

# ***Leveraging WBI (Broker) Monitoring***

# Introduction

**WBI (Broker) provides the ability to get considerable data about its operation and the behavior of your message flows.**

**This session will cover the various interfaces to request data from the broker. This will include command line and programming interfaces to request status information, monitoring data publishing and message flow events.**

**By attending this session, you will be able to better understand your broker operation and function.**

# About the Speaker

- **Richard Nikula**

- ▶ VP of Product Development and Support
- ▶ Over 20 years experience with MQ, Broker and related technologies

- **About Nastel Technologies [www.nastel.com](http://www.nastel.com)**

- ▶ Founded in 1994
- ▶ Middleware-centric Application Performance Management software supplier
- ▶ Core competency : Real-time monitoring and analytics, Messaging Middleware, Java Application Servers, ESB's and other SOA technologies
- ▶ Details on our Message Broker solution are here:  
<http://www.nastel.com/tech/middleware/websphere-message-broker.html>



# Disclaimers

- IBM periodically has changed the name of the Message Broker, from MQSI to WebSphere Message Broker, WebSphere Business Integrator, IBM Integration Bus, ...
- In this session, we are specifically focused on the broker, whatever you call it...
- All topics are applicable to Version 7 and higher
- Most examples are from Version 8 but are applicable to Version 9
- Some examples included are from Nastel products but there are several methods to leverage this data and vendor products are not required.

# Overview

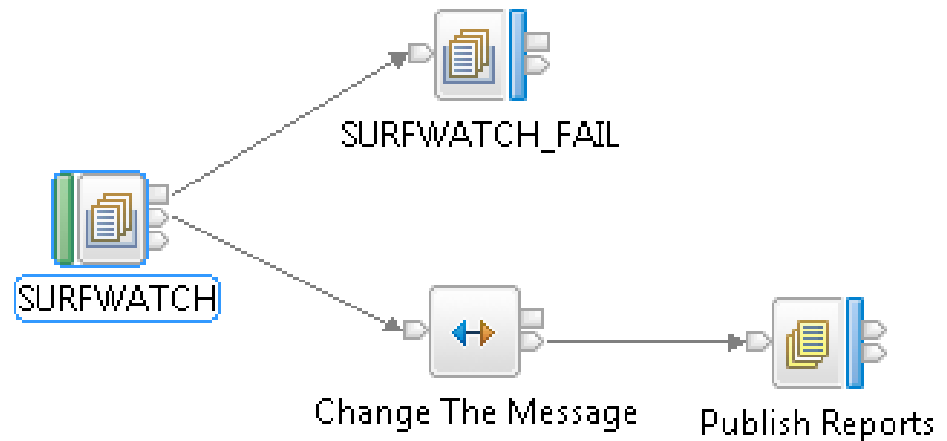
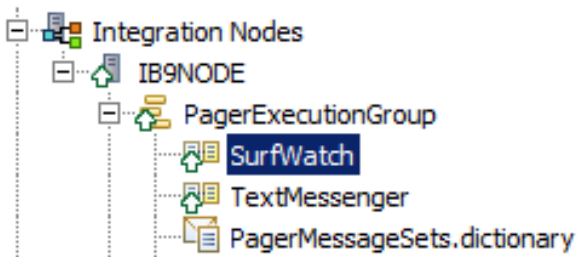
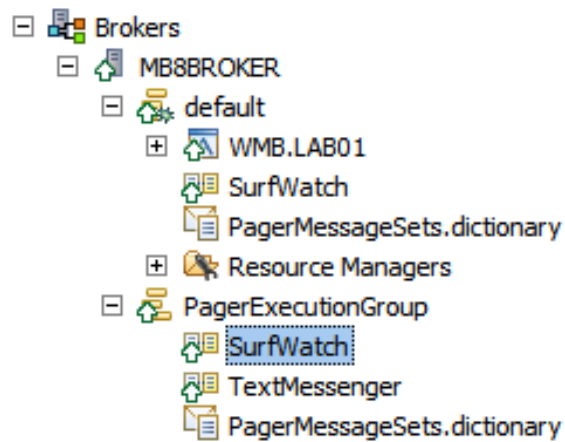
- **Agenda**
  - **Introduction to Broker Monitoring**
  - **Resource Statistics**
  - **Flow Statistics**
  - **Broker Tracking**
  - **Using the Statistics to monitor Broker Environments**
  - **Conclusion**

# Key Broker Terms (as People know them)

- **Broker** – Routing and Transformation Engine supporting various protocols such as http, MQ and files
- **Execution Group** – a grouping of related Message Flows (processes) that provides isolation, performance allocation, and control
- **Message Flow** – A specific logical flow through the message broker which may include logic, transformation and routing.
- **Threads** – processing threads allocated to an Execution Group which execute the message flows
- **Topic** – A hierarchy that describes the context of a specific message
- **Subscription** – A request to receive posts to all or part of a topic hierarchy
- **Queue Manager** – An IBM (WebSphere) MQ service that provides the underlying technology for Message Broker. Required even if the broker does not use WMQ Message Flows

# Simple Example

- This is one of the sample message flows that is shipped with the broker, which has a single input node, one transformation step, and an output. It also has a potential failure node.





# Execution Example

- This example shows a broker with 3 execution groups, each running a set of message flows. One of the execution groups is running 2 instances (threads) of the sample Surfwatch flow, as well as a TextMessage message flow.

Broker1



# Why Monitor the broker?

- “Sub-optimal” configuration
- “Sub-optimal” Application Design
- Contention for broker resources
- Actual usage compared to intended usage
- Planning for future growth
- Tracking
- ...

# Broker Got Monitoring Right

- **Integrated Monitoring**
- **Robust Statistics**
- **Dynamic Activation**
- **Multiple Collection Levels**
- **Consumers Subscribe / Broker Publishes**
- **Basic Monitoring Tools \***





# Type of Broker Management Data

- **Resource Statistics**
  - ▶ Resources used by execution groups
- **Monitoring Statistics**
  - ▶ Usage Statistics of execution groups, nodes and threads
- **Flow Tracking**
  - ▶ Tracking of execution flow through message flows



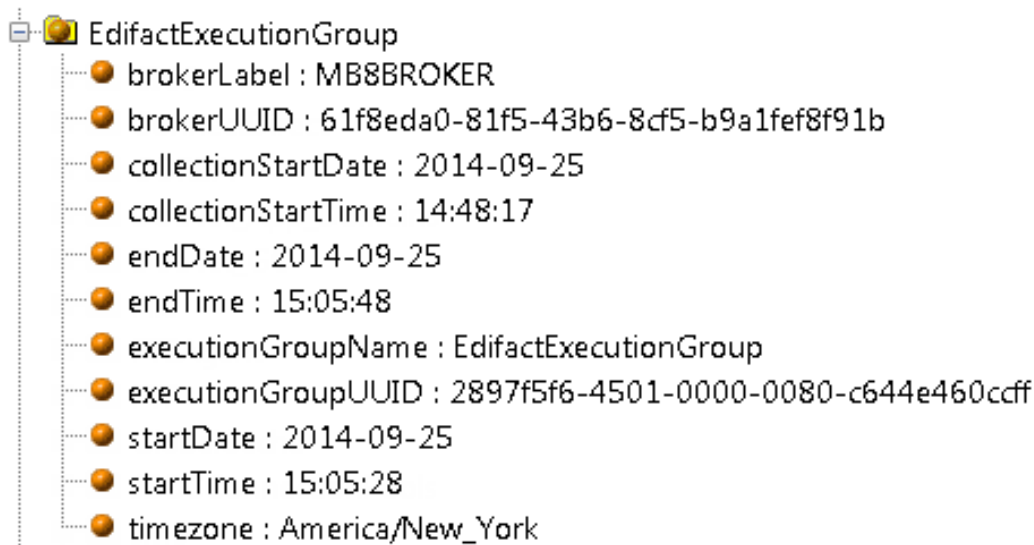
# RESOURCE STATISTICS

# Broker Resource Statistics

- **The broker provides detailed statistics**
  - **Base Data**
    - Collection Information
  - **Extended Data by domain**
    - JVM
    - Parsers
    - DotNet
    - Security
    - JMS
    - FTE
    - JDBC
    - Sockets
    - Soap
  - **Specific data for details varies by domain**

# Sample Statistics (Base)

```
<ResourceStatistics brokerLabel="MB8BROKER" brokerUUID="61f8eda0-81f5-43b6-8cf5-b9a1fef8f91b" executionGroupName="EdifactExecutionGroup" executionGroupUUID="2897f5f6-4501-0000-0080-c644e460ccff" collectionStartDate="2014-09-10" collectionStartTime="19:09:43" startDate="2014-09-21" startTime="10:32:09" endDate="2014-09-21" endTime="10:32:29" timezone="America/New_York">
```



EdifactExecutionGroup

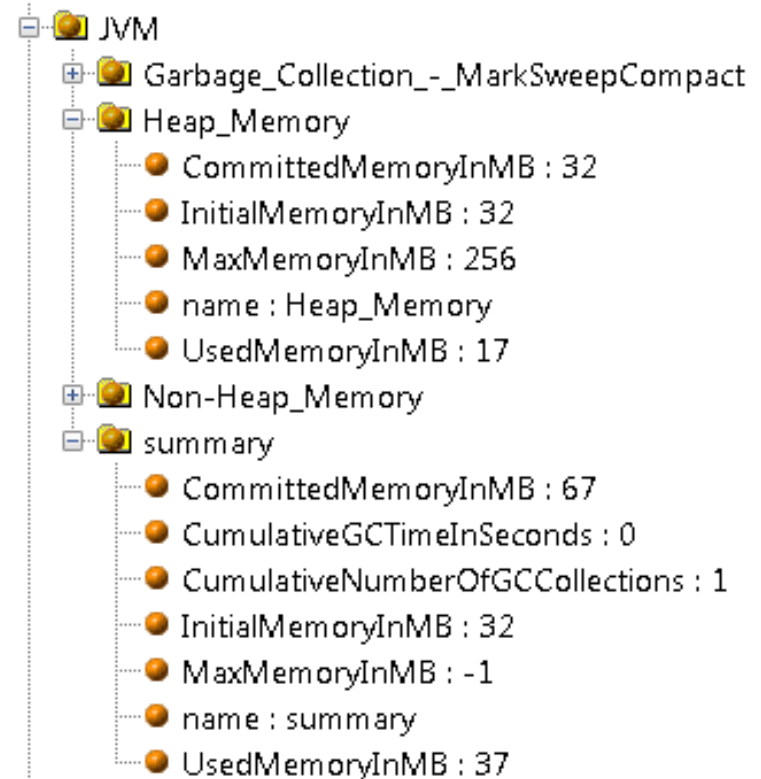
- brokerLabel : MB8BROKER
- brokerUUID : 61f8eda0-81f5-43b6-8cf5-b9a1fef8f91b
- collectionStartDate : 2014-09-25
- collectionStartTime : 14:48:17
- endDate : 2014-09-25
- endTime : 15:05:48
- executionGroupName : EdifactExecutionGroup
- executionGroupUUID : 2897f5f6-4501-0000-0080-c644e460ccff
- startDate : 2014-09-25
- startTime : 15:05:28
- timezone : America/New\_York



# Sample Statistics (Detail)

```
<ResourceType name="JVM">
  <resourceIdentifier name="summary">
    InitialMemoryInMB="32"
    UsedMemoryInMB="63"
    CommittedMemoryInMB="92"
    MaxMemoryInMB="-1"
    CumulativeGCTimeInSeconds="0"
    CumulativeNumberOfGCCollections="40" />
  <resourceIdentifier name="Heap Memory">
    InitialMemoryInMB="32"
    UsedMemoryInMB="43"
    CommittedMemoryInMB="59"
    MaxMemoryInMB="256" />
  <resourceIdentifier name="Non-Heap Memory">
    InitialMemoryInMB="0"
    UsedMemoryInMB="20"
    CommittedMemoryInMB="33"
    MaxMemoryInMB="-1" />
  ...

```



# Activating Resource Statistics

- `mqsichangeresourcestats`
- Collect for all or specific execution group
- Similar commands for distributed and z/OS
- Once Executed, set until reset
- Collection Interval is fixed at 20 seconds

# mqsichangeresourcestats Examples

- **mqsichangeresourcestats Broker -c active**
- **> Activate resource stats data collection for all execution groups**
  
- **mqsichangeresourcestats Broker -e EGRP -c active**
- **> Activate resource stats data collection for execution group EGRP**
  
- **mqsichangeresourcestats Broker -c inactive**
- **> Deactivate resource stats for everything**

# Turning on Collection is the First Step

- The resource statistics data is published
- `$SYS/Broker/brokerName/ResourceStatistics/ExecutionGroup`
- **Example Subscriptions**
  - `$SYS/Broker/Broker1/ResourceStatistics/#`
  - `$SYS/Broker/Broker2/ResourceStatistics/EGRP/#`

The resultant data is then processed directly by a subscribing application or placed on a queue for processing by an application.

# Sample Subscription

- Sample Broker Subscription

The screenshot shows the 'New Subscription' dialog box with the following fields and values:

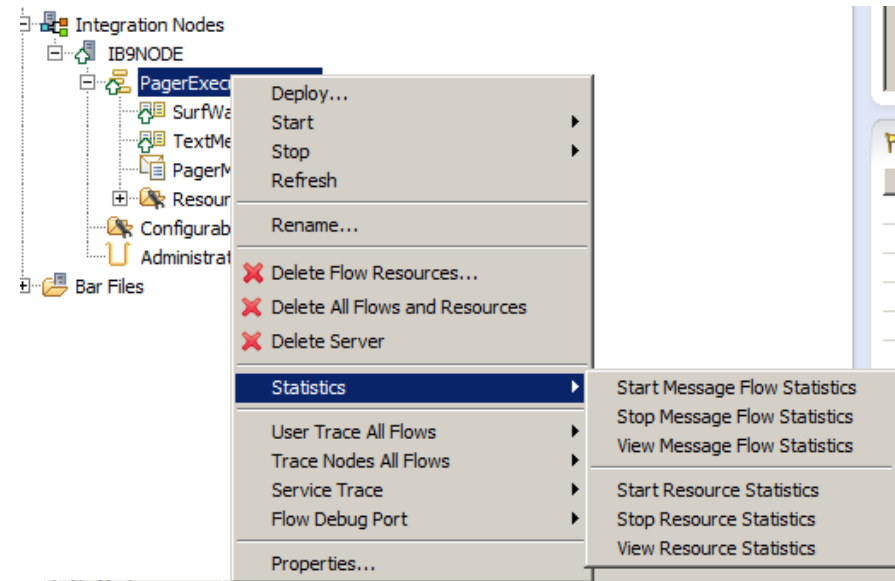
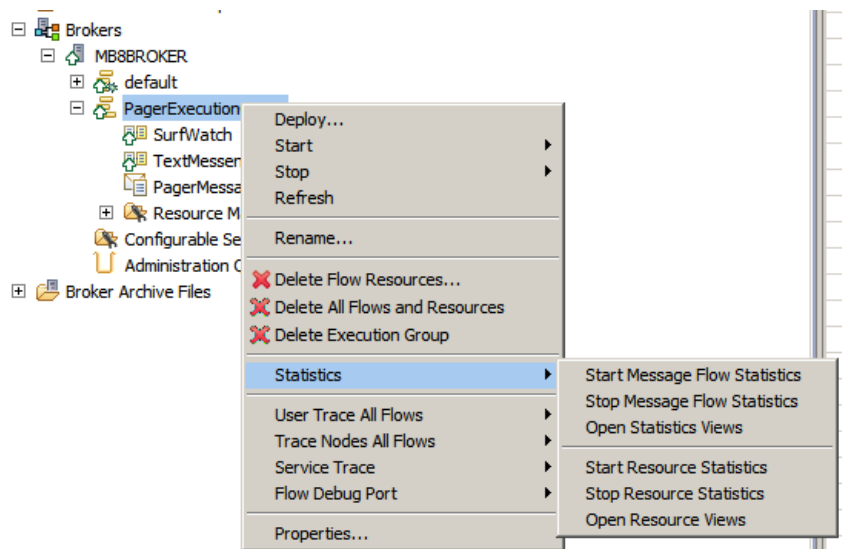
- Subscription name:** Broker\_Resources
- Topic:**
  - Topic name: (empty)
  - Topic string: \$SYS\Broker\MB8BROKER\ResourceStatistics\#
- Wildcard usage:** Topic level wildcard
- Scope:** All
- Destination:**
  - Destination class: Provided
  - Destination queue manager: (empty)
  - Destination name: \* BROKER\_RESOURCE\_QUEUE
  - Correlation identifier: 00000 00 00 00 00 00 00 00 00--  
00010 00 00 00 00 00 00 00 00--

At the bottom of the dialog, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'. A help icon (?) is also present in the bottom left corner.

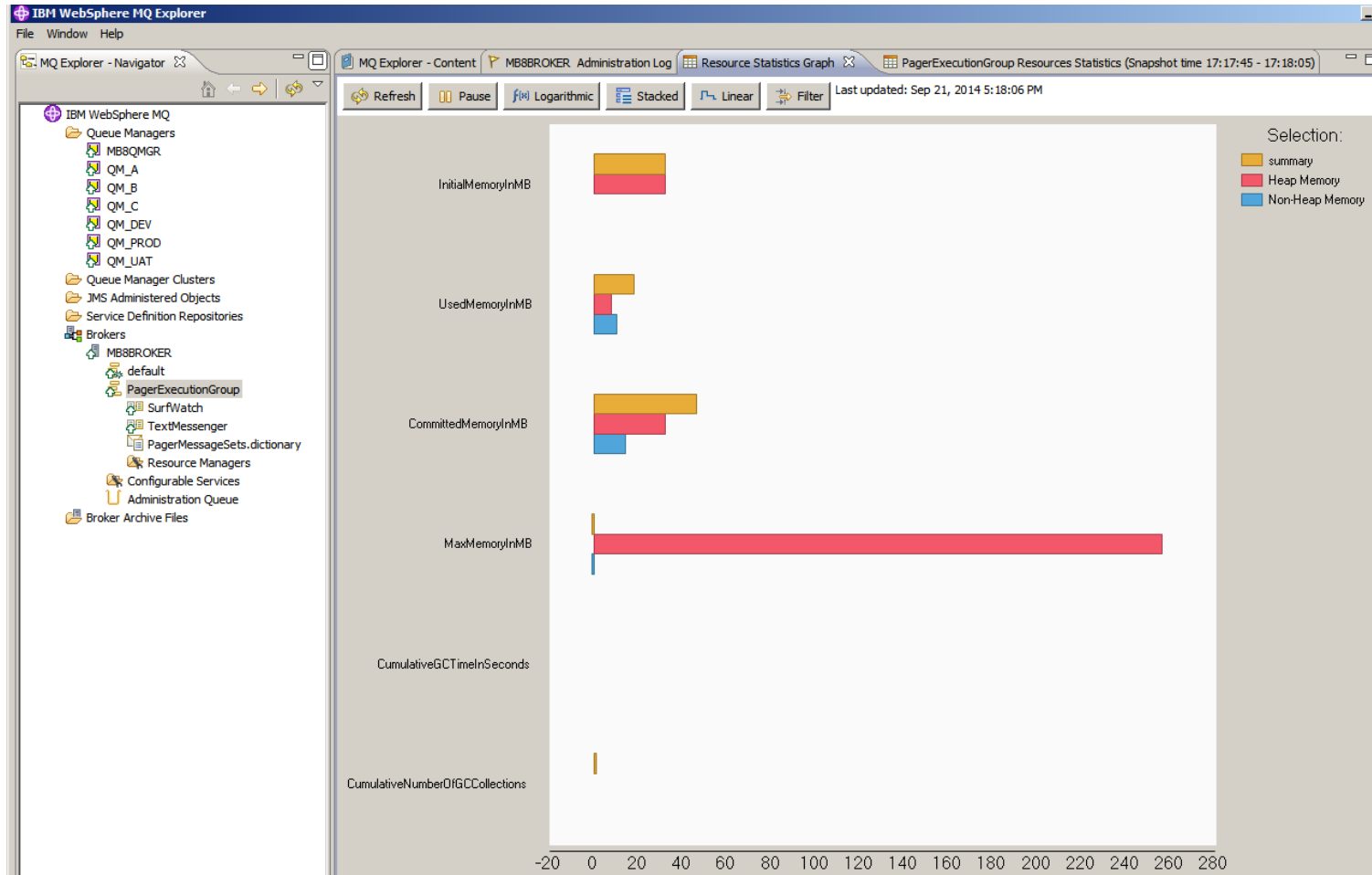
# Resource Statistics Usage

- **JVM**
  - ▶ Identify memory resource issues
  - ▶ Frequent Garbage Collection
- **Parsers**
  - ▶ Identify Resource cost of a parser
  - ▶ Identify inefficient parser architecture
- **Sockets**
  - ▶ Traffic patterns
- **Other**
  - ▶ Based on domains used

# Activating with Broker Explorer



# Displaying Resource Statistic Charts





# Resource Statistics Details

The screenshot shows the IBM WebSphere MQ Explorer interface. On the left is a tree view of the MQ environment, including Queue Managers (MB8QMGR, QM\_A, QM\_B, QM\_C, QM\_DEV, QM\_PROD, QM\_UAT), Queue Manager Clusters, JMS Administered Objects, Service Definition Repositories, and Brokers (MB8BROKER, default, PagerExecutionGroup, Configurable Services, Administration Queue). The main window displays the 'Resource Statistics' for MB8BROKER, with a snapshot time of 17:19:06 - 17:19:26. The table below shows the details of these statistics.

ExecutionGroup	name	InitialMemoryInMB	UsedMemory...	CommittedMemoryI...	MaxMemoryI...	CumulativeGCTimeInSeconds	CumulativeNumberOfGCCollections
PagerExecution...	summary	32	18	46	-1	0	1
PagerExecution...	Heap Memory	32	8	32	256		
PagerExecution...	Non-Heap Mem...	0	10	14	-1		
PagerExecution...	Garbage Collect...					0	1
default	summary	32	38	56	-1	0	1
default	Heap Memory	32	16	32	256		
default	Non-Heap Mem...	0	22	24	-1		
default	Garbage Collect...					0	1



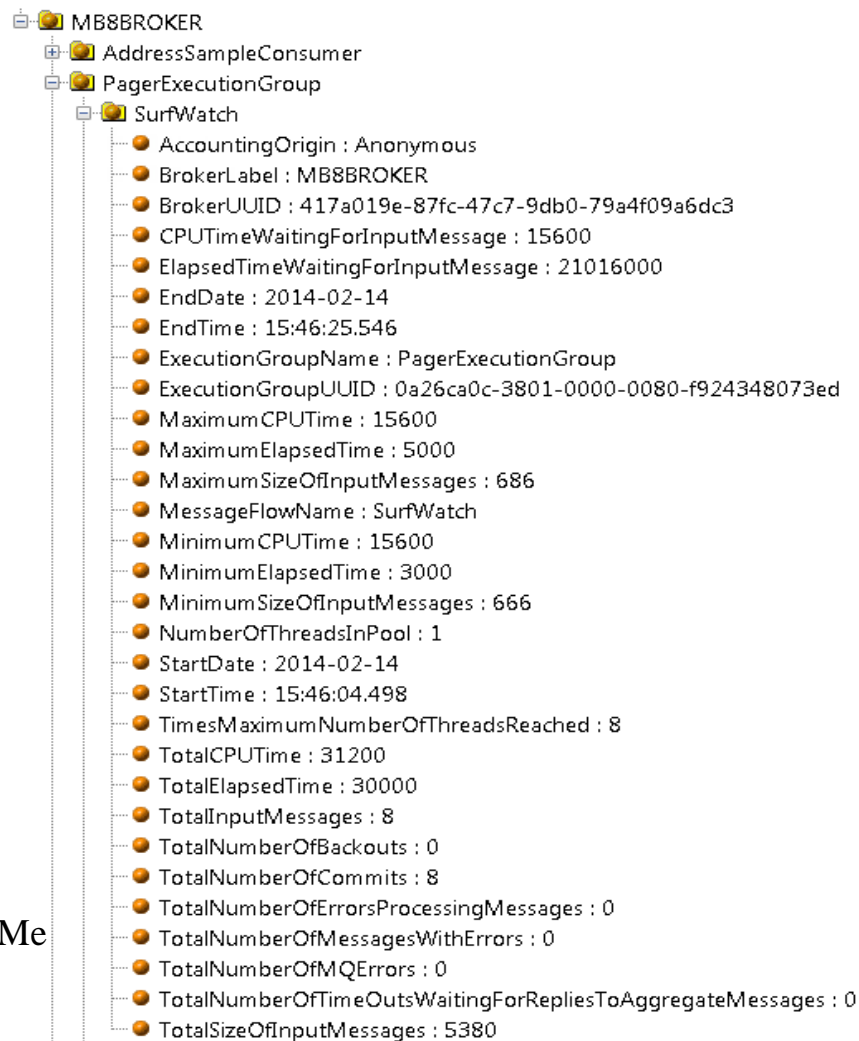
# MONITORING STATISTICS

# Broker Monitoring Statistics

- **The broker provides detailed statistics**
  - **Message Flow Level (for the flow)**
    - Total Messages Processed
    - Total Messages in Error
    - CPU Time Spent
    - Message Statistics
  - **Node Level (for each processing node)**
    - Invocations
    - CPU Time
    - Information
  - **Thread Level (for each thread)**
    - Messages Processed
    - CPU Time Spent
    - Message Statistics

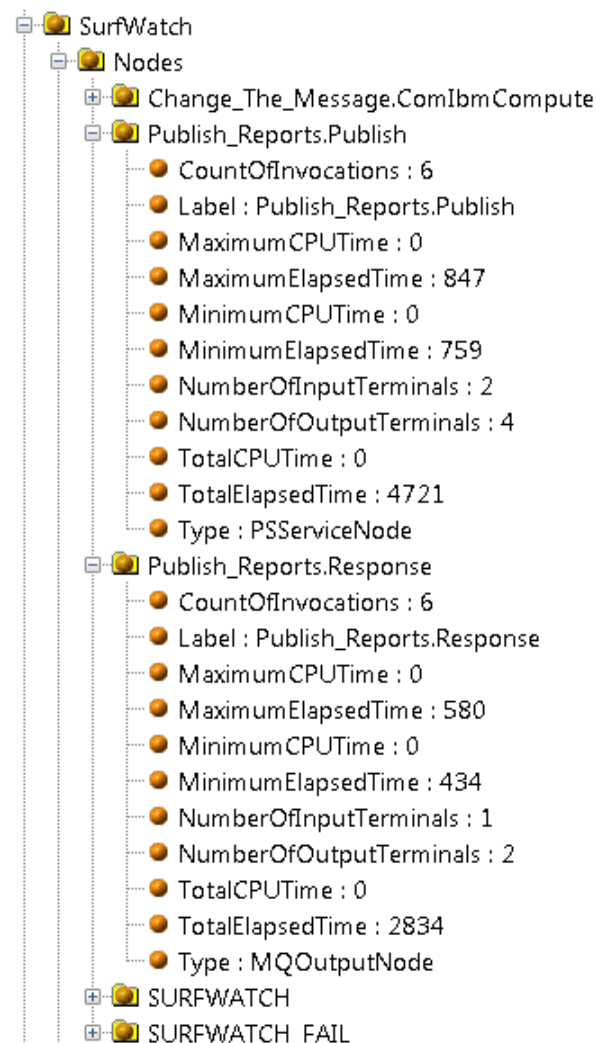
# Message Flow Statistics

```
<MessageFlow BrokerLabel="MB8BROKER"
BrokerUUID="61f8eda0-81f5-43b6-8cf5-b9a1fef8f91b"
ExecutionGroupName="PagerExecutionGroup"
ExecutionGroupUUID="a4f0fff6-4501-0000-0080-
c644e460ccff" MessageFlowName="SurfWatch"
StartDate="2014-09-25" StartTime="15:31:49.312"
EndDate="2014-09-25" EndTime="15:32:09.312"
TotalElapsedTime="0" MaximumElapsedTime="0"
MinimumElapsedTime="0" TotalCPUTime="0"
MaximumCPUTime="0" MinimumCPUTime="0"
CPUTimeWaitingForInputMessage="0"
ElapsedTimeWaitingForInputMessage="19999758"
TotalInputMessages="0" TotalSizeOfInputMessages="0"
MaximumSizeOfInputMessages="0"
MinimumSizeOfInputMessages="0"
NumberOfThreadsInPool="1"
TimesMaximumNumberOfThreadsReached="0"
TotalNumberOfMQErrors="0"
TotalNumberOfMessagesWithErrors="0"
TotalNumberOfErrorsProcessingMessages="0"
TotalNumberOfTimeOutsWaitingForRepliesToAggregateMe
ssages="0" TotalNumberOfCommits="0"
TotalNumberOfBackouts="0"
AccountingOrigin="Anonymous" />
```



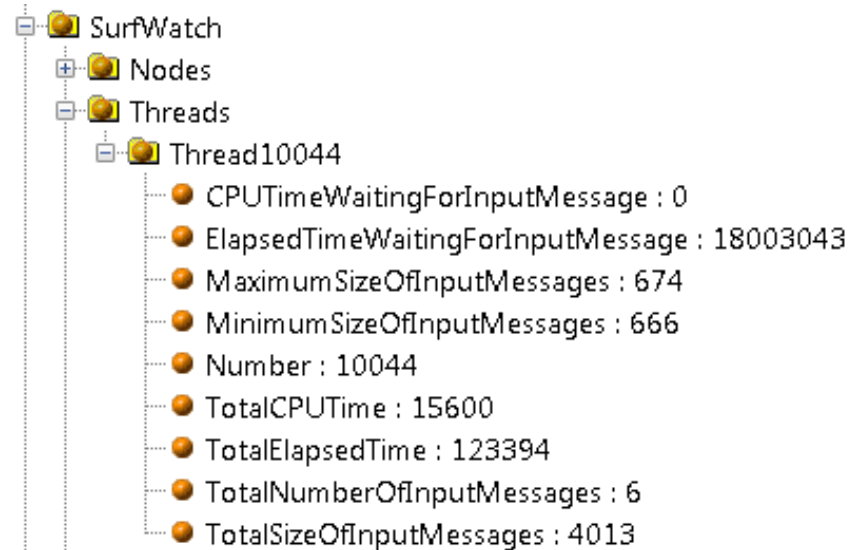
# Node Statistics

```
<Nodes Number="5">
  <NodeStatistics Label="Change The
Message.ComIbmCompute" Type="ComputeNode"
TotalElapsedTime="2724" MaximumElapsedTime="465"
MinimumElapsedTime="448" TotalCPUTime="0"
MaximumCPUTime="0" MinimumCPUTime="0"
CountOfInvocations="6" NumberOfInputTerminals="1"
NumberOfOutputTerminals="6" />
  <NodeStatistics Label="Publish Reports.Publish"
Type="PSServiceNode" TotalElapsedTime="4751"
MaximumElapsedTime="808" MinimumElapsedTime="773"
TotalCPUTime="0" MaximumCPUTime="0"
MinimumCPUTime="0" CountOfInvocations="6"
NumberOfInputTerminals="2"
NumberOfOutputTerminals="4" />
  <NodeStatistics Label="Publish Reports.Response"
Type="MQOutputNode" TotalElapsedTime="961"
MaximumElapsedTime="167" MinimumElapsedTime="157"
TotalCPUTime="0" MaximumCPUTime="0"
MinimumCPUTime="0" CountOfInvocations="6"
NumberOfInputTerminals="1"
NumberOfOutputTerminals="2" />
```



# Thread Level Statistics

```
<Threads Number="1">  
  <ThreadStatistics Number="10044"  
    TotalNumberOfInputMessages="6"  
    TotalElapsedTime="47679" TotalCPUTime="15600"  
    CPUTimeWaitingForInputMessage="0"  
    ElapsedTimeWaitingForInputMessage="18003043"  
    TotalSizeOfInputMessages="4013"  
    MaximumSizeOfInputMessages="674"  
    MinimumSizeOfInputMessages="666" />  
</Threads>
```



# Activating Statistics Collection

- `mqsichangeflowstats`
  
- Collect for all or specific flows
- Collect Node level statistics (optional)
- Collect Thread level statistics (optional)
  
- Similar commands for distributed and z/OS
  
- Once Executed, set until reset
- 2 forms
  - Snapshot – 20 second sample
  - Archive – longer term sample (default 60 minutes)

# mqsichangeflowstats Examples

- **mqsichangeflowstats Broker -s -g -j -c active**
- **> Activate snapshot data collection for all execution groups and flows (with the last options previously set)**
  
- **mqsichangeflowstats Broker -a -g -j -n basic -c active**
- **> Activate archive data collection for all execution groups and flows including basic node level statistics**
  
- **mqsichangeflowstats Broker -a -e EGRP -f MyFlow1 -c active -t basic**
- **> Activate archive data collection for execution group EGRP and only flow MyFlow1 include basic thread statistics**
  
- **mqsichangeflowstats Broker -s -g -j -c inactive**
- **> Deactivate snapshot data collection for everything**



# Some Notable Options

- r      Resets the archive interval
  
- k      Application name for fine-grain analysis
  
- o      json      Output is sent using JSON format (unique topics)  
         smf      zOS, written as SMF type 117  
         usertrace    written to log files for post processing  
         xml      xml formatted

# Turning on Collection is the First Step

- The statistics data is published
  - `$SYS/Broker/brokerName/StatisticsAccounting/recordType/executionGroupLabel/messageFlowLabel`
  - `$SYS/Broker/brokerName/Statistics/JSON/SnapShot/...`
  
- Example Subscriptions
  - `$SYS/Broker/Broker1/StatisticsAccounting/#`
  - `$SYS/Broker/+/StatisticsAccounting/SnapShot/#`
  - `$SYS/Broker/Broker2/StatisticsAccounting/+/EGRP/#`

The resultant data is then processed directly by a subscribing application or placed on a queue for processing by an application.

# Sample Subscription

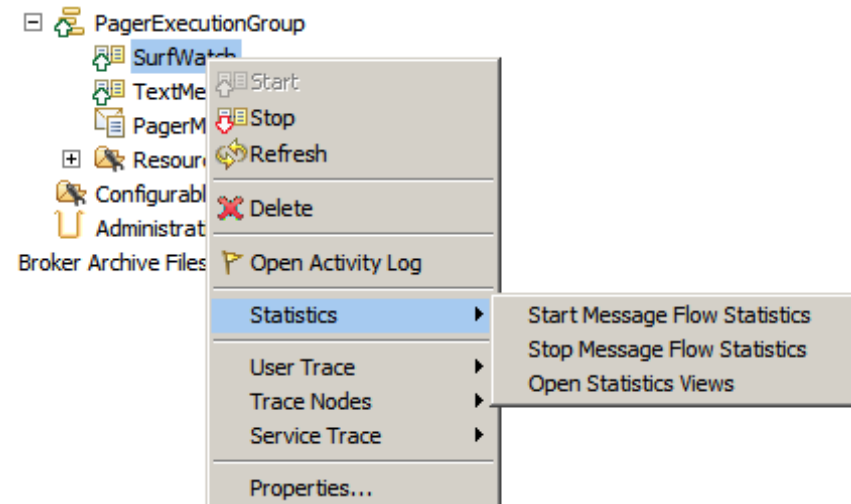
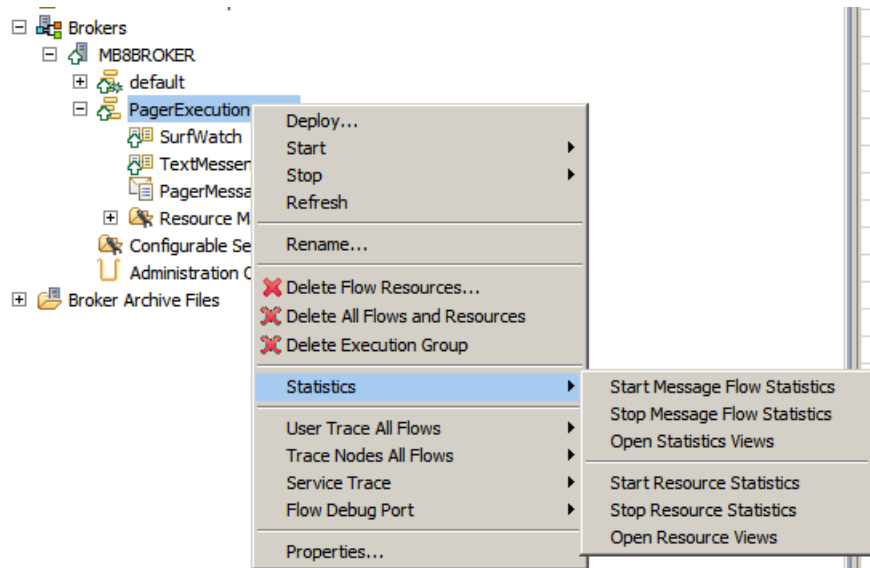
- Sample Monitoring Statistics Subscription

The screenshot shows a 'New Subscription' dialog box with the following fields and values:

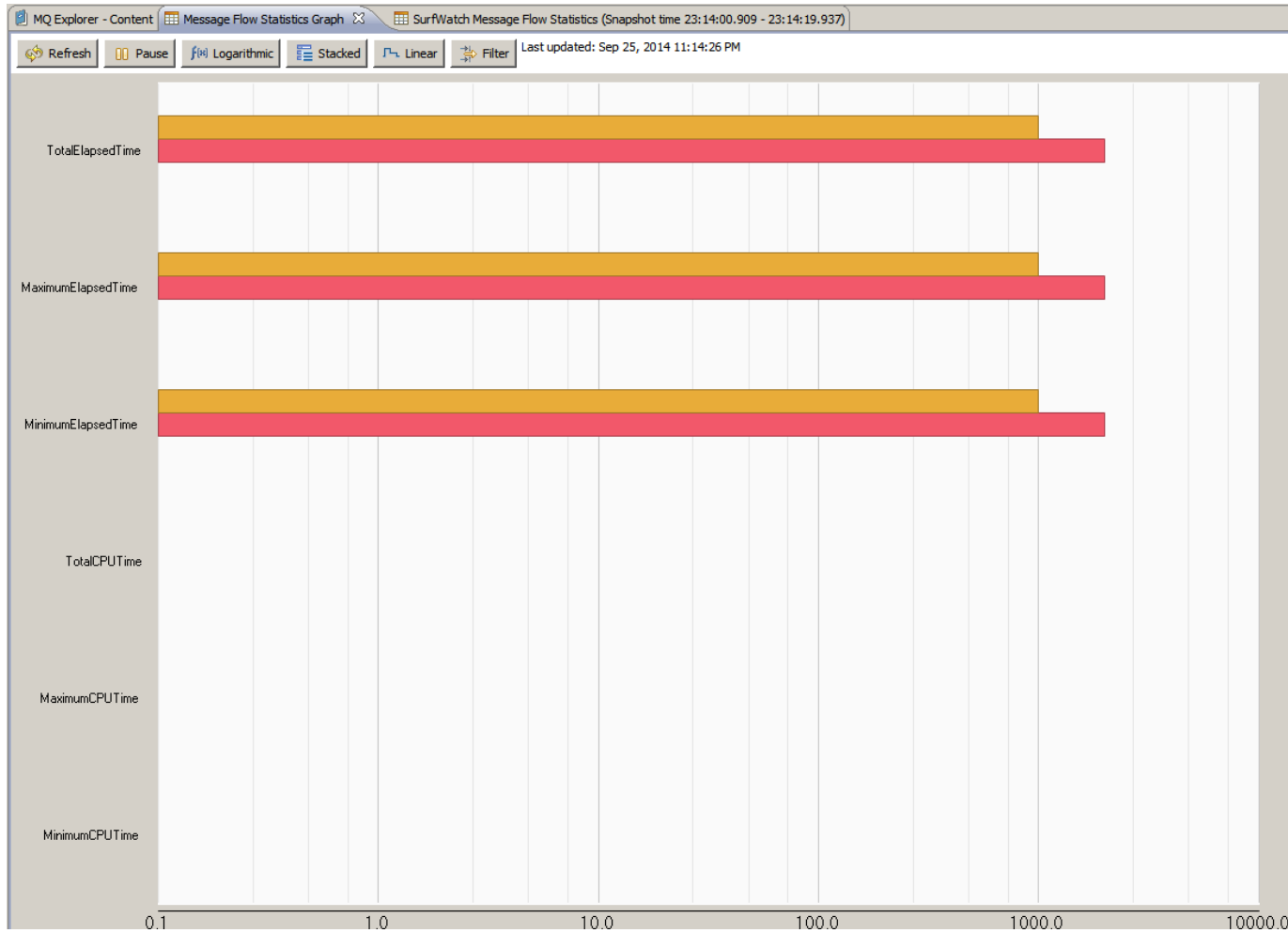
- Subscription name:** BROKER\_FLOW\_MONITORING
- Topic:**
  - Topic name: (empty)
  - Topic string: \$SYS/Broker/MB8BROKER/StatisticsAccounting/#
- Wildcard usage:** Topic level wildcard
- Scope:** All
- Destination:**
  - Destination class: Provided
  - Destination queue manager: (empty)
  - Destination name: \* BROKER\_MONITORING
  - Correlation identifier: 00000 00 00 00 00 00 00 00 00--  
00010 00 00 00 00 00 00 00 00--

At the bottom of the dialog, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'. A help icon (?) is also present in the bottom left corner.

# Activating with Broker Explorer



# Flow Statistics Charts



# Flow Statistics Details

MQ Explorer - Content | Message Flow Statistics Graph | SurfWatch Message Flow Statistics (Snapshot time 22:48:45.091 - 22:49:03.152)

Label	Type	TotalElapsedTime	MaximumElapsedTi...	MinimumElapsedTime	TotalCPU...	MaximumCP...	MinimumCPU...	CountOfInvocations	NumberOfInputTerminals	NumberOfOutputTerminals
Change The Mes...	ComputeNode	16000	16000	16000	15625	15625	15625	2	1	6
Publish Reports....	PSServiceNode	0	0	0				2	2	4
Publish Reports....	MQOutputNode	0	0	0				2	1	2
SURFWATCH	MQInputNode	0	0	0				2	0	3
SURFWATCH_FAIL	MQOutputNode	0	0	0				0	1	2

**Table Metrics**

- TotalElapsedTime
- MaximumElapsedTime
- MinimumElapsedTime
- TotalCPUTime
- MaximumCPUTime
- MinimumCPUTime
- CountOfInvocations
- NumberOfInputTerminals
- NumberOfOutputTerminals

Select All | Select None

OK | Cancel

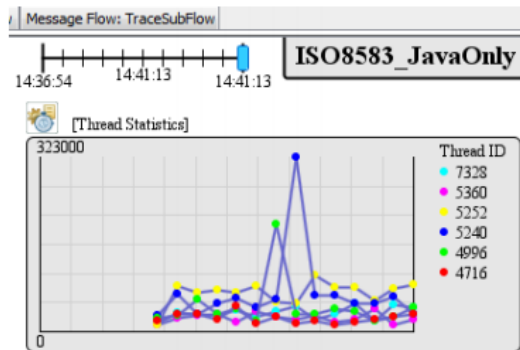
# Snapshot to File

Snapshot time	2014-09-25	23:13:40.849	-	2014-09-25	23:14:00.909		
Broker	MB8BROKER						
Execution Group	PagerExecutionGroup						
Flow	SurfWatch						
Label	Type	TotalElapsedTime	MaximumElapsedTime	MinimumElapsedTime		TotalCPUTime	
	MaximumCPUTime	MinimumCPUTime	CountOfInvocations	NumberOfInputTerminals		NumberOfOutputTerminals	
Change The Message.	ComIbmCompute	ComputeNode	0	0	0	0	0
	3	1	6				
Publish Reports.	Publish	PSServiceNode	0	0	0	0	0
	3	2	4				
Publish Reports.	Response	MQOutputNode	0	0	0	0	0
	3	1	2				
SURFWATCH	MQInputNode	0	0	0	0	0	3
	0	3					
SURFWATCH_FAIL	MQOutputNode	0	0	0	0	0	0
	1	2					

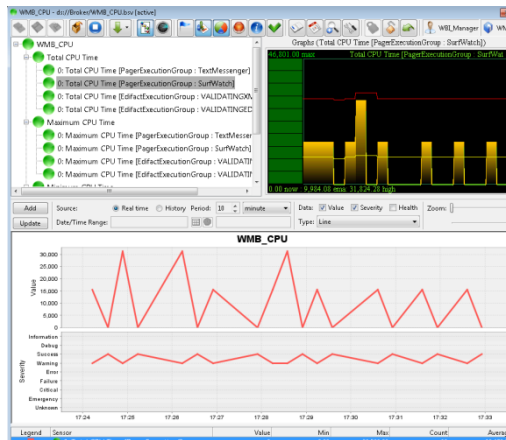
Snapshot time	2014-09-25	23:14:00.909	-	2014-09-25	23:14:19.937		
Broker	MB8BROKER						
Execution Group	PagerExecutionGroup						
Flow	SurfWatch						
Label	Type	TotalElapsedTime	MaximumElapsedTime	MinimumElapsedTime		TotalCPUTime	
	MaximumCPUTime	MinimumCPUTime	CountOfInvocations	NumberOfInputTerminals		NumberOfOutputTerminals	
Change The Message.	ComIbmCompute	ComputeNode	1000	1000	1000	0	0
	4	1	6				
Publish Reports.	Publish	PSServiceNode	2000	2000	2000	0	0
	4	2	4				
Publish Reports.	Response	MQOutputNode	0	0	0	0	0
	4	1	2				
SURFWATCH	MQInputNode	1000	1000	1000	0	0	4
	0	3					
SURFWATCH_FAIL	MQOutputNode	0	0	0	0	0	0
	1	2					

# Capture and Display

IS03



Via 3rd Party Products



Web Visualization and Analytics





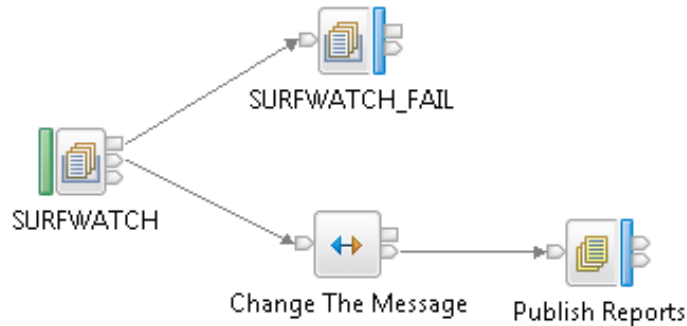


# FLOW MONITORING

# Tracking within the Message Flows

- **The Broker Supports Tracking within the Message Flows**
  - **Transaction Start / Stop (default)**
  - **See when a given node was processed**
  - **See details about the message being processed by the flow**
  - **Track message flows in and across brokers**
  
- **Activated at the Message Flow and Node Level**
  - **Whether to collect**
  - **Data to Collect**

# Configuring Message Flow Events



Graph User Defined Properties

Properties Problems Deployment Log

## Default Values for Message Flow Properties - SurfWatch

Description

Configure monitoring events.

**Monitoring**

Events

3 events defined. Events are defined via the Monitoring tab of a selected node in the message flow.

Enabled	Node	Event Source	Event Source Address	Event Name
<input checked="" type="checkbox"/>	Publish Reports	In terminal	Publish Reports.terminal.in	Publish Reports.InTerminal
<input checked="" type="checkbox"/>	SURFWATCH	Transaction start	SURFWATCH.transaction.St...	SURFWATCH.TransactionStart
<input checked="" type="checkbox"/>	SURFWATCH_FAIL	In terminal	SURFWATCH_FAIL.terminal....	SURFWATCH_FAIL.InTerminal

# Event Attributes – Basic

**Add event**

Basic | Correlation | Transaction

**Event Source**  
Select the source of the event.  
Transaction end

**Event Source Address**  
The broker identifies an event source using an event source address. Use this value when you enable and disable event sources using runtime commands.  
SURFWATCH.transaction.End

**Event Name**  
Provide the name by which events emitted from this source are to be known. Specify either a literal name, or the location of a character field in the message tree or elsewhere in the message assembly.  
 Literal SURFWATCH.TransactionEnd  
 Data location Edit...

**Event Filter**  
Provide an expression to control whether the event is emitted. The expression must evaluate to true or false, and can reference fields in the message tree or elsewhere in the message assembly. If you do not specify a value, the value true() is used.  
true() Edit...

**Event Payload**  
Most events need to contain data taken from fields in the message tree or from elsewhere in the message assembly. Data taken from simple fields or complex fields appears in the event in XML character format. An event can also contain bitstream data, which appears in the event as hexadecimal bytes.

Data location	

Add... Edit... Delete

Include bitstream data in payload

Content Encoding

# Event Attributes – Event Source

**Add event**

Basic Correlation Transaction

**Event Source**  
Select the source of the event.  
Transaction end

**Event Source Address**  
The broker identifies an event source address. You can enable or disable event sources using runtime configuration.  
SURFWATCH.transaction.End

**Event Name**  
Provide the name by which events are emitted, or the location of a character field in the message assembly.  
 Literal SURFWATCH.TransactionEnd  
 Data location

**Event Filter**  
Provide an expression to control whether the event is emitted. The expression must evaluate to true or false, and can reference fields in the message tree or elsewhere in the message assembly. If you do not specify a value, the value true() is used.

**Event Payload**  
Most events need to contain data taken from fields in the message tree or from elsewhere in the message assembly. Data taken from simple fields or complex fields appears in the event in XML character format. An event can also contain bitstream data, which appears in the event as hexadecimal bytes.

Data location		
		<input type="button" value="Add..."/>
		<input type="button" value="Edit..."/>
		<input type="button" value="Delete"/>

Include bitstream data in payload

Content  Encoding

# Event Attributes – Event Name

**Add event**

Basic | Correlation | Transaction

**Event Source**  
Select the source of the event.  
Transaction end

**Event Source Address**  
The broker identifies an event source using an event source address. Use this value when you enable and disable event sources using runtime commands.  
SURFWATCH.transaction.End

**Event Name**  
Provide the name by which events emitted from this source are to be known. Specify either a literal name, or the location of a character field in the message tree or elsewhere in the message assembly.

Literal SURFWATCH.TransactionEnd

Data location Edit...

**Event Filter**  
Provide an expression to control whether the event is emitted. The expression must evaluate to true or false, and can reference fields in the message tree or elsewhere in the message assembly. If you do not specify a value, the value true() is used.  
true() Edit...

**Event Payload**  
Most events need to contain data taken from fields in the message tree or from elsewhere in the message assembly. Data taken from simple fields or complex fields appears in the event in XML character format. An event can also contain bitstream data, which appears in the event as hexadecimal bytes.

Data location	

Add... Edit... Delete

Include bitstream data in payload

Content Encoding

# Event Attributes – Dynamic Event Name

The screenshot displays the XPath Expression Builder window, which is used for constructing XPath expressions. The interface is divided into several sections:

- Event Source:** Select the source of the event. A dropdown menu shows "Transaction start".
- Event Source Address:** The broker identifier for the event source. A dropdown menu shows "SURFWATCH.transaction".
- Event Name:** Provide the name of the event, or the location of the event. Radio buttons are present for "Literal" and "Data location", with "Data location" selected.
- Event Filter:** Provide an expression that evaluates to true or false, and can refer to the event source. A text input field contains "true()".
- Event Payload:** Most events need to be in a specific message assembly format. An event class is selected from a dropdown menu.

The main area of the XPath Expression Builder is titled "XPath Expression Builder" and contains the following elements:

- Data Types Viewer:** A tree view showing the following data types:
  - ▶ Data Types
    - ▶ (X)=\$Root
    - ▶ (X)=\$Properties
    - ▶ (X)=\$LocalEnvironment
      - (X)=\$DestinationList
      - (X)=\$ExceptionList
    - ▶ (X)=\$Environment
- Operators:** A list of operators including: |, /, <=, <, >=, >, =, !=, and, or, +, -.
- Show XML Schema groups:** A checkbox that is currently unchecked.
- XPath Expression:** A large text area for entering the XPath expression.
- Namespace settings:** A section for configuring namespace settings.

# Event Attributes – Event Filter

**Add event**

Basic | Correlation | Transaction

**Event Source**  
Select the source of the event.  
Transaction end

**Event Source Address**  
The broker identifies an event source using an event source address. Use this value when you enable and disable event sources using runtime commands.  
SURFWATCH.transaction.End

**Event Name**  
Provide the name by which events emitted from this source are to be known. Specify either a literal name, or the location of a character field in the message tree or elsewhere in the message assembly.  
 Literal SURFWATCH.TransactionEnd  
 Data location Edit...

**Event Filter**  
Provide an expression to control whether the event is emitted. The expression must evaluate to true or false, and can reference fields in the message tree or elsewhere in the message assembly. If you do not specify a value, the value true() is used.  
true() Edit...

**Event Payload**  
Most events need to contain data taken from fields in the message tree or from elsewhere in the message assembly. Data taken from simple fields or complex fields appears in the event in XML character format. An event can also contain bitstream data, which appears in the event as binary data.  

Data location		

Add... Edit... Delete

Include bitstream data in payload  
Content Encoding



# Event Attributes – Event Payload

The screenshot shows the 'Add event' dialog box with the following sections:

- Basic** (selected tab)
- Event Source**: Select the source of the event. Dropdown menu: Transaction end
- Event Source Address**: The broker identifies an event source using an event source address. Use this value when you enable and disable event sources using runtime commands. Text field: SURFWATCH.transaction.End
- Event Name**: Provide the name by which events emitted from this source are to be known. Specify either a literal name, or the location of a character field in the message tree or elsewhere in the message assembly.
  - Literal: SURFWATCH.TransactionEnd
  - Data location: [Empty field] [Edit...]
- Event Filter**: Provide an expression to control whether the event is emitted. The expression must evaluate to true or false, and can reference fields in the message tree or elsewhere in the message assembly. If you do not specify a value, the value true() is used. Text field: true() [Edit...]
- Event Payload** (highlighted with a green rounded rectangle):
  - Most events need to contain data taken from fields in the message tree or from elsewhere in the message assembly. Data taken from simple fields or complex fields appears in the event in XML character format. An event can also contain bitstream data, which appears in the event as hexadecimal bytes.
  - Table with 2 columns: Data location, [Empty]
  - Buttons: Add..., Edit..., Delete
  - Include bitstream data in payload
  - Content: [Empty dropdown] Encoding: [Empty dropdown]

# Event Attributes - Correlation

The screenshot shows a software window titled "Edit event" with a close button in the top right corner. Below the title bar are three tabs: "Basic", "Correlation", and "Transaction". The "Correlation" tab is selected and active. The main content area is titled "Event Correlation" and contains a descriptive paragraph about event correlators. Below this are three sections, each with a radio button selection and a description box:

- Local transaction correlator:**
  - Automatic  Specify location of correlator
  - Description: The local correlator used by the most recent event for this message flow invocation will be used. If no local correlator exists yet, a new unique value will be generated.
- Parent transaction correlator:**
  - Automatic  Specify location of correlator
  - Description: The parent correlator used by the most recent event for this message flow invocation will be used. If no correlator exists yet, no parent correlator will be used.
- Global transaction correlator:**
  - Automatic  Specify location of correlator
  - Description: The global correlator used by the most recent event for this message flow invocation will be used. If no correlator exists yet, no global correlator will be used.

# mqsichangeflowmonitoring Examples

- `mqsichangeflowmonitoring BROKER -c active -g -j`
- > Activate event flow monitoring for all execution groups and flows
  
- `mqsichangeflowmonitoring BROKER -c active -e default -k application1 -f myflow1`
- > Activate monitoring for message flow *myflow1* referenced by application *application1*, in execution group *default*
  
- `mqsichangeflowmonitoring BROKER -c inactive -g -j`
- > Deactivate event flow monitoring for everything

# Getting Tracking Data

- The statistics Tracking data is published
- `$SYS/Broker/<brokerName>/Monitoring/<executionGroupName>/<flowName>`
- Example Subscriptions
  - `$SYS/Broker/Broker1/Monitoring/#`
  - `$SYS/Broker+/Monitoring/#`
  - `$SYS/Broker/Broker2/Monitoring/EGRP/Flow1`

The resultant data is then processed directly by a subscribing application or placed on a queue for processing by an application.

# Tracking Data

```
<wmb:event xmlns:wmb="http://www.ibm.com/xmlns/prod/websphere/messagebroker/6.1.0/monitoring/event">
<wmb:eventPointData>
<wmb:eventData wmb:productVersion="8001" wmb:eventSchemaVersion="6.1.0.2">
  <wmb:eventIdentity wmb:eventName="transactionStart" />
  <wmb:eventSequence wmb:creationTime="2014-09-25T21:06:10.008Z" wmb:counter="1" />
  <wmb:eventCorrelation wmb:localTransactionId="414d51204d4238514d475220202020bf172454201458fe"
    wmb:parentTransactionId="" wmb:globalTransactionId="" />
</wmb:eventData>
<wmb:messageFlowData>
  <wmb:broker wmb:name="MB8BROKER" wmb:UUID="61f8eda0-81f5-43b6-8cf5-b9a1fef8f91b" />
  <wmb:executionGroup wmb:name="PagerExecutionGroup"
    wmb:UUID="a4f0fff6-4501-0000-0080-c644e460ccff" />
  <wmb:messageFlow wmb:uniqueFlowName="MB8BROKER.PagerExecutionGroup.SurfWatch"
    wmb:name="SurfWatch" wmb:UUID="6c0000f7-4501-0000-0080-d6b3e1d5c115"
    wmb:threadId="10044" />
  <wmb:node wmb:nodeLabel="SURFWATCH" wmb:nodeType="ComIbmMQInputNode"
    wmb:detail="SURFWATCH" />
</wmb:messageFlowData>
</wmb:eventPointData>
</wmb:event>
```

# Tracking Data

```
<wmb:event
xmlns:wmb="http://www.ibm.com/xmlns/prod/websphere/messagebroker/6.1.0/monitoring/event">
<wmb:eventPointData>
  <wmb:eventData wmb:productVersion="8001" wmb:eventSchemaVersion="6.1.0.2">
    <wmb:eventIdentity wmb:eventName="transactionEnd" />
    <wmb:eventSequence wmb:creationTime="2014-09-25T21:06:48.273998Z" wmb:counter="2" />
    <wmb:eventCorrelation wmb:localTransactionId=" 414d51204d4238514d47522020202020bf172454201458fe "
      wmb:parentTransactionId="" wmb:globalTransactionId="" />
  </wmb:eventData>
  <wmb:messageFlowData>
    <wmb:broker wmb:name="MB8BROKER" wmb:UUID="61f8eda0-81f5-43b6-8cf5-b9a1fef8f91b" />
    <wmb:executionGroup wmb:name="PagerExecutionGroup"
      wmb:UUID="a4f0fff6-4501-0000-0080-c644e460ccff" />
    <wmb:messageFlow wmb:uniqueFlowName="MB8BROKER.PagerExecutionGroup.SurfWatch"
      wmb:name="SurfWatch" wmb:UUID="6c0000f7-4501-0000-0080-d6b3e1d5c115"
      wmb:threadId="10044" />
    <wmb:node wmb:nodeLabel="SURFWATCH" wmb:nodeType="ComIbmMQInputNode"
      wmb:detail="SURFWATCH" />
  </wmb:messageFlowData>
</wmb:eventPointData>
</wmb:event>
```

# Example

- Message Flows can be tracked by capturing the flow tracking events



Transaction Group > Summary > Trace Details

Show: Select From: 2014-02-17 10:45:00 AM To: 2014-02-17 10:46:00 AM Show

Trace

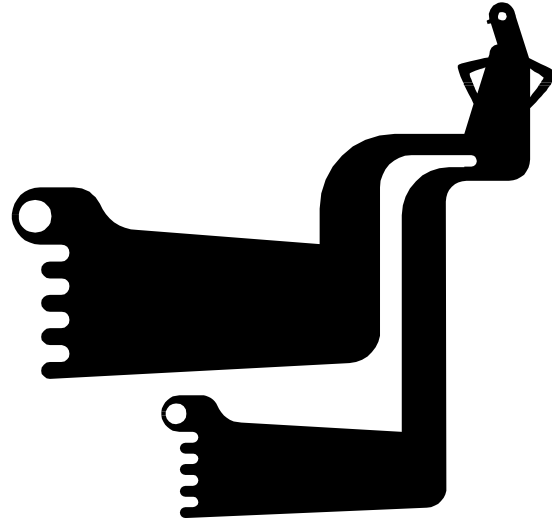
Start Date	Applications	Transaction Status	SLA Status	SLA Status Text	Workload (HH:MM:SS.mm)	Transaction Duration	Operations	Messages	Transaction ID	Transaction Groups	Servers	Resource Manager Typ
2014-02-17 10:45:20	MB8BROKER.P...	Complete ✓		Within SLA ●	0:00:00.000	0:00:00.025	2	1 3			DESKTOP99	Messaging S
2014-02-17 10:45:35	MB8BROKER.P...	Complete ✓		Within SLA ●	0:00:00.000	0:00:00.015	2	1 2			DESKTOP99	Messaging S
2014-02-17 10:45:17	MB8BROKER.P...	Complete ✓		Within SLA ●	0:00:00.000	0:00:00.008	2	1 1			DESKTOP99	Messaging S

Transaction ID 3

Transaction Flow Diagram | Transaction Timeline | Transaction Trace(2) | Transaction Milestones

Show Hierarchy

Time	Application	Operation Name	Resource	Message Id	Message Age (usec)	Completion Code	Correlator	Sen
2014-02-17 10:45:20.214	MB8BROKER.PagerExecutionGroup.SurfWatch	transactionStart	SURFWATCH	1	0	Succeeded	414d51204d4238514d47522020202035b8fb5220687f1d	DESKTOP
2014-02-17 10:45:20.239	MB8BROKER.PagerExecutionGroup.SurfWatch	transactionEnd	SURFWATCH		0	Succeeded	414d51204d4238514d47522020202035b8fb5220687f1d	DESKTOP



## A FEW RELATED ITEMS



# Activity Log

## A Useful related capability

The screenshot shows an IBM Activity Log window titled "IB9NODE\PagerExecutionGroup\SurfWatch - Activity Log". The window has a toolbar with buttons for "All Columns", "Apply filter", "Clear", "All Threads", "Select columns...", "Previous", "Next", and "18 entries". Below the toolbar is a table with two columns: "Message ..." and "Message Summary". The table contains 18 entries, each with a message ID and a description. A dialog box titled "Activity Log Entry for 'BIP11513W'" is open over the table, showing a detailed view of the message BIP11513W. The dialog box contains an information icon, the message ID, the message text, and an "OK" button.

Message ...	Message Summary
BIP11504I	Waiting for data from input node 'SURFWATCH'.
BIP11513W	Flow reconfiguration or redeployment is in progress; the flow is not waiting for data from input nod...
BIP11504I	Waiting for data from input node 'SURFWATCH'.
BIP11501I	Received data from input node 'SURFWATCH'.
BIP11507W	Rolled back a local transaction.
BIP11501I	Received data from input node 'SURFWATCH'.
BIP11506I	Committed a local transaction.
BIP11504I	Waiting for data from input node 'SURFWATCH'.
BIP11513W	Flow reconfiguration or redeployment is in progress; the flow
BIP11504I	Waiting for data from input node 'SURFWATCH'.
BIP11513W	Flow reconfiguration or redeployment is in progress; the flow
BIP11504I	Waiting for data from input node 'SURFWATCH'.
BIP11513W	Flow reconfiguration or redeployment is in progress; the flow is not waiting for data from input nod...
BIP11504I	Waiting for data from input node 'SURFWATCH'.
BIP11513W	Flow reconfiguration or redeployment is in progress; the flow is not waiting for data from input nod...
BIP11504I	Waiting for data from input node 'SURFWATCH'.
BIP11513W	Flow reconfiguration or redeployment is in progress; the flow is not waiting for data from input nod...
BIP11504I	Waiting for data from input node 'SURFWATCH'.

**Activity Log Entry for "BIP11513W"**

BIP11513W: Flow reconfiguration or redeployment is in progress; the flow is not waiting for data from input node 'SURFWATCH'.

The flow 'SurfWatch' is being reconfigured or redeployed, therefore this input node is not currently polling or waiting for data.

OK

# Sample Subscription

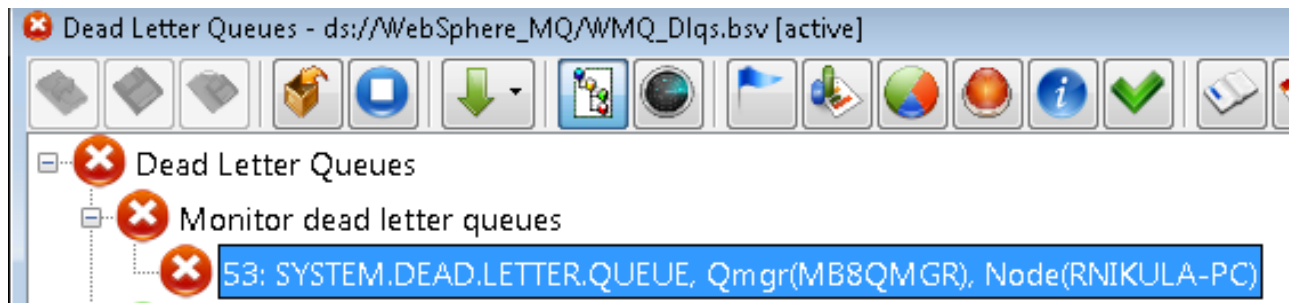
- **Subscribing for all Broker Data**

The screenshot shows a 'New Subscription' dialog box with the following configuration:

- Subscription name:** BROKER\_ALL\_MONITORING
- Topic:**
  - Topic name: (empty)
  - Topic string: \$SYS/Broker/MB8BROKER/#
- Wildcard usage:** Topic level wildcard
- Scope:** All
- Destination:**
  - Destination class: Provided
  - Destination queue manager: (empty)
  - Destination name: \* BROKER\_EVENTS
  - Correlation identifier: 00000 00 00 00 00 00 00 00 00--  
00010 00 00 00 00 00 00 00 00--

At the bottom of the dialog, there are buttons for '< Back', 'Next >', 'Finish', and 'Cancel'. A help icon (?) is also present in the bottom left corner.

# Bad Things Can Happen



SYSTEM.BROKER.MB.TOPIC - Status

Queue Manager: IB9QMGR Topic Name: SYSTEM.BROKER.MB.TOPIC

Topic status for the topic "SYSTEM.BROKER.MB.TOPIC":

Topic string	Publish	Subscribe	Durable subscriptions	Default priority	Default persistence	Model durable queue	Model non-durable queue	Default put response type
\$SYS/Broker	Allowed	Allowed	Allowed	0	Not persistent	SYSTEM.DURABLE.MODEL.QUEUE	SYSTEM.NDURABLE.MODEL.QUEUE	Synchronous
IB9NODE	Allowed	Allowed	Allowed	0	Not persistent	SYSTEM.DURABLE.MODEL.QUEUE	SYSTEM.NDURABLE.MODEL.QUEUE	Synchronous
ResourceStatistics	Allowed	Allowed	Allowed	0	Not persistent	SYSTEM.DURABLE.MODEL.QUEUE	SYSTEM.NDURABLE.MODEL.QUEUE	Synchronous
PagerExecutionGroup	Allowed	Allowed	Allowed	0	Not persistent	SYSTEM.DURABLE.MODEL.QUEUE	SYSTEM.NDURABLE.MODEL.QUEUE	Synchronous
StatisticsAccounting	Allowed	Allowed	Allowed	0	Not persistent	SYSTEM.DURABLE.MODEL.QUEUE	SYSTEM.NDURABLE.MODEL.QUEUE	Synchronous
Snapshot	Allowed	Allowed	Allowed	0	Not persistent	SYSTEM.DURABLE.MODEL.QUEUE	SYSTEM.NDURABLE.MODEL.QUEUE	Synchronous
PagerExecutionGroup								
SurfWatch								
TextMessenger								
Status								

Topic Name: SYSTEM.BROKER.MB.TOPIC

Sub count	Pub count	Retained publication	NPM delivery	PM delivery	Publication scope	Subscription scope	Cluster name	Cluster route	Communication information	Multicast	Use DLO
0	0	No	To all available subscribers	To all durable subscribers	All	All		None	SYSTEM.DEFAULT.COMMINFO.MULTICAST	Disabled	Yes
1	0	No	To all available subscribers	To all durable subscribers	All	All		None	SYSTEM.DEFAULT.COMMINFO.MULTICAST	Disabled	Yes
1	0	No	To all available subscribers	To all durable subscribers	All	All		None	SYSTEM.DEFAULT.COMMINFO.MULTICAST	Disabled	Yes
1	0	No	To all available subscribers	To all durable subscribers	All	All		None	SYSTEM.DEFAULT.COMMINFO.MULTICAST	Disabled	Yes
1	0	No	To all available subscribers	To all durable subscribers	All	All		None	SYSTEM.DEFAULT.COMMINFO.MULTICAST	Disabled	Yes
1	0	No	To all available subscribers	To all durable subscribers	All	All		None	SYSTEM.DEFAULT.COMMINFO.MULTICAST	Disabled	Yes
1	0	No	To all available subscribers	To all durable subscribers	All	All		None	SYSTEM.DEFAULT.COMMINFO.MULTICAST	Disabled	Yes
1	0	Yes	To all available subscribers	To all durable subscribers	All	All		None	SYSTEM.DEFAULT.COMMINFO.MULTICAST	Disabled	Yes
1	0	Yes	To all available subscribers	To all durable subscribers	All	All		None	SYSTEM.DEFAULT.COMMINFO.MULTICAST	Disabled	Yes
1	0	Yes	To all available subscribers	To all durable subscribers	All	All		None	SYSTEM.DEFAULT.COMMINFO.MULTICAST	Disabled	Yes



# SUMMARY

# Where to Start

- **Resource Statistics**
  - ▶ Identify resource usage
- **Flow Statistics**
  - ▶ How Broker is being used
- **Flow Tracking**
  - ▶ Advanced analysis
  - ▶ Special use cases (audit)

## Other sources

- **Session “Performance Considerations for IBM MQ and IBM Integration Bus”**
- **Redbooks/Web Sessions**

# Summary

- **Message Broker provides a wealth of statistics and performance data**
- **Collection is an integrated component of the broker**
- **You can select the level of data collection required**
- **Tracking data can be used to augment the statistical data**

# Questions & Answers

