IBM RATIONAL HATS & SOA INTEGRATION

The evolution of modern business technology leads to a need of collaborative systems that can be dynamically integrated for instant delivery of business values. SOA standard-based flexibility allows services (loosely coupled applications) to be distributed over a connected information technology infrastructure thus creating an ever growing, easily accessible repository of business functions.

In web services based SOA architecture; web services are independent of the communication layer and they describe a standards based flexibility for loose coupling and integrating application using XML, SOAP, WSDL and UDDI open standards over any protocols as shown in figure 1.

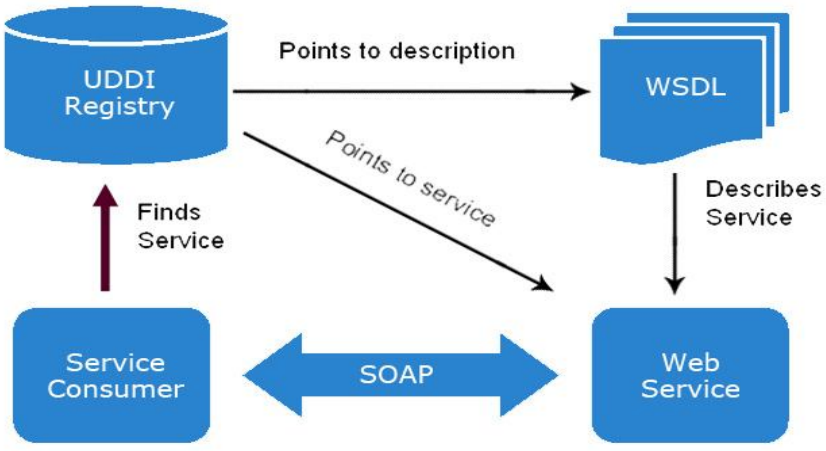


Fig. 1: Web Services & SOA

Web services allow applications to communicate with each other that enable them to work together by exchanging service messages back and forth. Even terminal access based applications can easily integrate into existing SOA-based environments. Web services and HATS offer enough room for developers to integrate the legacy terminal access based applications into the overall business processes. They are originally designed to minimize the total cost and time needed for systems collaboration and integration; which is the biggest IT expense of most companies.

To promote service orientation among legacy terminal access based applications, HATS provides full support for Application-to-Application integration via service-enabling of existing legacy assets. SOA Assets can be created from 5250 application business logic using HATS Web service tools. Macro recorder tool can be used to record business processes that will be later on exposed as Web services. HATS generates standard Web Service interfaces that can be easily integrated into SOA.

HATS provides a strong support for SOA by

- Building self-service transactions.
- Exposing host business processes as Web Services.
- Providing controlled access to vital host applications and host data.

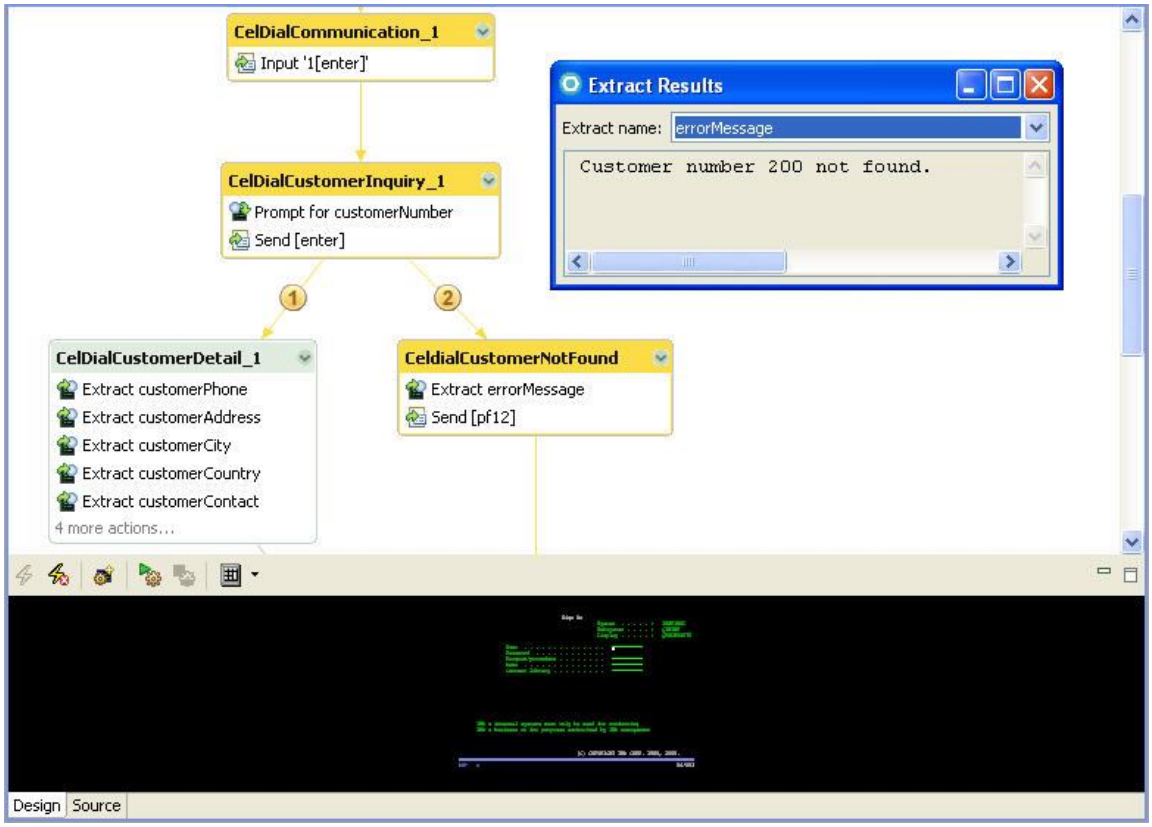
CREATING WEB SERVICES IN IBM-RATIONAL HATS

This section will show steps and tools available in IBM-Rational HATS for development of Web services that provide access to critical legacy green screen terminal-based application processes and data. These services can be easily published and hosted on WebSphere Application Server. Web services in IBM Rational HATS can be build using HATS macros and Integration Objects. Web services development and implementation in IBM Rational HATS comprises following steps.

I. Creating HATS Macros.
II. Creating HATS Integration Objects to drive Macros.
III. Creating the Web service Support Files.
IV. Creating the Web service.
V. Running & Testing Web service.

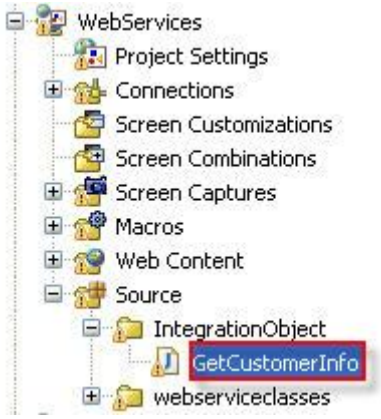
I) Creating HATS Macros

Visual Macro Editor provides a very user-friendly development environment to record new macros or to enhance the functionality of existing macros. During macro recording, every host screen is recognized using host screen recognition criteria. Prompt and Extract actions enables the macro to send and receive messages from host screens. Definition of alternate flows within self service transactions promotes the flexibility to handle multiple scenarios.



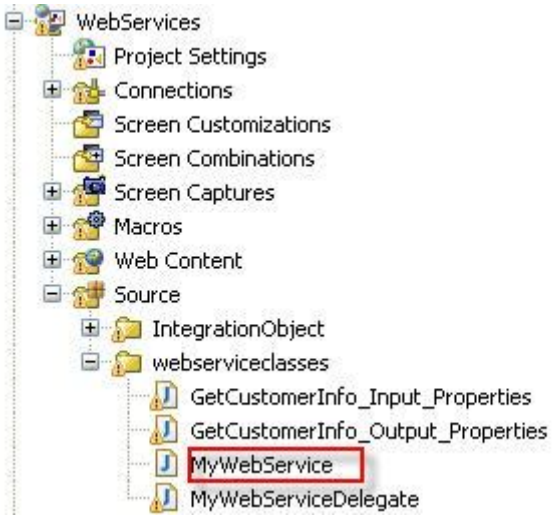
II) Creating HATS Integration Objects

Integration Objects are equivalent java codes i.e. JavaBeans for HATS macros that are used to execute the desired macros functionalities. They are Java classes that encapsulate programmed interactions with host application. Integration Objects can be used in multiple ways to integrate interaction with a host application into new Java or Web based programs. It is also used to provide the interaction with a host application for a Web service. HATS-Macro can easily transform into an HATS Integration Object.



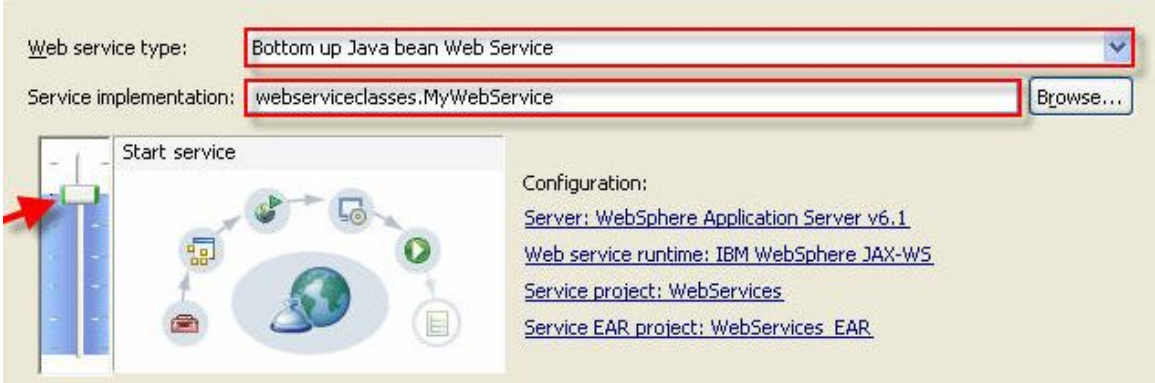
III) Creating Web service Support Files

Before creating the actual web service, HATS Integration Objects needs be transformed into their corresponding Web service Support Files. These Support Files are necessary for creating a Web service using IBM Rational Software Development Tools.



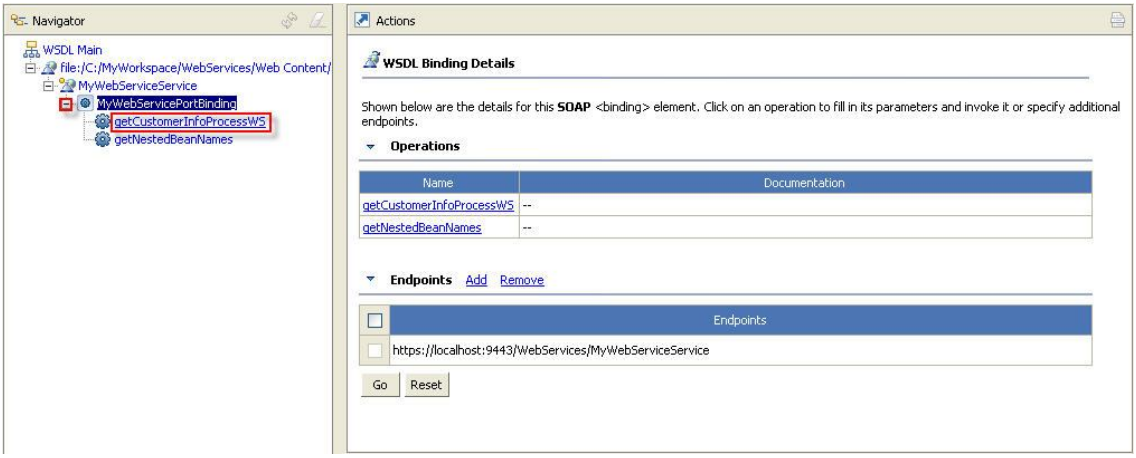
IV) Creating the Web service

In this phase, actual Web service including the WSDL (Web Services Description Language) will be generated with the help of Web Services support Files.



V) Testing Web service

IBM Rational Software Development platform provides the Web Services Explorer tool to test Web services.



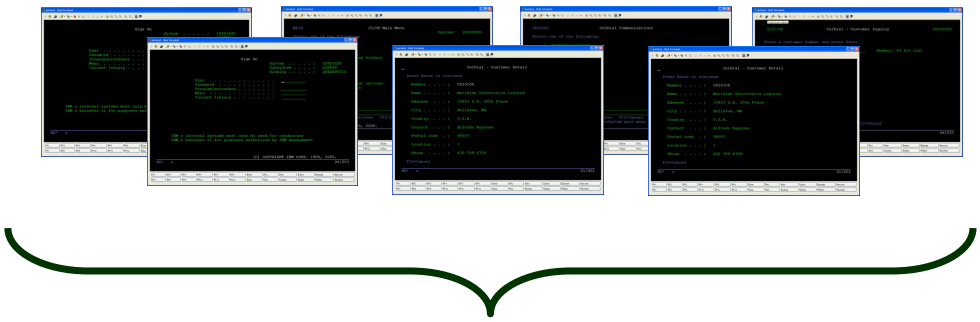
BUSINESS CASE STUDY

To promote “on-the-fly” integration among existing terminal based applications and to drive higher productivity and efficiencies; HATS simplifies the interaction of environment with host applications and provide a way for external parties to invoke business processes on terminal applications through Web services within or across the enterprise.

In this case study, a business process is required to extract Customer Information (i.e. Customer name, address, city, state, zip, phone, contact name, past order history, etc) from an iSeries terminal-based application on a given criteria. This business process should accept Customer number as an argument and should return corresponding detailed customer information. This business process will be exposed as a web service for external/internal service consumers.

To achieve this target, a macro was created to span over multiple host screens. During screen navigation, it will sign-on to a remote host application and will pass a message [Customer number] to a specific host screen. On the basis of that “Customer number”, required detailed information will be extracted from the host screen/screens. After extracting the required information, it will automatically sign-off and will release the host terminal connection.

Prompt Actions in the macro body will enable the HATS macro to take input message at execution time, whereas Extract Action will enable the macro to extract information from multiple host screens and to return the output message to service requestor as shown in following figure.

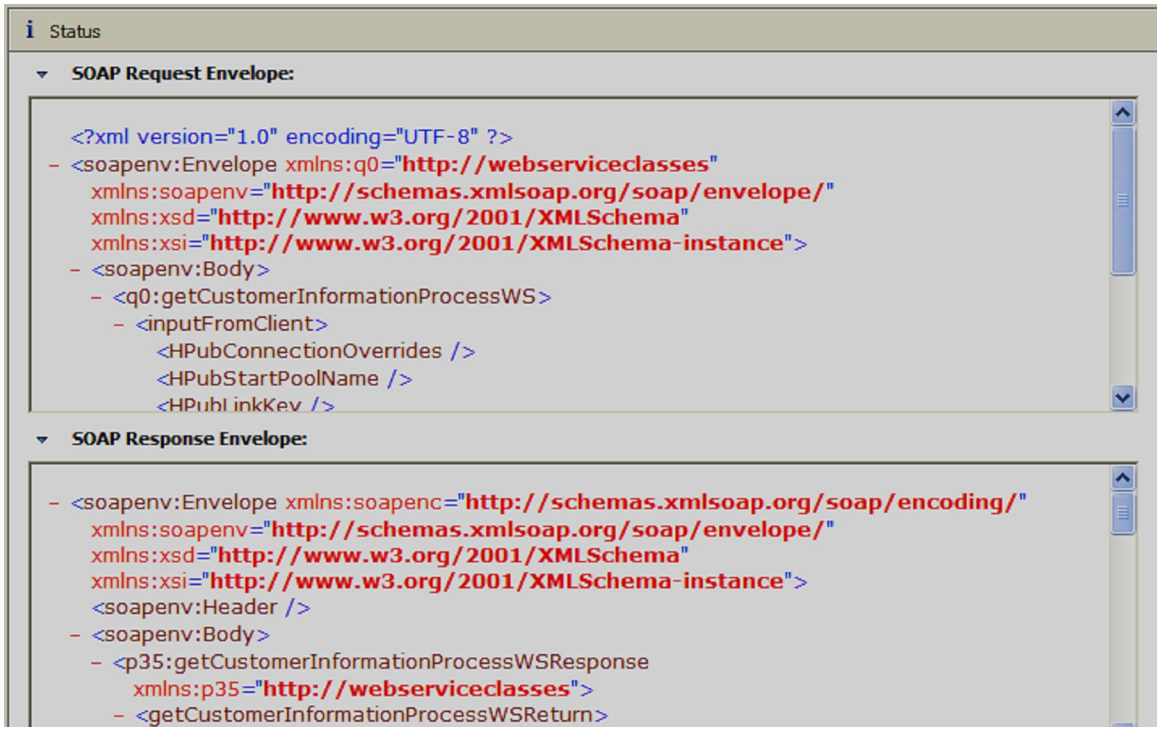


Input: Customer number

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

Now, this macro will be transformed into an HATS Integration Object. Corresponding Web services Support files will also be generated to aid the creation of Web service along its WSDL file. This web service will be published and hosted on WebSphere Application Server to be reused by business partners or internal users.

Sample SOAP Request Envelope for input message and SOAP Response Envelope for returning output message for this sample Web service are shown below.



In this business case study, it is proved that we can minimize the Application-to-Application integration cost by simply encapsulating the application business logic into HATS macros and exposing them to outside world in the form of Web services without making any modifications in underlying terminal-based host applications. These Web services can be orchestrated together to form large and complex business processes. These composite Web services would have standardized input/output SOAP interfaces that would promote interoperability among terminal based applications because they can be accessed from anywhere via internet.

Source: http://www.ibm.com

IBM Global Services
Route 100
Somers, NY 10589
U.S.A.
Produced in the United States of America
08-10