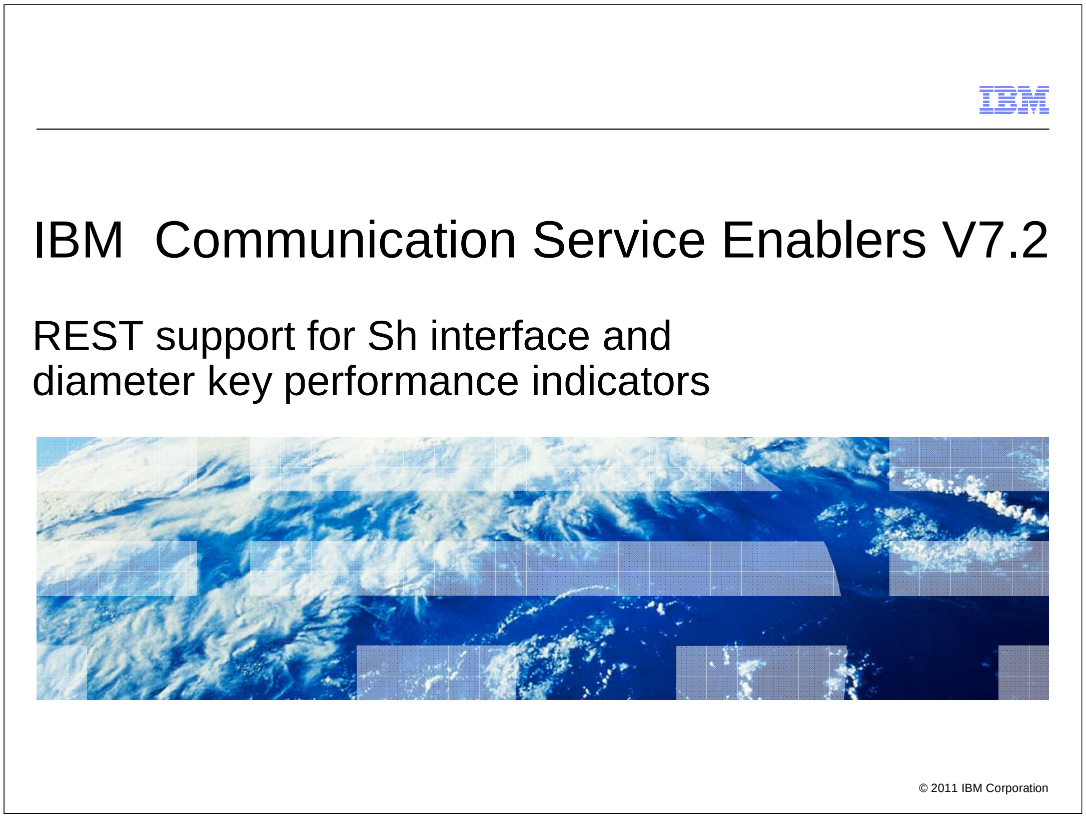


IBM Communication Service Enablers V7.2

REST support for Sh interface and diameter key performance indicators

This presentation deals with REST support for Sh Interface and with Key Performance Indicators (KPIs) for Diameter.

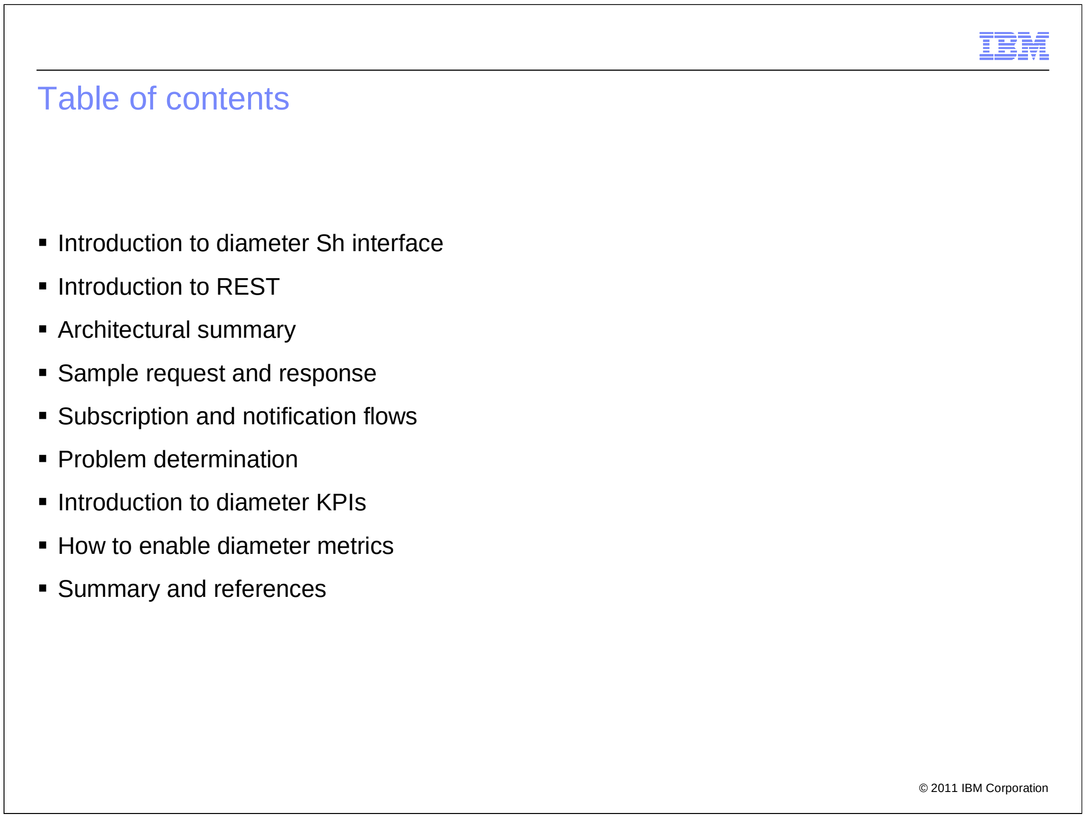


Table of contents

- Introduction to diameter Sh interface
- Introduction to REST
- Architectural summary
- Sample request and response
- Subscription and notification flows
- Problem determination
- Introduction to diameter KPIs
- How to enable diameter metrics
- Summary and references

The agenda includes:

- Introduction to Diameter Sh Interface

- Introduction to the REST

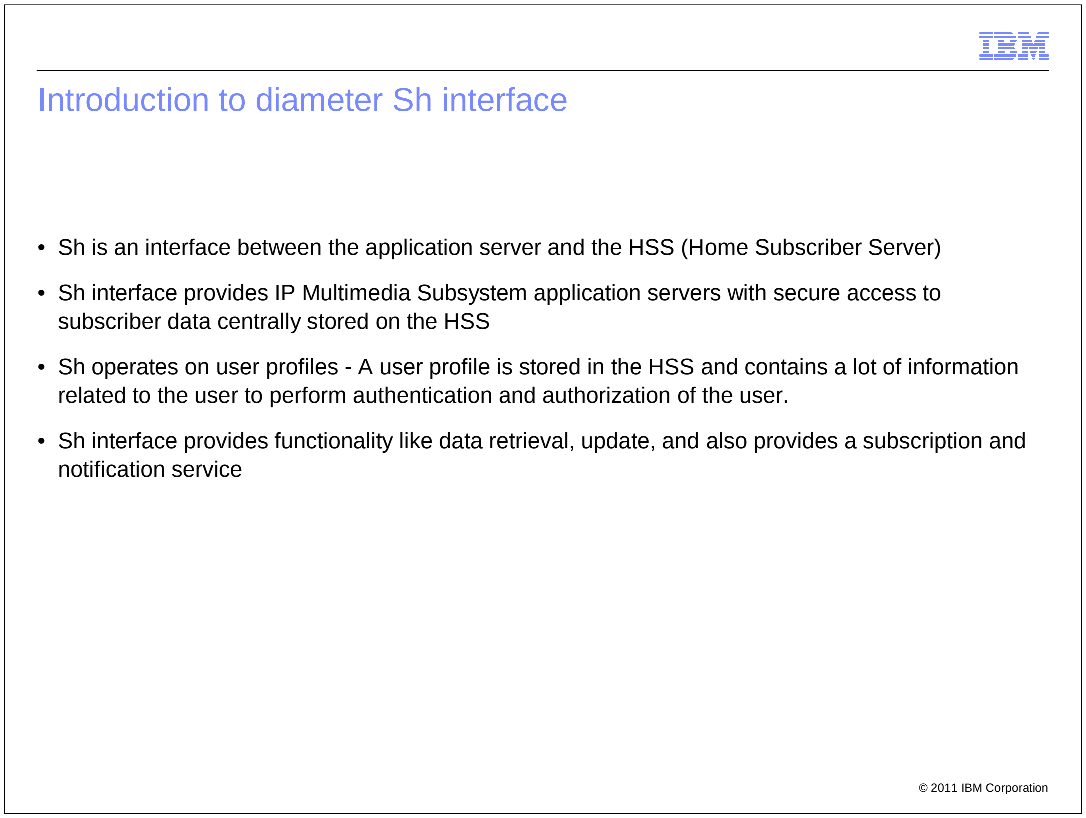
- Architectural Summary

- Sample request and response for the web service and the notification flows as well

- Problem Determination

- Introduction to Diameter Key Performance Indicators (KPIs)

- How to enable Diameter KPIs with focus on the metrics enabled in this release



Introduction to diameter Sh interface

- Sh is an interface between the application server and the HSS (Home Subscriber Server)
- Sh interface provides IP Multimedia Subsystem application servers with secure access to subscriber data centrally stored on the HSS
- Sh operates on user profiles - A user profile is stored in the HSS and contains a lot of information related to the user to perform authentication and authorization of the user.
- Sh interface provides functionality like data retrieval, update, and also provides a subscription and notification service

This slide provides the introduction to the Diameter interface. Sh is the Diameter interface between the application server and the Home Subscriber Server.

In IP Multimedia Subsystem parlance, the application server resides in the Service Plane and the Home Subscriber Server (HSS) resides in the Control Plane. Sh is an interface between these two components.

HSS is a master user database that supports the IP Multimedia Subsystem network entities and contains subscriber profiles. It can provide information about the subscriber's location and IP information and also helps to perform authentication and authorization of the user.

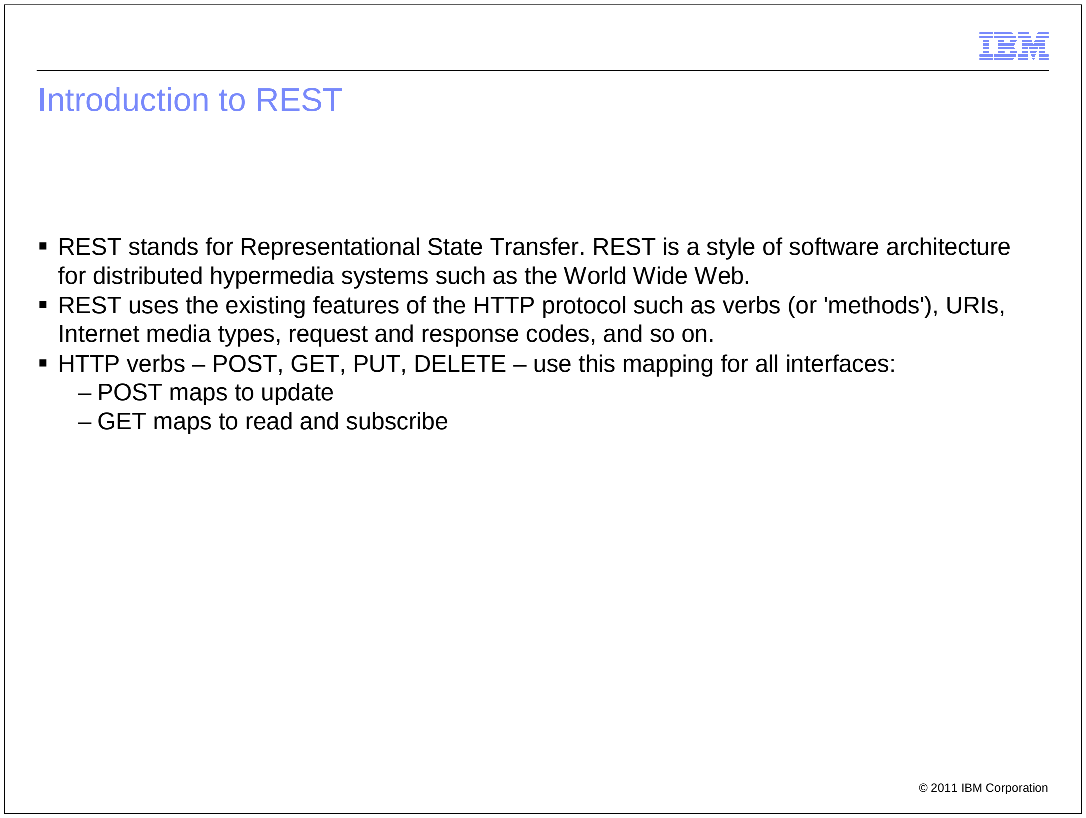
IP Multimedia Subsystem Connector Diameter component exposes the Sh interface as web service APIs.

From this release, REST supports four types of operation: Data Retrieval, Data Updating, Subscription, and Notification.

For Retrieval – you can use getChargingInfo, getRepositoryData, and so on.

For Update - update the HSS using updateRepositoryData, updatePSIActivation, and so on.

Subscribe and notification operations are complimentary. You can subscribe to an IP Multimedia Subsystem user state. If the user state changes at the HSS end, you get a notification - notifyIMSUserState.



Introduction to REST

- REST stands for Representational State Transfer. REST is a style of software architecture for distributed hypermedia systems such as the World Wide Web.
- REST uses the existing features of the HTTP protocol such as verbs (or 'methods'), URIs, Internet media types, request and response codes, and so on.
- HTTP verbs – POST, GET, PUT, DELETE – use this mapping for all interfaces:
  - – POST maps to update
  - – GET maps to read and subscribe

This slide gives an introduction to REST.

From this release, the Diameter Sh interface is being exposed through REST APIs, along with the already existing web service support.

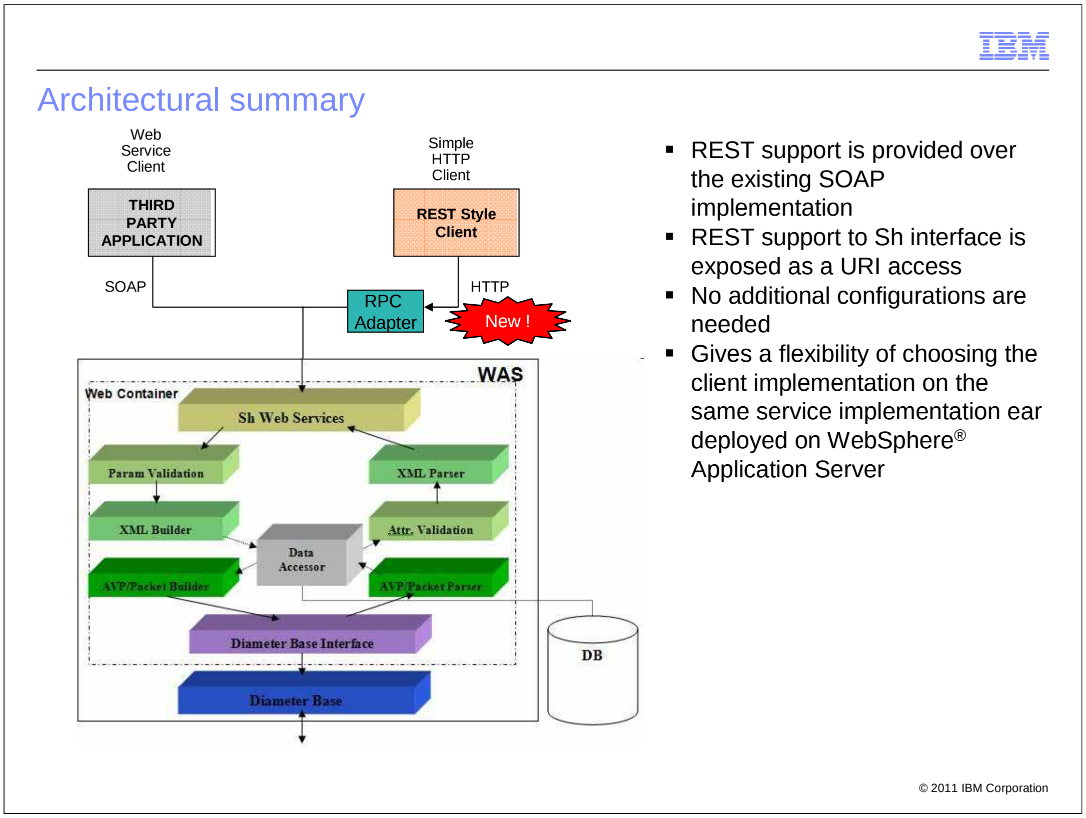
REST stands for Representational State Transfer. It is not a technology but a set of principles (or an architectural style) that makes resources available so that web applications can access and modify them.

In REST terms, everything is represented as resources.

The requests and responses are built around the transfer of representations of these resources.

All the services should be defined in terms of resources, which are addressable as simple HTTP URIs.

The HTTP verbs – POST, GET, PUT, and DELETE – are used to perform Create, Read, Update, and Delete operations on these resources.



Architectural summary

- REST support is provided over the existing SOAP implementation
- REST support to Sh interface is exposed as a URI access
- No additional configurations are needed
- Gives a flexibility of choosing the client implementation on the same service implementation ear deployed on WebSphere® Application Server

This slide gives the architectural summary of the Sh Interface in IP Multimedia Subsystem Connector.

The diagram shows that from this release, any simple HTTP client can also access the exact same functionality that is being provided through web service.

That is, the Sh Interface is exposed through the web service and HTTP.

The design uses the libraries defined in the IBM Web 2.0 feature pack, which contains the RPC Adapter for IBM (IBM implementation of Web-Remoting). This has been marked as "new" in the diagram.

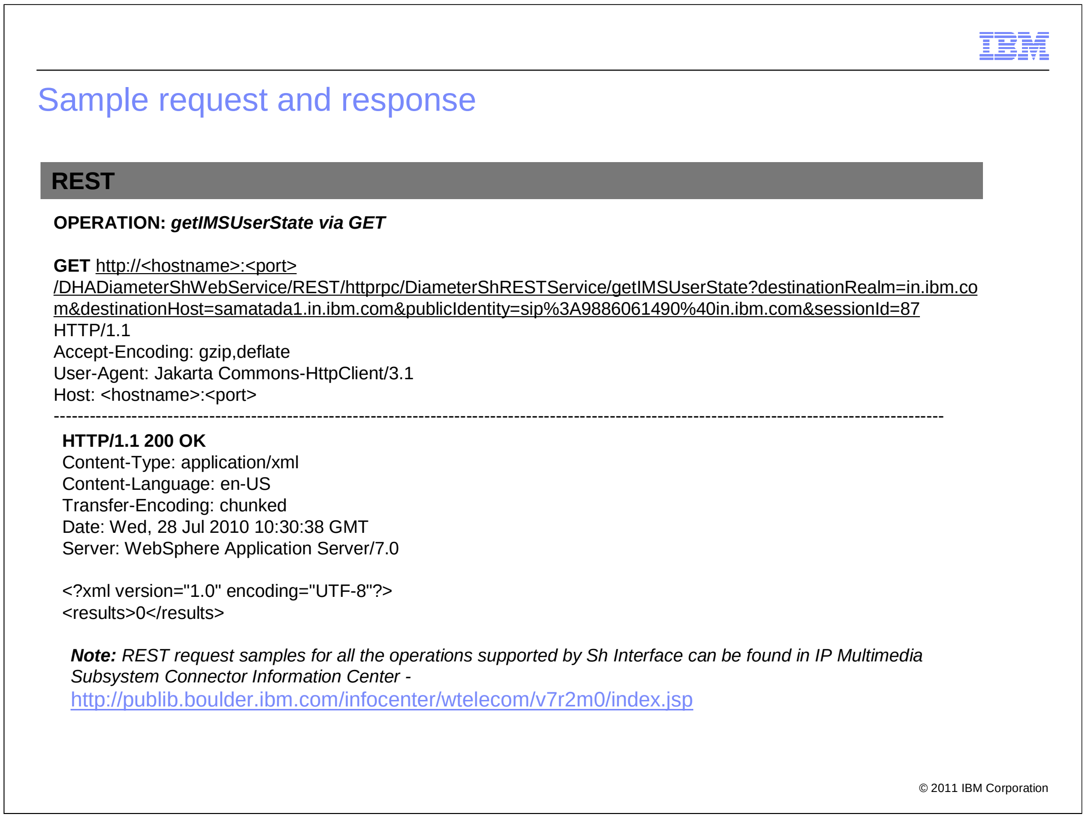
The RPC adapter implemented as a generic servlet, provides an HTTP interface to the registered JavaBeans.

It deserializes the input and calls the corresponding Java™ Bean method.

Conversely, it will serialize the output from Java Bean to XML format.

Nothing changes after it enters the web container. The exact same functionality exposed through web service is available as REST APIs.

Essentially, it gives the flexibility of choosing the client implementation on the same service implementation ear deployed on WebSphere Application Server.



Sample request and response

REST

OPERATION: ***getIMSUserState via GET***

```
GET http://<hostname>:<port>
/DHADiameterShWebService/REST/httprpc/DiameterShRESTService/getIMSUserState?destinationRealm=in.ibm.co
m&destinationHost=samatada1.in.ibm.com&publicIdentity=sip%3A9886061490%40in.ibm.com&sessionId=87
HTTP/1.1
Accept-Encoding: gzip,deflate
User-Agent: Jakarta Commons-HttpClient/3.1
Host: <hostname>:<port>
---------------------------------------------------------------------------------------------------------------------------------------------
 HTTP/1.1 200 OK
 Content-Type: application/xml
 Content-Language: en-US
 Transfer-Encoding: chunked
 Date: Wed, 28 Jul 2010 10:30:38 GMT
 Server: WebSphere Application Server/7.0

 <?xml version="1.0" encoding="UTF-8"?>
 <results>0</results>
```

Note: *REST request samples for all the operations supported by Sh Interface can be found in IP Multimedia Subsystem Connector Information Center -* http://publib.boulder.ibm.com/infocenter/wtelecom/v7r2m0/index.jsp

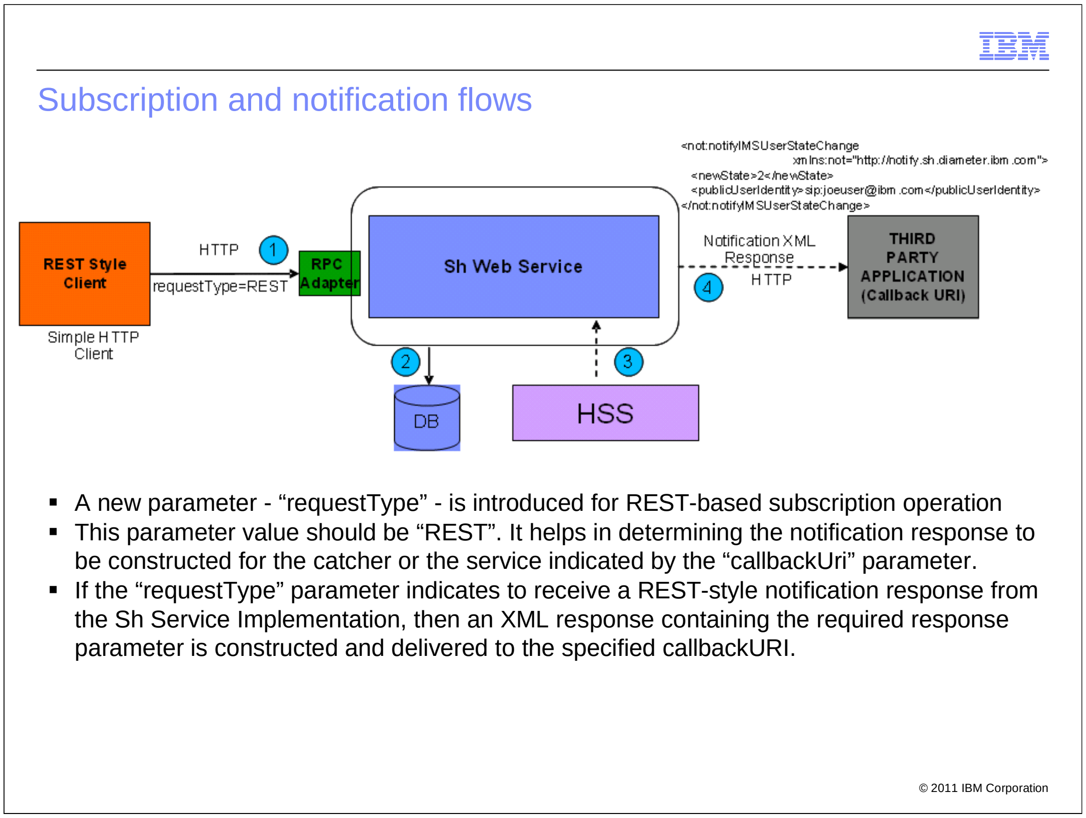
This slide provides a sample request and response.

In this example, IP Multimedia Subsystem user state is being retrieved through a simple GET operation and all the parameters are being sent through the URI itself.

In the response, the HTTP status returned is 200 OK.

And in the body part, the result returned is an XML object.

This is just one of the examples provided. There are about 17 operations in Sh interface, and all the examples are provided in the Information Center - http://publib.boulder.ibm.com/infocenter/wtelecom/v7r2m0/index.jsp.



Subscription and notification flows

```
<not:notifyIMSUserStateChange
                    xmlns:not="http://notify.sh.diameter.ibm.com">
  <newState>2</newState>
  <publicUserIdentity>sip:joeuser@ibm.com</publicUserIdentity>
</not:notifyIMSUserStateChange>
```

- A new parameter - “requestType” - is introduced for REST-based subscription operation
- This parameter value should be “REST”. It helps in determining the notification response to be constructed for the catcher or the service indicated by the “callbackUri” parameter.
- If the “requestType” parameter indicates to receive a REST-style notification response from the Sh Service Implementation, then an XML response containing the required response parameter is constructed and delivered to the specified callbackURI.

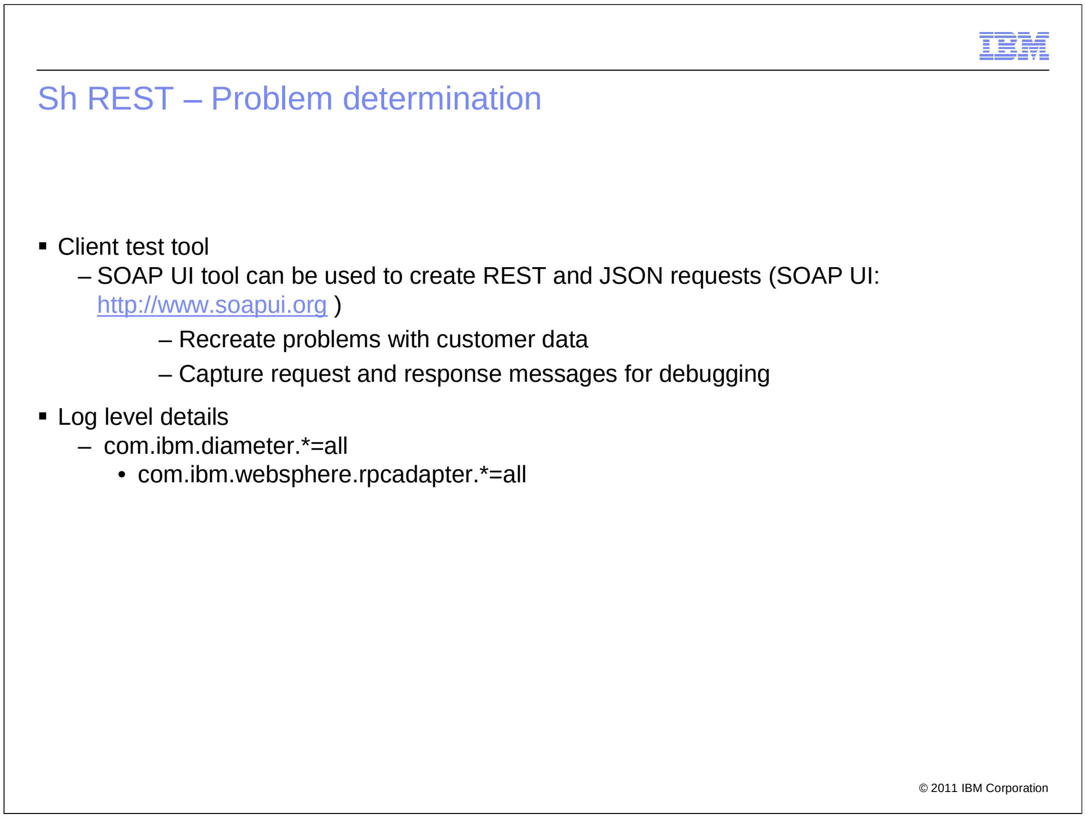
This slide deals with subscription and notification flows in the Sh interface.

There are four kinds of operations - Read, Update, Subscription, and Notification.

This diagram shows the subscription and notification flow in which a new parameter called requestType is introduced.

Steps 1 and 2 indicate that when a REST client sends a message, this field should be set to REST and this information is stored in the database along with the CallbackURI.

Steps 3 and 4 indicate that when a notification is triggered from the HSS and the Sh web service looks up the Callback URI, it comes to know if the request type was REST or not. If it is REST, then a Notification XML message is sent to the given Third Party Application Callback URI.



Sh REST – Problem determination

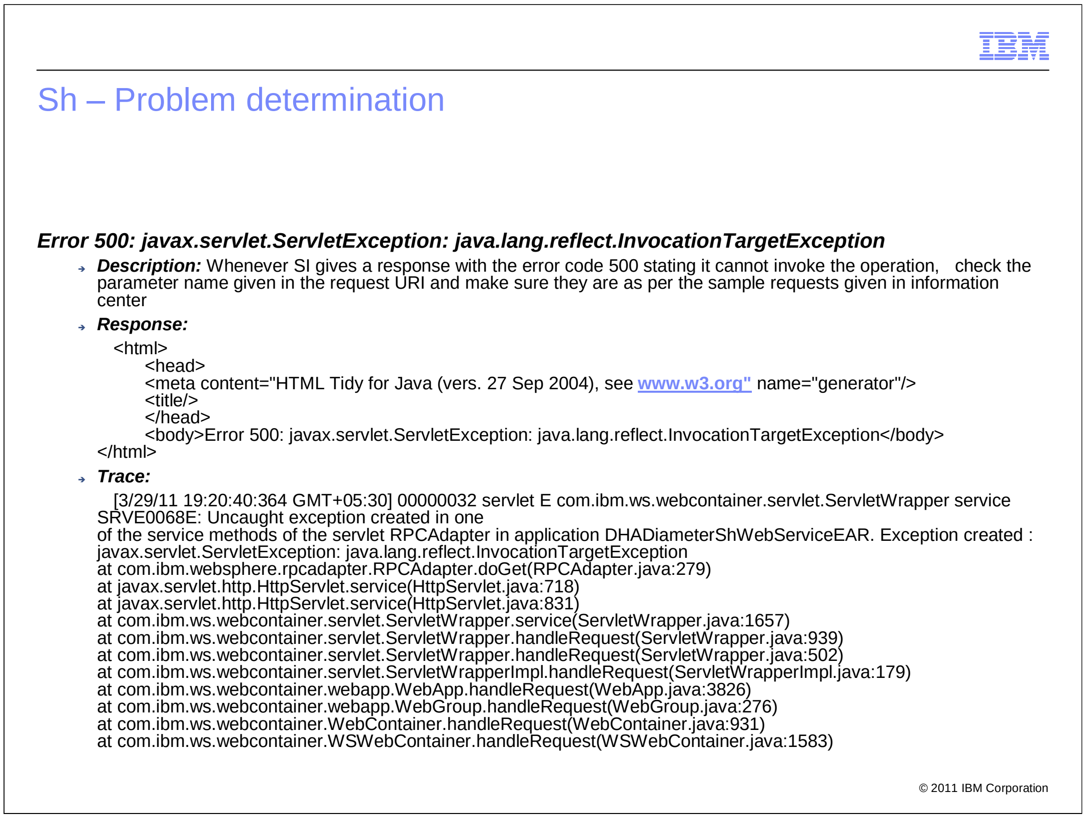
- Client test tool
  - SOAP UI tool can be used to create REST and JSON requests (SOAP UI: http://www.soapui.org)
    - Recreate problems with customer data
    - Capture request and response messages for debugging
- Log level details
  - com.ibm.diameter.*=all
    - com.ibm.websphere.rpcadapter.*=all

This slide deals with problem determination.

SOAP user interface (UI) can be used for creating REST request and for enabling log level details.

In case you have errors, debugging is required. For debugging, log on to the administration console; navigate to **Logging and Tracing** > **Choose the server** > **Change log detail levels**, and provide the trace string.

This gives an idea of what is happening in the system and helps in solving the problem.



Sh – Problem determination

Error 500: javax.servlet.ServletException: java.lang.reflect.InvocationTargetException

- ***Description:*** Whenever SI gives a response with the error code 500 stating it cannot invoke the operation,   check the parameter name given in the request URI and make sure they are as per the sample requests given in information center
- ***Response:***

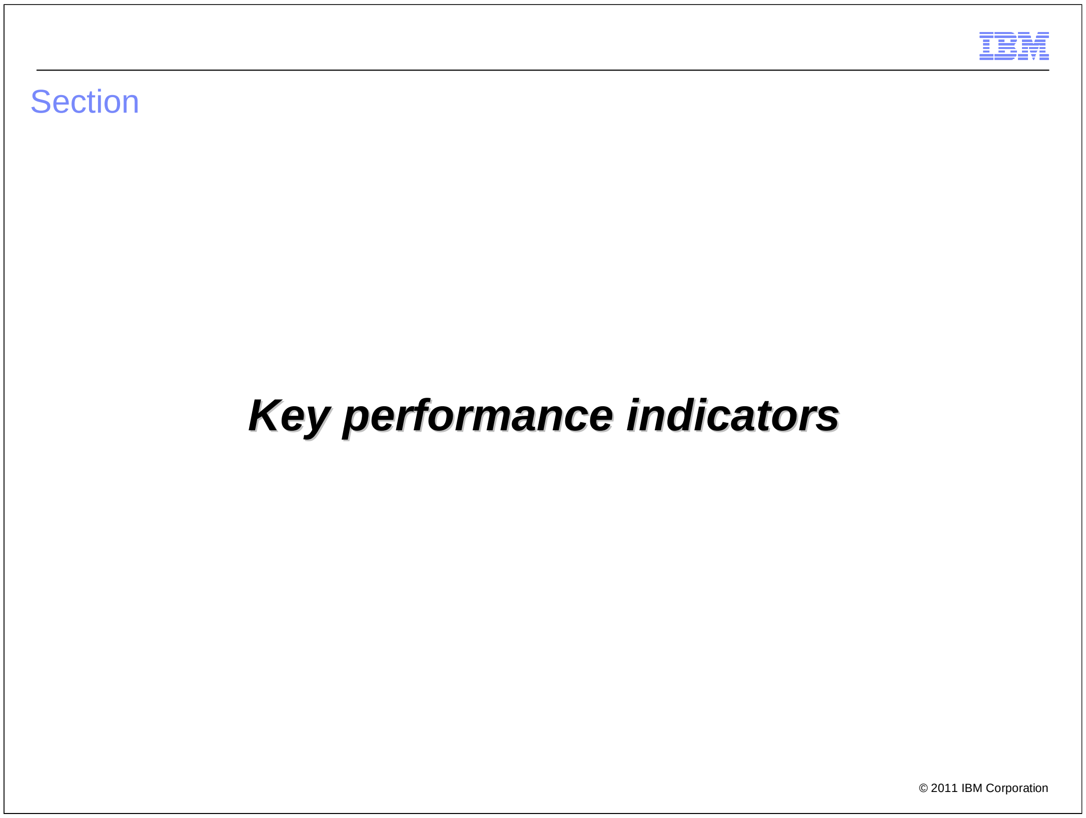
```
 <html>
      <head>
      <meta content="HTML Tidy for Java (vers. 27 Sep 2004), see www.w3.org" name="generator"/>
      <title/>
      </head>
      <body>Error 500: javax.servlet.ServletException: java.lang.reflect.InvocationTargetException</body>
</html>
```

- ***Trace:***

```
   [3/29/11 19:20:40:364 GMT+05:30] 00000032 servlet E com.ibm.ws.webcontainer.servlet.ServletWrapper service
SRVE0068E: Uncaught exception created in one
of the service methods of the servlet RPCAdapter in application DHADiameterShWebServiceEAR. Exception created :
javax.servlet.ServletException: java.lang.reflect.InvocationTargetException
at com.ibm.websphere.rpcadapter.RPCAdapter.doGet(RPCAdapter.java:279)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:718)
at javax.servlet.http.HttpServlet.service(HttpServlet.java:831)
at com.ibm.ws.webcontainer.servlet.ServletWrapper.service(ServletWrapper.java:1657)
at com.ibm.ws.webcontainer.servlet.ServletWrapper.handleRequest(ServletWrapper.java:939)
at com.ibm.ws.webcontainer.servlet.ServletWrapper.handleRequest(ServletWrapper.java:502)
at com.ibm.ws.webcontainer.servlet.ServletWrapperImpl.handleRequest(ServletWrapperImpl.java:179)
at com.ibm.ws.webcontainer.webapp.WebApp.handleRequest(WebApp.java:3826)
at com.ibm.ws.webcontainer.webapp.WebGroup.handleRequest(WebGroup.java:276)
at com.ibm.ws.webcontainer.WebContainer.handleRequest(WebContainer.java:931)
at com.ibm.ws.webcontainer.WSWebContainer.handleRequest(WSWebContainer.java:1583)
```

Here is a common error you might face. Whenever the service gives a response with an error code of “500”, check the parameter name.

The error is typically because the parameter names are not consistent with what the implementation expects.

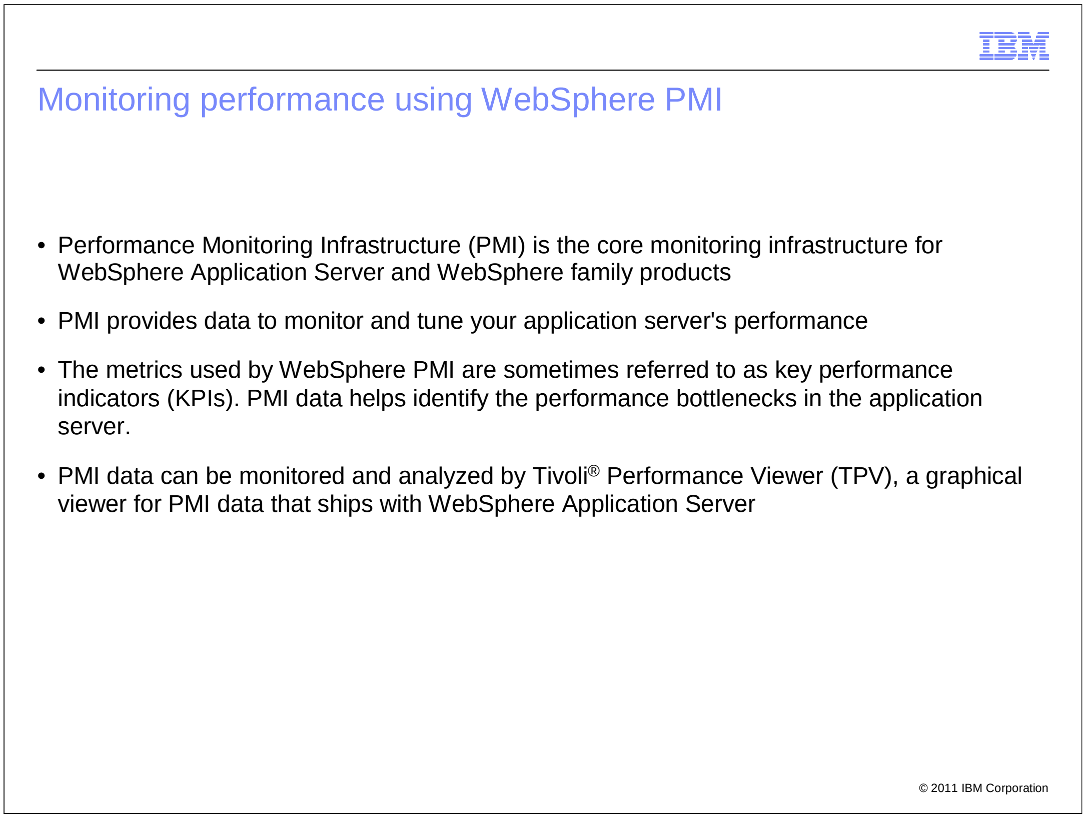


Section

Key performance indicators

Key Performance Indicators for IMS Connector

From this release, the Diameter component of IP Multimedia Subsystem Connector is publishing certain Key Performance Indicators.



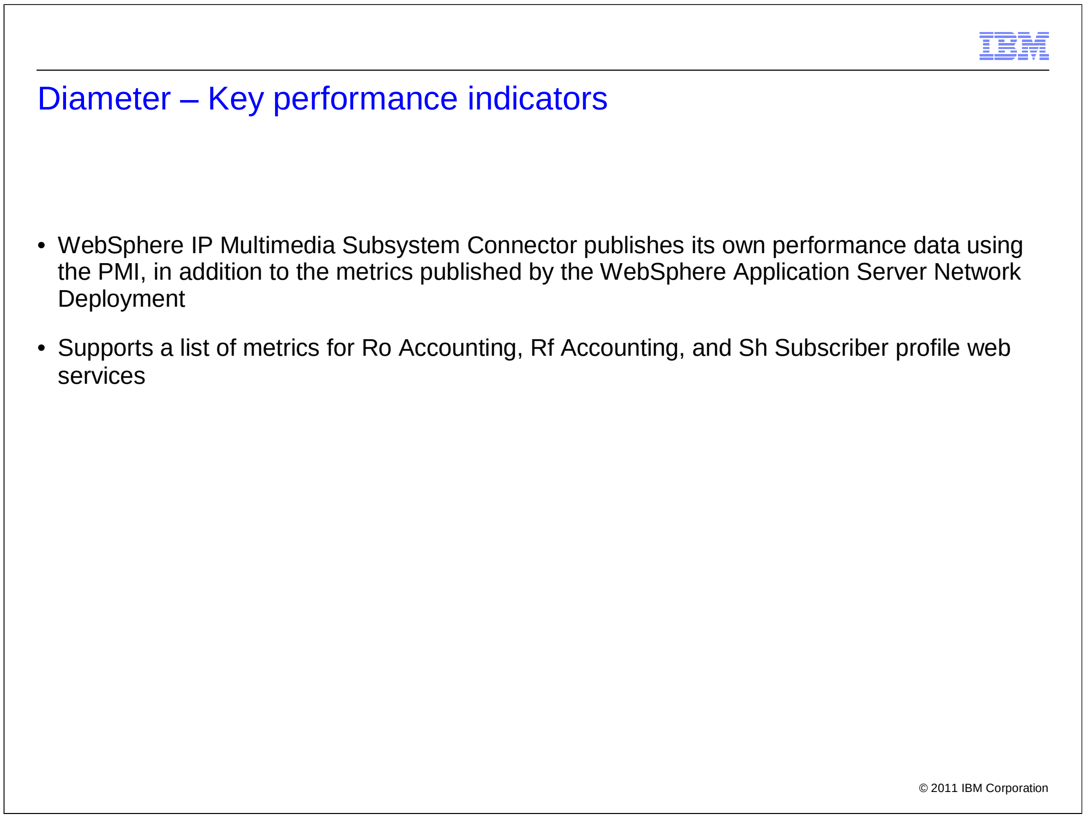
Monitoring performance using WebSphere PMI

- Performance Monitoring Infrastructure (PMI) is the core monitoring infrastructure for WebSphere Application Server and WebSphere family products
- PMI provides data to monitor and tune your application server's performance
- The metrics used by WebSphere PMI are sometimes referred to as key performance indicators (KPIs). PMI data helps identify the performance bottlenecks in the application server.
- PMI data can be monitored and analyzed by Tivoli® Performance Viewer (TPV), a graphical viewer for PMI data that ships with WebSphere Application Server

Performance Monitoring Infrastructure (PMI) is the core monitoring infrastructure for WebSphere Application Server and the WebSphere family of products.

PMI provides data to monitor and tune your application server's performance.

Using PMI data the performance bottlenecks in the application servers can be identified and fixed. These PMIs or KPIs can be monitored by a tool called Tivoli Performance Viewer. This tool is embedded in the WebSphere Administration Console. From the Tivoli Performance Viewer, you can view the current activity and summary reports. It provides a simple viewer for the performance data collected by the Performance Monitoring Infrastructure.



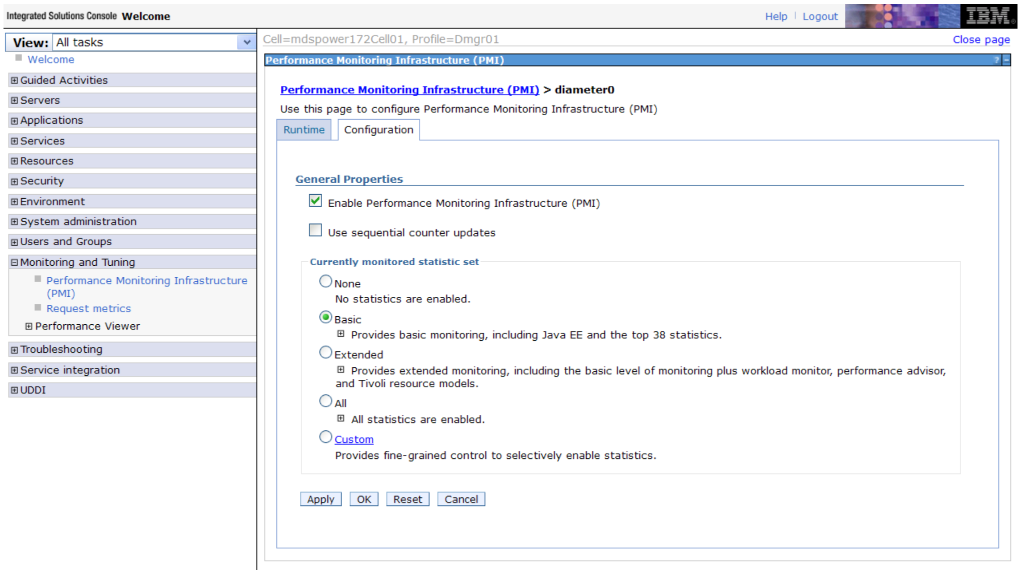
Diameter – Key performance indicators

- WebSphere IP Multimedia Subsystem Connector publishes its own performance data using the PMI, in addition to the metrics published by the WebSphere Application Server Network Deployment
- Supports a list of metrics for Ro Accounting, Rf Accounting, and Sh Subscriber profile web services

From this release, the Diameter component of WebSphere IP Multimedia Subsystem Connector publishes its own performance data using the PMI infrastructure provided by WebSphere Application Server. This is in addition to the metrics published by WebSphere Application Server Network Deployment.

It also supports a list of metrics for Ro, Rf, and Sh web services.

Enabling performance monitoring



Cell=mdspower172Cell01, Profile=Dmgr01

Performance Monitoring Infrastructure (PMI)

Performance Monitoring Infrastructure (PMI) > diameter0

Use this page to configure Performance Monitoring Infrastructure (PMI)

Runtime | Configuration

General Properties

☑ Enable Performance Monitoring Infrastructure (PMI)

□ Use sequential counter updates

Currently monitored statistic set

○ None
No statistics are enabled.

● Basic
⊞ Provides basic monitoring, including Java EE and the top 38 statistics.

○ Extended
⊞ Provides extended monitoring, including the basic level of monitoring plus workload monitor, performance advisor, and Tivoli resource models.

○ All
⊞ All statistics are enabled.

○ Custom
Provides fine-grained control to selectively enable statistics.

Apply | OK | Reset | Cancel

To enable Performance Monitoring in WebSphere Application Server, log on to the administration console. In the navigation pane, click **Monitoring and Tuning**, and then click **Performance Monitoring Infrastructure (PMI)**.

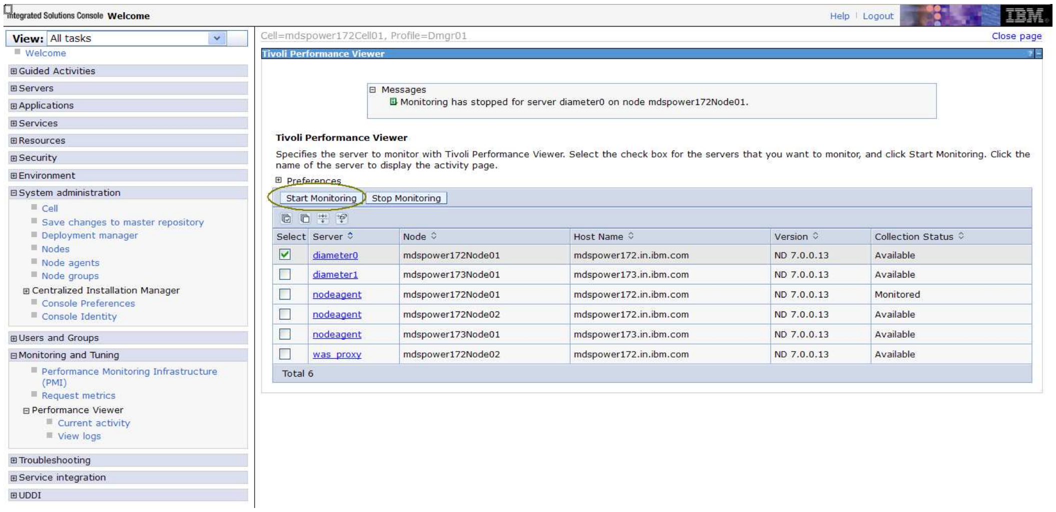
Select the application server instance.

In the **Configuration** tab, select the **Enable Performance Monitoring Infrastructure (PMI)** check box under **General Properties**.

Under **Currently monitored statistic set**, select one option.

Click **Save** to save changes to the master configuration.

Start monitoring



Cell=mdspower172Cell01, Profile=Dmgr01

Tivoli Performance Viewer

Messages
Monitoring has stopped for server diameter0 on node mdspower172Node01.

Tivoli Performance Viewer

Specifies the server to monitor with Tivoli Performance Viewer. Select the check box for the servers that you want to monitor, and click Start Monitoring. Click the name of the server to display the activity page.

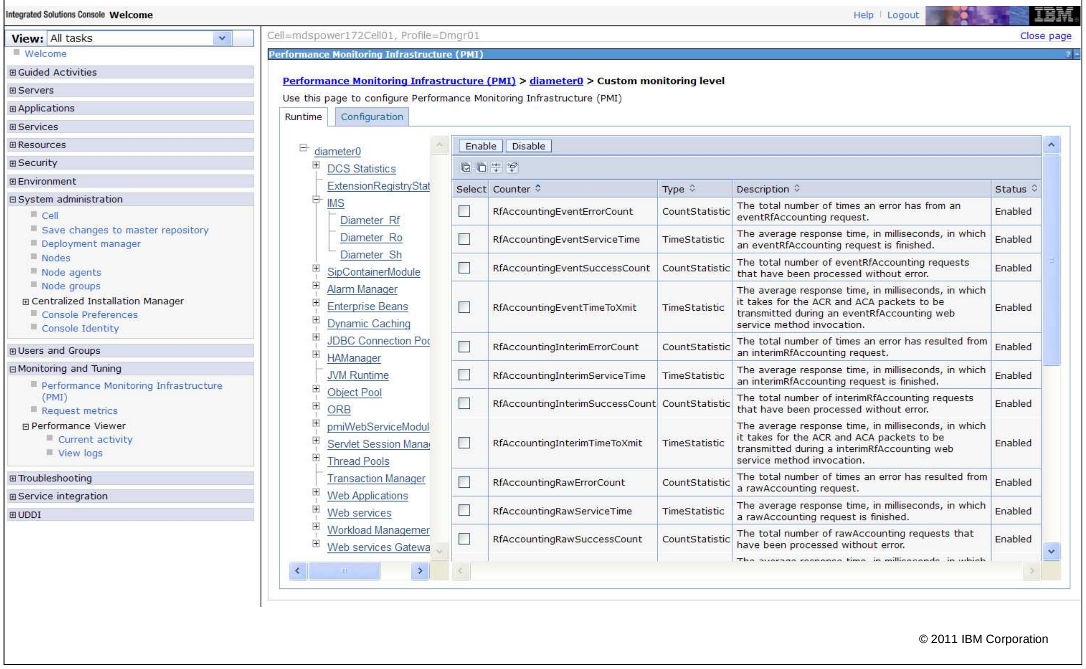
Select	Server	Node	Host Name	Version	Collection Status
☑	diameter0	mdspower172Node01	mdspower172.in.ibm.com	ND 7.0.0.13	Available
□	diameter1	mdspower173Node01	mdspower173.in.ibm.com	ND 7.0.0.13	Available
□	nodeagent	mdspower172Node01	mdspower172.in.ibm.com	ND 7.0.0.13	Monitored
□	nodeagent	mdspower172Node02	mdspower172.in.ibm.com	ND 7.0.0.13	Available
□	nodeagent	mdspower173Node01	mdspower173.in.ibm.com	ND 7.0.0.13	Available
□	was_proxy	mdspower172Node02	mdspower172.in.ibm.com	ND 7.0.0.13	Available

Total 6

To start monitoring, click **Performance Viewer** in the navigation pane, and then click **Current Activity** to go to the Tivoli Performance Viewer page.

Select the application server instance to monitor, and click **Start Monitoring**.

Enabling the KPIs



Performance Monitoring Infrastructure (PMI) > diameter0 > Custom monitoring level

Use this page to configure Performance Monitoring Infrastructure (PMI)

Select	Counter	Type	Description	Status
□	RfAccountingEventErrorCount	CountStatistic	The total number of times an error has from an eventRfAccounting request.	Enabled
□	RfAccountingEventServiceTime	TimeStatistic	The average response time, in milliseconds, in which an eventRfAccounting request is finished.	Enabled
□	RfAccountingEventSuccessCount	CountStatistic	The total number of eventRfAccounting requests that have been processed without error.	Enabled
□	RfAccountingEventTimeToXmit	TimeStatistic	The average response time, in milliseconds, in which it takes for the ACR and ACA packets to be transmitted during an eventRfAccounting web service method invocation.	Enabled
□	RfAccountingInterimErrorCount	CountStatistic	The total number of times an error has resulted from an interimRfAccounting request.	Enabled
□	RfAccountingInterimServiceTime	TimeStatistic	The average response time, in milliseconds, in which an interimRfAccounting request is finished.	Enabled
□	RfAccountingInterimSuccessCount	CountStatistic	The total number of interimRfAccounting requests that have been processed without error.	Enabled
□	RfAccountingInterimTimeToXmit	TimeStatistic	The average response time, in milliseconds, in which it takes for the ACR and ACA packets to be transmitted during a interimRfAccounting web service method invocation.	Enabled
□	RfAccountingRawErrorCount	CountStatistic	The total number of times an error has resulted from a rawAccounting request.	Enabled
□	RfAccountingRawServiceTime	TimeStatistic	The average response time, in milliseconds, in which a rawAccounting request is finished.	Enabled
□	RfAccountingRawSuccessCount	CountStatistic	The total number of rawAccounting requests that have been processed without error.	Enabled
			The average response time, in milliseconds, in which	

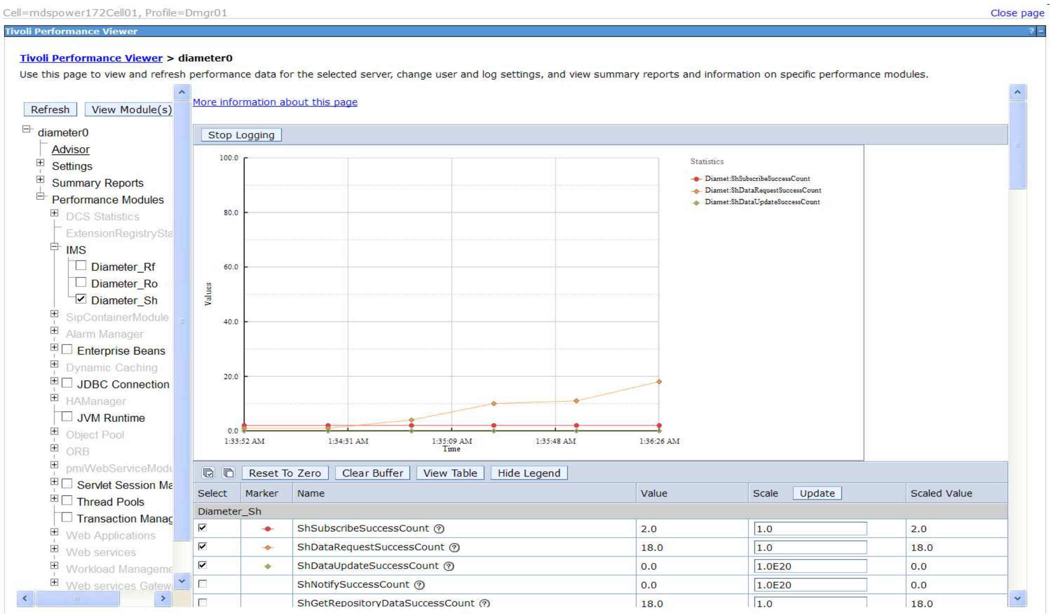
After you start monitoring, enable the Key Performance Indicators (KPIs).

To enable KPIs:

1. Click **Performance Monitoring Infrastructure (PMI)** in the navigation pane. This takes you back to the main PMI configuration page.
2. Select the application server instance.
3. In the **Runtime** tab, click **Custom** under **General Properties**.
4. Choose a component from the tree on the left side of the screen, and select the associated metrics in the table on the right side of the screen.
5. Click **Enable**, and then restart the application server.

In a clustered configuration, repeat these steps for each application server.

Tivoli Performance Viewer



Tivoli Performance Viewer > diameter0

Use this page to view and refresh performance data for the selected server, change user and log settings, and view summary reports and information on specific performance modules.

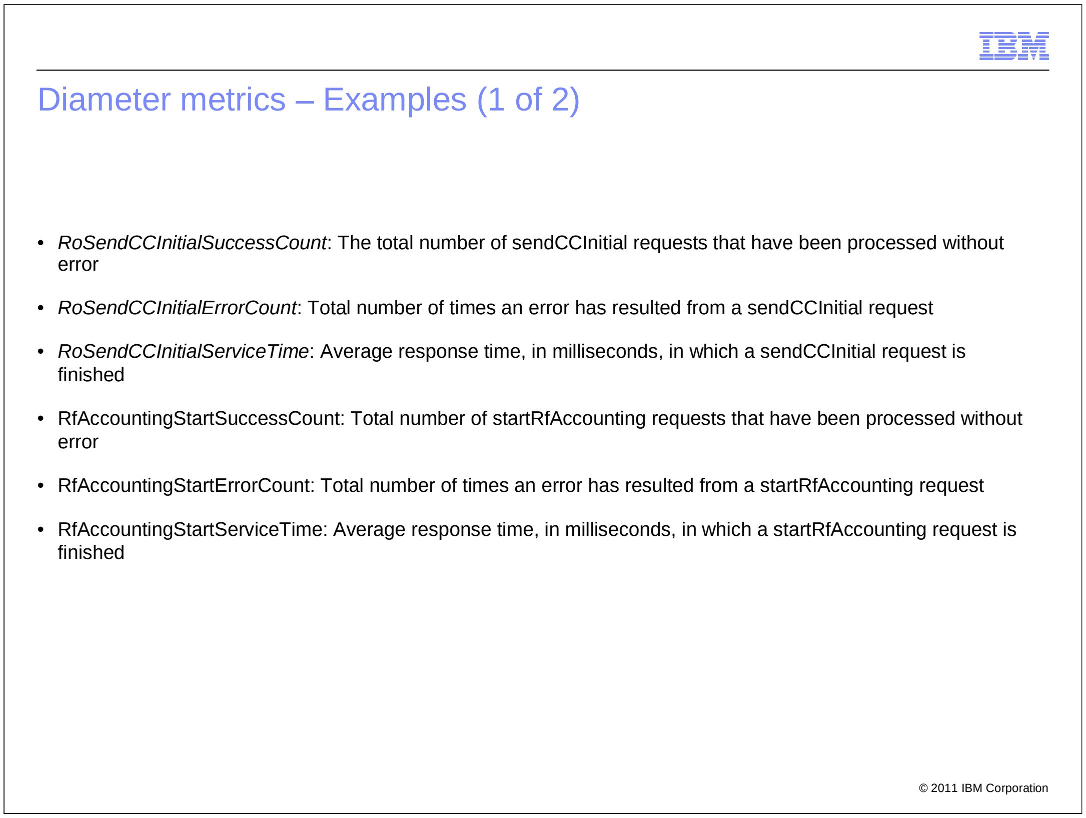
Select	Marker	Name	Value	Scale	Scaled Value
Diameter_Sh					
☑	●	ShSubscribeSuccessCount ⓘ	2.0	1.0	2.0
☑	◆	ShDataRequestSuccessCount ⓘ	18.0	1.0	18.0
☑	◆	ShDataUpdateSuccessCount ⓘ	0.0	1.0E20	0.0
□		ShNotifySuccessCount ⓘ	0.0	1.0E20	0.0
□		ShGetRepositoryDataSuccessCount ⓘ	18.0	1.0	18.0

This slide describes the Tivoli Performance Viewer, which is a representation of how the KPI data can be viewed.

The performance data for the selected server is provided as graphs or tables with the Diameter KPIs that you have enabled.

You can see a graphical representation or a tabular view and you have the option of resetting the values, viewing the legend, and so on.

You can see the graph changing and the values getting updated.



Diameter metrics – Examples (1 of 2)

- *RoSendCCInitialSuccessCount*: The total number of sendCCInitial requests that have been processed without error
- *RoSendCCInitialErrorCount*: Total number of times an error has resulted from a sendCCInitial request
- *RoSendCCInitialServiceTime*: Average response time, in milliseconds, in which a sendCCInitial request is finished
- RfAccountingStartSuccessCount: Total number of startRfAccounting requests that have been processed without error
- RfAccountingStartErrorCount: Total number of times an error has resulted from a startRfAccounting request
- RfAccountingStartServiceTime: Average response time, in milliseconds, in which a startRfAccounting request is finished

This slide has a few Diameter metrics examples that are published.

RoSendCCInitialSuccessCount: The total number of Send Charging Control Initial requests that have been processed without error

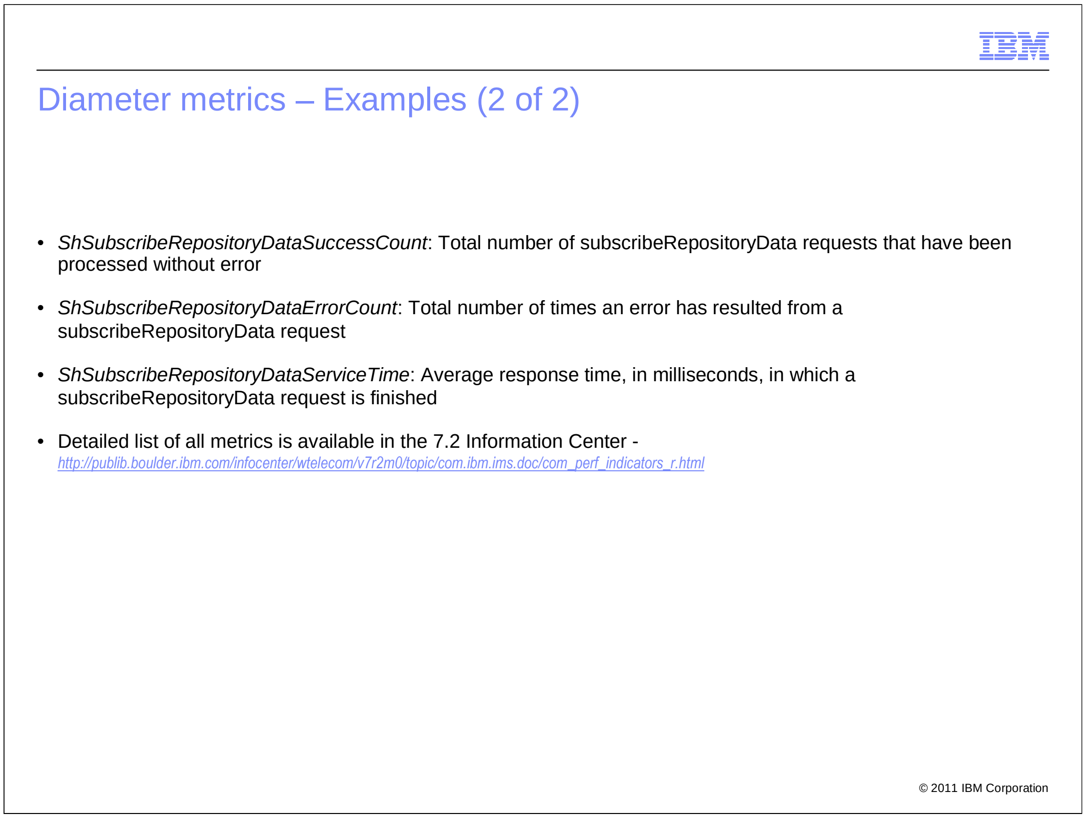
RoSendCCInitialErrorCount: is the same but an error count, so it is the number of failures that have occurred for the same operation

RoSendCCInitialServiceTime is the service time required for the same operation

The first three are for RoSEndCCInitial operation. And the next three are for RfAccountingStart operation.

So in general, there are separate metrics for success, failure, and service timing.

There are also other metrics that are published.



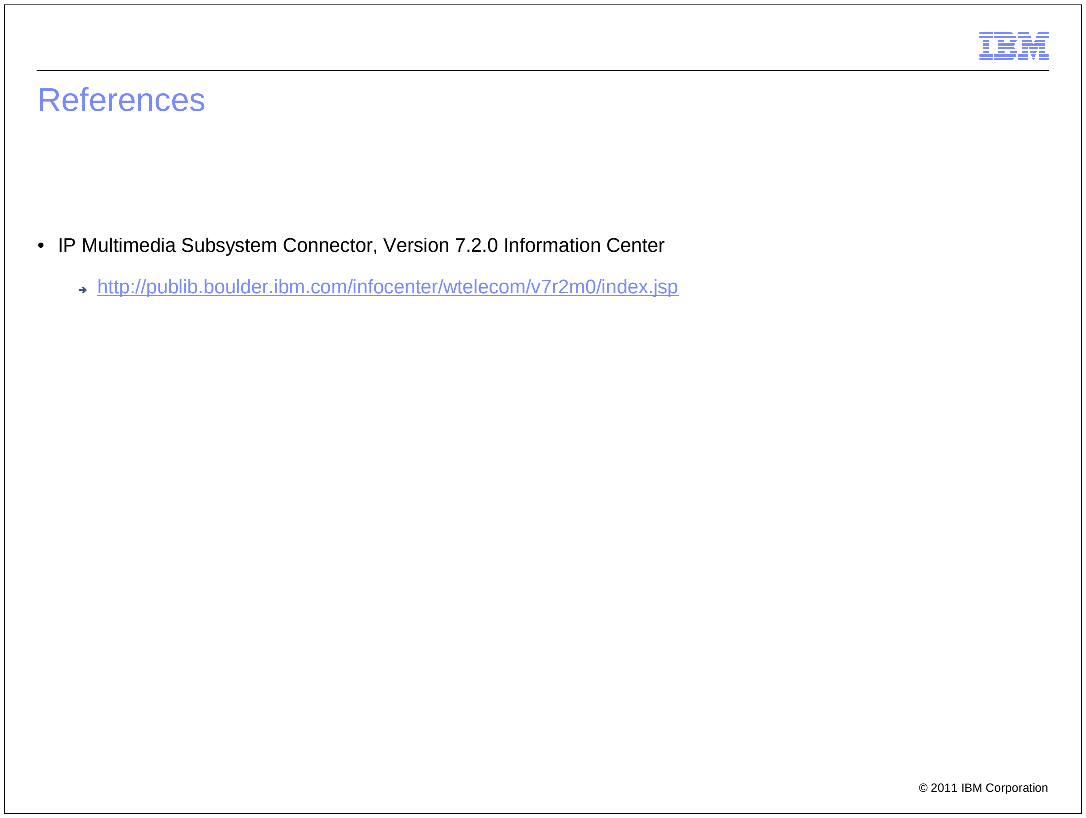
Diameter metrics – Examples (2 of 2)

- *ShSubscribeRepositoryDataSuccessCount*: Total number of subscribeRepositoryData requests that have been processed without error
- *ShSubscribeRepositoryDataErrorCount*: Total number of times an error has resulted from a subscribeRepositoryData request
- *ShSubscribeRepositoryDataServiceTime*: Average response time, in milliseconds, in which a subscribeRepositoryData request is finished
- Detailed list of all metrics is available in the 7.2 Information Center - http://publib.boulder.ibm.com/infocenter/wtelecom/v7r2m0/topic/com.ibm.ims.doc/com_perf_indicators_r.html

The three examples for Sh interface are for ShSubscribeRepositoryData - success count, error count, and service time. The total number of requests processed without error, total number of requests that have errored out, and the time that is taken to process this particular request.

There are 160-odd metrics that are published together in the Ro, Rf, and Sh components.

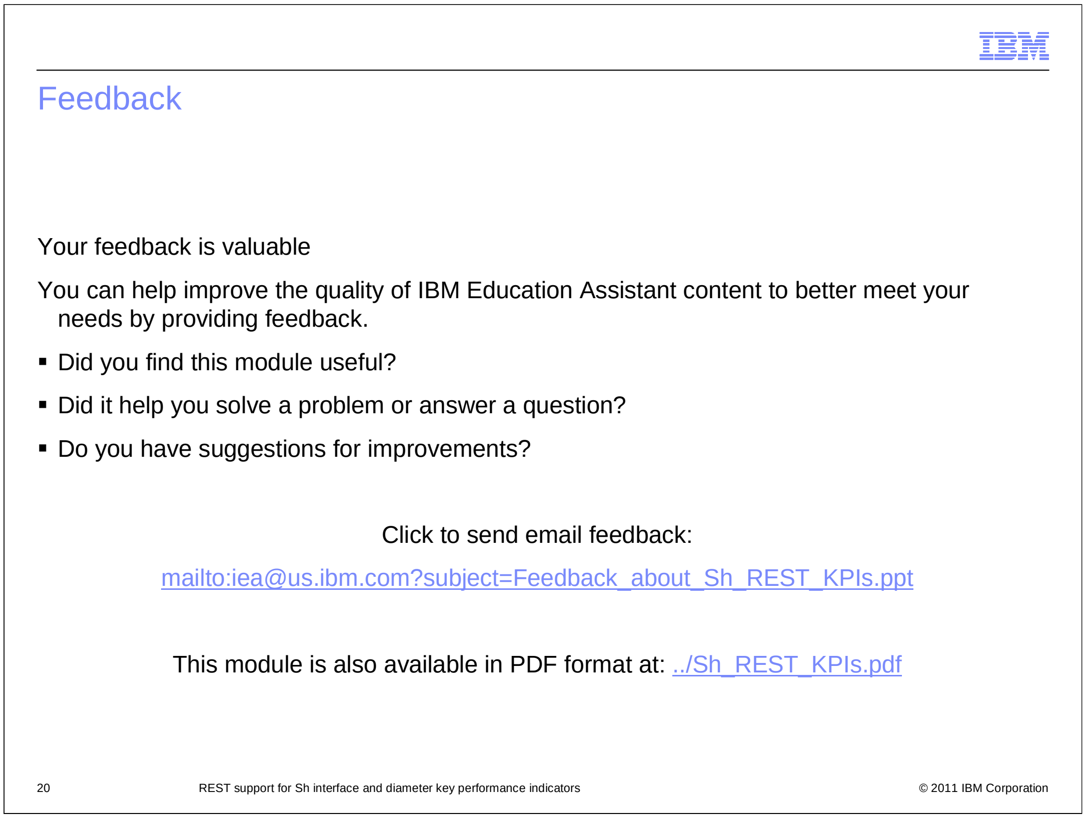
Detailed list of the metrics along with the descriptions are available in the 7.2 Information Center (http://publib.boulder.ibm.com/infocenter/wtelecom/v7r2m0/topic/com.ibm.ims.doc/com_perf_indicators_r.html).



References

- IP Multimedia Subsystem Connector, Version 7.2.0 Information Center
  - → http://publib.boulder.ibm.com/infocenter/wtelecom/v7r2m0/index.jsp

For more information, see the IMS Connector 7.2 Information Center - http://publib.boulder.ibm.com/infocenter/wtelecom/v7r2m0/index.jsp.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

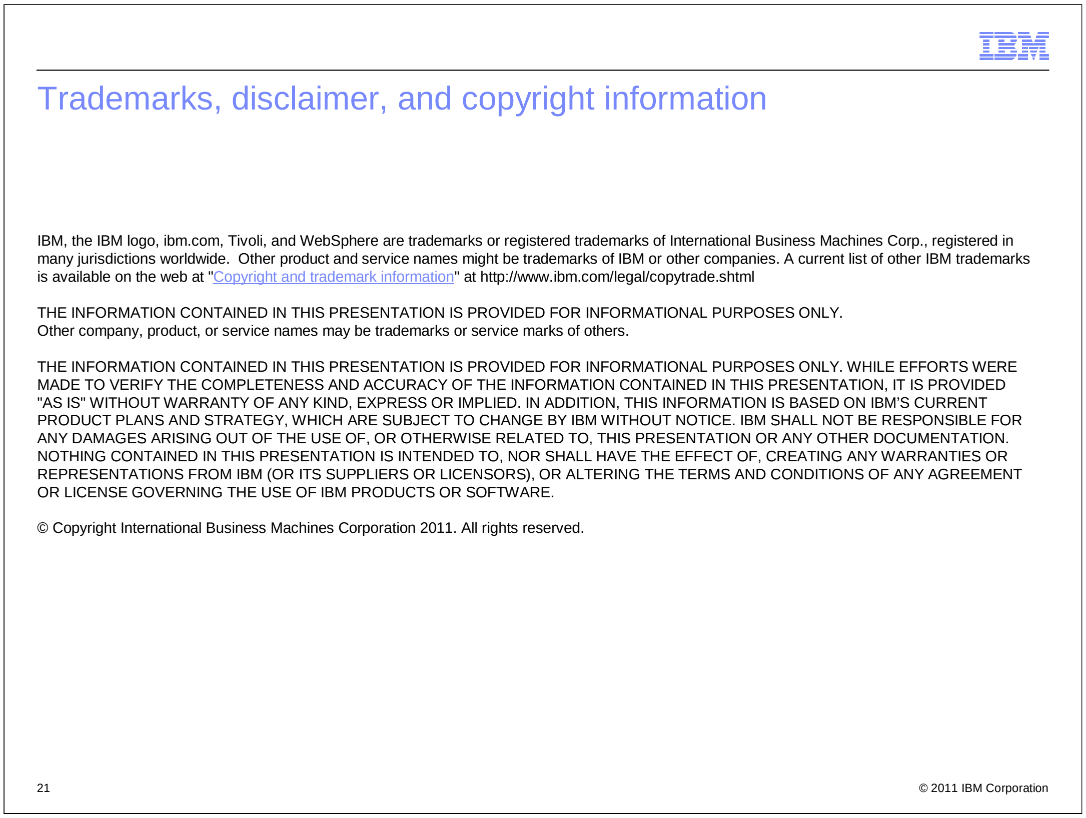
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_Sh_REST_KPIs.ppt

This module is also available in PDF format at: ../Sh_REST_KPIs.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, Tivoli, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.
Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.