

Rhapsody Interface Management and Administration

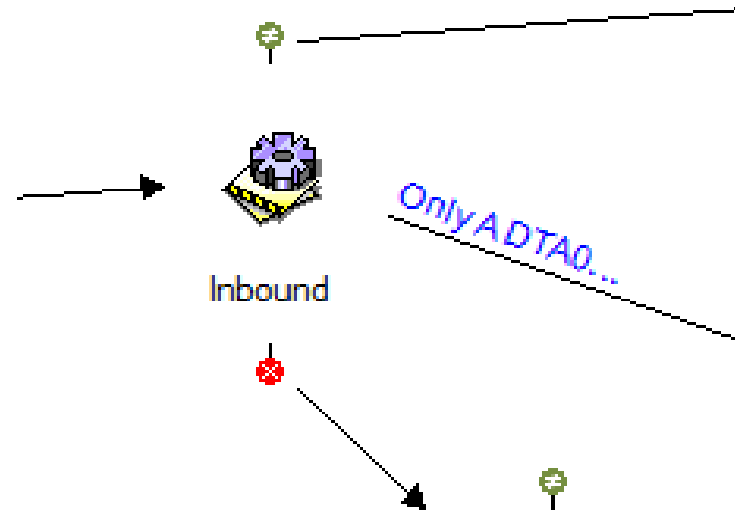
Presented by: Jamie Titak

Welcome

- The Rhapsody Framework
 - Rhapsody Processing Model
 - Application and persistence store files
- Web Management Console
 - Backups
 - Route, communication and message detail pages
- Monitoring and Management
- Engine Management, Best Practices and Common Performance Issues
- Users and Groups
- Notifications and Watch Lists
- Q&A – Chat and Live

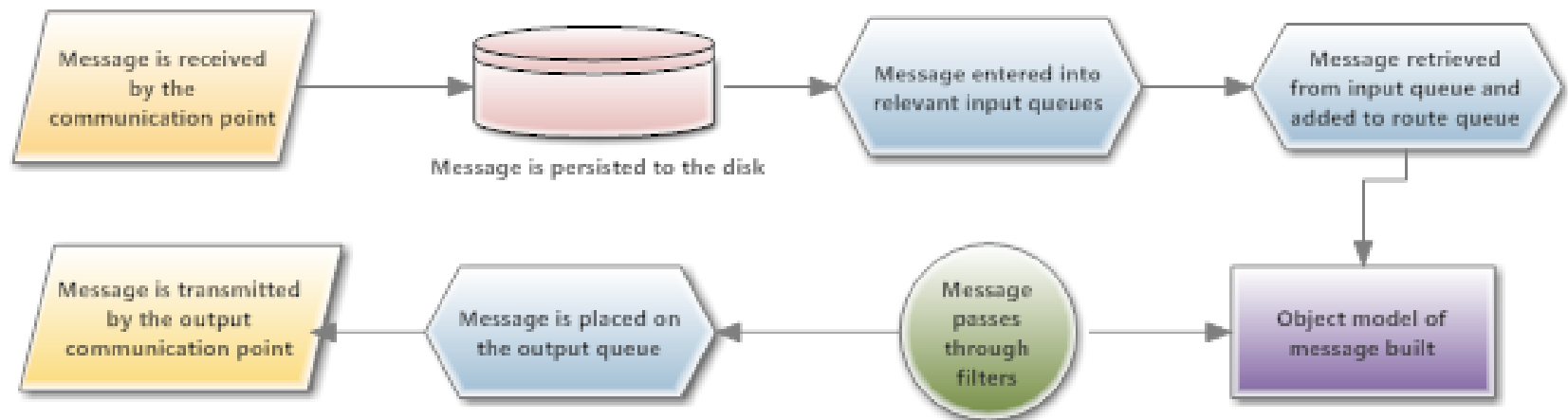
Rhapsody Processing Model

- Communication points
 - Each communication point is optimized to support
- Routes
 - Filters
 - Input/Output Connectors
 - No match connector
 - Error connector
 - Connectors
 - Standard Connector
 - Conditional Connector
 - Javascript Connector



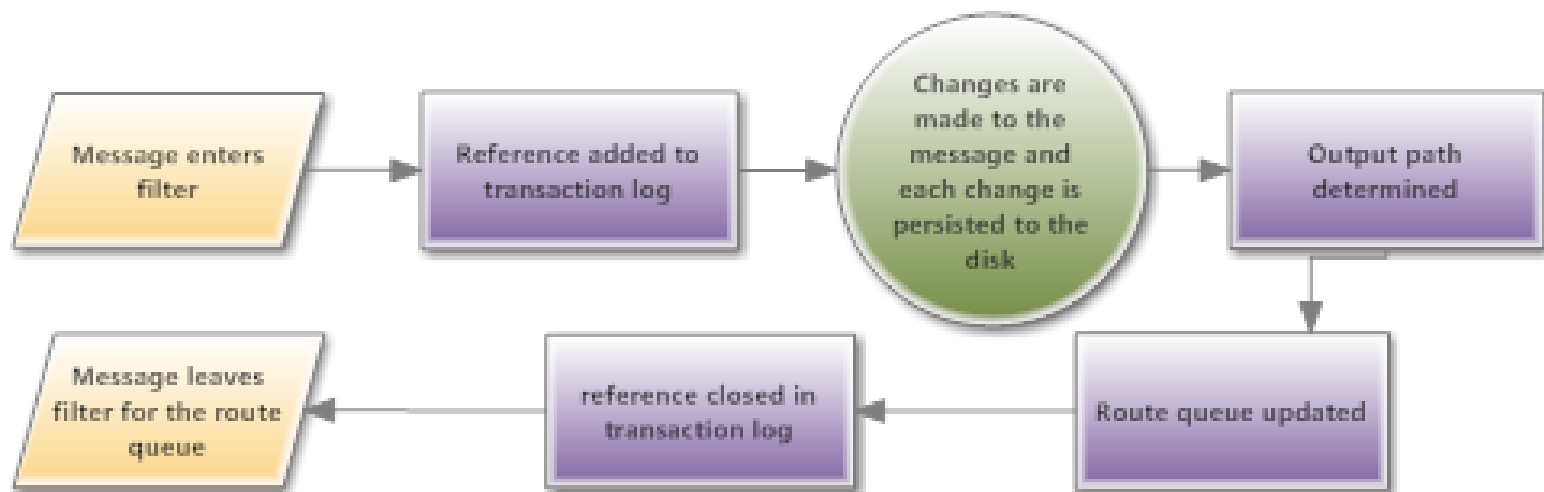
Rhapsody Message Pathway

1. Message is received by the communication point
2. Message is persisted to the disk
3. Message entered into relevant input queues
4. Reference retrieved from input queue by the route and added to the route queue
5. Object model of message built
6. Message passes through filters
7. Message is placed into the output queue(s) for the output communication point(s) to transmit.



Filter Processing

1. Message enters filter
2. Reference added to transaction log
3. Changes made to message properties are persisted to disk
4. Filter Processing complete
5. Output path/s determined
6. Next processing component/s identified and route queue updated
7. Reference closed in transaction log



Application File Layout

- **bin:** contains the Rhapsody startup files.
- **equinox:** contains support libraries and logs for those libraries.
- **jre:** contains the Java Runtime binaries that are used in the specific version of Rhapsody
- **licenses:** contains third party license detail for libraries used within Rhapsody.
- **log:** contains the text log files for the service wrapper and for Rhapsody.
- **plugins:** contains the Rhapsody Java library files.
- **rhapsody:** contains the run-time configuration file for Rhapsody as well as the engine license and log configuration; also contains the default location for the Rhapsody persistence store (data/).
- **run:** process details for a running engine.
- **tools:** support tools which may be required as standalone applications.

The Persistence Store

- The persistence (or data) store is a single repository with two primary purposes:
 1. As a repository for active messages and their associated meta-data.
 2. As a repository for historical messages
- The persistence store does not differentiate between active messages and messages for which processing has been completed
- The size of the persistence store is managed to constrain the size of the archive
- Messages older than the limit of the archiving period are normally backed up to system storage devices.

Archive Cleanup Process

- Once message processing is complete and no updates are written to the store the messages are available for historical analysis
- The persistence store can grow at a rate of 30-50 GB per day in high load environments
- The archive cleanup service can be configured to ensure that disk limits are not exceeded

Archive Cleanup

Automatic Cleanup

Keep archive data for * Hours Days

Keep logs for * days

☐ Cleanup notification event history

Cleanup Hourly Daily Weekly Monthly Advanced

Every * hours

At minutes past the hour

[Apply](#) [Discard Changes](#)

Cleanup Status

Available Disk Space	24.3 GB
Automatic Cleanup Status	Last cleanup finished at 24 September 2013 14:00:00. The cleanup took 0 secs and removed 0 files (0 MB).
Manual Cleanup Status	Never run since last Rhapsody startup.
Cleanup Now	

Web Management Console

- Monitoring and control tool for the Rhapsody engine.
- https://<server_name>:8444
- http://<server_name>:8081
- Landing page of the Web Management Console is the Engine Monitor page under the Monitoring menu
- Engine Monitor provides a snapshot of the Engine Uptime, total Messages Processed and message Throughput for the Rhapsody engine.

Backups

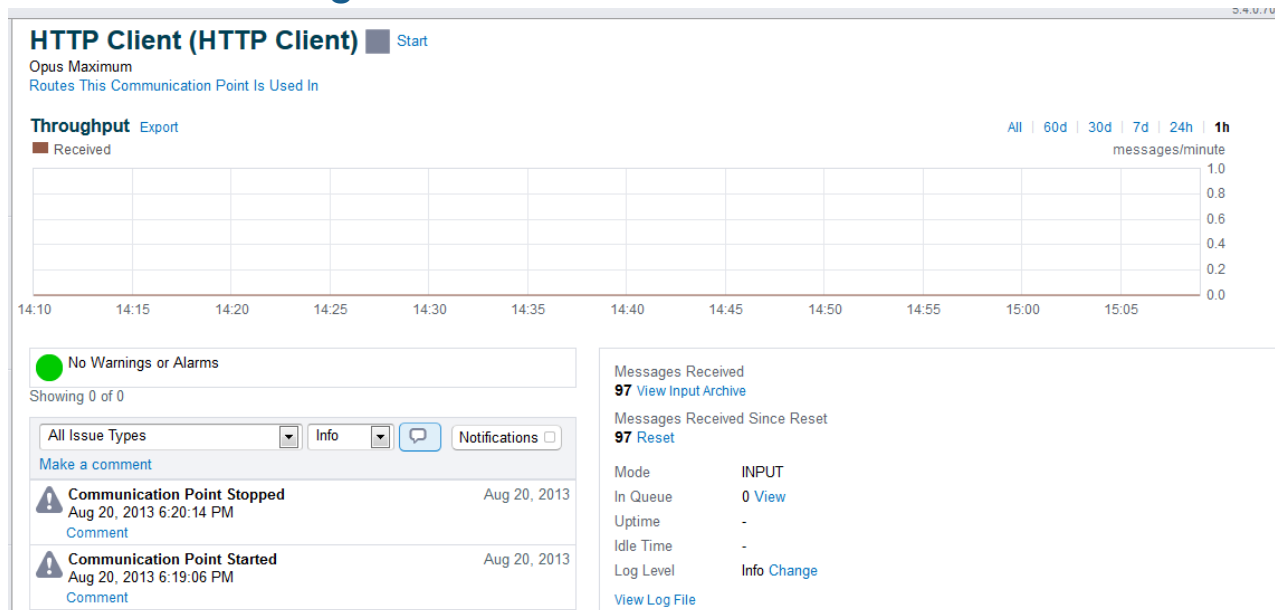
- Backups can be scheduled on a full or incremental cycle
- Forensic restores allow a non-production engine for analysis purposes
- Backup schedules may be defined for
 - Administration components: notably the user store and schedules for backup and cleanup
 - Configuration: the engine configuration including watch lists, notifications and tracking schemes
 - Message store including the active messaging and logs
 - All: the complete engine state
- **Good practice model-** weekly full back up followed by nightly incremental backups

Route Details

- Provides a summary view of all routes in the system, including their current state, traffic and processing load.
- Waiting column provides a numerical indicator of any messages queuing on the route
- Currently Processing column is an indicator of messages that have not yet completed processing
- Route Details - lists all Input and Output communication points associated with this route, along with any filters that may be present
- Ability to view throughput of route
- Warnings or Alarms - lists potential problems and issues

Communication Point Details

- Top section identifies whether the communication point is running or not and includes a link to changing this status
- **Warnings or Alarms** - lists potential problems and issues
- Ability to view Notification Settings
- Link to view messages



View Messages

- The message view has four sections
 1. **Message Events** - lists the path which the message has followed as it traverses the engine
 2. **Event Tree**-allow for navigating within the Message Events tree
 3. **Message Properties** - displays a read-only list of the properties (meta data) associated with the message at each stage of its processing
 4. **Message Body** - displays the content of the message
- The download link at the bottom left of the panel saves the message locally
 - These messages can then be viewed in another application such as HL7 spy

Handling Message Errors

- Unhandled processing errors result in the associated messages being placed onto the Error Queue
- Messages listed on the Error Queue can be selected for further processing
 - **Reinject** - The message is reinjected into the engine at its original point of entry
 - **Edit**-allows you to edit the message before redirection
 - **Redirect** - The message is redirected to a selected filter or communication point.
 - **Reprocess** - The message is sent back to the filter on which the error originally occurred
 - **Delete** - The message is removed (deleted) from the queue and no additional processing is performed on the message
 - **Download** - The message is downloaded as a compressed file

Message Searching

- Web Management Console provides the ability to search for messages held in the persistence store, including messages which are still active and have not yet completed processing
- The Find Messages section of the WMC menu provides a number of templates for search operations and also provides for re-use of common query structures:
 - Input Search: standard template for locating messages by the input communication point
 - Output Search: standard template for locating messages which have completed processing
 - Route Search: standard template for locating messages by route
 - Saved Searches: a search may be saved for re-use
 - Recent Results: presents a list of the 10 most recent searches

Monitoring Engine Health

- A well managed Rhapsody instance has high reliability and is not likely to require system attention. Continuous uptimes in excess of 400 days is not unusual
- Traits of a healthy engine system
 - Is configured defensively to ensure that processing exceptions are rapidly noted and acted upon
 - Utilizes memory effectively and has sufficient spare disk capacity to comfortably manage the work load
 - Ensures that information placed in log files efficiently reports issues
 - Maintains small to non-existent queues
 - The volume of logging is sufficiently low so that significant issues may be readily identified rather than being difficult to locate due to the volume of logging

Common Performance Issues

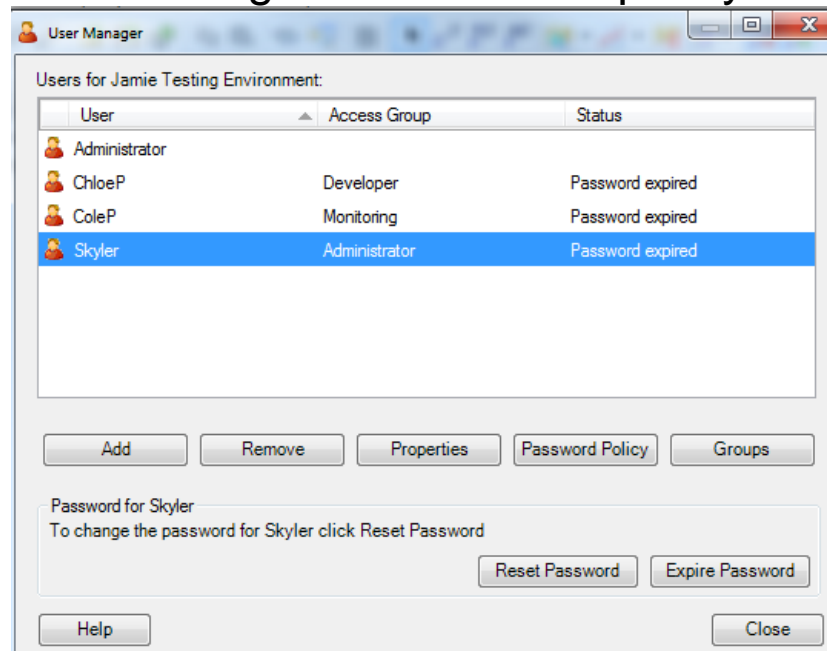
- “Random” restarts
 - Examination of the logs normally reveals numerous communication point failures prior to the restart are present in the System log
- Startup failure
 - Engine already running
 - Bind issues
 - Data Store corruption
- Slow processing
 - Abnormally heavy message load, resulting in queuing
 - Additional activity on the host
 - Insufficient host memory resulting Blocking route threads due to slow filter processing

Log Levels

- The operation of the Rhapsody environment is continually monitored by the logging components and events are reported as they occur
- Five different log level
 - FATAL: report only fatal (very severe) events
 - ERROR: report only error and fatal events
 - WARN: report warnings, errors and fatal events
 - INFO: include information events in the logs as well
 - TRACE: provide trace level logging as supported by the underlying code
 - DEBUG: provide more detailed information on the events normally useful for application debugging

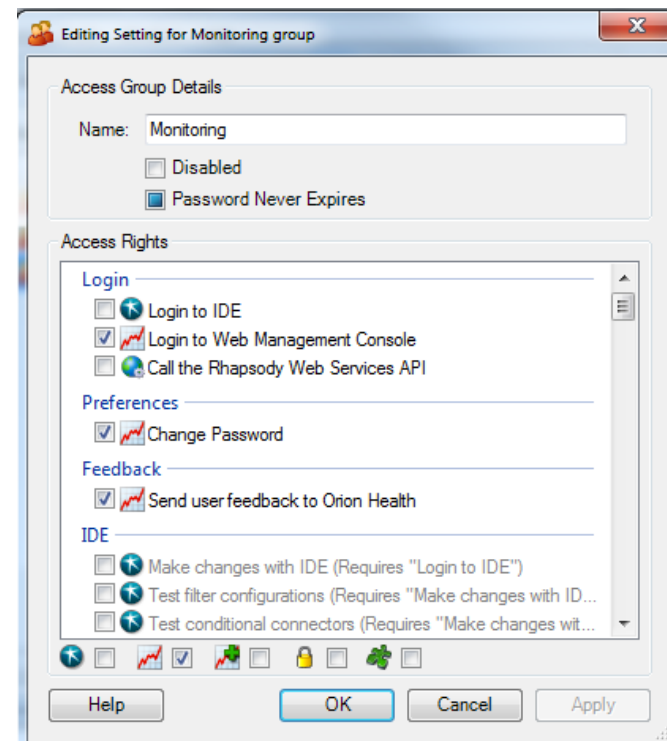
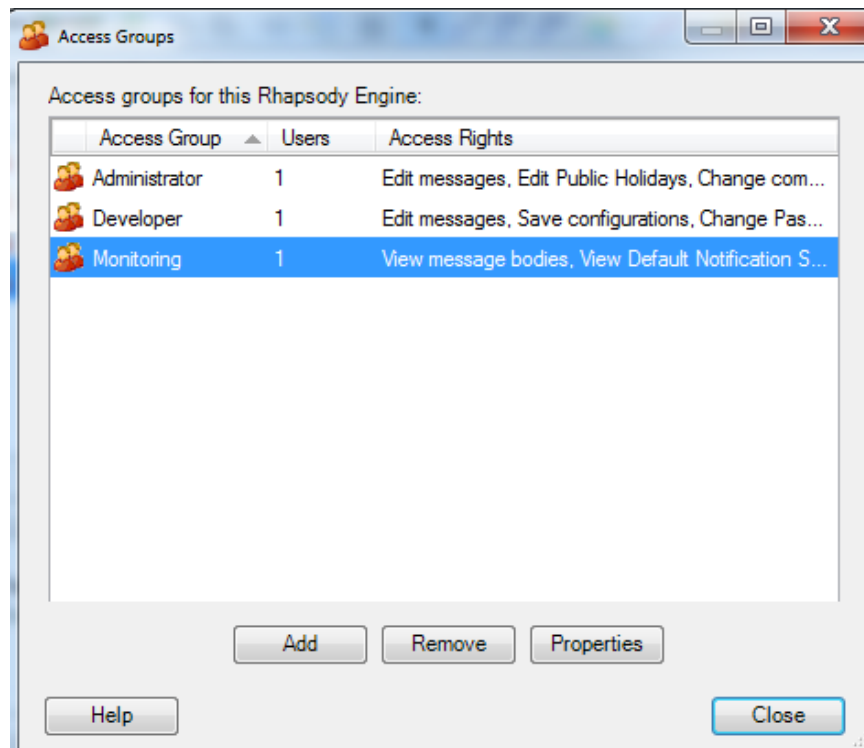
Users

- Assigning users provides many benefits for organizations
 - Permits the creation of user accounts with limited access to Rhapsody
 - The ability who has a Rhapsody component checked out
 - The ability to tailor the way each user receives notifications from the Rhapsody elements they have been subscribed to
 - Users and Groups are managed from the Rhapsody IDE via the User Manager



Groups

- The list of Access Groups is accessible by clicking the Groups button on the User Manager
- An administrator is able to add a new, modify an existing scheme or delete an access group when no longer required



Tasks and Responsibilities

- Maintaining and managing a Rhapsody site generally requires three groups of activities.
 - Site Administration tasks: responsible for the server and integration with site networks
 - Analyst / Developer: responsible for building and managing the configuration and ensuring the correct operation of interfaces
 - Monitoring / support: responsible for monitoring the state of the system and first point of contact for issues

Access Schemes

- Login to IDE - provides read access to the Rhapsody Integrated Development Environment (IDE).
- Login to Web Management Console - provides read access to the Management Console.
- Call the Rhapsody Web Services API - can call operations in the Rhapsody Web Services API.

Notification Schemes

- **View Default Notification Settings** - can view the *Default Settings* page for notifications.
- **Suspend and Resume Component and Issue Notifications** - can suspend and resume the notifications of components and issues from the *Suspend / Resume All* notifications page.
- **Notification stream comment** - can make comments in the notification stream.
- **Notification issue resolve** - can resolve notification issues.
- **Notification issue assign** - can assign notification issues.
- **Notification flag comment** - can flag comments in the notification stream as important.
- **Edit Current User's Notification Preferences** - can access and edit the *My Notifications* section for the current user.
- **Send test notifications** - can send test notification messages.
- **Edit Watch Lists** - can edit *Watchlists*.
- **Edit Component Notifications** - can change notification recipient and escalation resend periods for components from the *Resend and Escalation Periods* section in the Management Console.

Notification Schemes Continued

- **Edit Unmonitored Component Notifications** - can edit component notification thresholds from *Notifications for Unmonitored Components* section.
- **Edit Component Notification Thresholds** - can customize notifications thresholds for components from the component details page in the Management Console.
- **Edit Default Notification Thresholds** - can edit default notifications thresholds from the *Default Threshold Settings* section.
- **Redirect User Notifications** - can redirect user notification while they are on *holiday*.
- **Edit Notification Delivery Methods** - can enable, disable and configure the *Delivery Methods* available for delivering notifications.
- **Suspend and Resume Global Notifications** - can *suspend and resume* all component and issues notifications across the engine.
- **Edit Other User's Notification Preferences** - can access and edit the *My Notifications Section* page for other users.
- **Edit Public Holidays** - can add, remove, import and export public holidays from the *Public Holidays* page.

Notifications

- The issue event
 - Warning-indicates an issue considered relatively benign but requires attention (ops team receive emails)
 - Alarm-indicates an issue which requires immediate attention (Team Lead receives SMS)
 - Escalation-indicates an issue raised in an Alarm state which has not been resolved in a defined time period (Manager)
- Managing Notification Delivery
 - Email, SMS via an Email bridge, Pager via an Email bridge, SNMP
- Default Notifications
- Default Thresholds
 - Trigger Event
 - Threshold

Watch Lists

- Building and managing a watch list
 - Each watchlist may be managed by clicking on its link on the Notifications / Watchlists page
 - Management of the recipient list for notifications
 - Management of the components associated with the list.
- Building a watch list schedule
- Watch lists and public holidays
- Watch lists and user schedules
 - The vacation support is available in the My Leave section of the Notifications / My Notifications page for the user

Why Integrate to 3rd Party Monitoring

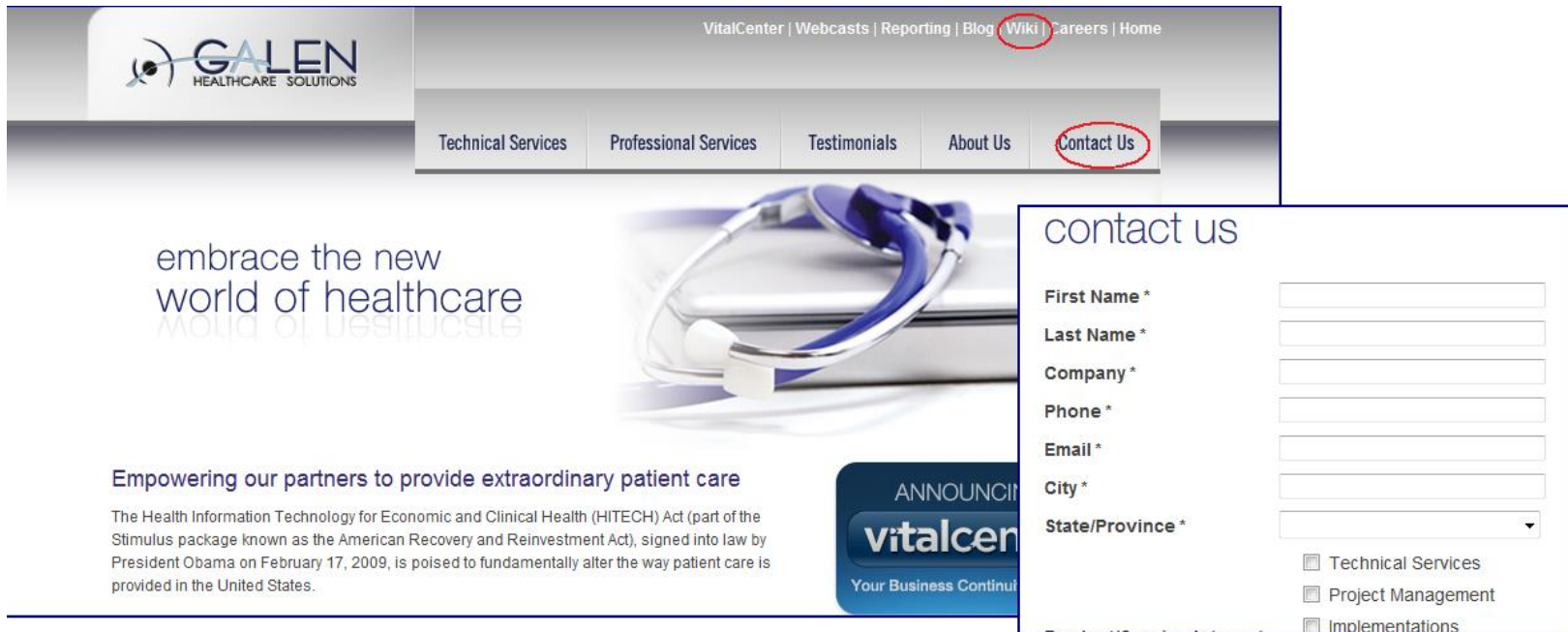
- There is already an enterprise wide monitoring system in use
- There are many Rhapsody engines that need to be monitored
- Rhapsody 'push' based notifications can not notify if something critical happens
- Options
 - Restful API
 - Rhapsody Watchlist Notifier
 - Web Services API

Best Practices

- Ensure the persistence store is maintained in a secure environment
- Ensure the archive cleanup parameters are correctly defined, and that message queues are dealt with efficiently
- Ensure that disk usage is monitored
- Understand the purpose of backups and implement appropriately
- Establish procedures for Disaster Recovery and restoration of the configuration
- Ensure the configuration is tuned to minimize the volume of logging

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

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



The screenshot shows the Galen Healthcare Solutions website. The header includes the company logo and navigation links: VitalCenter | Webcasts | Reporting | Blog | **Wiki** | Careers | Home. A secondary navigation bar contains: Technical Services | Professional Services | Testimonials | About Us | **Contact Us**. The main content area features the slogan "embrace the new world of healthcare" with a stethoscope image. Below this is a section titled "Empowering our partners to provide extraordinary patient care" with text about the HITECH Act. A "vitalcenter" announcement box is also visible. An inset shows the "contact us" form with fields for First Name, Last Name, Company, Phone, Email, City, and State/Province, followed by checkboxes for Technical Services, Project Management, and Implementations.

VitalCenter | Webcasts | Reporting | Blog | **Wiki** | Careers | Home

Technical Services | Professional Services | Testimonials | About Us | **Contact Us**

embrace the new world of healthcare

Empowering our partners to provide extraordinary patient care

The Health Information Technology for Economic and Clinical Health (HITECH) Act (part of the Stimulus package known as the American Recovery and Reinvestment Act), signed into law by President Obama on February 17, 2009, is poised to fundamentally alter the way patient care is provided in the United States.

ANNOUNCING
vitalcenter
Your Business Continuity

contact us

First Name *
Last Name *
Company *
Phone *
Email *
City *
State/Province *

☐ Technical Services
☐ Project Management
☐ Implementations

Disk, Heap and CPU Utilization

Rhapsody Threads

- **Route Execution Threads**

1. Route executor thread becomes free
2. Next processing transaction assigned to the thread
3. Transaction starts and message is processed
4. Transaction complete and route executor thread becomes free

- **Message Collector Timeout Threads**

- manage the timeout states of messages held by the message collectors

- **Administration Threads**

- manage the communications with connected IDE sessions (

- **Communication Point Threads**

- The Controller thread monitors the operation of the communication point and manages the behavior if the connection fails
- The Connection threads manage the sending and receiving of messages, the number of threads used being dependent on the mode and the number of connections configured

Engine Startup Processes

- **Active Processes**

- Wrapper
- The Rhapsody JVM

- **The Tanuki Wrapper**

- manages the engine startup and shutdown processes and ensures that the engine remains responsive by sending a service ping to the engine every second

- **Normal Startup**

- 1. Start webservices 2. Start the core Rhapsody service 3. Load modules 4. Roll back transactions which may have been in process when the engine last stopped 5. Start components as defined by the startup state defined for each 6. Resume normal processing.

Retry Management

- **Rhapsody provides a range of tools to manage communication point connections, and more significantly, connection failure.**
 - Immediate retry: suitable for protocols where the target is expected to be permanently available with no need to delay connection attempts
 - Linear retry: suitable for protocols such as network protocols where a connection failure may result from a transient outage lasting a short period, the re-connection attempt being made at regular intervals.
 - Exponential retry: suitable for protocols where the target may have irregular and unknown outages, the re-connection attempts being made at successively longer intervals until connection is achieved; the utility of this method may be lost as the interval grows.

The Persistence Store

- **The default location for the persistence store is in the rhapsody/data folder in the application folder tree**
 - Backup and cleanup: schedule details for configured backups and cleanups
 - config: the Rhapsody configuration
 - head: the persisted version of the configuration
 - running: the currently running version of the configuration
 - lib: user modifiable; support libraries required for filters may be placed here (for example JDBC drivers). The engine checks this location regularly for new libraries.
 - logs: binary event and audit log data, accessible through the log pages in the WMC
- **the data/ folder also contains the engine.version and engine.running files**