



IBM Software Group

# Tracing WebSphere's Service Integration Bus For Your Own Use

Paul O'Donnell (paulod@us.ibm.com)

Mohamed Najih (najih@us.ibm.com)

December 11, 2014



WebSphere® Support Technical Exchange



# Agenda

- Service Integration Bus: A quick overview
- Tools used for diagnostic data collection and analysis:
  - ▶ IBM Support Assistant for Data Collection (ISADC)
  - ▶ WebSphere Application Server (WAS) Trace Analyzer
- Preparation for WAS Tracing
- SIB Trace Strings
- Tracing JMS™ Format And Protocol (JFAP) Channel and SIB Communications
- Tracing Messages
- Tracing SIB Database Interaction



# Agenda

- **Service Integration Bus: A quick overview**
- Tools used for diagnostic data collection and analysis:
  - ▶ IBM Support Assistant for Data Collection (ISADC)
  - ▶ WAS Trace Analyzer
- Preparation for WAS Tracing
- SIB Trace Strings
- Tracing JFAP Channel and SIB Communications
- Tracing Messages
- Tracing SIB Database Interaction

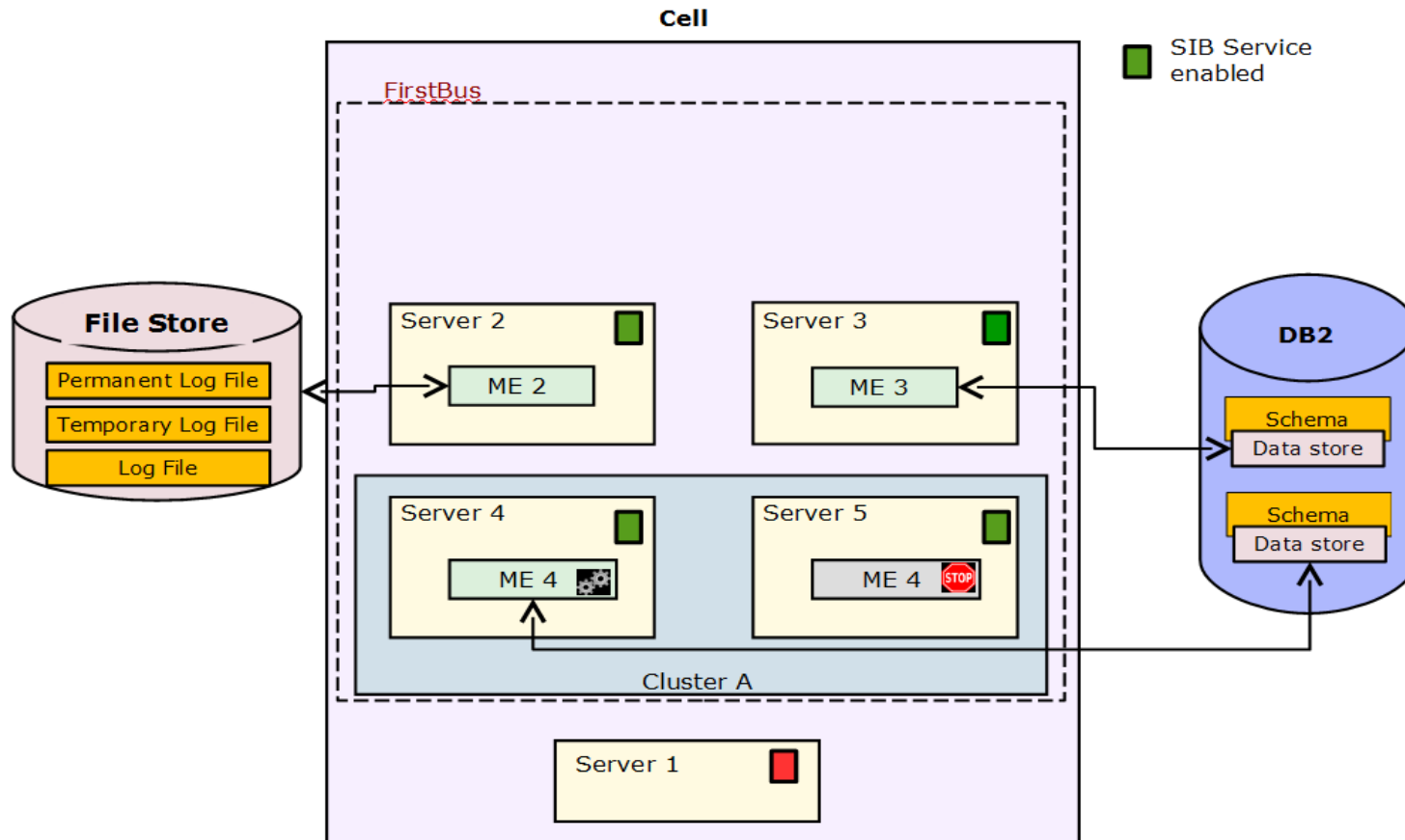


# Service Integration Bus (SIB): Overview

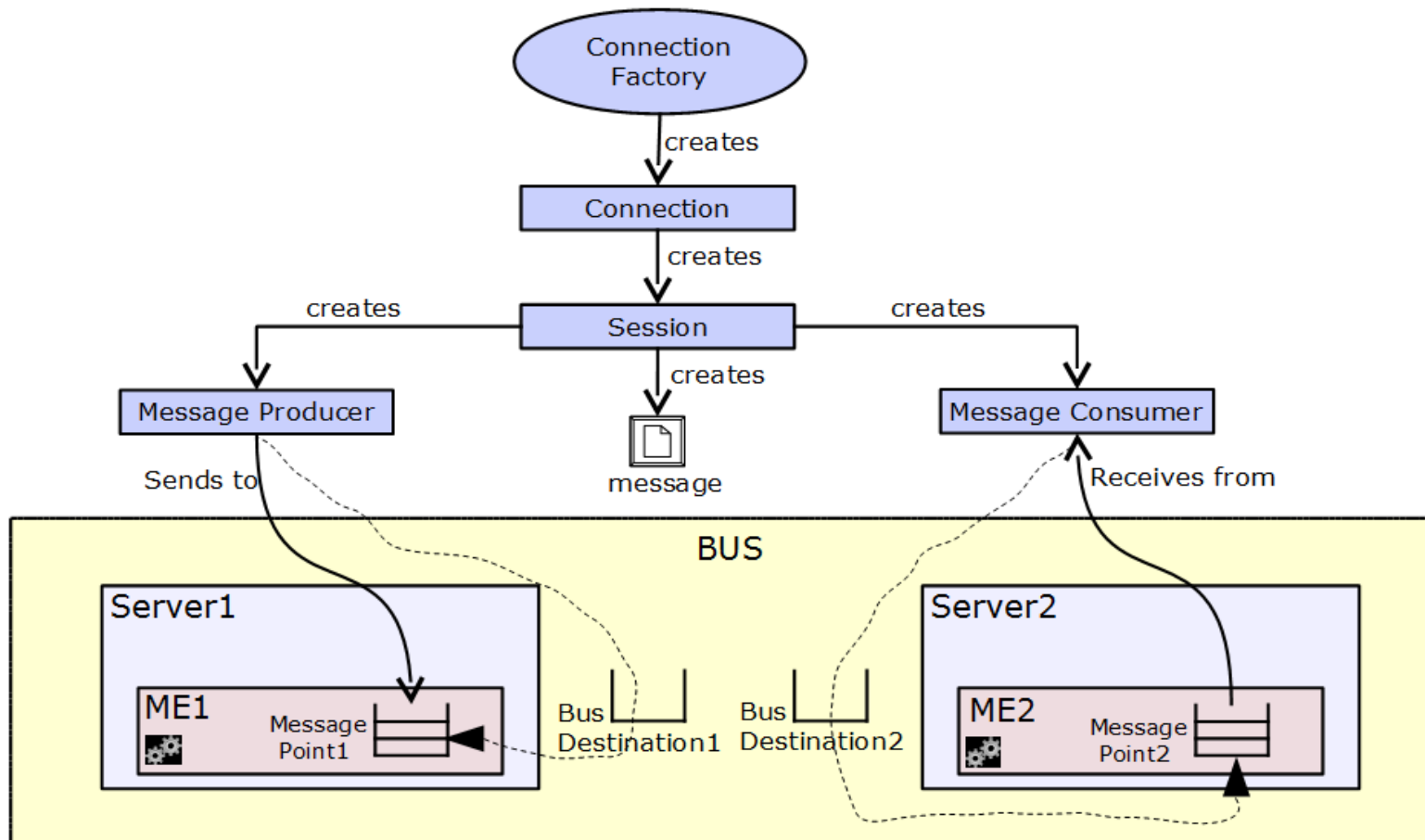
The Service Integration Bus is the implementation of the WAS Default Messaging Provider. It is therefore a key component on which other IBM products rely, including WebSphere Process Server, WebSphere Portal, and Maximo<sup>®</sup> Asset Management. Its main components are:

- **Bus:** A group of interconnected servers, clusters and WMQ servers that become bus members.
- **Messaging Engine (ME):** Lightweight runtime component that runs inside an application server. It provides core messaging functionality of a bus. MEs inside a bus work together to create the illusion of the concept of a bus.
- **Sib Service:** An application server component responsible for managing (i) all of the messaging resources within an application server, (ii) the life cycle of any defined messaging-related transport chains and (iii) handles inbound connection requests from external messaging applications.
- **Destination (queue or topic):** A logical address to which applications can attach to in order to exchange messages. It gets associated with a physical address, called **message point**, where messages actually end up in.
- **Message store (file store or data store):** Used by the ME to preserve both, its volatile and durable data.

# SIB Topology: Example



# JMS API Programming Model



JMS™

# Agenda

- Service Integration Bus: A quick overview
- Tools used for diagnostic data collection and analysis:
  - ▶ IBM Support Assistant for Data Collection (ISADC)
  - ▶ WAS Trace Analyzer
- Preparation for WAS Tracing
- SIB Trace Strings
- Tracing JFAP Channel and SIB Communications
- Tracing Messages
- Tracing SIB Database Interaction



## IBM Support Assistant Data Collector (ISADC)

- Bundled in WAS V8.5 and later: Interactive way of collecting a COMPLETE mustgather data, for IBM and/or your own use. <http://www.ibm.com/support/docview.wss?uid=swg21145599>
- Can be invoked through web browser or command line prompt: index.html and isadc.bat/sh are both located in the <WAS\_ROOT>\ISADC directory.
- Useful for diagnosing problems of over 50 WAS components.
- For SIB problems, choose options 'Connectors' than 'JMS Problems'.





# <WAS\_ROOT>\isadc\isadc.bat(.sh)

```
c:\Program Files (x86)\IBM\WebSphere\AppServer8.5.5\isadc>isadc
```

```
Starting IBM Support Assistant Data Collector in console mode... NOTE: On supported platforms you have the option to run data collection in a web browser. This can be done by opening the file index.html in a web browser. See the release notes for a list of platforms where data collection is supported in a web browser.
```

```
Licensed Materials - Property of IBM Corporation Copyright, International Business Machines, 2004, 2011. IBM is a trademark or registered trademark of IBM Corporation in the United States, other countries, or both.
```

```
ApplicationServer: 2.0.11.20140701  
IBM Support Assistant Data Collector:  
2.0.2.GA20140408-1111  
Common Inventory Sub Agent: 6.3.0.20120123
```

```
Provide a file name for saving the collected data or press enter to generate a unique file name:
```

```
> _
```

# <WAS\_ROOT>\isadc\index.html

[Feedback](#) [Help](#)

Select another type of collection you would like to perform from the options below.

- Default Collection
- General
- Administration
- Security
- Connectors
  - Database Connection Pooling Problem
- JMS Problem
- Containers
- JDK (Distributed Only)
- Runtime
- HTTP
- Service Oriented Architecture
- WAS\_INSTALL



Start

# Agenda

- Service Integration Bus: A quick overview
- Tools used for diagnostic data collection and analysis:
  - ▶ IBM Support Assistant for Data Collection (ISADC)
  - ▶ WAS Trace Analyzer
- Preparation for WAS Tracing
- SIB Trace Strings
- Tracing JFAP Channel and SIB Communications
- Tracing Messages
- Tracing SIB Database Interaction



# Trace Analyzer

- Trace Analyzer for WebSphere Application Server was initially created by IBM
- Trace Analyzer is not supported by IBM
- It is now available on the Internet, as is, at: [www.softpedia.com](http://www.softpedia.com)
- The tool is very easy to use, experimenting with the different search options is a good place to start
- It can also be used to search SystemOut.log files
- Trace Analyzer is still used internally by some IBMers



Trace Analyzer for WebSphere Application Server

FileViewRefineMarkupHelp

Stack

```

com.ibm.ws.sib.comms.client.ClientSideConnection.connect (com.ibm.ws.sib.comms.client.ClientSideConnection) [/3a2a1a2()]
com.ibm.ws.sib.comms.client.ClientSideConnection.mpfHandshakeExchange (com.ibm.ws.sib.comms.client.ClientSideConnection) [/3a2a1a2()]
com.ibm.ws.sib.comms.common.JFAPCommunicator.jfapExchange (com.ibm.ws.sib.comms.common.JFAPCommunicator) [/]
com.ibm.ws.sib.jfapchannel.impl.ConversationImpl.exchange (com.ibm.ws.sib.jfapchannel.impl.ConversationImpl) [/5120db33()]
com.ibm.ws.sib.jfapchannel.impl.ConversationImpl.sendInternal (com.ibm.ws.sib.jfapchannel.impl.ConversationImpl) [/5120db33()]
com.ibm.ws.sib.jfapchannel.impl.Connection.send (com.ibm.ws.sib.jfapchannel.impl.Connection) [/8623cb20()]
com.ibm.ws.sib.jfapchannel.impl.JFAPUtils

```

Unfiltered (6190)Filter (10)Filter (2)Filter (3730)Filter (1859)Filter (589)

Date/Time	Thread	Origin
[12/2/14 23:59:53:351]	00000036 >	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.RequestIdTableEntry.hashCode (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.RequestIdTableEntry)
[12/2/14 23:59:53:351]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.containsId (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:351]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.ConversationImpl.checkRequestNumberIsFree (com.ibm.ws.sib.jfapchannel.impl.ConversationImpl)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.comms.client.ClientJfapCommunicator.getRequestNumber (com.ibm.ws.sib.comms.client.ClientJfapCommunicator)
[12/2/14 23:59:53:352]	00000036 ?	com.ibm.ws.sib.comms.common.JFAPCommunicator
[12/2/14 23:59:53:352]	00000036 ?	com.ibm.ws.sib.comms.common.JFAPCommunicator
[12/2/14 23:59:53:352]	00000036 >	com.ibm.ws.sib.jfapchannel.impl.ConversationImpl.exchange (com.ibm.ws.sib.jfapchannel.impl.ConversationImpl)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.JFAPByteBuffer.setReadOnly (com.ibm.ws.sib.jfapchannel.JFAPByteBuffer) [/a6]
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.JFAPByteBuffer.setReadOnly (com.ibm.ws.sib.jfapchannel.JFAPByteBuffer) [/a6]
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.ExchangeReceiveListener.setExpectedRequestNumber (com.ibm.ws.sib.jfapchannel.impl.ExchangeReceiveListener)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.ExchangeReceiveListener.setExpectedRequestNumber (com.ibm.ws.sib.jfapchannel.impl.ExchangeReceiveListener)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.ConversationImpl.sendInternal (com.ibm.ws.sib.jfapchannel.impl.ConversationImpl)
[12/2/14 23:59:53:352]	00000036 ?	com.ibm.ws.sib.jfapchannel.JFAPByteBuffer
[12/2/14 23:59:53:352]	00000036 ?	com.ibm.ws.sib.jfapchannel.impl.ConversationImpl
[12/2/14 23:59:53:352]	00000036 ?	com.ibm.ws.sib.jfapchannel.impl.ConversationImpl
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.ConversationImpl.getDefaultReceiveListener (com.ibm.ws.sib.jfapchannel.impl.ConversationImpl)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.containsId (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.RequestIdTableEntry.hashCode (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.RequestIdTableEntry.hashCode (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.containsId (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.add (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.containsId (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.RequestIdTableEntry.hashCode (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.RequestIdTableEntry.hashCode (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.containsId (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.RequestIdTableEntry.<init> (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.RequestIdTableEntry.<init> (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.RequestIdTableEntry.hashCode (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.RequestIdTableEntry.hashCode (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.RequestIdTable.add (com.ibm.ws.sib.jfapchannel.impl.RequestIdTable)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.Connection.send (com.ibm.ws.sib.jfapchannel.impl.Connection) [/8623cb20]
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.ConversationImpl.getId (com.ibm.ws.sib.jfapchannel.impl.ConversationImpl)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.ConversationImpl.getId (com.ibm.ws.sib.jfapchannel.impl.ConversationImpl)
[12/2/14 23:59:53:352]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.JFAPUtils

Standard Search

Enter Search Details

Type

Thread

Method

Class

String

☐ Perform Exclusive Search
☒ Ignore case
☐ Fuzzy Method Search

Search

Clear

Cancel

Search(41)

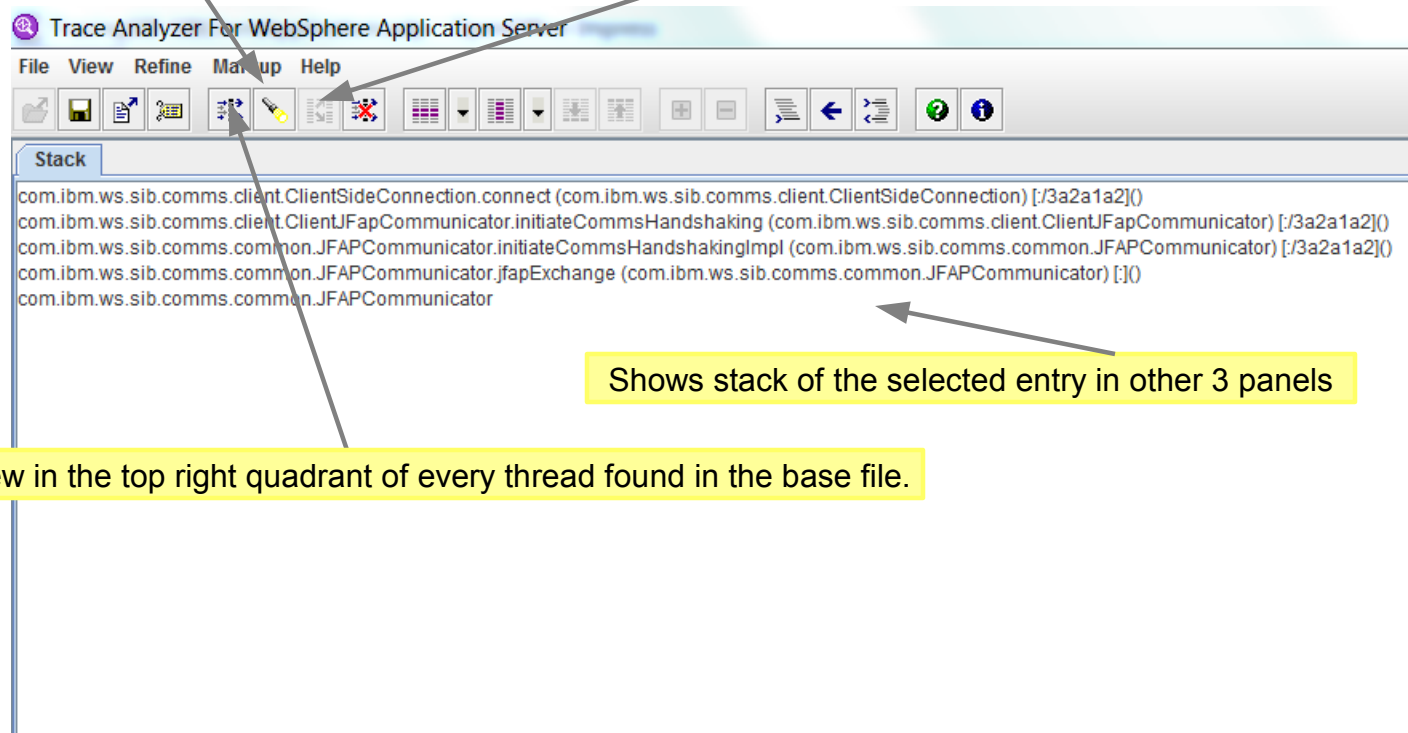
Date/Time	Thread	Origin
[12/2/14 23:59:52:511]	00000036 ?	com.ibm.ws.sib.comms.common.JFAPCommunicator
[12/2/14 23:59:52:524]	00000036 <	com.ibm.ws.sib.jfapchannel.impl.JFAPUtils
[12/2/14 23:59:52:823]	00000039 ?	com.ibm.ws.sib.jfapchannel.impl.JFAPUtils
[12/2/14 23:59:52:897]	00000036 ?	com.ibm.ws.sib.comms.common.JFAPCommunicator
[12/2/14 23:59:53:352]	00000036 ?	com.ibm.ws.sib.comms.common.JFAPCommunicator
[12/2/14 23:59:53:363]	00000036 ?	com.ibm.ws.sib.jfapchannel.impl.JFAPUtils
[12/2/14 23:59:53:477]	00000039 ?	com.ibm.ws.sib.jfapchannel.impl.JFAPUtils
[12/2/14 23:59:53:516]	00000036 ?	com.ibm.ws.sib.comms.common.JFAPCommunicator
[12/2/14 23:59:53:572]	00000036 ?	com.ibm.ws.sib.comms.common.JFAPCommunicator
[12/2/14 23:59:53:577]	00000036 ?	com.ibm.ws.sib.jfapchannel.impl.JFAPUtils
[12/2/14 23:59:53:665]	00000039 ?	com.ibm.ws.sib.jfapchannel.impl.JFAPUtils
[12/2/14 23:59:53:691]	00000039 ?	com.ibm.ws.sib.jfapchannel.impl.JFAPUtils
[12/2/14 23:59:53:730]	00000036 ?	com.ibm.ws.sib.comms.common.JFAPCommunicator

(com

# Trace Analyzer : Top left panel

Search

Filters the current view: useful to search within search result



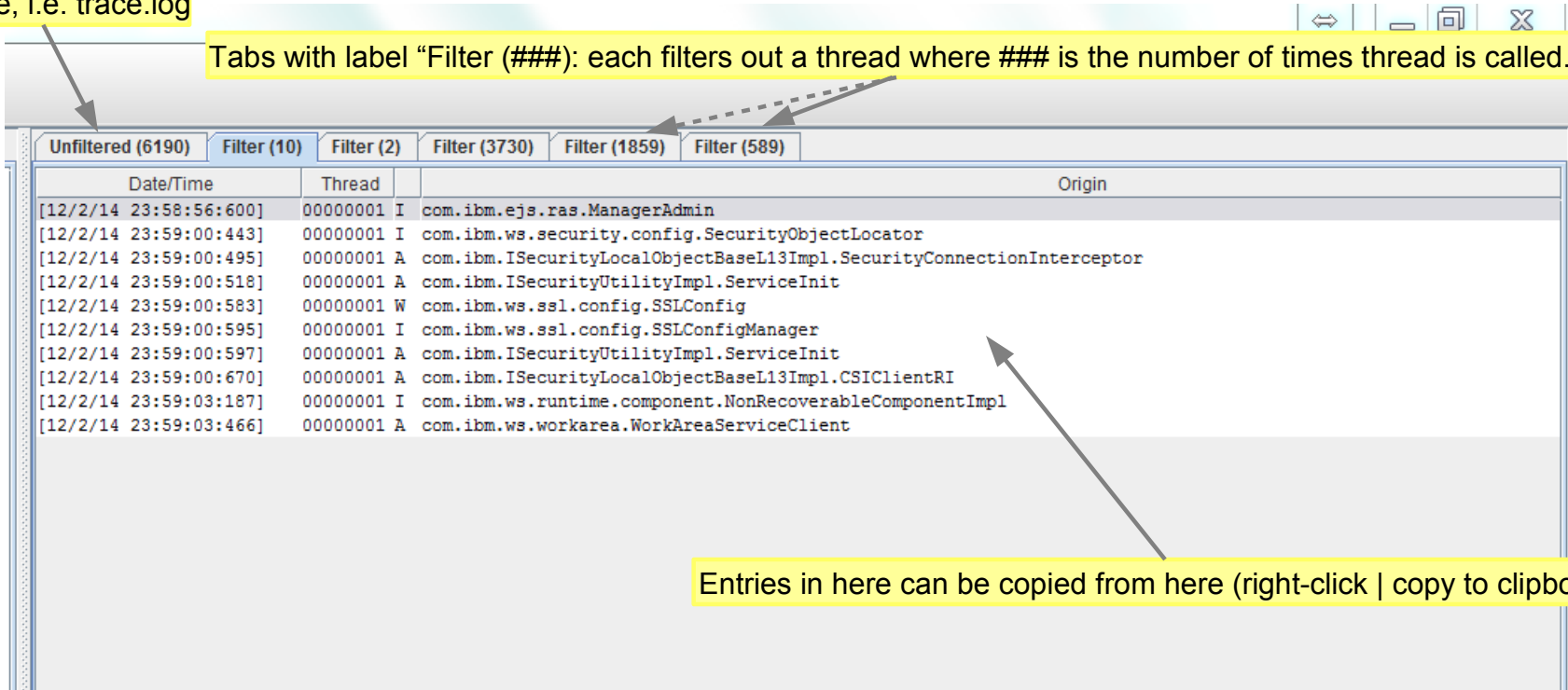
Shows stack of the selected entry in other 3 panels

Opens a view in the top right quadrant of every thread found in the base file.

# Trace Analyzer : Top right panel

Base file, i.e. trace.log

Tabs with label "Filter (###)": each filters out a thread where ### is the number of times thread is called.



The screenshot shows the Trace Analyzer interface with a list of threads. The tabs at the top are: Unfiltered (6190), Filter (10), Filter (2), Filter (3730), Filter (1859), and Filter (589). The table below has columns for Date/Time, Thread, and Origin. The threads listed are:

Date/Time	Thread	Origin
[12/2/14 23:58:56:600]	00000001 I	com.ibm.ejs.ras.ManagerAdmin
[12/2/14 23:59:00:443]	00000001 I	com.ibm.ws.security.config.SecurityObjectLocator
[12/2/14 23:59:00:495]	00000001 A	com.ibm.ISecurityLocalObjectBaseL13Impl.SecurityConnectionInterceptor
[12/2/14 23:59:00:518]	00000001 A	com.ibm.ISecurityUtilityImpl.ServiceInit
[12/2/14 23:59:00:583]	00000001 W	com.ibm.ws.ssl.config.SSLConfig
[12/2/14 23:59:00:595]	00000001 I	com.ibm.ws.ssl.config.SSLConfigManager
[12/2/14 23:59:00:597]	00000001 A	com.ibm.ISecurityUtilityImpl.ServiceInit
[12/2/14 23:59:00:670]	00000001 A	com.ibm.ISecurityLocalObjectBaseL13Impl.CSIClientRI
[12/2/14 23:59:03:187]	00000001 I	com.ibm.ws.runtime.component.NonRecoverableComponentImpl
[12/2/14 23:59:03:466]	00000001 A	com.ibm.ws.workarea.WorkAreaServiceClient

Entries in here can be copied from here (right-click | copy to clipboard)

# Trace Analyzer: Bottom left & right panels

Search tab gets created to display search result.

Displays stack results of a selected entry from the top right panel

THIS IS AN EDITABLE BOX: USER CAN PUT COMMENT(S) HERE.

```
(com.ibm.ws.sib.comms.common.JFAPCommunicator) [:/3a2a1a2] About to Exchange  
segment conversation: class  
com.ibm.ws.sib.jfapchannel.impl.ConversationImpl@1361107763 id: 1 first: true  
STATE: open connection: -2044474592 onClientSide: true SEG_HANDSHAKE - 6 (0x6)  
using request number 1
```

Date/Time	Thread	
[12/2/14 23:59:51:970]	00000036	> com.ibm.ws.sib.comms.client.ClientS
[12/2/14 23:59:52:424]	00000036	> com.ibm.ws.sib.comms.client.ClientJ
[12/2/14 23:59:52:424]	00000036	> com.ibm.ws.sib.comms.common.JFAPCom
[12/2/14 23:59:52:506]	00000036	> com.ibm.ws.sib.comms.common.JFAPCom
[12/2/14 23:59:52:511]	00000036	? com.ibm.ws.sib.comms.common.JFAPCom

Displays return value (result) of the selected entry in other panel. If no applicable return, the clock time it took for that call to execute is displayed.



# Agenda

- Service Integration Bus: A quick overview
- Tools used for diagnostic data collection and analysis:
  - ▶ IBM Support Assistant for Data Collection (ISADC)
  - ▶ WAS Trace Analyzer
- **Preparation for WAS Tracing**
- SIB Trace Strings
- Tracing JFAP Channel and SIB Communications
- Tracing Messages
- Tracing SIB Database Interaction



# Preparation for WAS Tracing

- Tracing records code actions
- All activity is stored on disk
- Potential Performance Hit
- Needs to run where, and when, the problem occurs
- Better to have too much trace than too little
- The default sizes for trace files may result in the files being overwritten
- They can be increased by navigating to this location in the admin console:
  - ▶ Troubleshooting > Logging and tracing > YOUR\_JVM > Diagnostic trace service

JVM™



# Trace File sizes

Configuration

**General Properties**

**Trace Output**

☐ None

☐ Memory Buffer

\* Maximum Buffer Size  
8 thousand entries

☒ File

\* Maximum File Size  
50 MB

\* Maximum Number of Historical Files  
10

\* File Name  
\${SERVER\_LOG\_ROOT}/trace.log

**Trace Output Format**  
Basic (Compatible)

Apply OK Reset Cancel

# JVM Logs

- JVM logs should be set so that they will correspond to whatever happens in the Trace.
- Use the same settings: 10 files at 50 MB each
  - ▶ Troubleshooting > Logging and tracing > JVM\_NAME > JVM Logs



# Entering Trace Strings

- In the admin console, navigate to:  
**Troubleshooting > Logging and tracing > JVM\_NAME > Change log detail levels**
- Select either Configuration or Runtime tab and enter the string in the box

Configuration Runtime

### General Properties

#### Change log detail levels

☐ Disable logging and tracing of potentially sensitive data (WARNING: This might cause the log detail level setting to be modified when it is applied on the server.)

Select components and specify a log detail level. Log detail levels specified here will apply to the entire server. Expand Components and Groups and click Components to specify a log detail level for individual components, or click Groups to specify a log detail level for a predefined group of components. Click a component or group name to select a log detail level. Log detail levels are cumulative.

`*=info:SIBMessageTrace=all:SIBMessageStore=all:SIBProcessor=all`

[+ Components and Groups](#)

# IBM Level 2 Support

- If you need L2 support the SIB team has the following recommendations:
- Use 50 MB as your standard trace size when submitting to IBM
- Be aware that Level 3 support may still ask for files up to 200 MB in size
- The best link to use when submitting files to IBM:  
<https://www.ecurep.ibm.com/app/upload>
- Always send complete files, never edit them
- Let IBM decide what should be sent



# Agenda

- Service Integration Bus: A quick overview
- Tools used for diagnostic data collection and analysis:
  - ▶ IBM Support Assistant for Data Collection (ISADC)
  - ▶ WAS Trace Analyzer
- Preparation for WAS Tracing
- **SIB Trace Strings**
- Tracing JFAP Channel and SIB Communications
- Tracing Messages
- Tracing SIB Database Interaction



# SIB Trace Strings

- Trace strings tell WebSphere what code action to record. For example:
  - ▶ `SIBSecurity=all:SIBJms*=all`
- The above string records all SIB security code actions and anything which begins with SIBJms, such as: SIBJmsRa and SIBJms\_External
- The strings are separated by a “:”
- `*=info` is automatically pre-pended, so the above string becomes `*=info:SIBSecurity=all:SIBJms*=all`





# More Trace Strings

- The SIB mustgather site contains instructions, which include trace strings, for many aspects of SIB
- Check out the website:  
<http://www.ibm.com/support/docview.wss?uid=swg21266769>



# Message Prefixes 1

Message prefix	SIB Component	Trace String
CWSIA	API	SIBJms*=all
CWSIB	Message formatting and parsing core	SIBMfpApi=all
CWSIC	Communications	SIBCommunications*=all
CWSID	Administration and system management	SIBAdmin=all
CWSIE	Message formatting and parsing SPI	SIBMfpApi=all
CWSIF	Message formatting and parsing	SIBMfp*=all
CWSIG	Example	SIBExample=all
CWSIH	Match space	SIBMatchSpace=all
CWSII	Security	SIBSecurity=all
CWSIJ	Communications formats and protocol	SIBJFapChannel=all

# Message Prefixes 2

Message prefix	SIB Component	Trace String
CWSIA	API	SIBJms*=all
CWSIB	Message formatting and parsing core	SIBMfpApi=all
CWSIC	Communications	SIBCommunications*=all
CWSID	Administration and system management	SIBAdmin=all
CWSIE	Message formatting and parsing SPI	SIBMfpApi=all
CWSIF	Message formatting and parsing	SIBMfp*=all
CWSIG	Example	SIBExample=all
CWSIH	Match space	SIBMatchSpace=all
CWSII	Security	SIBSecurity=all
CWSIJ	Communications formats and protocol	SIBJFapChannel=all

# Message Prefixes 3

Message prefix	SIB Component	Trace String
CWSIK	Common messages	SIBAdmin=all:SIBMfp*=all: SIBProcessor=all
CWSIL	Publish and subscribe bridge	SIBPsb=all
CWSIM	Mediations	SIBMediations=all
CWSIN	Mediation services	SIBMediations=all
CWSIO	Administration migration	SIBMigrationUtil=all
CWSIP	Message processor	SIBProcessor=all
CWSIQ	MQ formats and protocol	SIBMqFapChannel=all
CWSIR	Core programming interface	SIBProcessor=all
CWSIS	Message store	SIBMessageStore=all
CWSIT	Topology routing and management	SIBTrm=all



# Message Prefixes 4

Message prefix	SIB Component	Trace String
CWSIU	Utilities	SIBUtils=all
CWSIV	Resource adapter	SIBJmsRaCommon=all
CWSIX	Core beans	Not traceable at present
CWSIY	Mediation handler framework	SIBMediations=all
CWSIZ	Mediation framework	SIBMediations=all
CWSJA	Administration commands	SIBAdmin=all
CWSJB	Inter-bus link	SIBIbl=all
CWSJC	Core selector	Check with SIB Level 2 Support
CWSJD	Administration security	SIBSecurity=all

# Message Prefixes 5

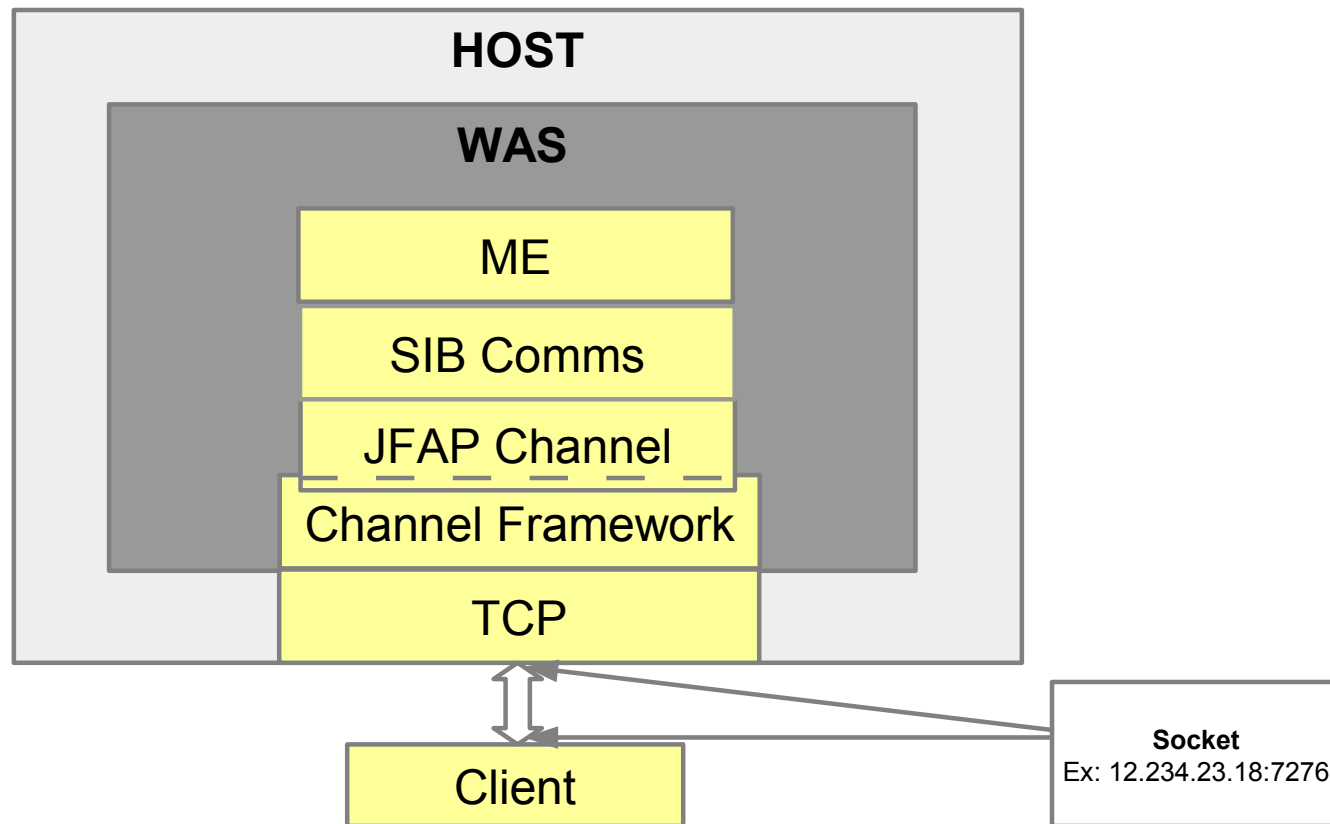
Message prefix	SIB Component	Trace String
CWSJO	Service Data Objects configuration	SIBSdoRepository=all
CWSJQ	Message formatting and parsing MQ interoperability	SIBMfpMq=all
CWSJR	Resource adapter (JMS)	SIBJmsRa=all
CWSJU	Message tracing	SIBProcessor=all
CWSJW	WLM classifier for z/OS	SIBWlmClassifier=all

# Agenda

- Service Integration Bus: A quick overview
- Tools used for diagnostic data collection and analysis:
  - ▶ IBM Support Assistant for Data Collection (ISADC)
  - ▶ WAS Trace Analyzer
- Preparation for WAS Tracing
- SIB Trace Strings
- **Tracing JFAP Channel and SIB Communications**
- Tracing Messages
- Tracing SIB Database Interaction



# Networking Services, Protocol and I/O operations between client and ME





# Test Environment

- Client: Launched SIB client application that:
  - ▶ Creates a physical connection
  - ▶ Creates a producer session,
  - ▶ Sends 3 messages in a row and then
  - ▶ Closes everything.
- ME server: ME (non-local to the client) runs on a server bus member.
- Trace specifications used on both sides:

\*=info:SIBCommunications=all:SIBJFapChannel=all



# Trace Strings

- Trace groups used on both ends:

`*=info:SIBJFapChannel=all:SIBCommunications=all`

- ▶ JFAP Exchange: Tracking physical connection and its details, conversations and, requests.
- ▶ SIB Comms: Tracking communications between the JfapChannel and the ME



# Trace Strings - continue

- JFapChannel trace group tracks the actual connection between the ME and the client:
  - ▶ JFAP: Communications between a SIB application and a remote ME are supported by the WAS channel framework via the JMS Format And Protocol (JFAP).
- SIBCommunications trace group (SIB comms) is the layer that sits between JFAP and the ME:
  - ▶ As far as message processing, the ME always sees the client as a local client.



# JFap Exchange: On the client side

- Using Trace Analyzer, the best string to start your search with is: "SEG\_"
- These are all communications (information exchange/commands) going on between the client and the ME to establish a connection, create/close producer/consumer session, send a message, get a response back and close the connection... : SEG\_HANDSHAKE, SEG\_MESSAGE\_FORMAT\_INFO, SEG\_CONNECTION\_INFO, SEG\_TOPOLOGY, SEG\_CREATE\_PRODUCER\_SESS\_R, SEG\_CREATE\_PRODUCER\_SESS

<pre>(com.ibm.ws.sib.comms.common.JFAPCommunicator) [:/3a2a1a2] About to Exchange segment conversation: class com.ibm.ws.sib.jfapchannel.impl.ConversationImpl@1361107763 id: 1 first: true STATE: open connection: -2044474592 onClientSide: true SEG_HANDSHAKE - 6 (0x6) using request number 1</pre>		<b>Search(41)</b>	
Date/Time	Thread		
[12/2/14 23:59:52:511]	00000036 ?	com.ibm.ws.sib.comms.common.JFAPCommunicator	
[12/2/14 23:59:52:524]	00000036 ?	com.ibm.ws.sib.jfapchannel.impl.JFapUtils	
[12/2/14 23:59:52:823]	00000039 ?	com.ibm.ws.sib.jfapchannel.impl.JFapUtils	
[12/2/14 23:59:52:897]	00000036 ?	com.ibm.ws.sib.comms.common.JFAPCommunicator	
[12/2/14 23:59:53:352]	00000036 ?	com.ibm.ws.sib.comms.common.JFAPCommunicator	

# Tracking a Request

```
[12/2/14 23:59:52:316] 00000036 com.ibm.ws.sib.jfapchannel.impl.JFapUtils  
[(com.ibm.ws.sib.jfapchannel.impl.JFapUtils) [:] {JFAPSUMMARY}  
[9.65.164.108:60491:9.65.164.108:7276:-1] 8623cb20[OUT 9.65.164.108:60491-  
>9.65.164.108:7276, closeDeferred=false, invalidateDeferred=false]:null[] New outbound connection  
established]
```

...

```
[12/2/14 23:59:53:934] 00000036 com.ibm.ws.sib.jfapchannel.impl.JFapUtils  
(com.ibm.ws.sib.jfapchannel.impl.JFapUtils) [:] {JFAPSUMMARY}  
[9.65.164.108:60491:9.65.164.108:7276:1:5] 8623cb20[CLIENT-ME OUT 9.65.164.108:60491-  
>9.65.164.108:7276, closeDeferred=false, invalidateDeferred=false]:5120db33[ConvId:1] Exchange  
[Request Id:5] 48 (SEG_CREATE_PRODUCER_SESS)
```

- JFAPSUMMARY information shows the following:
  - ▶ “9.65.164.108:60491:9.65.164.108:7276” – The actual physical connection (socket to socket)
  - ▶ “9.65.164.108”: IP address of the client, same as that of the ME.
  - ▶ “60491”: Port used by the client.
  - ▶ “7276”: Port on ME side, the SIB\_ENDPOINT\_ADDRESS port.
  - ▶ 1: Conversation id
  - ▶ 5: Request id



# Example Request: Producer session

- [12/2/14 23:59:53:928] **00000036** com.ibm.ws.sib.comms.common.JFAPCommunicator (com.ibm.ws.sib.comms.common.JFAPCommunicator) [:/2b0e1b19] About to **Exchange** segment conversation: class com.ibm.ws.sib.jfapchannel.impl.ConversationImpl@1361107763 id: 1 first: true STATE: open connection: -2044474592 **onClientSide: true SEG\_CREATE\_PRODUCER\_SESS** - 72 (0x48) using **request number 5**
- [12/2/14 23:59:53:934] 00000036 com.ibm.ws.sib.jfapchannel.impl.JFapUtils (com.ibm.ws.sib.jfapchannel.impl.JFapUtils) [:] {JFAPSUMMARY}[9.65.164.108:60491:9.65.164.108:7276:1:5] 8623cb20[CLIENT-ME OUT 9.65.164.108:60491->9.65.164.108:7276, closeDeferred=false, invalidateDeferred=false]:5120db33[ConvId:1] Exchange [Request Id:5] 48 (SEG\_CREATE\_PRODUCER\_SESS)
- [12/2/14 23:59:53:982] **00000039** com.ibm.ws.sib.jfapchannel.impl.JFapUtils (com.ibm.ws.sib.jfapchannel.impl.JFapUtils) [:] {JFAPSUMMARY}[9.65.164.108:60491:9.65.164.108:7276:1:5] 8623cb20[CLIENT-ME OUT 9.65.164.108:60491->9.65.164.108:7276, closeDeferred=false, invalidateDeferred=false]:5120db33[ConvId:1] received conversation data with segment c8 (SEG\_CREATE\_PRODUCER\_SESS\_R)
- [12/2/14 23:59:53:992] 00000036 com.ibm.ws.sib.comms.common.JFAPCommunicator (com.ibm.ws.sib.comms.common.JFAPCommunicator) [:/2b0e1b19] **Exchange completed** successfully. Segment returned SEG\_CREATE\_PRODUCER\_SESS\_R - 200 (0xc8)

# JFap Exchange: On the ME side

- Using Trace Analyzer, the best string to start your search with is the string “rcv” to see what the ME has received from the client.
- For my simple client application, I got 31 hits in a matter of a fraction of second.
- Look for the entry of method call “rcvCreateOrderContext” to view the new established connection (see bottom-left panel of the Trace Analyzer):

```
(com.ibm.ws.sib.comms.server.clientsupport.StaticCATConnection) [:] Entry
    JFapByteBuffer@e24ecae6: { valid=false, released=false, dataList=[], receivedData=null,
receivedBuffer=RichByteBufferImpl@c9aa64c8: PooledWsByteBufferImpl: ID: 33
java.nio.HeapByteBuffer[pos=0 lim=2 cap=32] Owner Count: 1 From pool: WSByteBufferPool:
com.ibm.ws.buffermgmt.impl.WsByteBufferPool@2ba1756a buffer size: 32 }
    class com.ibm.ws.sib.jfapchannel.impl.ConversationImpl@1275416129 id: 1 first: true
STATE: open connection: 1355265371 onClientSide: false
4
```

true



# Trace Analyzer Search Engine

Stack

ReceiveListen.run()  
ConversationR.invoke()  
GenericTransp.dataReceived()  
GenericTransp

Unfiltered (5508)

Date/Time	Thread	Origin
[12/2/14 23:59:52:700]	000000c1 ?	GenericTransp
[12/2/14 23:59:52:700]	000000bf <	InboundTransm.parse ()
[12/2/14 23:59:52:700]	000000c1 >	ServerTransp.acceptConnection ()
[12/2/14 23:59:52:700]	000000bf >	Connection.getHeartbeatInterval ()
[12/2/14 23:59:52:701]	000000c1 >	ConversationI.getConnectionReference ()
[12/2/14 23:59:52:701]	000000bf <	Connection.getHeartbeatInterval ()
[12/2/14 23:59:52:701]	000000c1 <	ConversationI.getConnectionReference ()
[12/2/14 23:59:52:701]	000000bf ?	ConnectionRea
[12/2/14 23:59:52:701]	000000bf <	RichByteBuffe.release ()
[12/2/14 23:59:52:701]	000000bf <	RichByteBuffe.release ()
[12/2/14 23:59:52:701]	000000c1 >	ConversationI.getConnectionClosedListener ()
[12/2/14 23:59:52:701]	000000bf >	WsByteBufferP.getInstance ()
[12/2/14 23:59:52:701]	000000c1 >	Connection.getConnectionClosedListener ()
[12/2/14 23:59:52:701]	000000bf <	WsByteBufferP.getInstance ()
[12/2/14 23:59:52:701]	000000c1 <	Connection.getConnectionClosedListener ()
[12/2/14 23:59:52:701]	000000bf <	RichByteBuffe.allocateDirect ()
[12/2/14 23:59:52:701]	000000c1 <	ConversationI.getConnectionClosedListener ()
[12/2/14 23:59:52:702]	000000c1 >	ConversationI.addConnectionClosedListener ()
[12/2/14 23:59:52:702]	000000bf >	RichByteBuffe.getFromPool ()
[12/2/14 23:59:52:702]	000000c1 >	Connection.addConnectionClosedListener ()
[12/2/14 23:59:52:702]	000000bf <	RichByteBuffe.getFromPool ()
[12/2/14 23:59:52:702]	000000c1 <	Connection.addConnectionClosedListener ()
[12/2/14 23:59:52:702]	000000bf <	RichByteBuffe.allocateDirect ()
[12/2/14 23:59:52:702]	000000c1 <	ConversationI.addConnectionClosedListener ()
[12/2/14 23:59:52:702]	000000bf >	CFWIOReadRequ.setBuffer ()
[12/2/14 23:59:52:702]	000000c1 >	ConversationI.getAttachment ()
[12/2/14 23:59:52:702]	000000bf <	CFWIOReadRequ.setBuffer ()
[12/2/14 23:59:52:702]	000000c1 <	ConversationI.getAttachment ()
[12/2/14 23:59:52:702]	000000bf <	CFWIOReadRequ.read ()
[12/2/14 23:59:52:702]	000000c1 ?	ServerTransp
[12/2/14 23:59:52:715]	000000bf <	CFWIOReadRequ.read ()
[12/2/14 23:59:52:715]	000000bf >	ConversationI.getConversationType ()
[12/2/14 23:59:52:715]	000000c1 ?	ConversationS
[12/2/14 23:59:52:715]	000000bf >	Connection.getConversationType ()
[12/2/14 23:59:52:715]	000000bf <	Connection.getConversationType ()
[12/2/14 23:59:52:715]	000000c1 >	ConversationS.<init> ()

Search(2)

Search(31)

(com.ibm.ws.sib.comms.server.GenericTransportReceiveListener) [:d18f7bbd] class

com.ibm.ws.sib.jfapchannel.impl.ConversationImpl[1275416129 id: 1 first: true

STATE: open connection: 1355265371 onClientSide: false 1 conversation events

recorded in total

timestamp/sequence/thread/type/description

23:59:52:596 0 e06Sec79D << Data rcvd: Segment 6 (0x6), Request No: 1

Date/Time	Search For (string=rcv)	Origin
[12/2/14 23:59:52:700]	000000c1 ? GenericTransp	(com.ibm.w
[12/2/14 23:59:52:727]	000000c1 ? ServerTransp	(com.ibm.w
[12/2/14 23:59:52:727]	000000c1 > CommonServerR.rcvHandshake ()	(com.ibm.w
[12/2/14 23:59:52:787]	000000c1 < CommonServerR.rcvHandshake ()	(com.ibm.w
[12/2/14 23:59:53:436]	000000c2 ? ServerTransp	(com.ibm.w
[12/2/14 23:59:53:436]	000000c2 > ServerTransp	(com.ibm.w



# Trace Analyzer Search Engine - Cont

```

[12/2/14 23:59:53:842] 000000c2 < CommsByteBuff.<init>() (com.ibm.ws.sib.comms.common.CommsByteBufferPool) [:/565edcbb
[12/2/14 23:59:53:842] 000000c2 > StaticCATConn.rcvCreateOrderContext() (com.ibm.ws.sib.comms.server.clientsupport.StaticCATConnection
[12/2/14 23:59:53:842] 000000c2 > ConversationI.getAttachment() (com.ibm.ws.sib.jfapchannel.impl.ConversationImpl) [:/4c054e4:
[12/2/14 23:59:53:842] 000000c2 < ConversationI.getAttachment() (com.ibm.ws.sib.jfapchannel.impl.ConversationImpl) [:/4c054e4:
[12/2/14 23:59:53:842] 000000c2 ? StaticCATConn (com.ibm.ws.sib.comms.server.clientsupport.StaticCATConnection
[12/2/14 23:59:53:842] 000000c2 > ConversationS.getObject() (com.ibm.ws.sib.comms.server.ConversationState) [:/a87482b4] !
[12/2/14 23:59:53:842] 000000c2 < ConversationS.getObject() (com.ibm.ws.sib.comms.server.ConversationState) [:/a87482b4] !

```

Search(2)		Search(31)	Stack(4)	Search(2)
Date/Time		Thread	Search For (string=rcvCreateOrderContext)	
[12/2/14 23:59:53:842]		000000c2 >	StaticCATConn.rcvCreateOrderContext() (com.ibm.ws.sib.)	
[12/2/14 23:59:53:854]		000000c2 <	StaticCATConn.rcvCreateOrderContext() (com.ibm.ws.sib.)	

```

(com.ibm.ws.sib.comms.server.clientsupport.StaticCATConnection) [:] Entry
    JFapByteBuffer@e24ecae6: { valid=false,
released=false, dataList=[], receivedData=null,
receivedBuffer=RichByteBufferImpl@c9aa64c8: PooledWsByteBufferImpl: ID: 33
java.nio.HeapByteBuffer[pos=0 lim=2 cap=32] Owner Count: 1 From pool:
WSByteBufferPool: com.ibm.ws.buffermgmt.impl.WsByteBufferPool@2ba1756a buffer
size: 32 }

    class
com.ibm.ws.sib.jfapchannel.impl.ConversationImpl@1275416129 id: 1 first: true
STATE: open connection: 1355265371 onClientSide: false
    4
    true

```

# Tracking Request on ME Side

- The {JFAPSUMMARY} information on the client side can be used for tracking purposes on the ME side:
  - ▶ Example: Tracking a producer request:  
“9.65.164.108:60491:9.65.164.108:7276:1:5”
  - ▶ Or the whole conversation:  
“9.65.164.108:60491:9.65.164.108:7276:1”



# Tracking Request on ME Side - Cont

- Searching for “9.65.164.108:60491:9.65.164.108:7276:1:5” (obtained from client side), we get 2 hits:
  - ▶ JFapchannel side receiving the create producer session command:

```
[12/2/14 23:59:53:954] 000000bf JFapUtils [(com.ibm.ws.sib.jfapchannel.impl.JFapUtils) [:]  
{JFAPSUMMARY}[9.65.164.108:60491:9.65.164.108:7276:1:5] 50c7b55b[CLIENT-ME IN  
9.65.164.108:7276<-9.65.164.108:60491, closeDeferred=false, invalidateDeferred=false  
]:4c054e41[ConvId:1] received conversation data with segment 48  
(SEG_CREATE_PRODUCER_SESS)]
```

- ▶ and that gets dispatched to another thread (000000c2).

```
[12/2/14 23:59:53:968] 000000c2 JFapUtils [(com.ibm.ws.sib.jfapchannel.impl.JFapUtils) [:]  
{JFAPSUMMARY}[9.65.164.108:60491:9.65.164.108:7276:1:5] 50c7b55b[CLIENT-ME IN  
9.65.164.108:7276<-9.65.164.108:60491, closeDeferred=false, invalidateDeferred=false  
]:4c054e41[ConvId:1] Send [Request Id:5] c8 (SEG_CREATE_PRODUCER_SESS_R)]
```

...



# What is a “Local” ME?

- If the client application connects to a “local” ME, expect to see JMS and RA calls instead of Jfap, until this very call called 'createProducerSession()'.
- The ME is considered local to the client application if the latter is deployed in the same server when that ME runs.



# ME Communications (SIB Comms)

- The ME comms trace group (SIBCommunications=all) should allow you to see your destination being resolved:

```
[12/2/14 23:59:53:964] 000000c2 CommsByteBuff  
(com.ibm.ws.sib.comms.common.CommsByteBuffer) [:] Destination name:  
Qnaj01  
[12/2/14 23:59:53:964] 000000c2 CommsByteBuff.getSIDestinationAddress()  
[(com.ibm.ws.sib.comms.common.CommsByteBuffer) [:/3a569d4a] Exit]  
[12/2/14 23:59:54:215] 000000c2 StaticCATProd  
[(com.ibm.ws.sib.comms.server.clientsupport.StaticCATProducer) [:] Destination: [ QNaj01, false, null,  
null ]]...
```

- CommsByteBuff call extends a JFap particular method to allow many useful methods that Comms requires for putting data into buffers for transmission or reception. It can also be used to put / get Strings, Xld's, transaction information, messages of all kinds, destination addresses and selection criterias ...ect.
- StaticCATProducer class takes responsibility for dealing with all FAP flows relating to message producers.



# Agenda

- Service Integration Bus: A quick overview
- Tools used for diagnostic data collection and analysis:
  - ▶ IBM Support Assistant for Data Collection (ISADC)
  - ▶ WAS Trace Analyzer
- Preparation for WAS Tracing
- SIB Trace Strings
- Tracing JFAP Channel and SIB Communications
- **Tracing Messages**
- Tracing SIB Database Interaction



# Tracing Messages

- We will now review the following:
- Messages going to a local ME
- Messages going to a remote ME
- Messages going through a bridge
- Identifying a specific message



# Tracing Messages

- Publish/Subscribe Test Environment
- JVMs: PubSubServer1 (PS1), PubSubServer2 (PS2), PubSubServer3 (PS3)
- The topic destination is: JMSMonitorTopicSpace
- PS1 and PS2 are on a different bus than PS3
- The PubSub test application used for this presentation publishes a message on PS1, and the subscriber application prints a message in SystemOut.log on PS3
- Trace string:  
**SIBCommunications=all:SIBMessageTrace=all:SIBProcessor=all**





# Tracing Messages

- When a message is put to a local destination several UserTrace messages are generated
- One or more of those messages will have a system message ID (SysMsgId)
- A SysMsgId can be used to track a message throughout a SIB network
- Sample SysMsgId: **395540756E349BC6\_34500004**



# Tracing - Messages

## ■ Test results

### Important messages on PS1

(com.ibm.ws.sib.processor.utils.UserTrace) [:] **CWSJU0021I**: A message with ID ID:855e785fd2e9a48ed3a0ec59110a134f00000000000000001 **system message ID 395540756E349BC6\_34500004** and correlation ID null is transmitting to messaging engine with ID RAMANDUNode\_01.**PubSubServer2**-GEMS for destination **JMSMonitorTopicSpace**

(com.ibm.ws.sib.processor.impl.**PubSubOutputHandler**) [:] Entry  
MessageItem@d594765c[395540756E349BC6\_34500004,ID:855e785fd2e9a48ed3a0ec59110a134f00000000000000001,69000017,'**My message ==> 1**']



# Tracing - Messages

## Important messages on PS2:

com.ibm.ws.sib.processor.utils.UserTrace) [:] CWSJU0020I: A message with ID ID:855e785fd2e9a48ed3a0ec59110a134f00000000000000001 system message ID **395540756E349BC6\_34500004** and correlation ID null **was received** from the messaging engine with ID 395540756E349BC6 with a destination **JMSMonitorTopicSpace**

(com.ibm.ws.sib.processor.impl.**PubSubOutputHandler**) [:] Entry MessageItem@f8733b89[395540756E349BC6\_34500004,ID:855e785fd2e9a48ed3a0ec59110a134f00000000000000001,-1,'**My message ==> 1**']



# Tracing - Messages

## Important messages on PS3

12/3/14 19:36:46:578 MST 0000008f I SibMessage [:]  
CWSIV0777I: A connection to messaging engine  
RAMANDUNode\_01.**PubSubServer2**-GEMS for destination  
**JMSMonitorTopicSpace** on bus GEMS has been successfully created.

12/3/14 19:39:33:194 MST 000000c7 < CommsByteBuff.getMessage  
(com.ibm.ws.sib.comms.common.CommsByteBuffer) [:] Exit  
[com.ibm.ws.sib.mfp.impl.JsJmsTextMessageImpl@76ee7d8a](#){**SysMsgId=3955407**  
**56E349BC6\_34500004**}

12/3/14 19:39:33:404 MST 000000cf SystemOut  
CONSUMED MESSAGE ==> **My message** ==> 1

# Agenda

- Service Integration Bus: A quick overview
- Tools used for diagnostic data collection and analysis:
  - ▶ IBM Support Assistant for Data Collection (ISADC)
  - ▶ WAS Trace Analyzer
- Preparation for WAS Tracing
- SIB Trace Strings
- Tracing JFAP Channel and SIB Communications
- Tracing Messages
- **Tracing SIB Database Interaction**



# Tracing – Message Stores

- Every ME has a message store
- A message store can be either a data store, which uses a commercial database product such as DB2, or a file store which uses three flat files
- The default is to use a file store



# Tracing – Message Stores

## Data Stores

- SIB uses SQL when working with a database
- Use **SIBMessageStore=all** to capture the SQL statements in a trace



# Tracing – Message Stores

## File Stores

- Use this trace string: `SIBMessageTrace=all`

- Watch for UserTrace messages:

(com.ibm.ws.sib.processor.utils.UserTrace) [:] CWSJU0004I: A message with ID ID:414d5120514d30312020202020202020205f7280542000be02, system message ID D4ACA879059BD2C5\_2000003 and correlation ID null is **put to queue**  
**MQ.GATEWAY.OM.IN**

(com.ibm.ws.sib.processor.utils.UserTrace) [:/4015a55b] CWSJU0003I: A message with ID ID:414d5120514d30312020202020202020205f7280542000be02, system message ID D4ACA879059BD2C5\_2000003 and correlation ID null is **committed to destination**  
**MQ.GATEWAY.OM.IN**, which is targetted for messaging engine  
RAMANDUNode\_01.OMSERVER-BUS1.



# Summary

- We have discussed the basics of tracing SIB communications, messaging and message stores
- We have introduced the Trace Analyzer
- The skills acquired today apply to other components of WebSphere Application Server, the major difference being the trace strings used
- These skills will make you less dependent on IBM and give you more flexibility in managing your Websphere installation



# Connect with us!

## 1. Get notified on upcoming webcasts

Send an e-mail to [wsehelp@us.ibm.com](mailto:wsehelp@us.ibm.com) with subject line “wste subscribe” to get a list of mailing lists and to subscribe

## 2. Tell us what you want to learn

Send us suggestions for future topics or improvements about our webcasts to [wsehelp@us.ibm.com](mailto:wsehelp@us.ibm.com)



# Questions and Answers



# Additional WebSphere Product Resources

- Learn about upcoming WebSphere Support Technical Exchange webcasts, and access previously recorded presentations at:  
[http://www.ibm.com/software/websphere/support/supp\\_tech.html](http://www.ibm.com/software/websphere/support/supp_tech.html)
- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at:  
<http://www.ibm.com/developerworks/websphere/community/>
- Join the Global WebSphere Community:  
<http://www.websphereusergroup.org>
- Access key product show-me demos and tutorials by visiting IBM Education Assistant:  
<http://www.ibm.com/software/info/education/assistant>
- View a webcast replay with step-by-step instructions for using the Service Request (SR) tool for submitting problems electronically:  
<http://www.ibm.com/software/websphere/support/d2w.html>
- Sign up to receive weekly technical My Notifications emails:  
<http://www.ibm.com/software/support/einfo.html>

