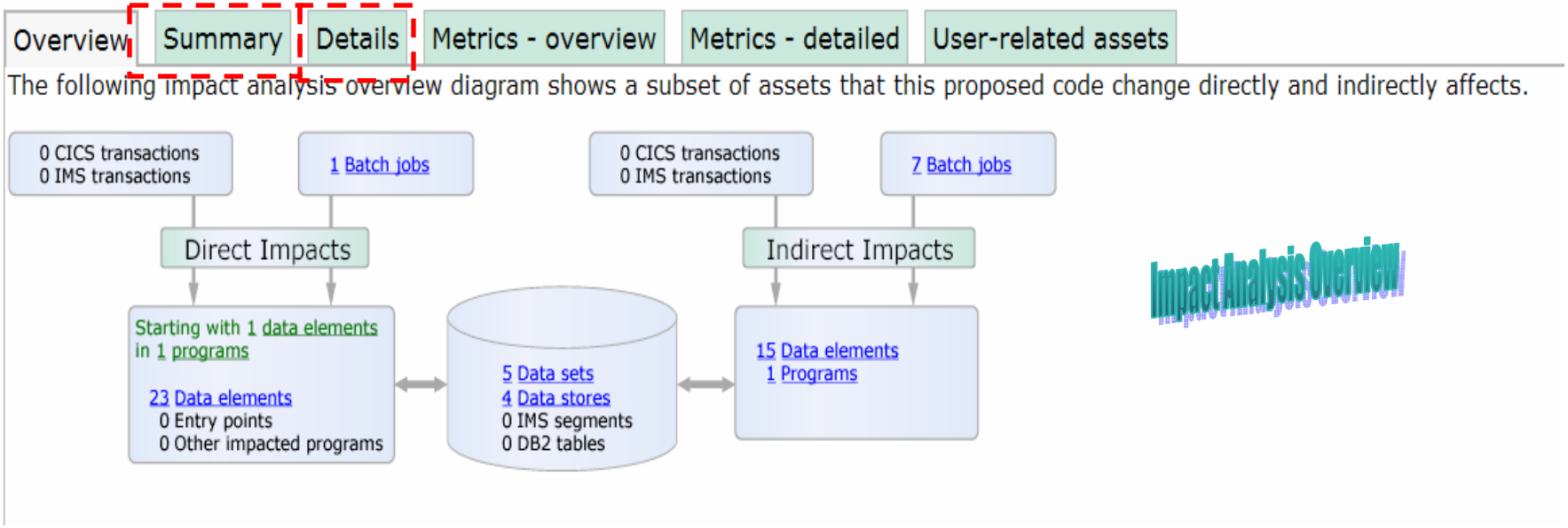
- Use **annotations** to capture knowledge from SMEs e.g. Business function, description, etc.
- Create **user-defined relationships** for situations where relationships cannot be determined through static analysis
- Perform **enterprise-level keyword searches**



Rational Asset Analyzer – End to End Impact Analysis

Reduce time to market & risk of downtime by understanding change impact upfront

- Reduce time to determine **scope of change** whether for new enhancements, or even maintenance efforts
- View the metrics for impacted artifacts to determine the **risk of change** i.e. cyclomatic complexity, lines in file, etc.
- Traverse user-defined relationships to determine **impacts across platforms** i.e. follow dependencies from mainframe to J2EE and back.
- Create a "**bill of materials**" of impacted artifacts by evaluating the details page



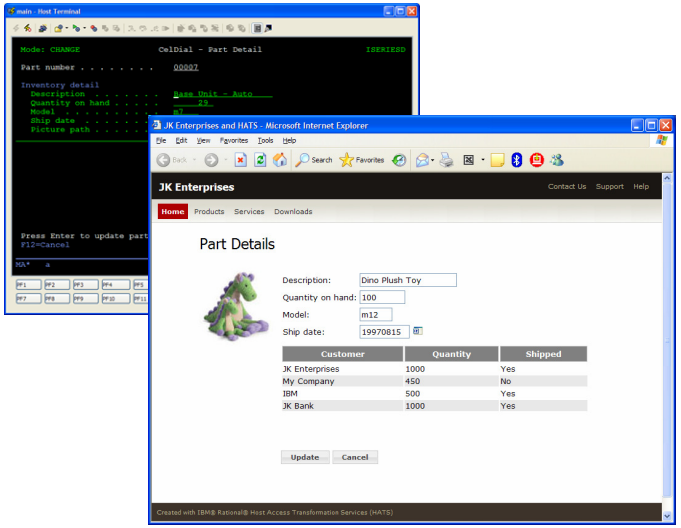
Rational Asset Analyzer – Synergies

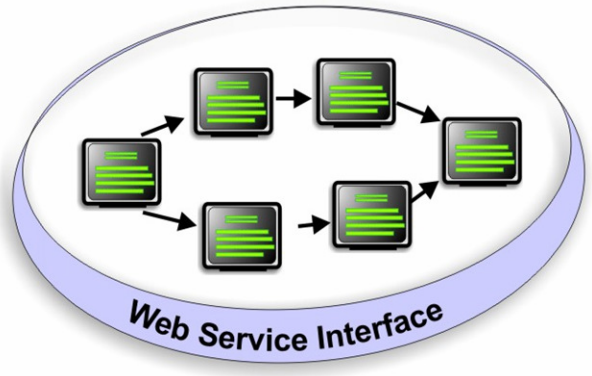
RAA enhances value of other tools

- **Rational Developer for System z (RDz)**
  - *Combine productivity gains of RDz with the enterprise-level insight in RAA to minimize time-to-market and minimize risk due to lack of visibility into impacted artifacts*
- **Rational Asset Manager (RAM)**
  - *Use RAA to identify artifacts/assets of interest, then use RAM to publish those assets for reuse, and manage/govern the development process around changes to those assets. That is, jumpstart RAM deployments for a quicker ROI.*
  - *Use RAA to perform deeper level analysis for change requests coming in to RAM.*
- **Rational Transformation Workbench (RTW)**
  - *Use RAA to gain visibility and control at enterprise level, then*
    - *Use RTW Business Rules Extension to extract, document and manage business rules.*
    - *Use RTW Architect Extension to identify and remove dead code, and accelerate componentization efforts.*

IBM Rational HATS

- **What can you do with HATS?**
  - Quickly and easily transform green screen applications to the Web, Portal, mobile device, or rich client without rewriting
    - Easy to use
    - Highly customizable
    - Iterative development process
    - Transformation “on the fly”
  - Extend terminal application tasks as Web services
- **Benefits**
  - Extend host applications to new users
  - Improve the navigation of your host applications
  - Reuse your existing assets in a Service Oriented Architecture
  - Low risk



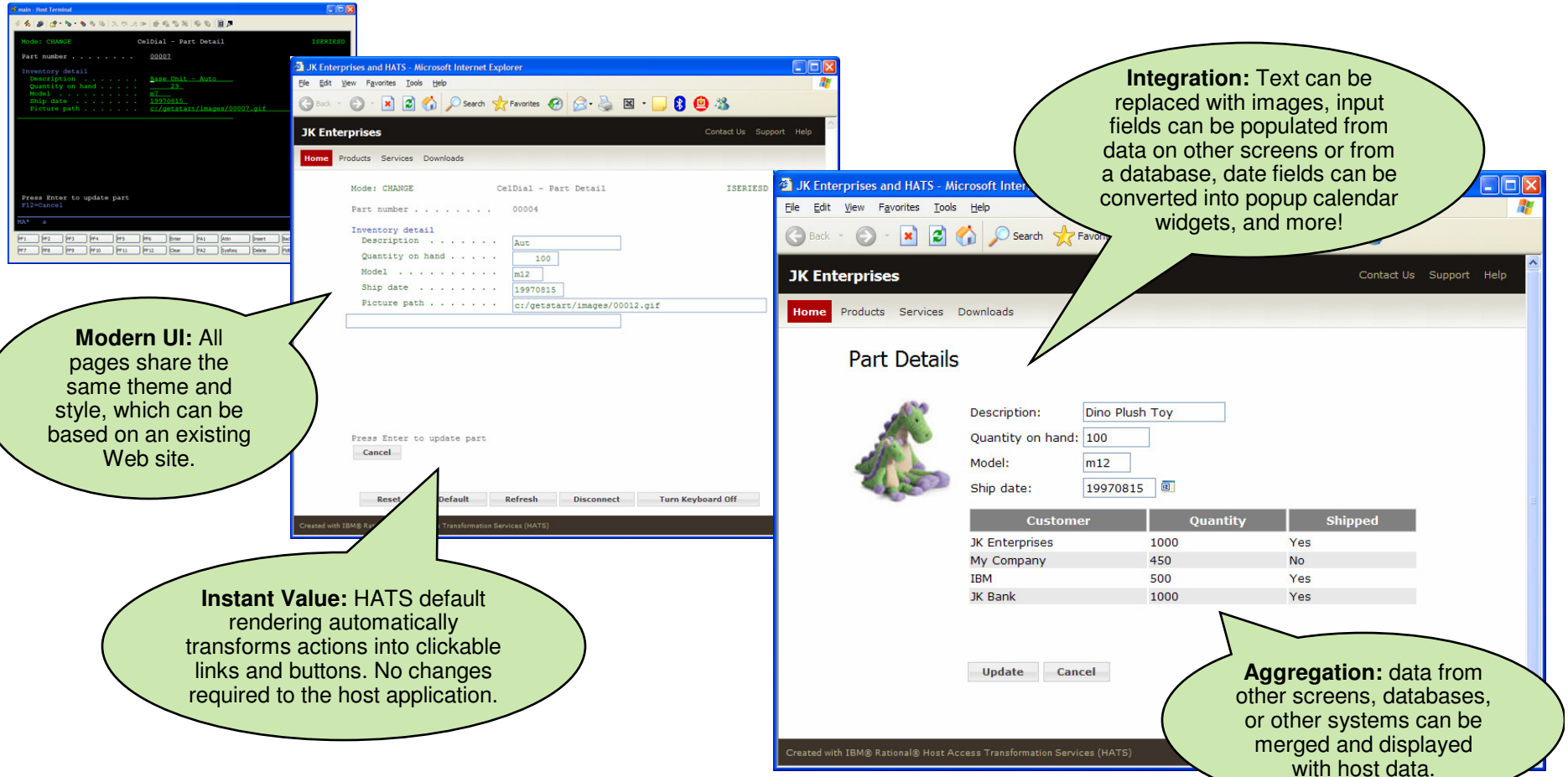


Modernization At Your Own Pace

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.

Business Value of HATS

- **Improved appearance, usability and navigation of existing, proven applications**
  - Reduce transaction times
  - Reduce training time
  - Reduce employee turnover
- **Speed to web, with less cost and less risk**
  - Reusing existing assets rather than re-writing
  - No changes to existing applications
  - Build on existing knowledge and skills
- **Externalize legacy transactions and data**
  - Reach new end users, markets, sales channels through web self-service
  - 24 X 7 access
  - Viewable, printable
- **Build self-service transactions**
  - Improve the availability of information and transactions
  - Off-load call centers, voice response units and clerical employees
- **Integrate information from multiple applications and databases "at the glass", into Portals**
- **Make existing applications "integration ready", using Web Services to link existing applications to new web applications**

Environments

Modernizing green screen applications without rewriting, refactoring, or reengineering

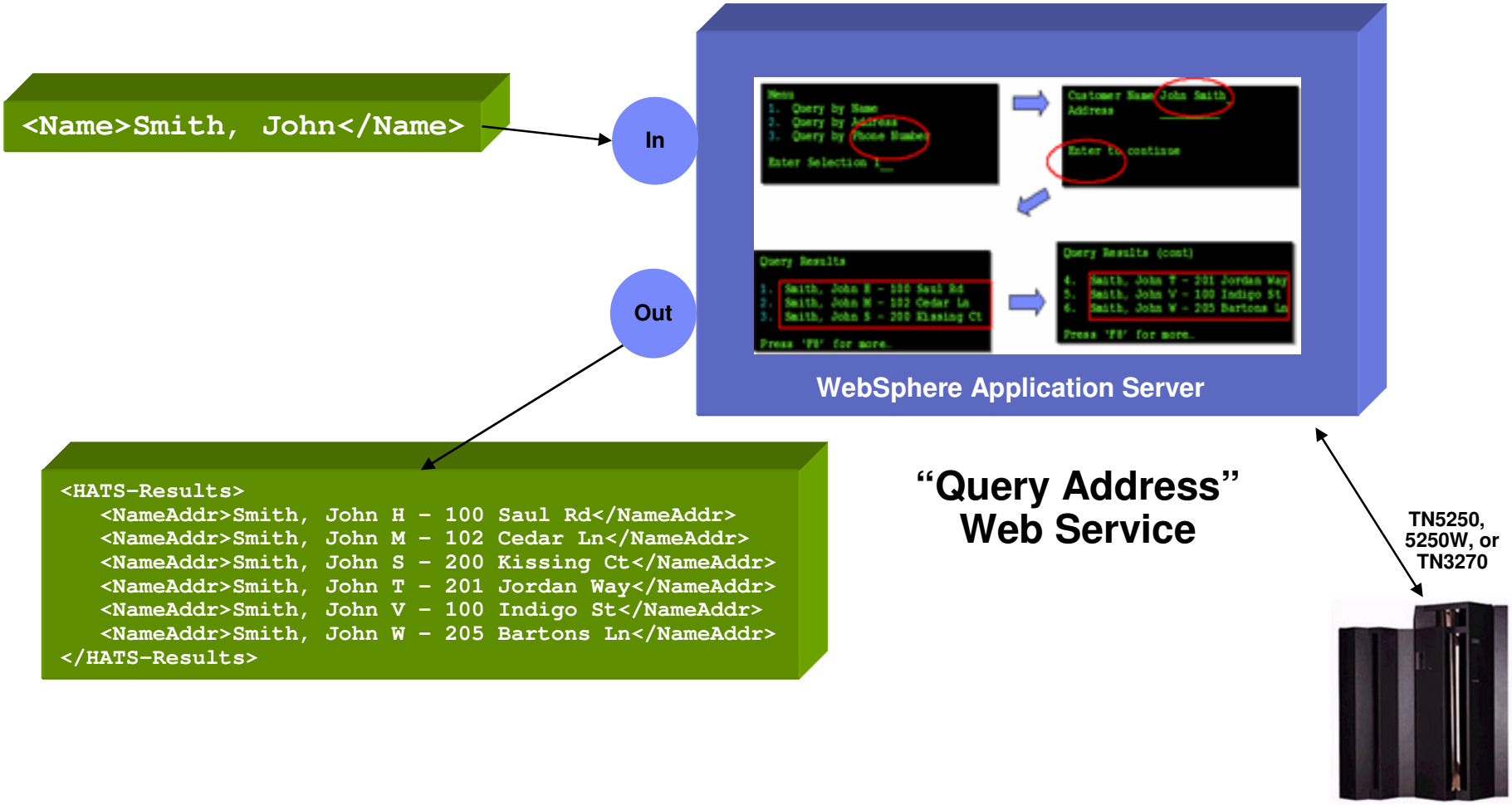
Browser	Rich Client	Portal	Mobile	Web Services
				
• Zero footprint • HTML and JavaScript only (no applet or browser plug-in required) • Access using all popular browsers, including Internet Explorer and Firefox.	• Integration at the desktop with other Eclipse-based applications • Client side processing • Rich set of user interface widgets • Support for Lotus Expeditor	• Integration at the glass • Cooperative portlet support • JSR 168 compliant	• Access host applications from mobile devices • Extend applications to mobile users	• 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*

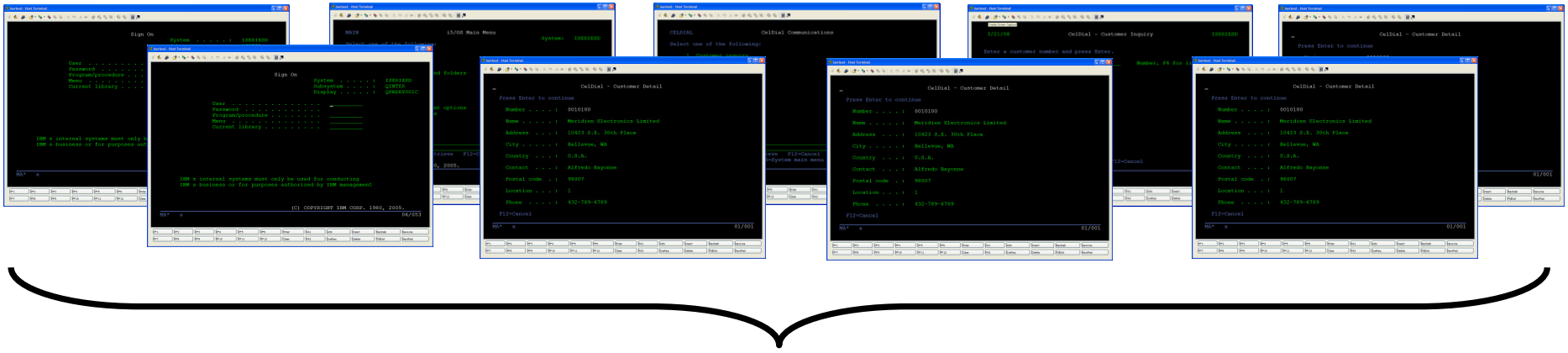
The Value of HATS SOA

- HATS does not require application reengineering 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 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 It Works - A simple example



Hiding Screen Navigation Complexity

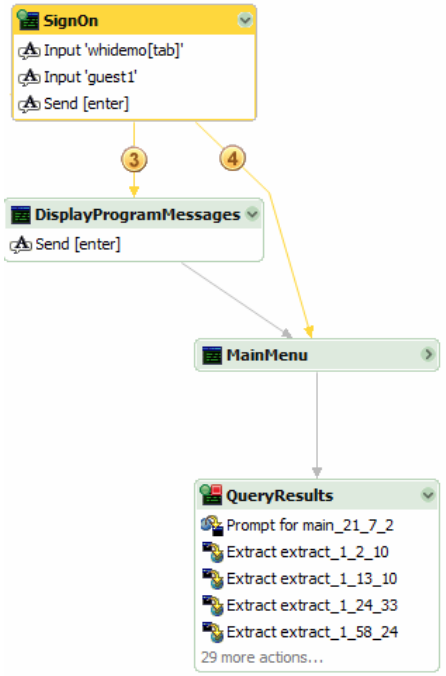


Input: Customer number

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

Visual Macro Editor (VME)

- **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 analyzed by both developers, expert users, and business analysts).
- **Fully supported in HATS 7.5 (November 2008)**

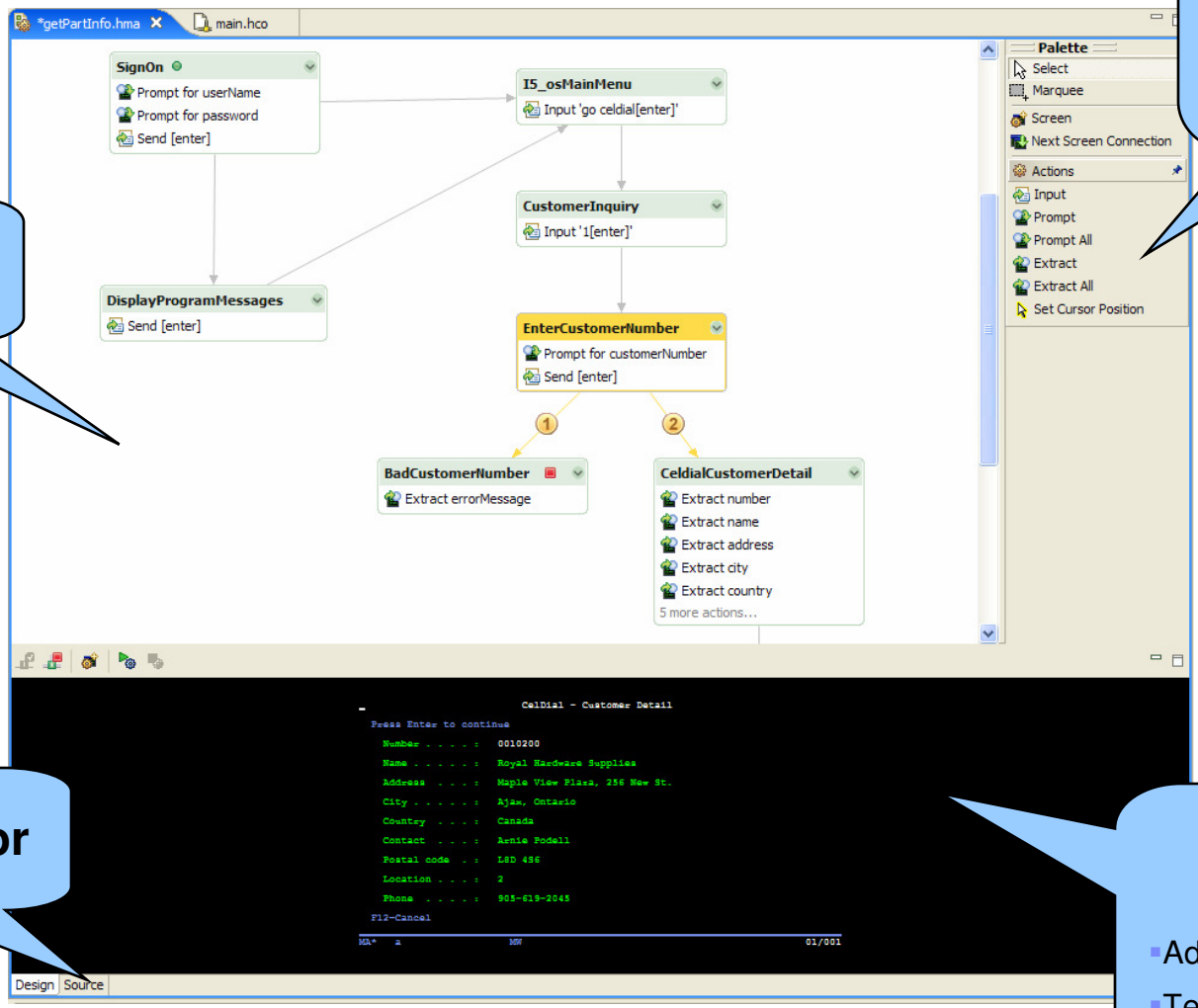


HATS Visual Macro Editor

Palette

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

Design Canvas



Source Editor

Integrated Terminal

- Add new screens
- Test macro

HATS