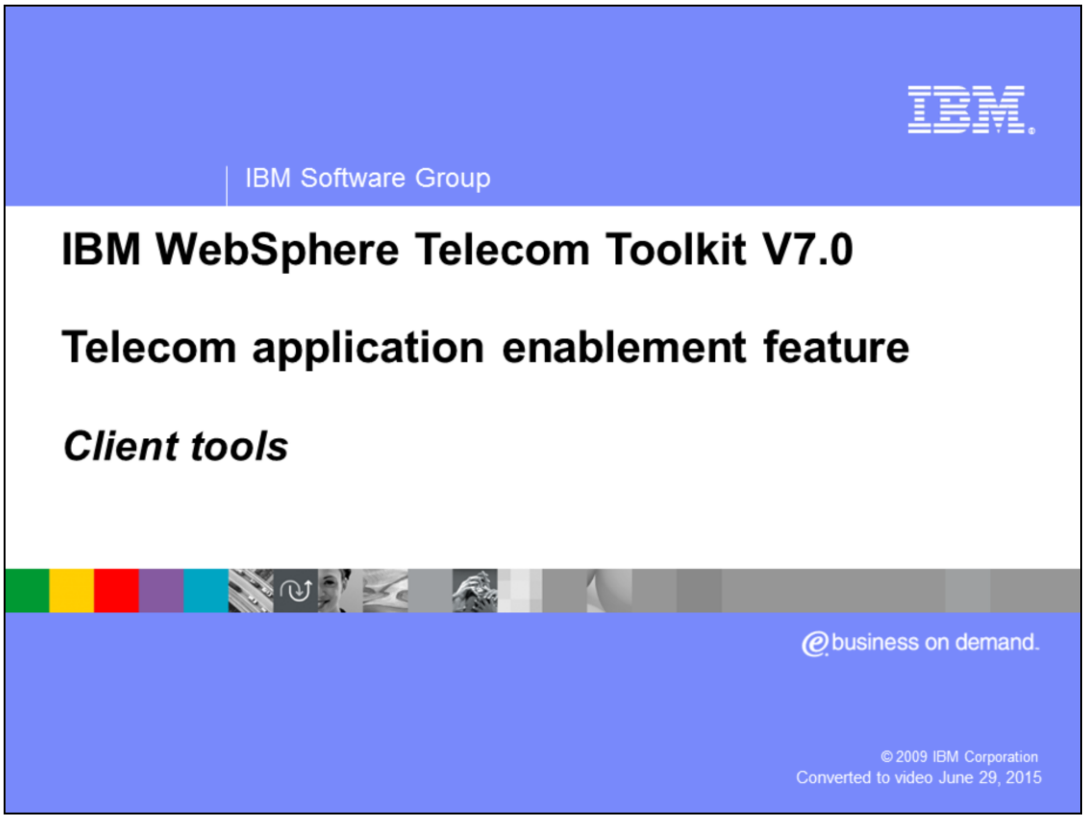


IBM Software Group

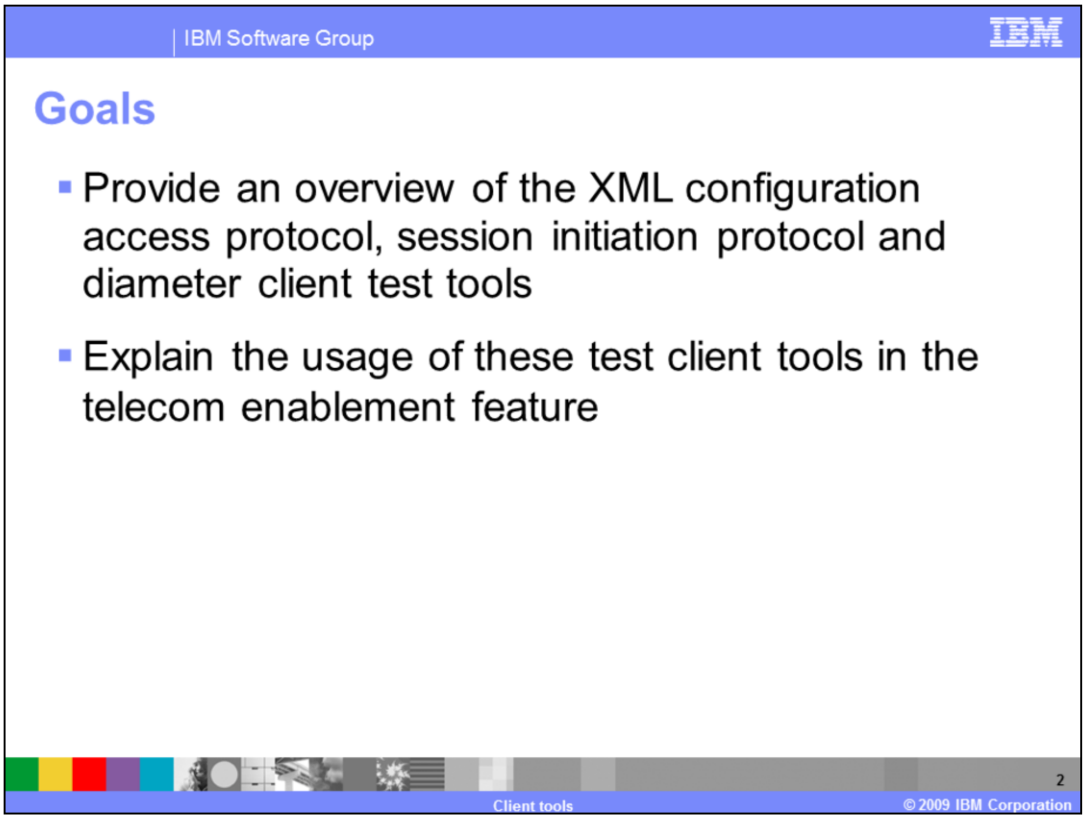
IBM WebSphere Telecom Toolkit V7.0

Telecom application enablement feature

Client tools

business on demand.

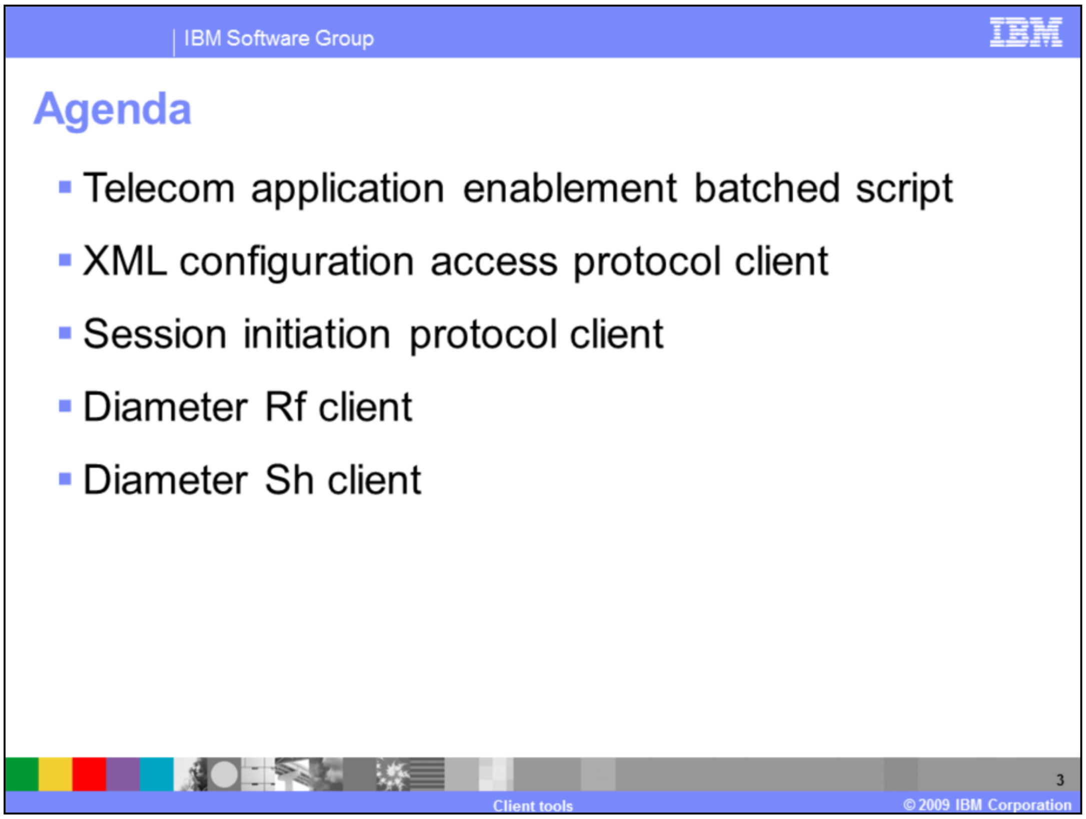
This presentation provides an overview of the client test tools available in the Telecom Enablement feature of the IBM WebSphere® Telecom Toolkit V 7.0. To get a better understanding on the content of this presentation, first review the overview presentation.



Goals

- Provide an overview of the XML configuration access protocol, session initiation protocol and diameter client test tools
- Explain the usage of these test client tools in the telecom enablement feature

The goal of this presentation is to provide an overview of the client test tools available in the Telecom Enablement Feature of the WebSphere Telecom Toolkit V 7.0. These client test tools are for the XML Configuration Access Protocol, Session Initiation Protocol, and Diameter protocols. The presentation also explains the usage of these tools to test the protocol specific applications to which the tools are geared.



Agenda

- Telecom application enablement batched script
- XML configuration access protocol client
- Session initiation protocol client
- Diameter Rf client
- Diameter Sh client

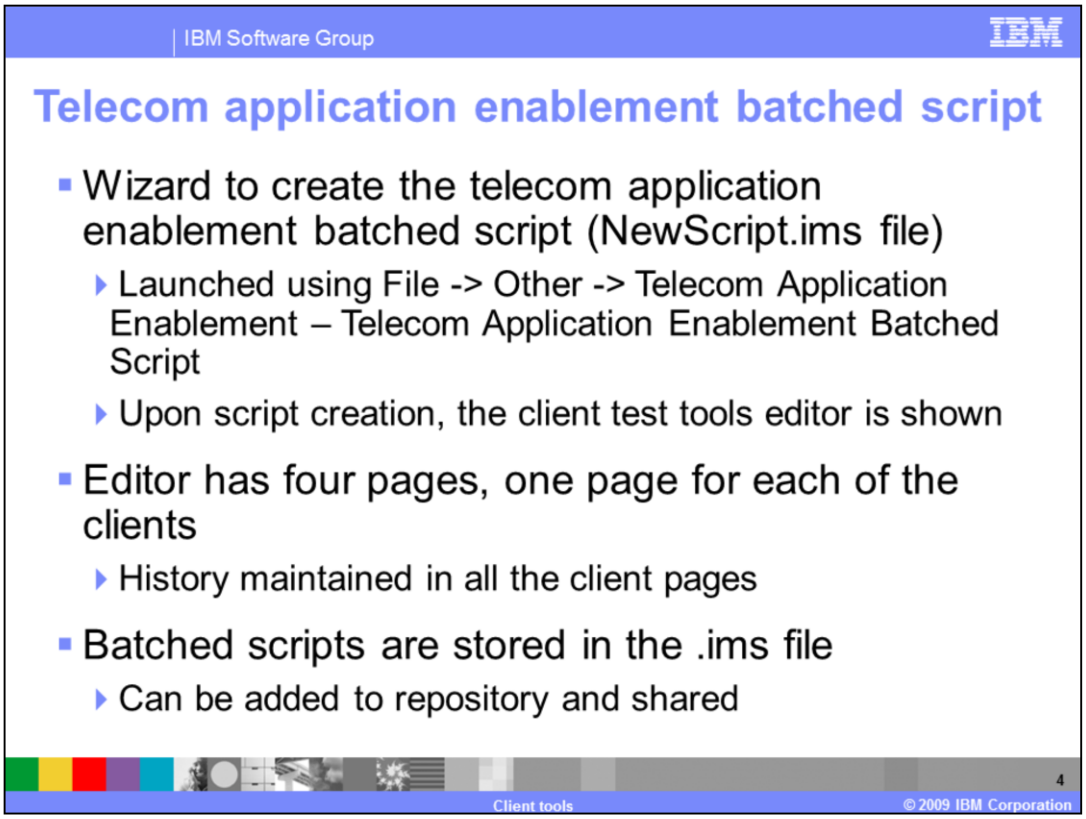
The agenda of this presentation covers the various client test tools which include:

- The Telecom Application Enablement Batched Request Script creation wizard.

- The XML Configuration Access Protocol Client used to test applications based on the XML Configuration Access Protocol

- The Session Initiation Protocol Client used to test applications based on the Session Initiation Protocol.

- The Diameter Rf and Sh clients used as Web services clients for the offline charging Rf and subscriber profile Sh Web services protocols.

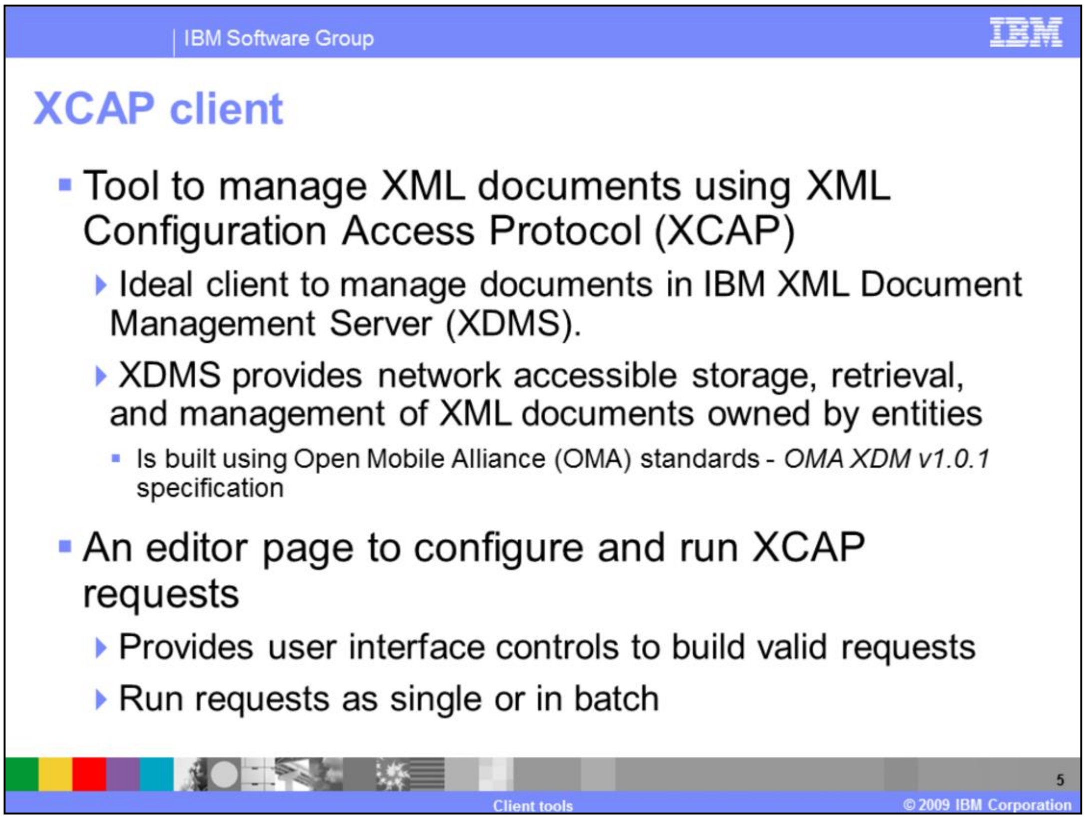


Telecom application enablement batched script

- Wizard to create the telecom application enablement batched script (NewScript.ims file)
  - Launched using File -> Other -> Telecom Application Enablement – Telecom Application Enablement Batched Script
  - Upon script creation, the client test tools editor is shown
- Editor has four pages, one page for each of the clients
  - History maintained in all the client pages
- Batched scripts are stored in the .ims file
  - Can be added to repository and shared

All four test client tools are driven and managed by a Telecom Application Enablement batched request script. The tools provide a wizard to create a new Telecom Application Enablement Batched Request Script. The wizard can be launched by following the menu actions File -> Other -> Telecom Application Enablement -> Telecom Application Enablement Batched Request Script

The wizard creates a script file with an .ims extension and launches the editor. The editor has four pages, one for each client. Tests performed from the four test clients can be run as single or batched tests. All test clients maintain history of the information entered and the history size is configurable (refer to slide 20 for this information). The batched tests are stored in the script file for reuse, for instance the script file can also be stored in a repository and can be shared and used by multiple users.



IBM Software Group

XCAP client

- Tool to manage XML documents using XML Configuration Access Protocol (XCAP)
  - Ideal client to manage documents in IBM XML Document Management Server (XDMS).
  - XDMS provides network accessible storage, retrieval, and management of XML documents owned by entities
    - Is built using Open Mobile Alliance (OMA) standards - *OMA XDM v1.0.1* specification
- An editor page to configure and run XCAP requests
  - Provides user interface controls to build valid requests
  - Run requests as single or in batch

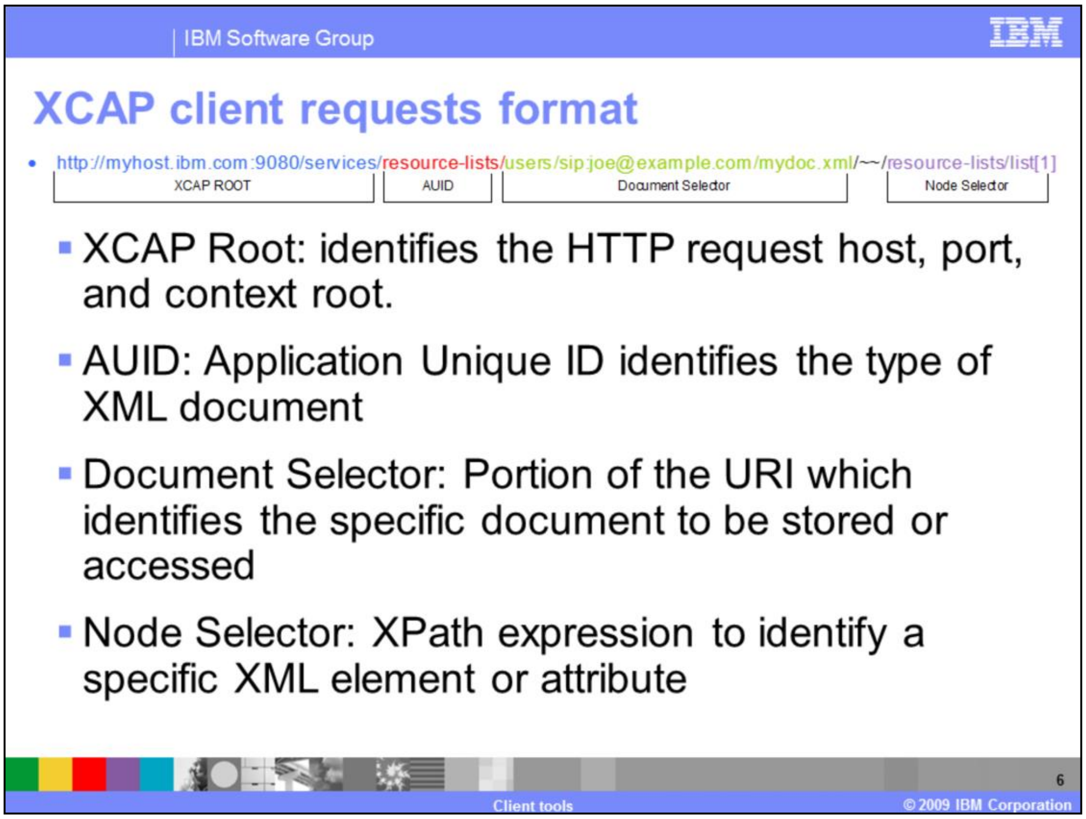
5

Client tools

© 2009 IBM Corporation

The XCAP client is built based on XML Configuration Access Protocol (XCAP) standards and is used to manage XML documents using the XCAP protocol. XCAP is a specification defined by Internet Engineering Task Force (IETF) that defines how XML documents are transported and managed over Hyper Text Transfer Protocol (HTTP). The XCAP client is the client test tool well suited for managing documents in the IBM XML document management server (XDMS) that uses XCAP over HTTP for managing its XML documents. The XDMS is a group list management server from IBM that is built using Open Mobile Alliance (OMA) standards. It provides network accessible storage, retrieval, and management of XML documents owned by various entities or users on a network. You can learn more about the IBM XDMS by visiting this URL: http://publib.boulder.ibm.com/infocenter/wtelecom/v6r2m0/index.jsp.

The XCAP client, built as an editor page, provides the user interface controls to build and validate the XCAP requests as described in the following slides. The requests can be run as single or batched requests.



IBM Software Group

XCAP client requests format

- http://myhost.ibm.com:9080/services/resource-lists/users/sip:joe@example.com/mydoc.xml/~~/resource-lists/list[1]

XCAP ROOT
AUID
Document Selector
Node Selector

- XCAP Root: identifies the HTTP request host, port, and context root.
- AUID: Application Unique ID identifies the type of XML document
- Document Selector: Portion of the URI which identifies the specific document to be stored or accessed
- Node Selector: XPath expression to identify a specific XML element or attribute

6

Client tools

© 2009 IBM Corporation

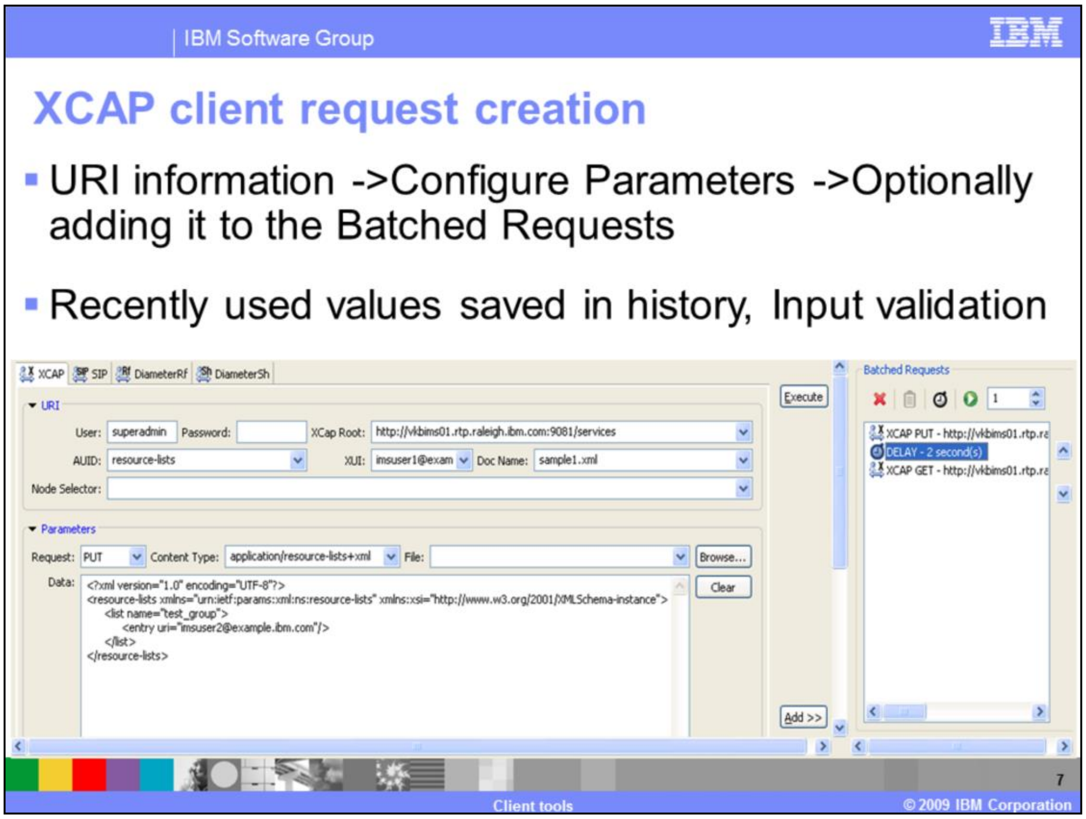
The figure in this slide shows the general format of an XCAP request to access XML documents. The XCAP request is made up of these elements:

- The XCAP Root that identifies the HTTP request host, port, and context root of the services

- The AUID which is the application unique ID that identifies the type of XML document being accessed. AUID's can be predefined IDs defined by the XCAP specification or user defined AUID's. For example resource-lists in the figure is a user defined AUID.

- The Document Selector portion of the URI identifies the specific document to be stored or accessed in the users or global space. Global space documents are only accessible to users with administrative access.

- The Node Selector is a limited XPath expression that can be used to identify a specific XML element or attribute which is to be updated, retrieved, or deleted.


IBM Software Group

XCAP client request creation

- URI information ->Configure Parameters ->Optionally adding it to the Batched Requests
- Recently used values saved in history, Input validation

User: superadmin
Password:
XCap Root: http://vkbims01.rtp.raleigh.ibm.com:9081/services
AUID: resource-lists
XUI: imsuser1@exam
Doc Name: sample1.xml
Node Selector:

Request: PUT
Content Type: application/resource-lists+xml
File:
Data:

```
<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <list name="test_group">
        <entry uri="imsuser2@example.ibm.com"/>
    </list>
</resource-lists>
```

1

XCAP PUT - http://vkbims01.rtp.ra
DELAY - 2 second(s)
XCAP GET - http://vkbims01.rtp.ra

7

Client tools

© 2009 IBM Corporation

The figure in the slide shows the screen capture of the XCAP requests section and the various user interface controls available to configure XCAP requests. Building an XCAP request involves three steps as follows:

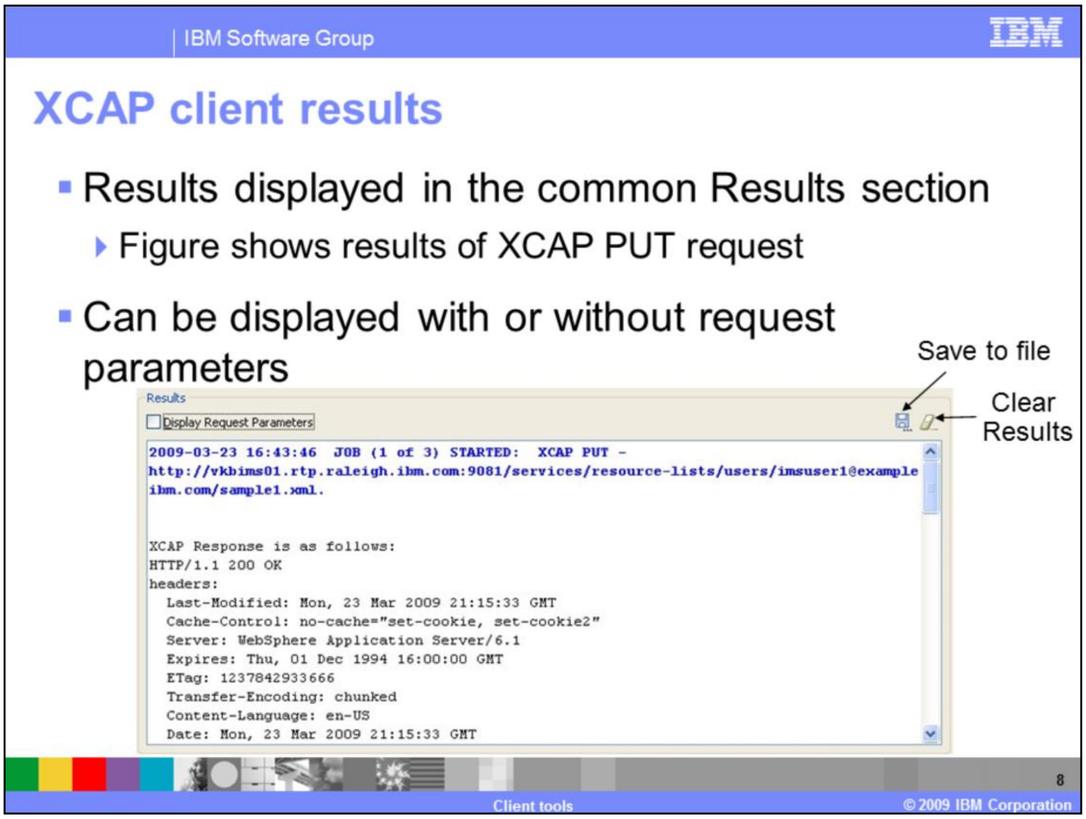
- Entering the URI information involved to build the XCAP URI with the XCAP Root, AUID and XUI values. The User and Password information are used for authenticating the request. The doc name is the name of the document to add, delete or modify

- Configuring the Parameters as follows:

> Selecting the type of operation to be performed using the Request field, and

> Selecting the Content Type for the request and populating the Data from a pre-existing file using the Browse button for the File field or manually entering the data in the Data field.

- Running the request after filling in the URI and Parameter sections as a single request or adding it to the Batch list using the Add button and running it as a Batched request. More details about the Batched requests is explained in slide 19 of this presentation.



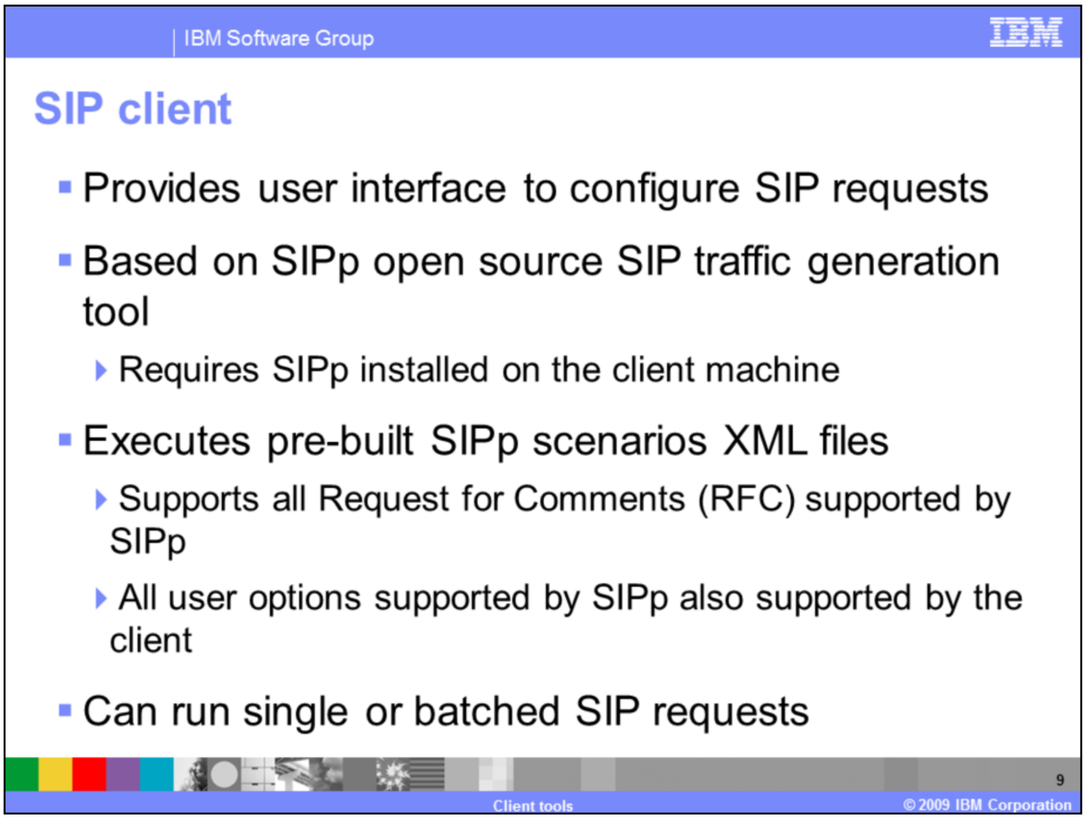
XCAP client results

- Results displayed in the common Results section
  - Figure shows results of XCAP PUT request
- Can be displayed with or without request parameters

```
2009-03-23 16:43:46  JOB (1 of 3) STARTED:  XCAP PUT -
http://vkbims01.rtp.raleigh.ibm.com:9081/services/resource-lists/users/imsuser1@example
ibm.com/sample1.xml.


XCAP Response is as follows:
HTTP/1.1 200 OK
headers:
  Last-Modified: Mon, 23 Mar 2009 21:15:33 GMT
  Cache-Control: no-cache="set-cookie, set-cookie2"
  Server: WebSphere Application Server/6.1
  Expires: Thu, 01 Dec 1994 16:00:00 GMT
  ETag: 1237842933666
  Transfer-Encoding: chunked
  Content-Language: en-US
  Date: Mon, 23 Mar 2009 21:15:33 GMT
```

The result of running single or batched XCAP requests is displayed in the common Results section. The figure in the slide shows the results of an XCAP PUT request of a sample.xml file as resource-lists document for imsuser1. The XCAP response shows the HTTP response code of 200 OK which means the document was inserted successfully.

The Results section is common for all the test clients and displays the results of all the client requests. The Results can be displayed with or without the request parameters. By default the Display Request Parameters check box is checked. The results displayed can be saved to an external file using the save to file button and the contents can be cleared using the clear button.


IBM Software Group

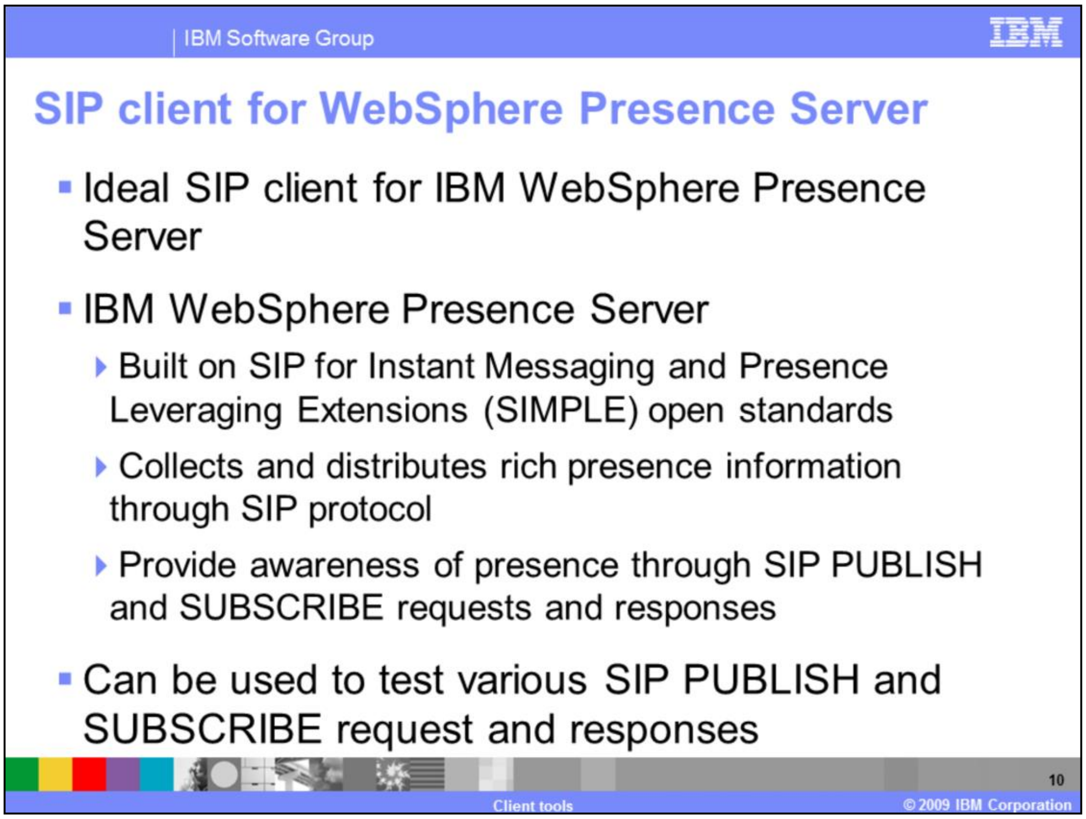
SIP client

- Provides user interface to configure SIP requests
- Based on SIPp open source SIP traffic generation tool
  - Requires SIPp installed on the client machine
- Executes pre-built SIPp scenarios XML files
  - Supports all Request for Comments (RFC) supported by SIPp
  - All user options supported by SIPp also supported by the client
- Can run single or batched SIP requests

9

Client tools © 2009 IBM Corporation

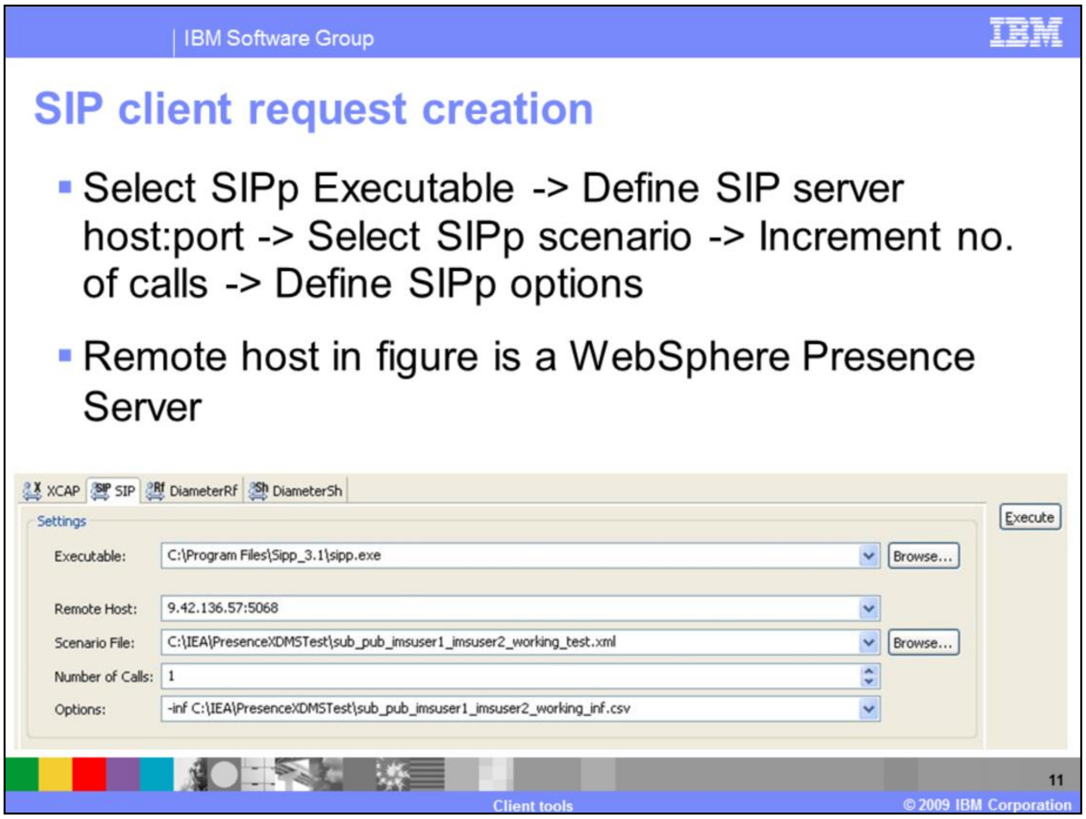
The SIP Client provides a user interface for configuring and running SIP requests. The client is based on the SIPp open source SIP traffic generation tool and requires SIPp installation on the client machine where the SIP client is installed. The client is capable of executing all pre-built SIPp scenario XML files. The client supports all the SIP Request for Comments (RFCs) supported by the SIPp, such as RFC 3261, 2823, 3725 and so on. The client supports all the configuration options provided by SIPp, and supports running SIP requests as single or batched requests.



SIP client for WebSphere Presence Server

- Ideal SIP client for IBM WebSphere Presence Server
- IBM WebSphere Presence Server
  - Built on SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) open standards
  - Collects and distributes rich presence information through SIP protocol
  - Provide awareness of presence through SIP PUBLISH and SUBSCRIBE requests and responses
- Can be used to test various SIP PUBLISH and SUBSCRIBE request and responses

The SIP Client is a client tool that can act as a SIP user agent for the IBM WebSphere Presence Server. The IBM WebSphere Presence Server is built on SIP for Instant Messaging and Presence Leveraging Extensions (SIMPLE) open standards. It collects and distributes rich presence information using the SIP protocol. It uses the SIP PUBLISH and SUBSCRIBE requests and responses to provide and distribute the presence information of a presentity on the network. The SIP client can be used to test various SIP PUBLISH and SUBSCRIBE scenarios by building SIPp scenarios following the SIPp document type definition.



SIP client request creation

- Select SIPp Executable -> Define SIP server host:port -> Select SIPp scenario -> Increment no. of calls -> Define SIPp options
- Remote host in figure is a WebSphere Presence Server

Executable:	C:\Program Files\Sipp_3.1\sipp.exe
Remote Host:	9.42.136.57:5068
Scenario File:	C:\IEA\PresenceXDMSTest\sub_pub_imsuser1_imsuser2_working_test.xml
Number of Calls:	1
Options:	-inf C:\IEA\PresenceXDMSTest\sub_pub_imsuser1_imsuser2_working_inf.csv

Building a SIP request using the SIP Client involves these steps:

- Selecting the SIPp executable file from the location where SIPp is installed on the local machine where the client is running.

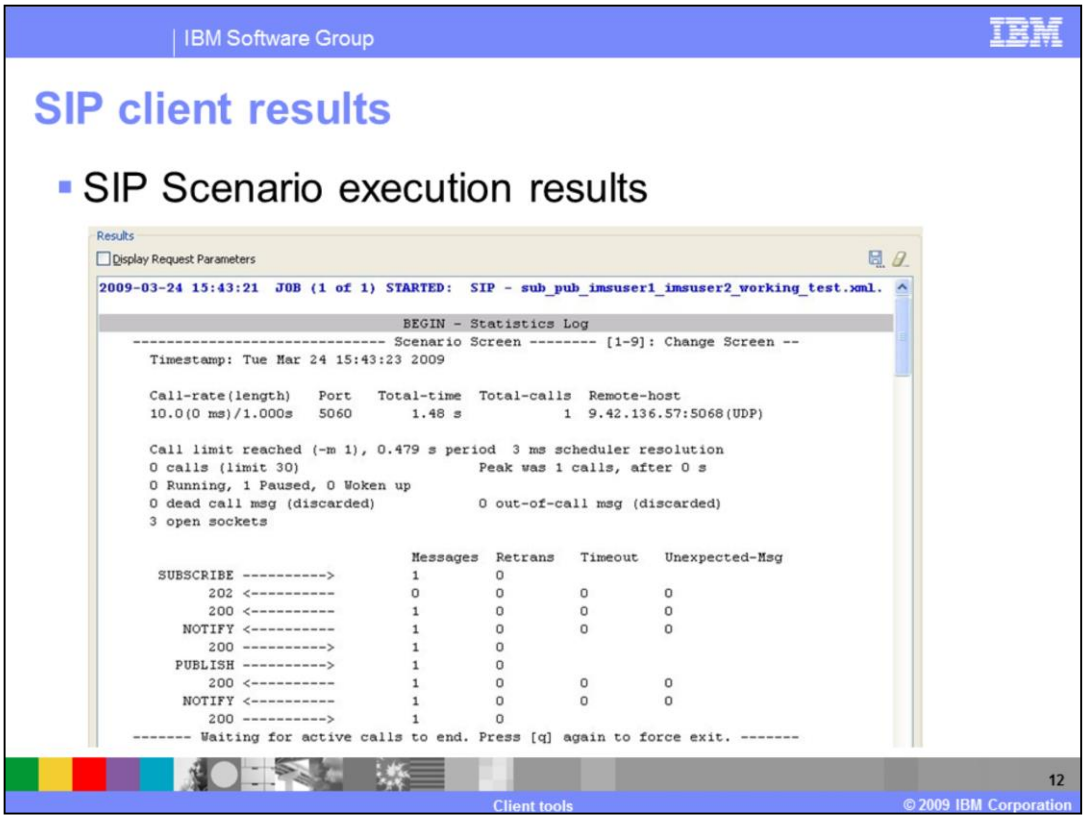
- Defining the Remote SIP Server host followed by a colon and its SIP listening port,

- Defining the SIPp scenario file, which is a prebuilt scenario file built following the sipp.dtd,

- Incrementing the number of SIP calls to generate, the client defaults to one call,

- Providing any SIPp options that need to be fed along with the SIP request. By default some SIPp options like -trace_err, -trace_msg, -trace_screen are provided by the client and need not be explicitly mentioned in the Options.

To demonstrate the use of the client, the values in the figure depict one SIP call using a PUBLISH and SUBSCRIBE request response SIPp scenario against a WebSphere Presence Server. WebSphere Presence Server is installed on the host with IP address 9.42.136.57 with its SIP listening port as 5068. An injection file is passed as an option to run the scenario across two SIP user agents. The results are presented in the next slide.



SIP client results

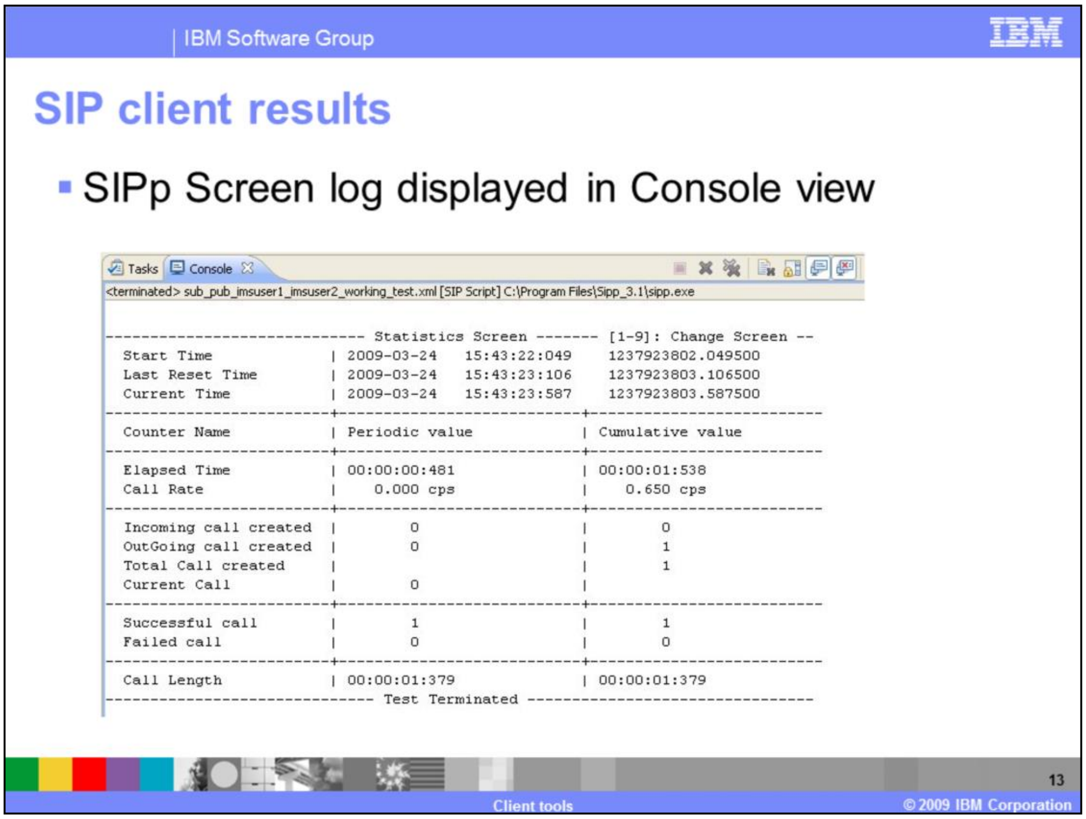
- SIP Scenario execution results

Results

☐ Display Request Parameters

```
2009-03-24 15:43:21  JOB (1 of 1) STARTED:  SIP - sub_pub_imsuser1_imsuser2_working_test.xml.

                                BEGIN - Statistics Log
    ------------------------------ Scenario Screen -------- [1-9]: Change Screen --
      Timestamp: Tue Mar 24 15:43:23 2009

      Call-rate(length)   Port   Total-time  Total-calls  Remote-host
      10.0(0 ms)/1.000s   5060       1.48 s            1  9.42.136.57:5068(UDP)

      Call limit reached (-m 1), 0.479 s period  3 ms scheduler resolution
      0 calls (limit 30)                   Peak was 1 calls, after 0 s
      0 Running, 1 Paused, 0 Woken up
      0 dead call msg (discarded)          0 out-of-call msg (discarded)
      3 open sockets

                                Messages  Retrans   Timeout   Unexpected-Msg
       SUBSCRIBE ---------->         1         0
             202 <----------         0         0         0         0
             200 <----------         1         0         0         0
          NOTIFY <----------         1         0         0         0
             200 ---------->         1         0
         PUBLISH ---------->         1         0
             200 <----------         1         0         0         0
          NOTIFY <----------         1         0         0         0
             200 ---------->         1         0
    ------- Waiting for active calls to end. Press [q] again to force exit. -------
```

The results of the SIP scenario run are displayed in the common Results section. The figure shows the results of the SIP PUBLISH and SUBSCRIBE scenario run in the previous slide. The Results section also displays the complete set of SIP messages sent and received between the SIP client and SIP server.

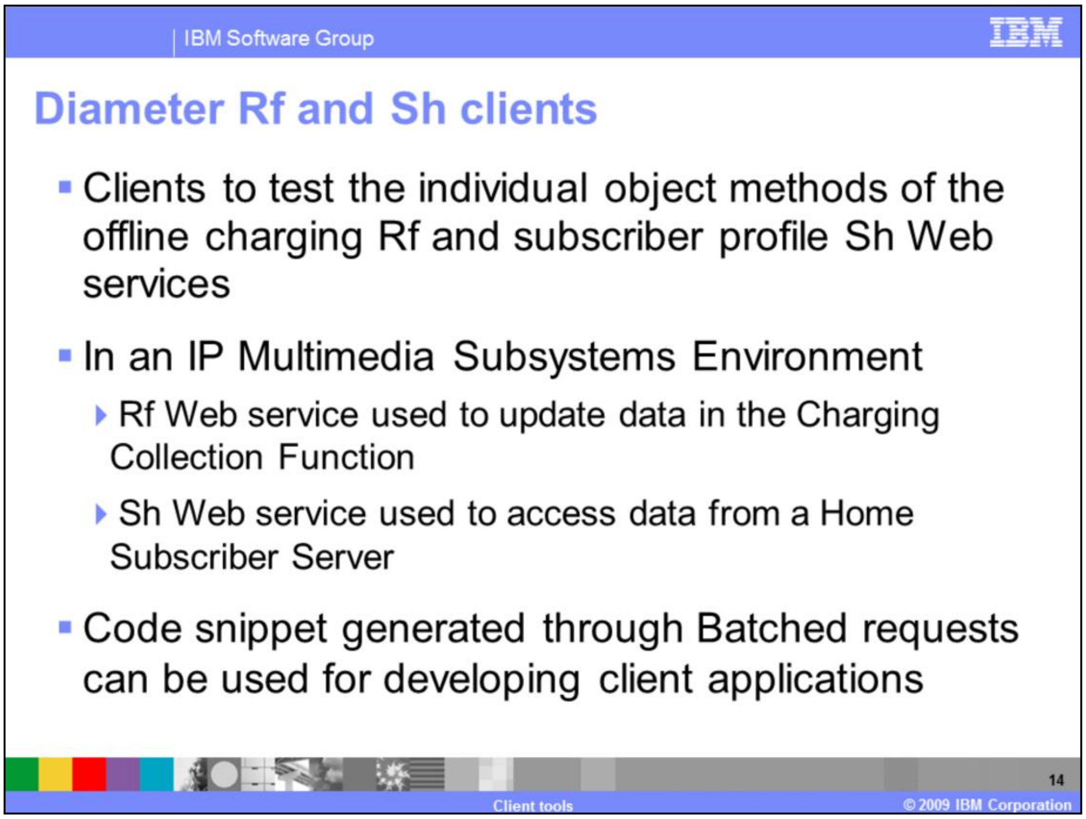


SIP client results

- SIPp Screen log displayed in Console view

```
<terminated> sub_pub_imsuser1_imsuser2_working_test.xml [SIP Script] C:\Program Files\Sipp_3.1\sipp.exe

----------------------------- Statistics Screen ------- [1-9]: Change Screen --
  Start Time             | 2009-03-24   15:43:22:049    1237923802.049500
  Last Reset Time        | 2009-03-24   15:43:23:106    1237923803.106500
  Current Time           | 2009-03-24   15:43:23:587    1237923803.587500
-------------------------+---------------------------+--------------------------
  Counter Name           | Periodic value            | Cumulative value
-------------------------+---------------------------+--------------------------
  Elapsed Time           | 00:00:00:481              | 00:00:01:538
  Call Rate              |    0.000 cps              |    0.650 cps
-------------------------+---------------------------+--------------------------
  Incoming call created  |        0                  |        0
  OutGoing call created  |        0                  |        1
  Total Call created     |                           |        1
  Current Call           |        0                  |
-------------------------+---------------------------+--------------------------
  Successful call        |        1                  |        1
  Failed call            |        0                  |        0
-------------------------+---------------------------+--------------------------
  Call Length            | 00:00:01:379              | 00:00:01:379
------------------------------ Test Terminated --------------------------------
```

In addition to the Results view, the SIPp screen log of the SIPp Scenario is displayed in the Console view of Rational® Application Developer where the toolkit is installed. The Figure shows the success or failure of the SIP call from the end of the screen log in the Console view.



IBM Software Group

Diameter Rf and Sh clients

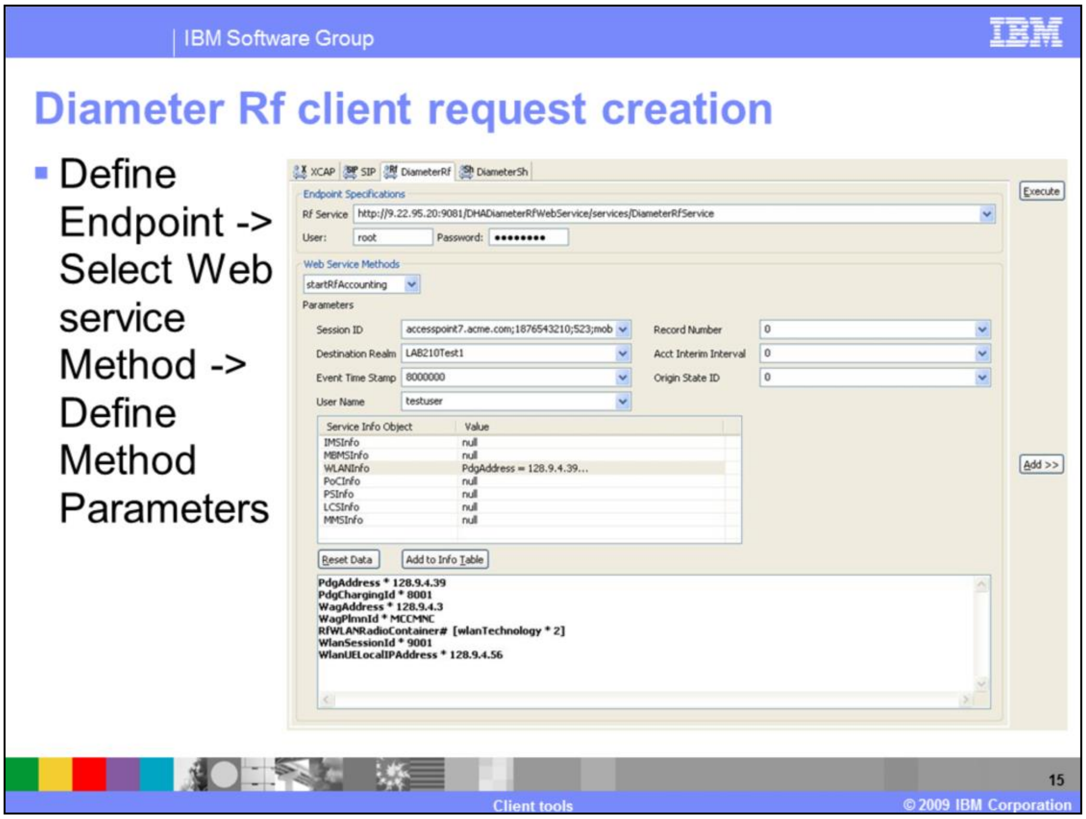
- Clients to test the individual object methods of the offline charging Rf and subscriber profile Sh Web services
- In an IP Multimedia Subsystems Environment
  - Rf Web service used to update data in the Charging Collection Function
  - Sh Web service used to access data from a Home Subscriber Server
- Code snippet generated through Batched requests can be used for developing client applications

14

Client tools

© 2009 IBM Corporation

The Diameter Rf and Sh client tools provide a user interface driven way to test the individual object methods of the Web services description language (WSDLs) of the offline charging Rf and subscriber profile Sh Web services. In an IP Multimedia Subsystem environment the Rf Web service is used to update offline charging data in the Charging Collection Function and the Sh Web service is to access subscriber profile data from the Home Subscriber Server. Just like the other client tools, the Batch request tool provides a feature to copy the code snippet from the test that can be reused in developing client applications. A more detailed description about the Batch requests tool is provided on slide 19.



Diameter Rf client request creation

- Define Endpoint -> Select Web service Method -> Define Method Parameters

XCAP | SIP | DiameterRf | DiameterSh

Endpoint Specifications

Rf Service: http://9.22.95.20:9081/DHADiameterRfWebService/services/DiameterRfService

User: root Password: ••••••••

Web Service Methods

startRfAccounting

Parameters

Session ID	accesspoint7.acme.com;1876543210;523;mob	Record Number	0
Destination Realm	LAB210Test1	Acct Interim Interval	0
Event Time Stamp	8000000	Origin State ID	0
User Name	testuser		

Service Info Object	Value
IMSInfo	null
MBMSInfo	null
WLANInfo	PdgAddress = 128.9.4.39...
PoCInfo	null
PSInfo	null
LCSInfo	null
MMSInfo	null

```
PdgAddress * 128.9.4.39
PdgChargingId * 8001
WagAddress * 128.9.4.3
WagPlmnId * MCCMNC
RfWLANRadioContainer# [wlanTechnology * 2]
WlanSessionId * 9001
WlanUELocalIPAddress * 128.9.4.56
```

The figure in the slide shows the screen capture of the Diameter Rf offline charging request section and the various user interface controls available to configure Rf requests. Creating a Rf Web service request involves three steps that include:

- Defining the Endpoint Specification for the Rf service which is the URL of the Diameter server where the Rf service endpoint is configured. The User and Password information are used for authenticating the request.

- Selecting a method from the Web service Methods drop down.

- Defining all the parameters of the method. The common parameters for all the Rf Web service methods include:

Session ID – Is a globally unique identifier that indentifies a user session. For example aaa://host.example.com;protocol=diameter;-117302099;1

Destination Realm – Is the realm the subscriber belongs to. The destination realm is a required parameter and must be a fully qualified domain name. The value specified must match the specific realmName property that is defined in the Diameter_Rf.properties file.

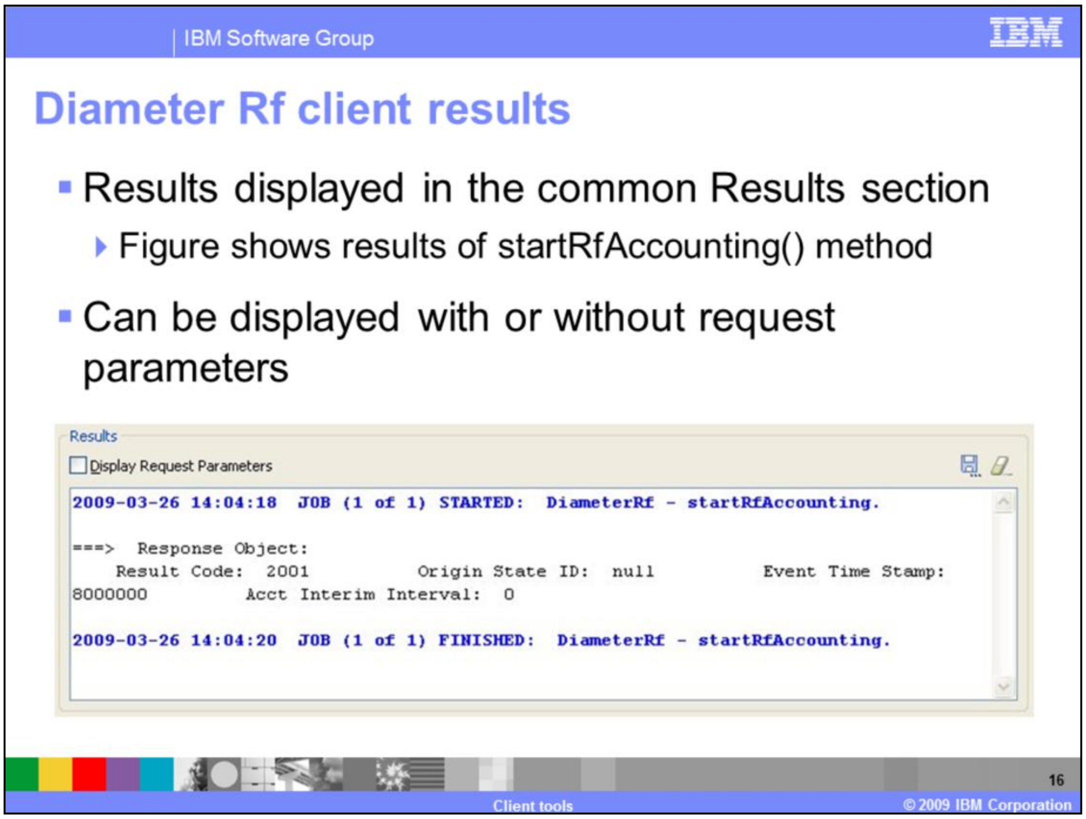
Event Time Stamp – Is the timestamp to record the time the event occurred.

User Name – Is the private user identity if available in the node.

Record Number – Is zero for a single session and increments by one for follow on messages.

Acct Interim Interval – Is the start time for the session, in seconds.

Origin State ID – holds the value to track the incremented value of possible times the client has lost the state or a possible reboot has occurred.



Diameter Rf client results

- Results displayed in the common Results section
  - Figure shows results of startRfAccounting() method
- Can be displayed with or without request parameters

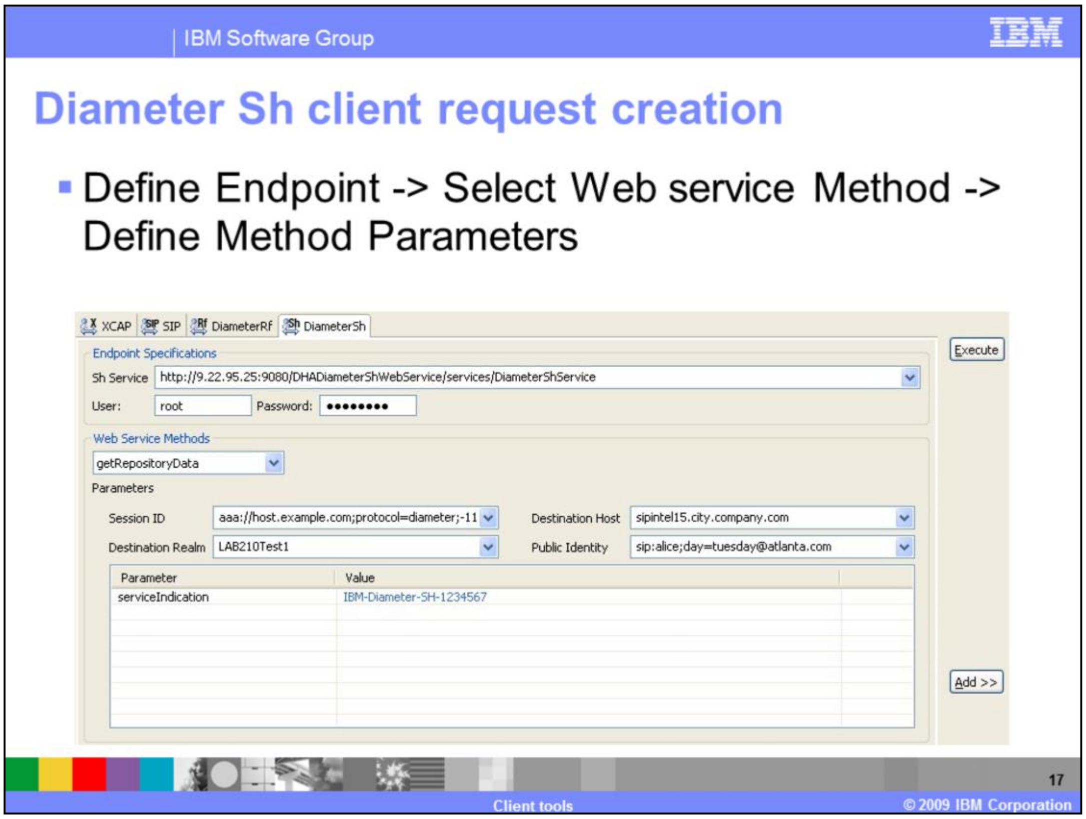
Results

☐ Display Request Parameters

```
2009-03-26 14:04:18  JOB (1 of 1) STARTED:  DiameterRf - startRfAccounting.

===>  Response Object:
    Result Code:  2001          Origin State ID:  null          Event Time Stamp:
8000000          Acct Interim Interval:  0

2009-03-26 14:04:20  JOB (1 of 1) FINISHED:  DiameterRf - startRfAccounting.
```

The results of running single or batched Diameter Rf request are displayed in the common Results section. The figure in the slide shows the results of a Diameter Rf request. The Response Object shows the service data returned for the startRfAccounting method.

The Results section is common for all the test clients and displays the results of all the client requests. The Results can be displayed with or without the request parameters. By default the Display Request Parameters check box is checked. The results displayed can be saved to an external file using the save to file button and the contents can be cleared using the Clear button. You can change the font or color of the text by right clicking and selecting the Font menu item.



Diameter Sh client request creation

- Define Endpoint -> Select Web service Method -> Define Method Parameters

XCAP | SIP | DiameterRf | DiameterSh

Endpoint Specifications

Sh Service: http://9.22.95.25:9080/DHADiameterShWebService/services/DiameterShService

User: root Password: ●●●●●●●●

Execute

Web Service Methods

getRepositoryData

Parameters

Session ID: aaa://host.example.com;protocol=diameter;-11

Destination Host: sipintel15.city.company.com

Destination Realm: LAB210Test1

Public Identity: sip:alice;day=tuesday@atlanta.com

Parameter	Value
serviceIndication	IBM-Diameter-SH-1234567

Add >>

The figure in the slide shows the Diameter Sh subscriber profile request section and the various user interface controls available to configure Sh requests. Creating a Sh Web service request involves three steps that include:

- Defining the Endpoint Specification for the Sh service which is the URL of the Diameter server where the Sh service endpoint is configured. The User and Password information are used for authenticating the user initiating the request.

- Selecting a method from the Web service Methods drop down.

- Defining all the parameters to the method. The common parameters for all the Sh Web service methods include:

Session ID – Is a globally unique identifier that indentifies a user session. For example aaa://host.example.com;protocol=diameter;-117302099;1

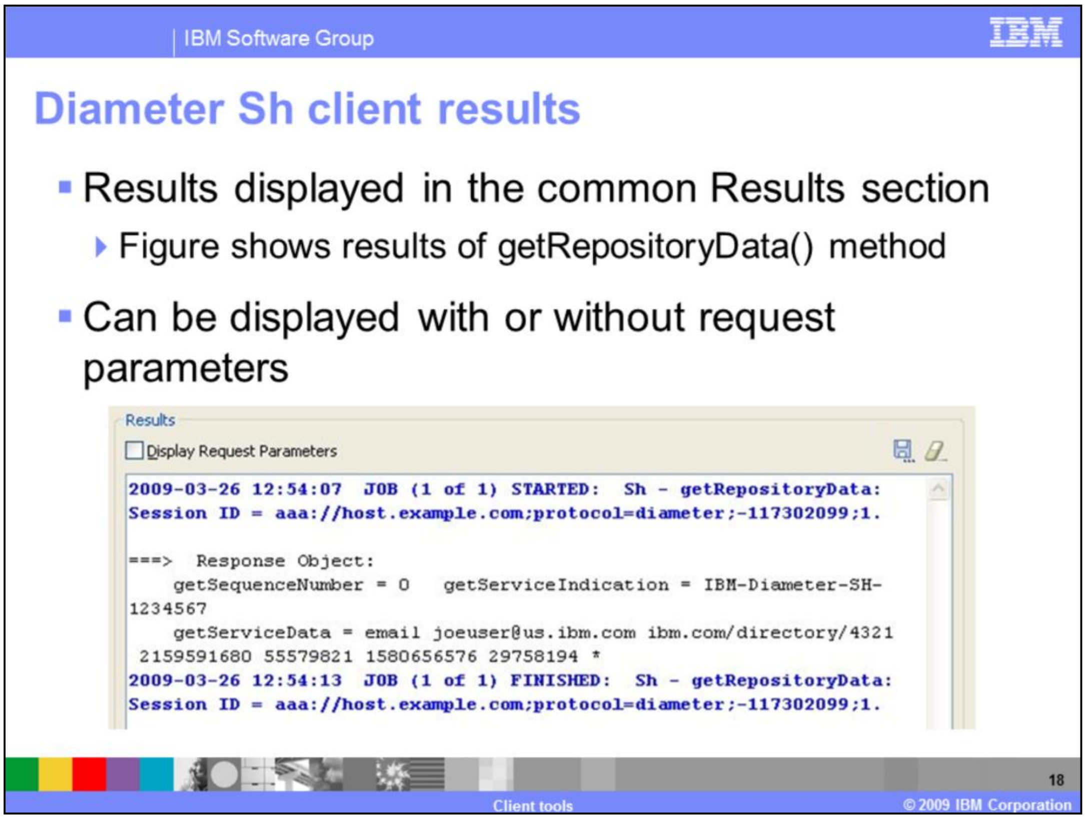
Destination Realm – Is the realm the subscriber belongs to. The destination realm is a required parameter and must be a fully qualified domain name. The value specified must match the specific realmName property that is defined in the Diameter_Sh.properties file.

Destination Host – Is the qualified domain name of the HSS that the Sh subscriber profile Web service will send this request to. This input parameter is optional. For example: sipintel15.city.example.com or diameter.example.com

Public Identity - Specifies the public user identity or public service identity. This can be either a SIP URL or a TEL URL. Example: sip:alice;day=tuesday@example.com

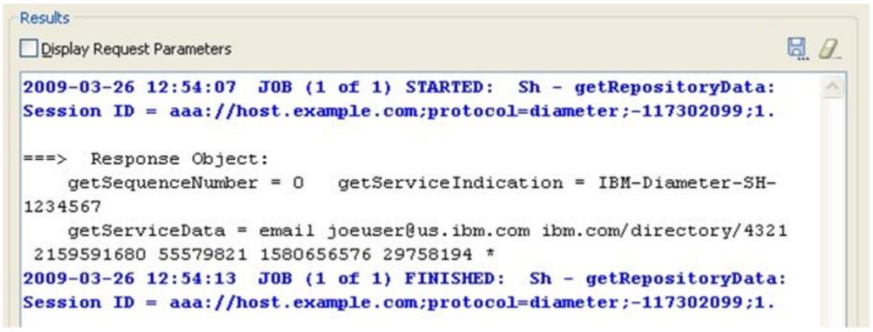
The figure shows the getRepositoryData method selected and its parameters which include:

serviceIndication – Is a unique identifier for the requested service data. Example: IBM-Diameter-SH-1234567



Diameter Sh client results

- Results displayed in the common Results section
  - Figure shows results of getRepositoryData() method
- Can be displayed with or without request parameters

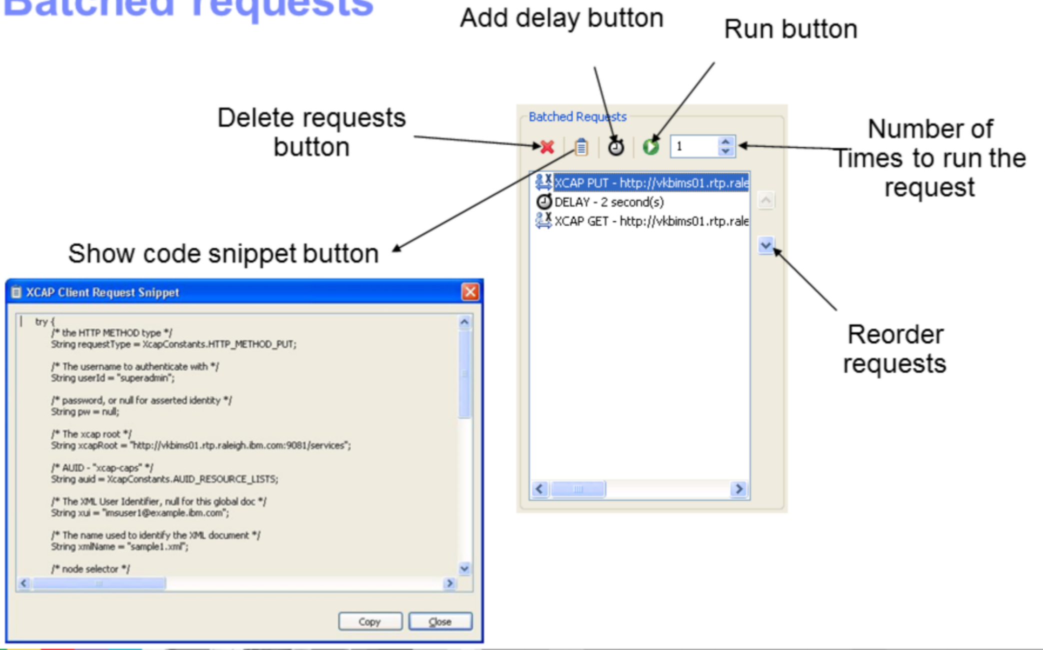


```
2009-03-26 12:54:07  JOB (1 of 1) STARTED:  Sh - getRepositoryData:
Session ID = aaa://host.example.com;protocol=diameter;-117302099;1.

===>  Response Object:
    getSequenceNumber = 0   getServiceIndication = IBM-Diameter-SH-
1234567
    getServiceData = email joeuser@us.ibm.com ibm.com/directory/4321
 2159591680 55579821 1580656576 29758194 *
2009-03-26 12:54:13  JOB (1 of 1) FINISHED:  Sh - getRepositoryData:
Session ID = aaa://host.example.com;protocol=diameter;-117302099;1.
```

The results of running single or batched Diameter Sh request are displayed in the common Results section. The figure in the slide shows the results of a Diameter Sh request. The Response Object shows the service data returned for the getRepositoryDate method along with the service identification information passed in with the request from the Diameter Sh service endpoint as shown in the previous slide.

The Results section is common for all the test clients and displays the results of all the client requests. The Results can be displayed with or without the request parameters. By default the Display Request Parameters check box is checked. The results displayed can be saved to an external file using the save to file button and the contents can be cleared using the Clear button.

Batched requests



XCAP PUT - http://vkbims01.rtp.rale
DELAY - 2 second(s)
XCAP GET - http://vkbims01.rtp.rale

```
try {
    /* the HTTP METHOD type */
    String requestType = XcapConstants.HTTP_METHOD_PUT;

    /* The username to authenticate with */
    String userId = "superadmin";

    /* password, or null for asserted identity */
    String pw = null;

    /* The xcap root */
    String xcapRoot = "http://vkbims01.rtp.raleigh.ibm.com:9081/services";

    /* AUID - "xcap-caps" */
    String auid = XcapConstants.AUID_RESOURCE_LISTS;

    /* The XML User Identifier, null for this global doc */
    String xui = "imsuser1@example.ibm.com";

    /* The name used to identify the XML document */
    String xmlName = "sample1.xml";

    /* node selector */
```

The Batched Request section of the client editor is common to all client test tools. Multiple client requests from different clients can be run together through the Bathed Requests section. The Bathed Requests section provides some additional useful features that are explained through the figure in the slide. These additional features include:

- Performing validation on request parameters in the request before adding to the request batch

- Reorder Requests button for reordering requests already added to the batch

- Number of Execution Iterations field to increase or decrease the number of times the requests need to be run

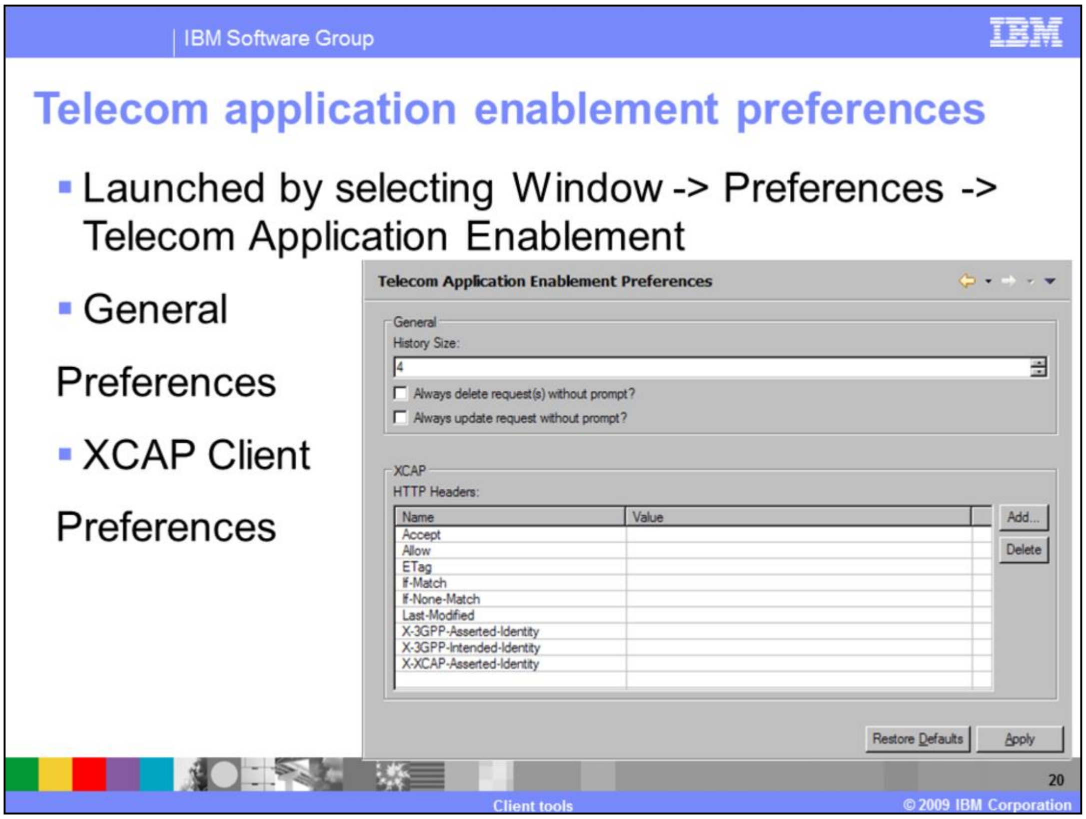
- Execute button to run the batched requests

- Add delay button to add time delay in seconds between requests

- Show Code Snippet button which shows the Java™ code that is built and used to run the request using the parameter values entered by the user. The code in this dialog can be copied using the Copy button and pasted in a Java class and reused for building a Java client for the request.

- Delete Requests button to delete any requests already in the batch.

Upon running the tests, moving the mouse up and down and selecting different requests shows the corresponding output in the results window. This is a useful feature, especially if the batch consists of a number of requests.



Telecom application enablement preferences

- Launched by selecting Window -> Preferences -> Telecom Application Enablement
- General Preferences
- XCAP Client Preferences

Telecom Application Enablement Preferences

General

History Size: 4

☐ Always delete request(s) without prompt?

☐ Always update request without prompt?

XCAP

HTTP Headers:

Name	Value
Accept	
Allow	
ETag	
If-Match	
If-None-Match	
Last-Modified	
X-3GPP-Asserted-Identity	
X-3GPP-Intended-Identity	
X-XCAP-Asserted-Identity	

The Telecom Application Enablement feature in the toolkit provides two sets of preferences in the Telecom Application Enablement Preferences page. These preferences can be launched from Rational Application Developer by selecting Window -> Preferences -> Telecom Application Enablement or by way of the pop-up menu of the results window. The two sets of preferences include General and XCAP preferences.

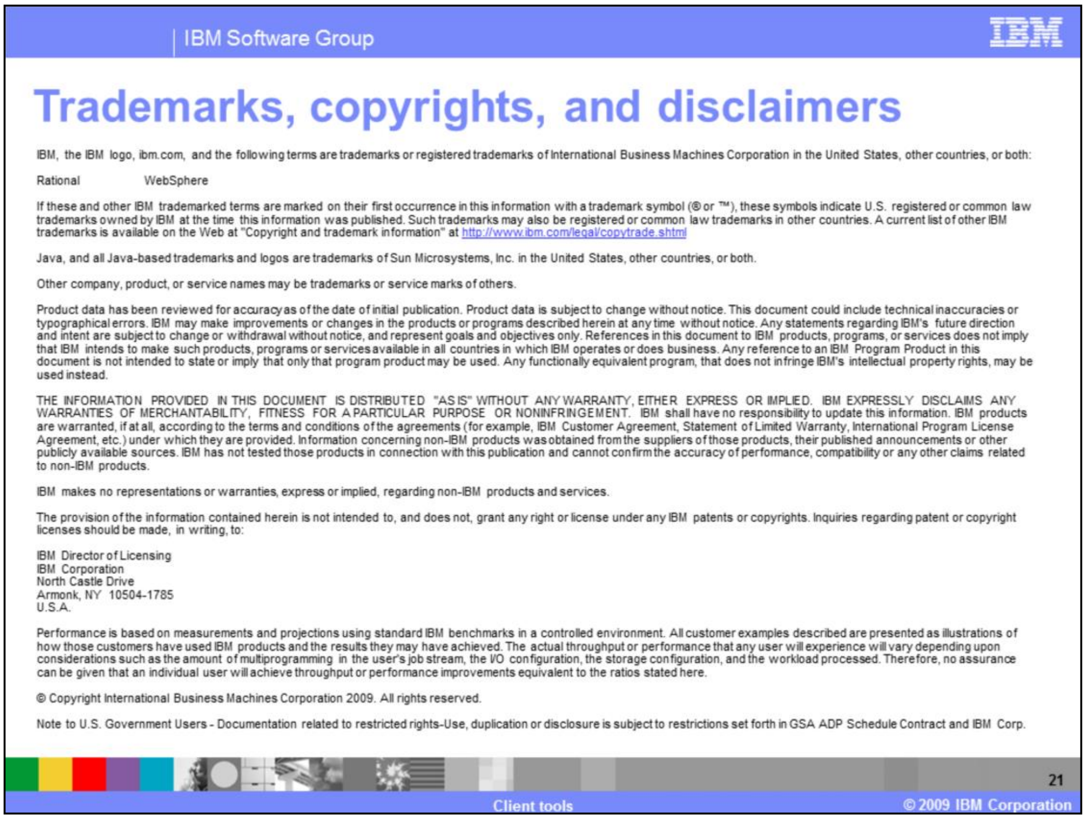
The General preferences include the following:

- The History Size field provides the number of previous user entries entered in all user interface controls of the toolkit that need to be stored so that previously entered values can be viewed and selected.

- The Always delete requests without prompt check box provides the option to delete the requests from the Batched requests section without a prompt

- The Always update request without prompt check box provides the option to update a request from the Batched request section without a prompt.

The XCAP Preferences page provides a way to configure existing HTTP Header values for XCAP requests in addition to adding new headers and deleting existing headers. The list of existing HTTP headers are listed under the Name column of the HTTP Headers table of the preference page.



Trademarks, copyrights, and disclaimers