

Galen Public Webcast: Orion Rhapsody 201

Boston, MA

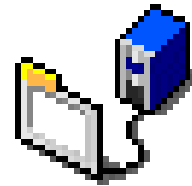
Outline

- **Communicating with Rhapsody**
- **Messages**
- **Lookup Tables**
- **Advanced Filters**
- **Message Collecting**
- **Message Tracking**
- **Error Handling**
- **Web Services**
- **Databases**

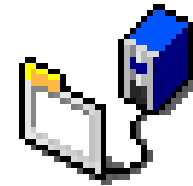
HTTP

- **HTTP Client**

- Input
- Output
- Out->In Mode



HTTP Client



HTTP Server

- **HTTP Server**

- Input
- Output
- Out->In Mode
- In->Out Mode

FTP and Database Communication Points

- **FTP and FTPS**

- Supports all communication modes



- **Database Communication Point**

- Input
- Output
- Out->In



- **Database Insertion Communication Point**

- Output Only
- optimized to insert or update records in a database



The Message Object

- **The message body-the content of the message as received on a communication point**
- **Properties-meta-data extracted during processing by the route pipeline, either explicitly or implicitly**
- **Other meta-data-other data about the message derived by the engine and maintained with the message object.**

The Message Object

- **The message object is persisted (saved) to the persistence store whenever meta-data or the body change**
- **Each message is persisted a minimum of three times**
 1. When the message is received by the Input communication point
 2. When the message is passed to a route queue
 3. When the message is passed from the route queue to an Output communication point.
- **By default, Rhapsody will assume that a message is encoded in UTF8.**
- **Rhapsody supports the full set of character encodings supported by Java**

Message Structure

- **Rhapsody makes no assumptions about the content of a message and is able to accept binary messages and messages containing text or structured text**
- **Implicit Message Structure**
 - XML
 - DOM
 - XSLT
- **Explicit Message Structure**
 - Symphonia message definition (.s3d)
 - XML Schema definition (.xsd)

Lookup Tables

- **Substitutes one code in a message with one or more other codes.**
- **Lookup tables can be accessed from the following filters:**
 - JavaScript
 - Message Modifier
 - Mapper
- **Managed by tools provided in the IDE and the WMC**
- **Entries can be added to a Lookup Table one of three ways:**
 1. By performing a bulk import from a CSV file
 2. By manually editing the Lookup Table
 3. By adding previously failed lookup attempts

Advanced Filters

- **Database Lookup**
 - used to look up values in a record from a table and to create or update property values and field values as required
- **Database Message Extraction**
 - replaces the body of the message with the records returned
 - locates specified data in an incoming message then generates an xml message containing matching information from a database
- **Generic Code Translation**
 - performs a database lookup to replace or populate the content of specified fields in a message

Advanced Filters

- **XSLT Stylesheet Filter**
 - applies an XSLT filter to an XML document transforming an XML Document
- **XML-FO Filters**
 - XML to PDF
 - XML to RTF
- **Zip/Unzip Filter**
 - zip separates messages into a single zip file, extracts individual messages from a zip file.
- **Batch/De-batch**
 - Batch - incoming messages are combined into a single file.
 - De-batch - an incoming file is split into its component messages.

Message Collecting

- **Provide the ability to manage sequences, groups and batches of messages input to a filter**
- **Message Sequencing**
 - Input Collation
 - Message Set
 - Message Set Trigger
- **Message Grouping**
 - Property
 - Size and/or Time
 - Timed Hold

Message Tracking

- **Used to ensure that an external system that has been interfaced to is processing messages correctly and that errors conditions are managed correctly**
- **Managed from within the Rhapsody IDE by creating and maintaining a Tracking Scheme**
- **Implicit Tracking**
 - Communications point mode
 - In->Out
 - Out->In
- **Explicit Tracking**
 - Use message tracking on the communication point
 - In->Out
 - Out->In
 - Bidirectional

Error Handling

- **Error Queue**

- a component of the Rhapsody Web Management Console where messages can be viewed and potentially edited before being resent back into the route for continued processing.



Send message to WS

- **Error Connector**

- selectively route errors occurring on different components of the route



- **Route Error Handler**

- capture all errors and route them together to the same location for further scrutiny.

Web Services

Web Services Overview

- **Rhapsody provides an easy-to-use interface for harnessing Web services, Independent of**
 - Whether the Web service is hosted by Rhapsody
 - Whether Rhapsody uses an external Web service.
- **Vendors can exploit Rhapsody's easy-to-use, secure Web services for application integration**
- **Providers can develop SOA strategies to simplify the control and management of their networks**

Web services

- **Provides support for SOAP / WSDL web services and provides a limited set of REST-based status operations through the Web API**
- **Web Services are created and managed utilizing the Web Services Management dialog in the IDE environment**
- **Activity and state of web services may be monitored utilizing the Monitoring / Web Services item in the Web Management Console.**



**Web Service
Client**



**Web Service
Hosting**

Web Services in Rhapsody

- **A Web service definition contains the configuration information to host the Web service, including its name and namespace, addressing information, the operations and types, and any security.**
- **Rhapsody supports two methodologies for creating Web services**
 - Top-down
 - Bottom-up.

Top-down approach

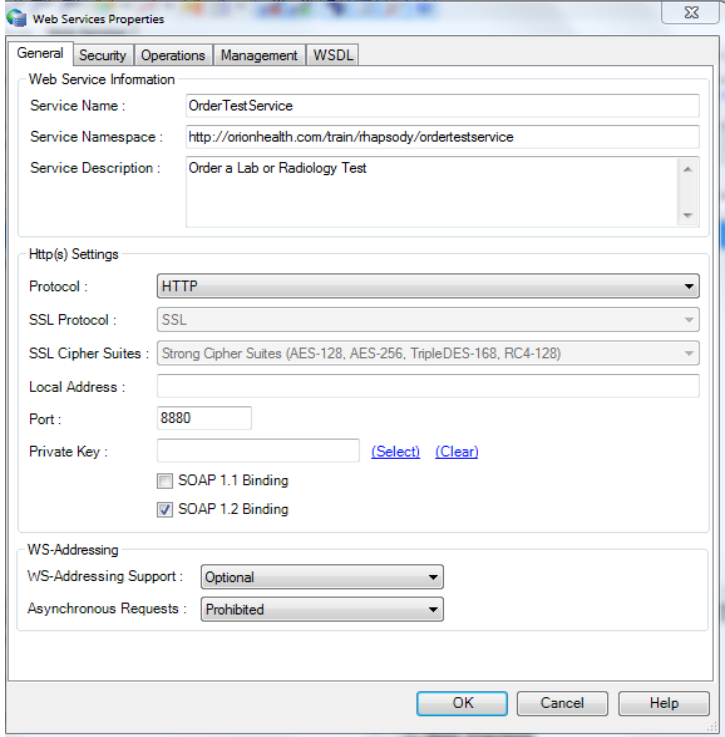
- **Uses an existing Web Services Description Language (WSDL).**
- **The WSDL is imported and is used to generate the Web service configuration. You can then modify the configuration, and your changes will be reflected in the WSDL that is hosted by Rhapsody.**

Bottom-Up approach

- **The bottom-up approach does not require a WSDL.**
- **Instead you create a new Web service, and configure all the settings manually.**
- **We recommend this approach when exposing an existing Rhapsody route as a Web service.**

Web services

- **Web services can be created by WSDL or by manually configuring the settings**
 - **General Settings**
 - **Operations**
 - **In-Only**
 - **Request-Response**
 - **WSDL**
 - **Monitoring Web servies**



Web Services Properties

General Security Operations Management WSDL

Web Service Information

Service Name : OrderTestService

Service Namespace : http://orionhealth.com/train/rhapsody/orderstestservice

Service Description : Order a Lab or Radiology Test

Http(s) Settings

Protocol : HTTP

SSL Protocol : SSL

SSL Cipher Suites : Strong Cipher Suites (AES-128, AES-256, TripleDES-168, RC4-128)

Local Address :

Port : 8880

Private Key : (Select) (Clear)

SOAP 1.1 Binding

SOAP 1.2 Binding

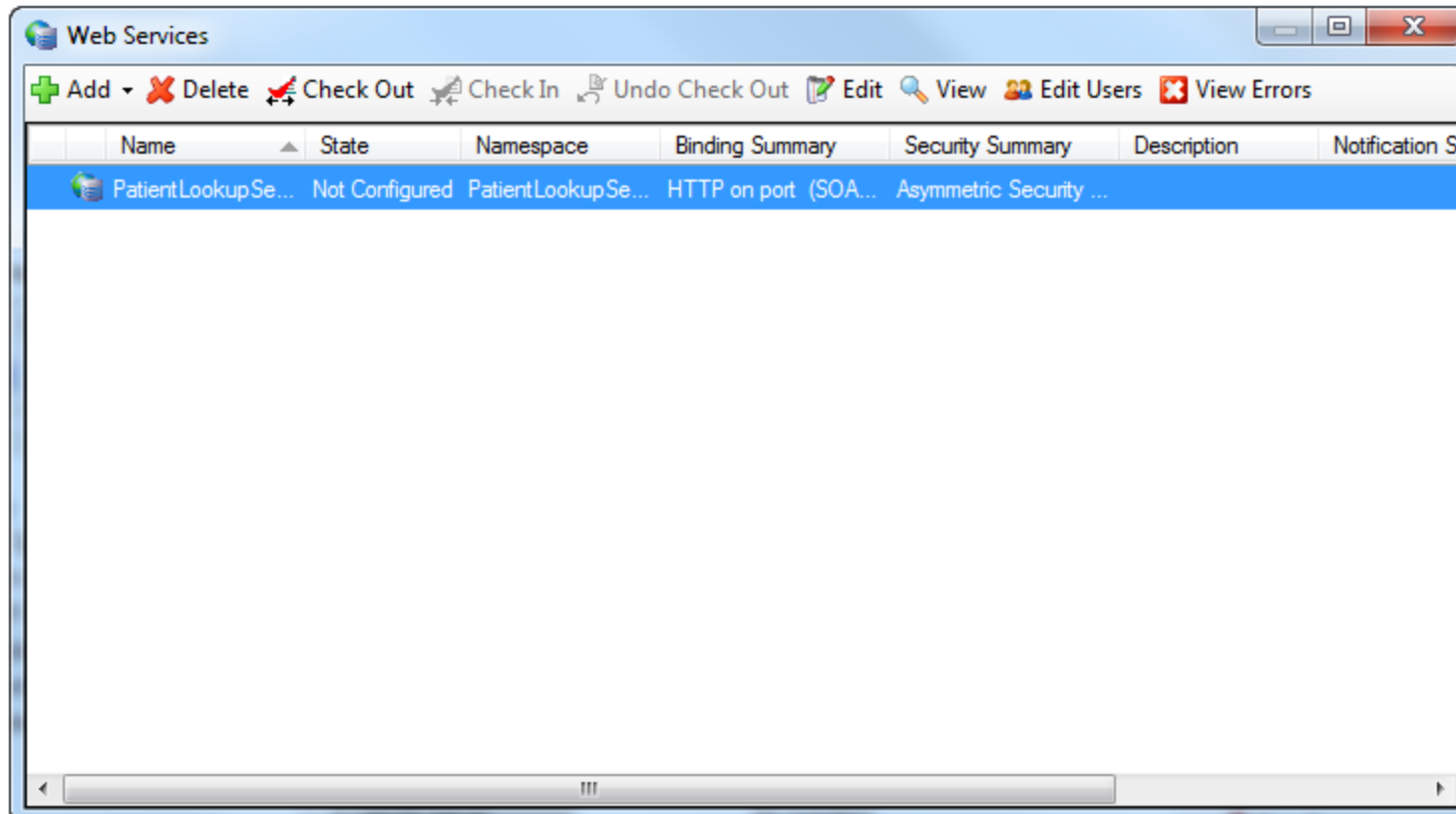
WS-Addressing

WS-Addressing Support : Optional

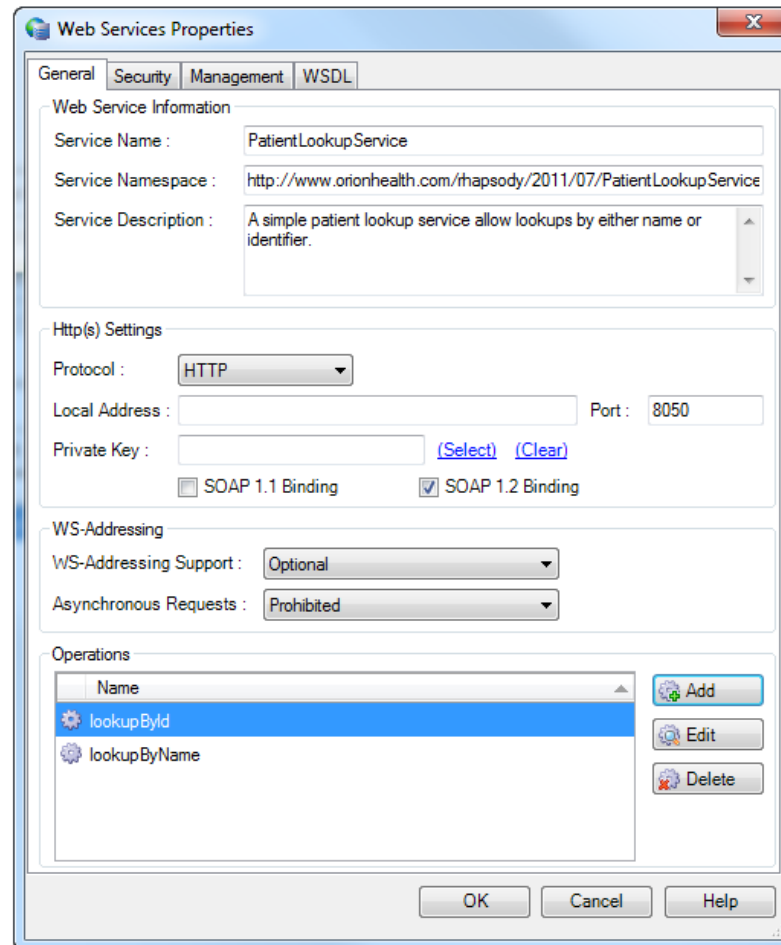
Asynchronous Requests : Prohibited

OK Cancel Help

Rhapsody Web Services Screen



Rhapsody Web Services Properties



Web Services Properties

General | Security | Management | WSDL

Web Service Information

Service Name : PatientLookupService

Service Namespace : http://www.orionhealth.com/rhapsody/2011/07/PatientLookupService

Service Description : A simple patient lookup service allow lookups by either name or identifier.

Http(s) Settings

Protocol : HTTP

Local Address : Port : 8050

Private Key : (Select) (Clear)

SOAP 1.1 Binding SOAP 1.2 Binding

WS-Addressing

WS-Addressing Support : Optional

Asynchronous Requests : Prohibited

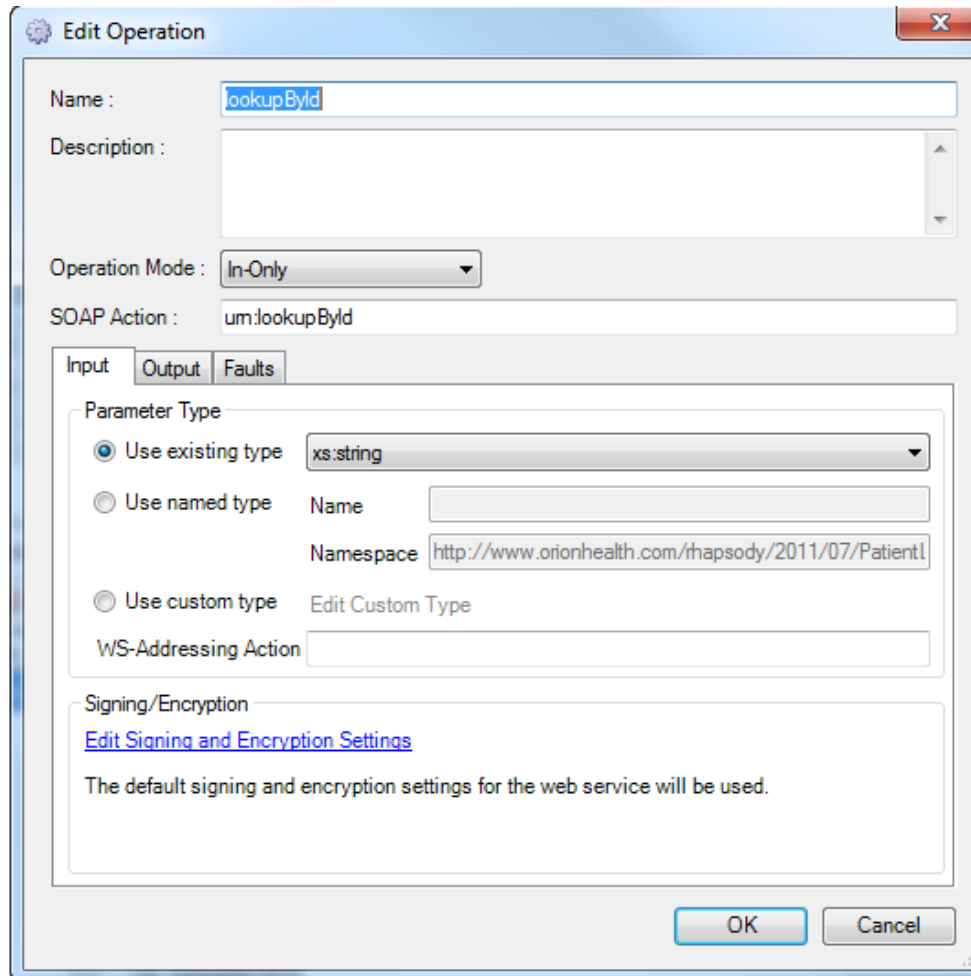
Operations

Name
lookupById
lookupByName

Add Edit Delete

OK Cancel Help

Edit Operation



Edit Operation

Name :

Description :

Operation Mode :

SOAP Action :

Input | Output | Faults

Parameter Type

Use existing type

Use named type Name Namespace

Use custom type Edit Custom Type

WS-Addressing Action

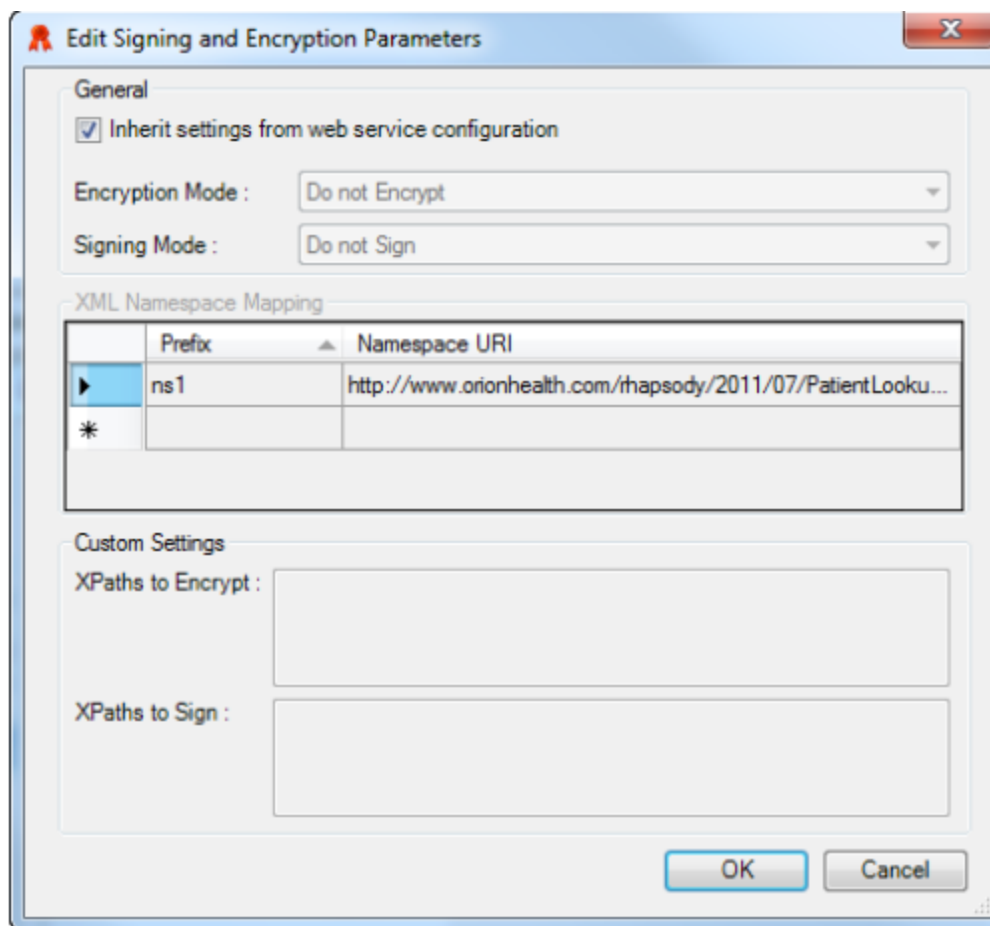
Signing/Encryption

[Edit Signing and Encryption Settings](#)

The default signing and encryption settings for the web service will be used.

OK Cancel

Edit Signing and Encryption Parameters



Edit Signing and Encryption Parameters

Inherit settings from web service configuration

Encryption Mode : Do not Encrypt

Signing Mode : Do not Sign

XML Namespace Mapping

	Prefix	Namespace URI
▶	ns1	http://www.orionhealth.com/rhapsody/2011/07/PatientLooku...
*		

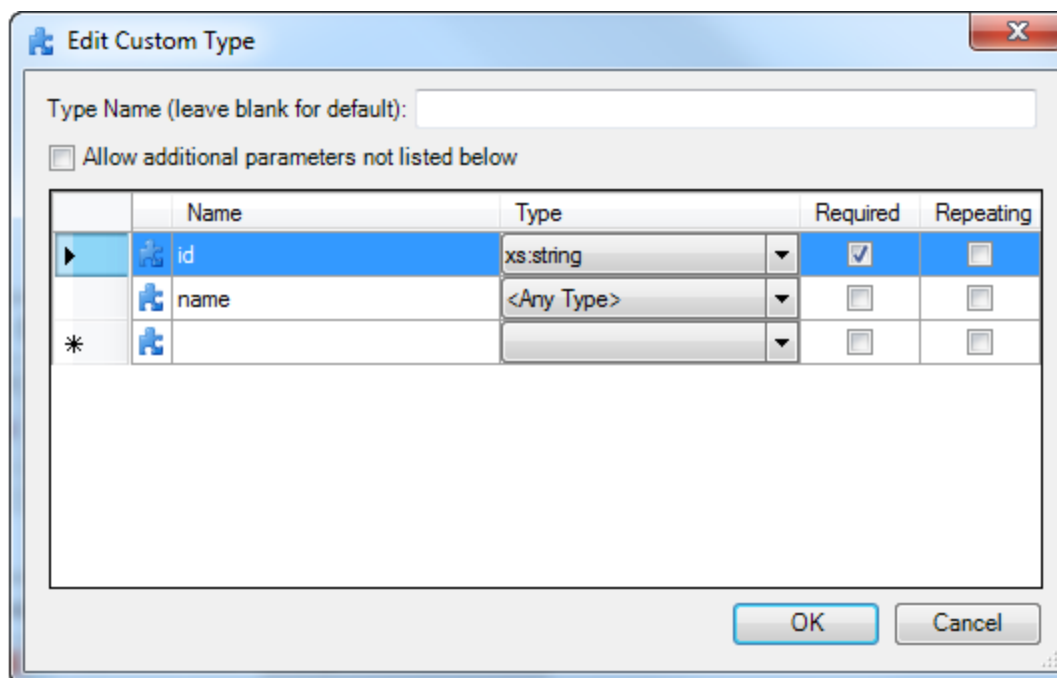
Custom Settings

XPaths to Encrypt :

XPaths to Sign :

OK Cancel

Edit Custom Type



Type Name (leave blank for default):

Allow additional parameters not listed below

	Name	Type	Required	Repeating
▶	id	xs:string	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	name	<Any Type>	<input type="checkbox"/>	<input type="checkbox"/>
*			<input type="checkbox"/>	<input type="checkbox"/>

OK Cancel

Web Services Demonstration

- **Route which takes in an HL7 message via TCP/IP and leverages a publicly available web-service to return the corresponding forecast for the ZipCode in the patient's address and output to a directory**

Databases

Database Support in Rhapsody

- **Communication points**
 - Database communication point
 - Input mode
 - Output mode
 - Out -> In
 - In -> Out
 - Database Insertion communication point
 - Output mode only
- **Filters**
 - Database Lookup filter
 - Message Extraction filter
 - Code Translation filter

Communication Point Input Mode


- **Designed to support rapid clearance of queued records when input query returns a single record each invocation**
 - If a query returns records, the call will be re-issued immediately
 - If no records are returned, the polling period delay is invoked
- **Ensure logic matches this requirement**
 - Retrieve only records input since the last time the table was polled (needs timestamp in input table)
 - Retrieve all records in the table and retrieve only those with a sequence number higher than the last record read

Using the Database Components

- **Configuring**

- Configuration tab of the communication point or filter dialog
- Database connection details specified either in the tab or the configuration file
- SQL statements and logic specified in configuration file
- Configuration file XML formatted, attached to Auxiliary Files tab

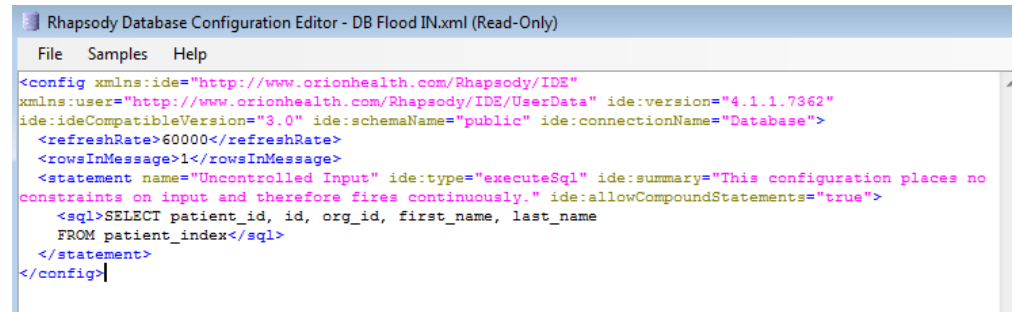
- **Managing the configuration file**

- Database Configuration Editor
- Manual creation 

The configuration file - structure

- **XML**

- One or more database elements
- Defined parameters (polling delay for input, message structure ...)
- One or more statement elements
 - Statements may be nested
- Each statement contains separate statement types for:
 - SQL
 - Stored procedures

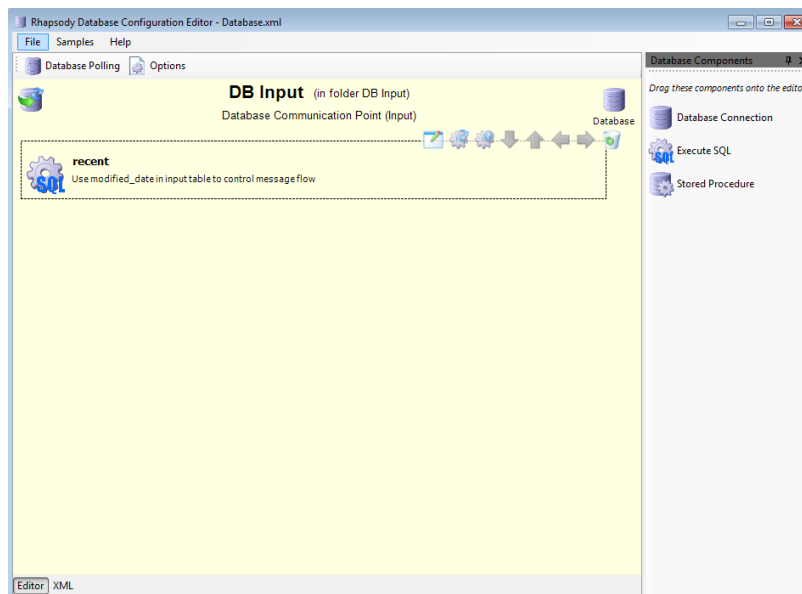


```
Rhapsody Database Configuration Editor - DB Flood IN.xml (Read-Only)
File Samples Help
<config xmlns:ide="http://www.orionhealth.com/Rhapsody/IDE"
xmlns:user="http://www.orionhealth.com/Rhapsody/IDE/UserData" ide:version="4.1.1.7362"
ide:ideCompatibleVersion="3.0" ide:schemaName="public" ide:connectionName="Database">
  <refreshRate>60000</refreshRate>
  <rowsInMessage>1</rowsInMessage>
  <statement name="Uncontrolled Input" ide:type="executeSql" ide:summary="This configuration places no
constraints on input and therefore fires continuously." ide:allowCompoundStatements="true">
    <sql>SELECT patient_id, id, org_id, first_name, last_name
FROM patient_index</sql>
  </statement>
</config>
```

The database configuration editor

- **Overview**

- External application invoked from the Configuration dialog
- Parameters from configuration tab and configuration file passed to the editor on open and passed back to the IDE on close
- Some ... idiosyncrasies ...



The database configuration editor cont.

- **Major characteristics**

- Two views of configuration – GUI / XML
- Drag and Drop in GUI mode
- Complete support for building the configuration except for
 - Defining Rhapsody Variables in connection detail
 - Defining more than one database connection
- Needs JDBC driver supplied if accessing a third party database

Connection details

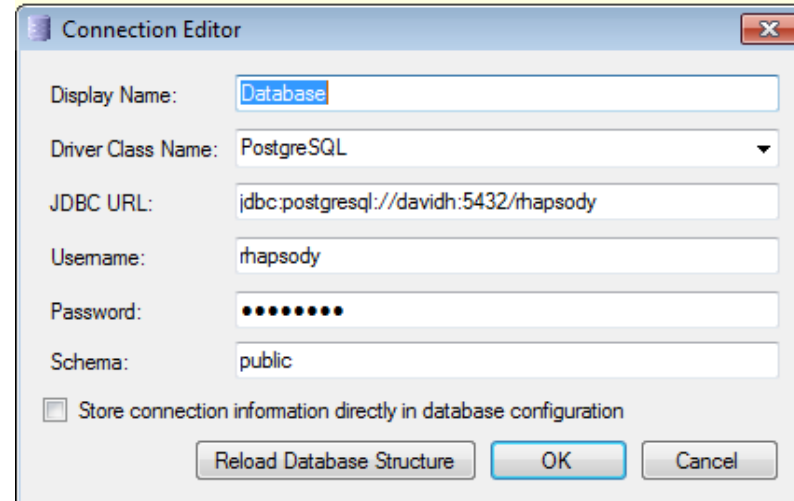
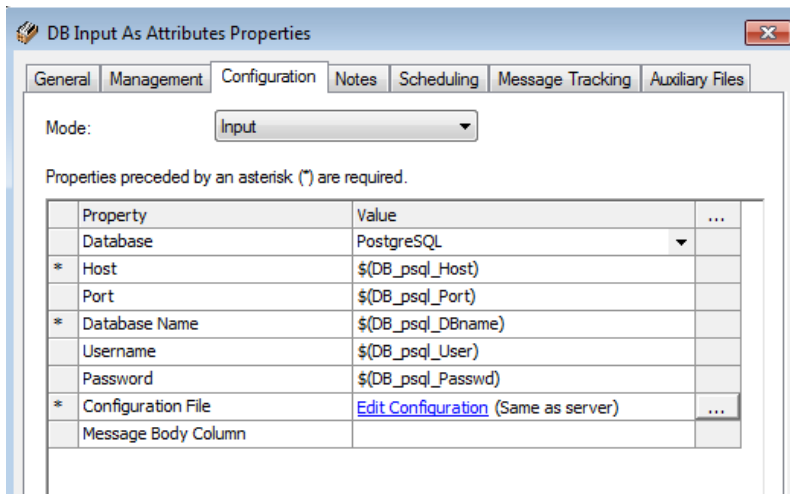
- **Requirements**

- Connection type
 - Oracle
 - SQL Server
 - My SQL (note, driver must be downloaded due to licensing)
 - MS Access
 - Third party databases require JDBC driver (and details in configuration file)
- Hostname of database server
- Database name
- Port to connect on
- Username to connect as
- Password

Connection details

- **Configuration Tab/Configuration File**

- Configuration Tab – more visible at a glance
- Configuration File – simpler, also test the connection once details are entered.



Connection details

- **Good Practice**

- Use Rhapsody Variables for consistency
- Protects details as the configuration is migrated between hosts
- Not recognized in the Configuration Editor (evaluated and passed in)
- Specify in Configuration Tab if using Rhapsody Variables

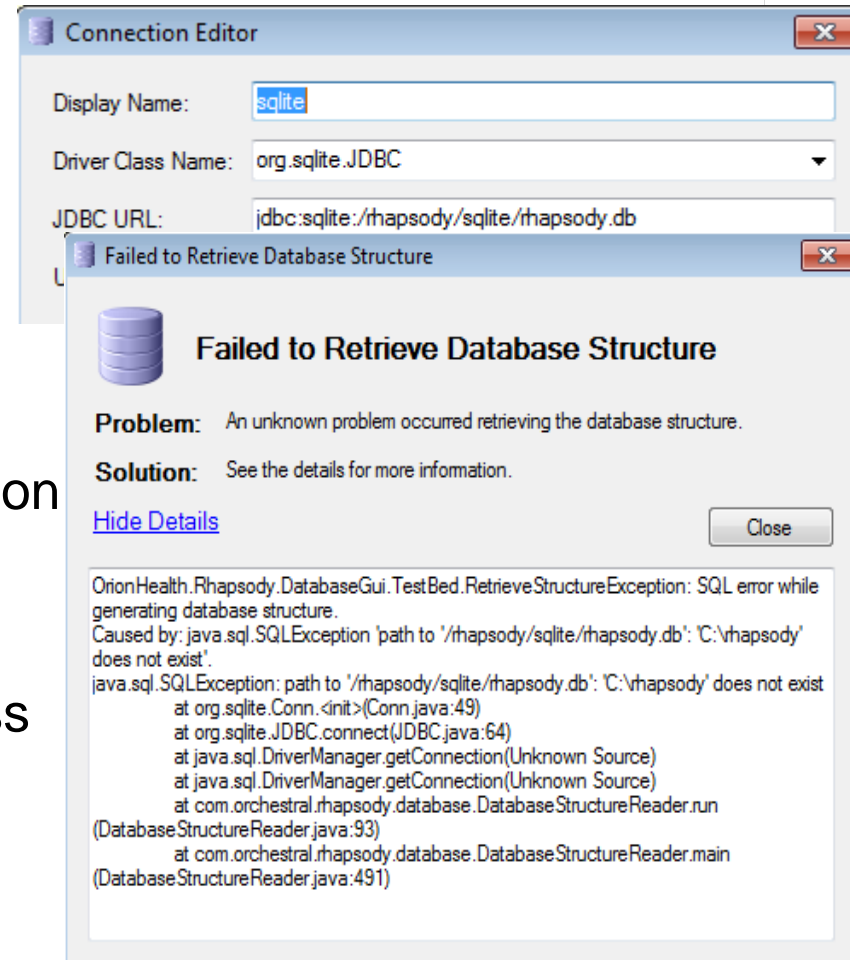
Connection details

• Connection visibility

- Database connection may not be network visible from IDE PC
- Filter testing in IDE occurs on engine
- Connectivity testing in the Configuration Editor occurs on client PC

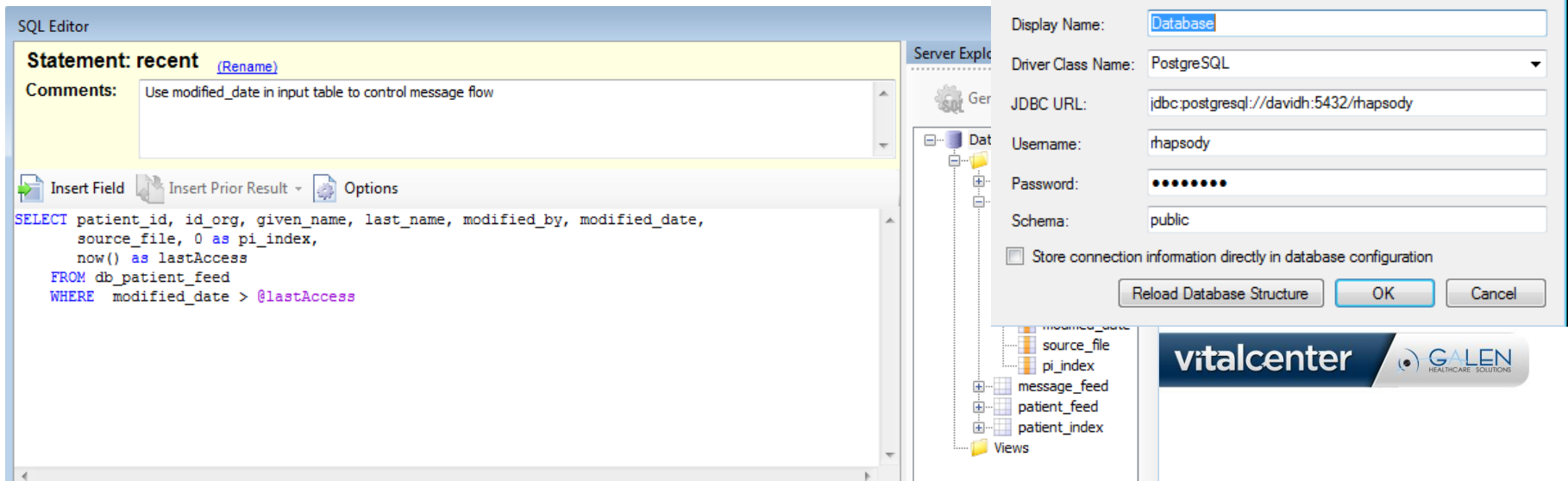
• Issues

- Host name resolution (use IP Address or add record to client PC hosts file)
- Firewall (use port forwarding on ssh session on server or attempt negotiation with site network security team)



Connection details

- **Testing in the configuration editor**
 - Validate Connection
 - Use table schema while constructing configuration
 - Supply input message definition and use while constructing configuration
- **Testing in the IDE**
 - Supply test input message and use filter testing before checking in configuration changes

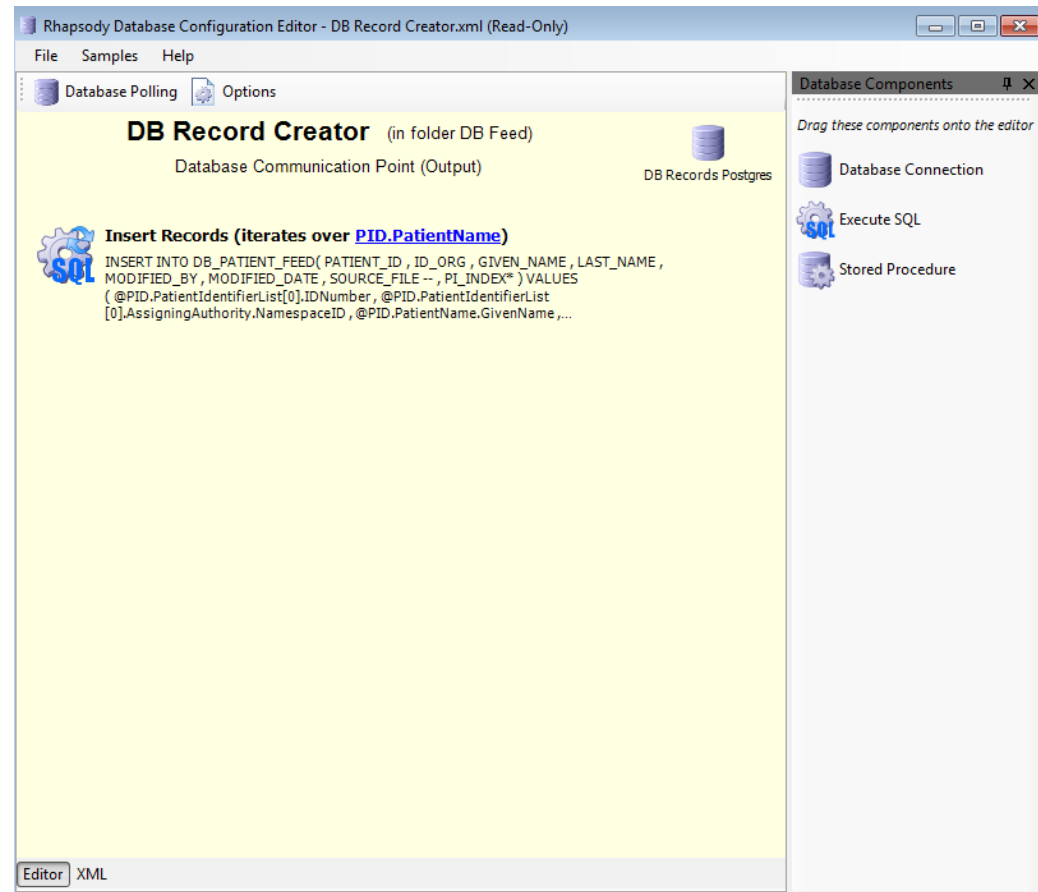


The screenshot displays two overlapping windows from a software application. The background window is the 'SQL Editor', showing a SQL query in a text area. The query is: `SELECT patient_id, id_org, given_name, last_name, modified_by, modified_date, source_file, 0 as pi_index, now() as lastAccess FROM db_patient_feed WHERE modified_date > @lastAccess`. Above the query, there is a 'Statement: recent' header and a 'Comments:' field containing the text 'Use modified_date in input table to control message flow'. The foreground window is the 'Connection Editor' dialog box. It contains several input fields: 'Display Name' (Database), 'Driver Class Name' (PostgreSQL), 'JDBC URL' (jdbc:postgresql://davidh:5432/rhapsody), 'Username' (rhapsody), 'Password' (masked with dots), and 'Schema' (public). There is a checkbox for 'Store connection information directly in database configuration' which is currently unchecked. At the bottom of the dialog are three buttons: 'Reload Database Structure', 'OK', and 'Cancel'. In the bottom right corner of the overall image, there is a 'vitalcenter' logo and the 'GALEN HEALTHCARE SOLUTIONS' logo.

Building a configuration

Purpose

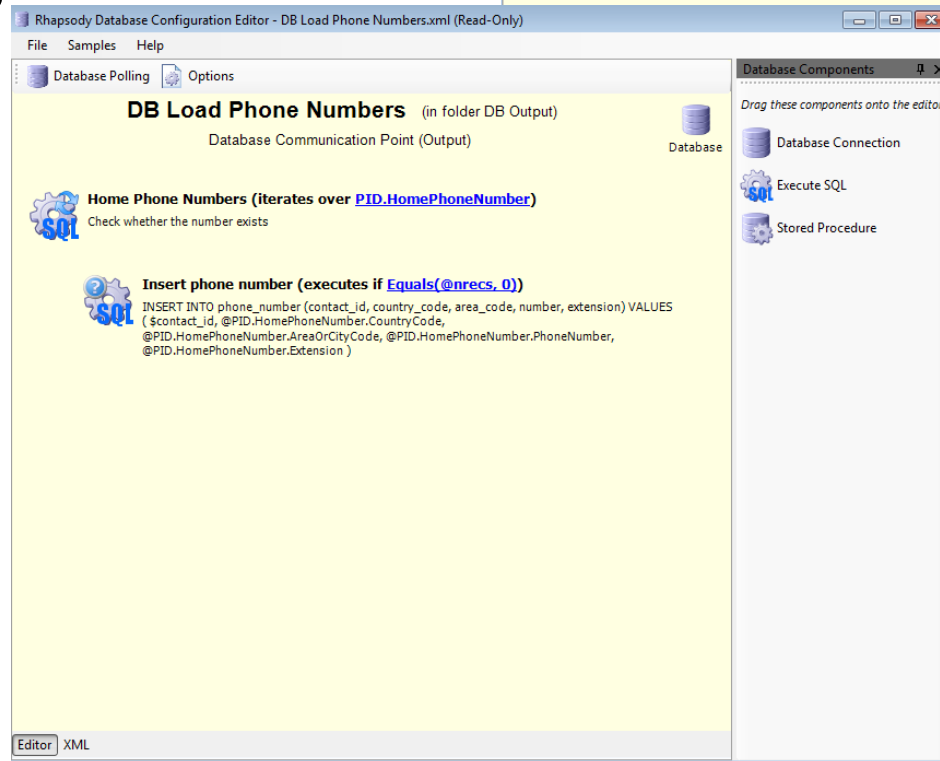
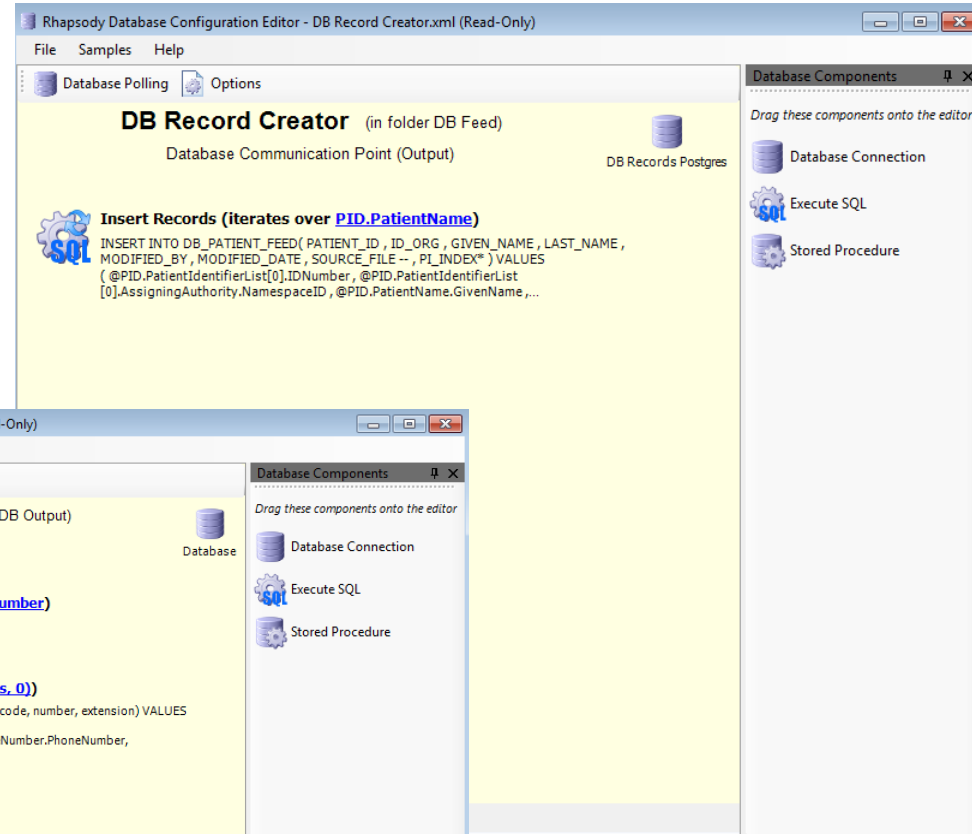
- Demonstrate the process of building a robust configuration
- Depends on:
 - Testing connection validity
 - Using schema and message definition
 - For filters, testing in the filter dialogue



Iteration & Logic

Purpose

- Handling iterations in the datastream
- Utilizing logic in statement execution



Database Demonstration

- **Route which takes in an HL7 message via TCP/IP, parses the message and writes the structured data components to the custom table in a SQL Server database.**

Top-down approach

- **Uses an existing Web Services Description Language (WSDL).**
- **The WSDL is imported and is used to generate the Web service configuration. You can then modify the configuration, and your changes will be reflected in the WSDL that is hosted by Rhapsody.**

Bottom-Up approach

- **The bottom-up approach does not require a WSDL.**
- **Instead you create a new Web service, and configure all the settings manually.**
- **We recommend this approach when exposing an existing Rhapsody route as a Web service.**

For the Galen take on industry news, EHR developments, interesting solutions we have developed, and new Galen products check out our blog...



The screenshot shows the Galen Healthcare Solutions blog interface. At the top, there is a navigation bar with the Galen logo, the tagline "embrace the new world of healthcare", and the word "blog". To the right of the navigation bar are links for "Wiki", "VitalCenter", and "Galen Healthcare Solutions".

The main content area features a blog post titled "The Three Types of Organizations that Need New EHRs". The post is attributed to Christopher Libby, dated February 18, 2013, and categorized under Business Intelligence, Data Conversion, Electronic Health Record, General, Healthcare IT, Implementations, Industry News, Meaningful Use, and Uncategorized Edit.

The post text discusses the era of electronic health records and the opportunities for innovative uses of data. It mentions that organizations that best position their data to help providers deliver patient-centered care will flourish. It also notes that physician organizations are becoming larger through growth, acquisitions, and mergers. These growth milestones provide organizations with opportunities to reflect on the capabilities of their current medical record system in order to decide if their current system will keep them competitive in the future. The post concludes by stating that these are the three types of organizations that will be purchasing a new Electronic Health Record (EHR) system in 2013.

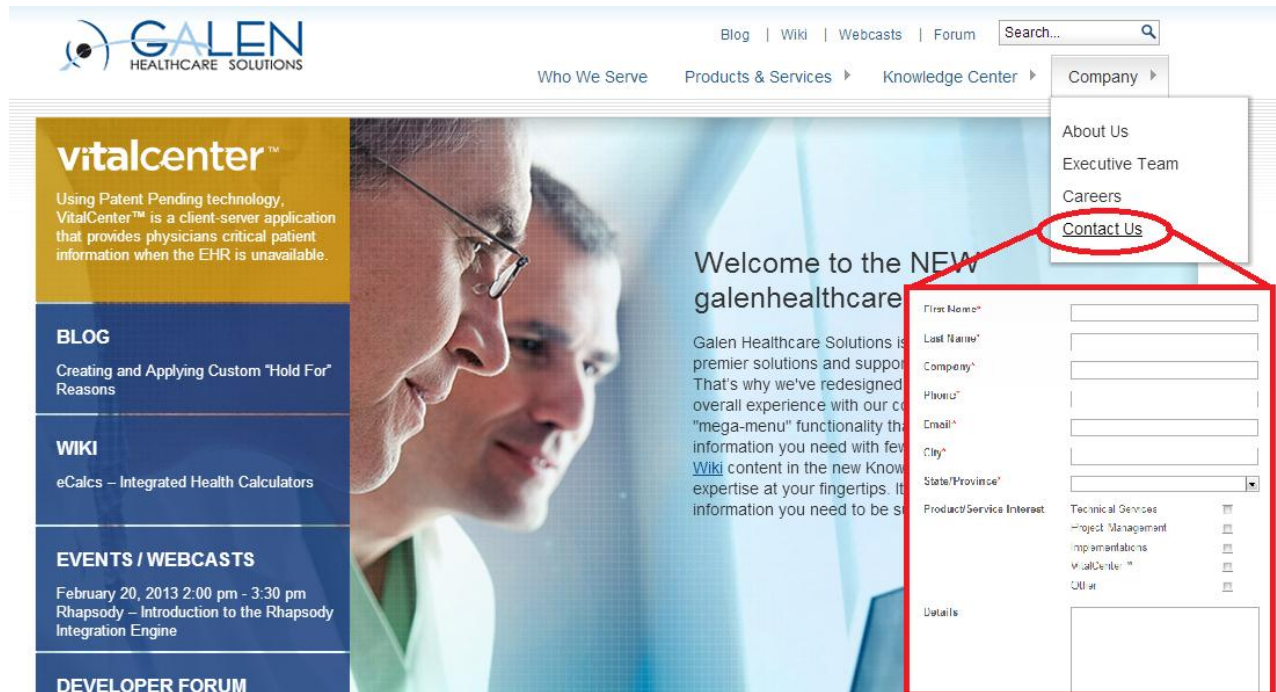
Below the text is a section titled "The New EHR Adopter" which includes a pie chart and a text block. The pie chart shows that 38% of Unregistered Professionals and 43% of Medicare Registrants are the new EHR adopters. The text block explains that buying a new EHR is not cheap, which explains why many smaller physician practices have held out on the investment as long as they could. It states that only around 20% of providers are attesting to Meaningful Use in the US. However, government mandates and incentives are forcing providers to overcome their resistance to EHRs. While larger practices initially lead the charge into the digital...

On the right side of the page, there is a "Sign up" section with an email input field and a "Submit" button. Below that is a "Search for:" section with a search input field and a "Search" button. At the bottom right, there is a "Tags" section with a list of tags including "Adoption", "Allscripts", "Allscripts Consultants", "Allscripts Conversion", "Allscripts Enterprise EHR", "Allscripts PM Business Continuity", "Business Intelligence Clinical", "Conversion", "ConnectR", "Crystal Reports", and "Custom Reports".

blog.galenhealthcare.com

Thank you for joining us today, for additional assistance....

You can contact us through our **NEW** website at www.galenhealthcare.com



The screenshot shows the Galen Healthcare Solutions website. The top navigation bar includes links for Blog, Wiki, Webcasts, Forum, and a search box. A dropdown menu is open under 'Company', with 'Contact Us' highlighted by a red circle. Below the navigation, there is a banner for 'vitalcenter' with a description: 'Using Patent Pending technology, VitalCenter™ is a client-server application that provides physicians critical patient information when the EHR is unavailable.' To the left of the banner are four blue boxes: 'BLOG' (Creating and Applying Custom "Hold For" Reasons), 'WIKI' (eCalcs – Integrated Health Calculators), 'EVENTS / WEBCASTS' (February 20, 2013 2:00 pm - 3:30 pm Rhapsody – Introduction to the Rhapsody Integration Engine), and 'DEVELOPER FORUM'. The main content area features a large image of two men looking at a screen, with the text 'Welcome to the NEW galenhealthcare.com'. Below this is a contact form with fields for First Name, Last Name, Company, Phone, Email, City, and State/Province. There are also checkboxes for 'Product/Service Interest' (Technical Services, Project Management, Implementations, VitalCenter™, Other) and a 'Details' section.