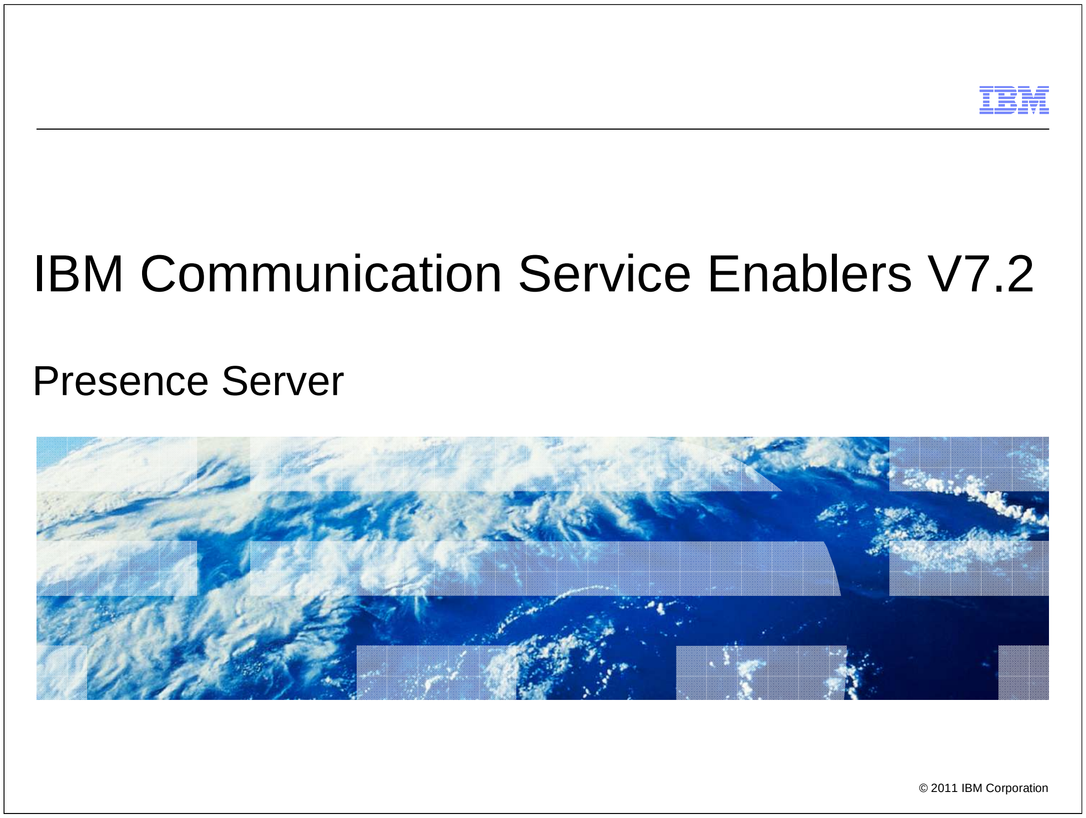


IBM Communication Service Enablers V7.2

Presence Server

This presentation deals with Presence Server version 7.2 and the new features available in this release.

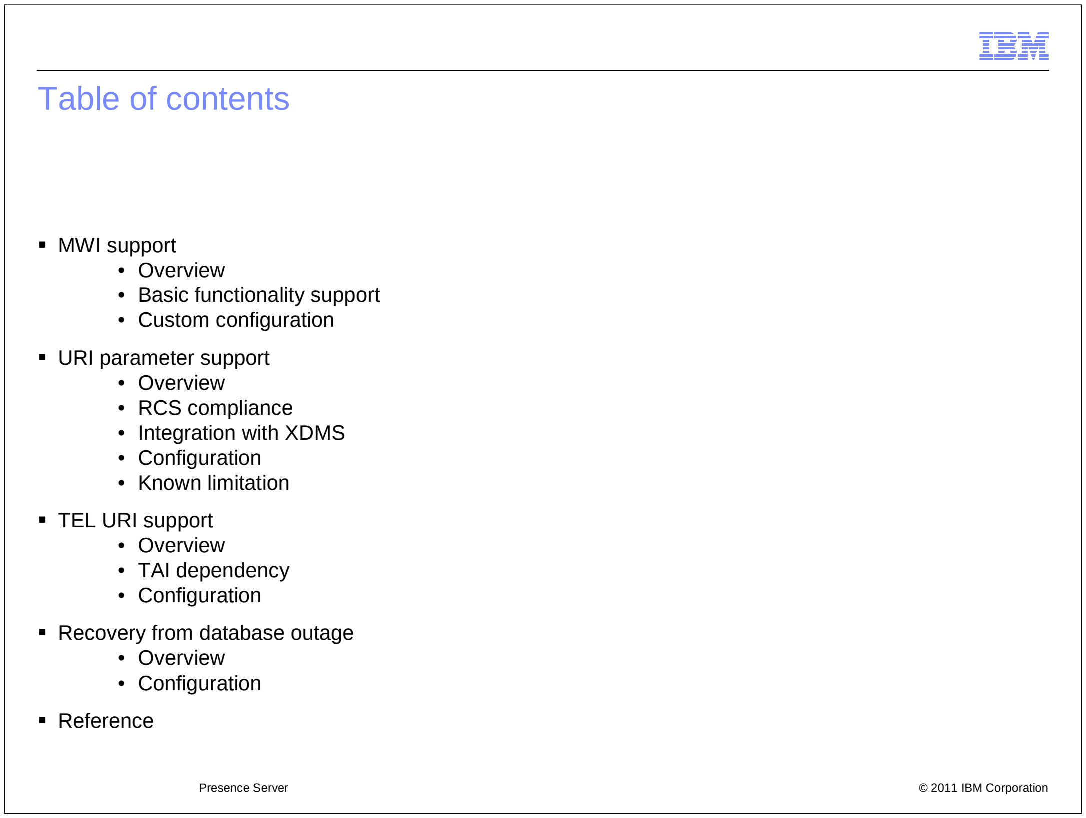


Table of contents

- MWI support
  - Overview
  - Basic functionality support
  - Custom configuration
- URI parameter support
  - Overview
  - RCS compliance
  - Integration with XDMS
  - Configuration
  - Known limitation
- TEL URI support
  - Overview
  - TAI dependency
  - Configuration
- Recovery from database outage
  - Overview
  - Configuration
- Reference

Presence Server

The agenda includes the four new features in the 7.2 release: Message Waiting Indicator support, URI parameter support, TEL URI support, and recovery from database outage.

- **Message Waiting Indicator support**:

-- Overview

-- Basic functionality support

-- Custom configuration

- **URI parameter support**:

-- Overview

-- RCS compliance

-- Integration with XDMS

-- Configuration

-- Known limitation

- **TEL URI support**:

-- Overview

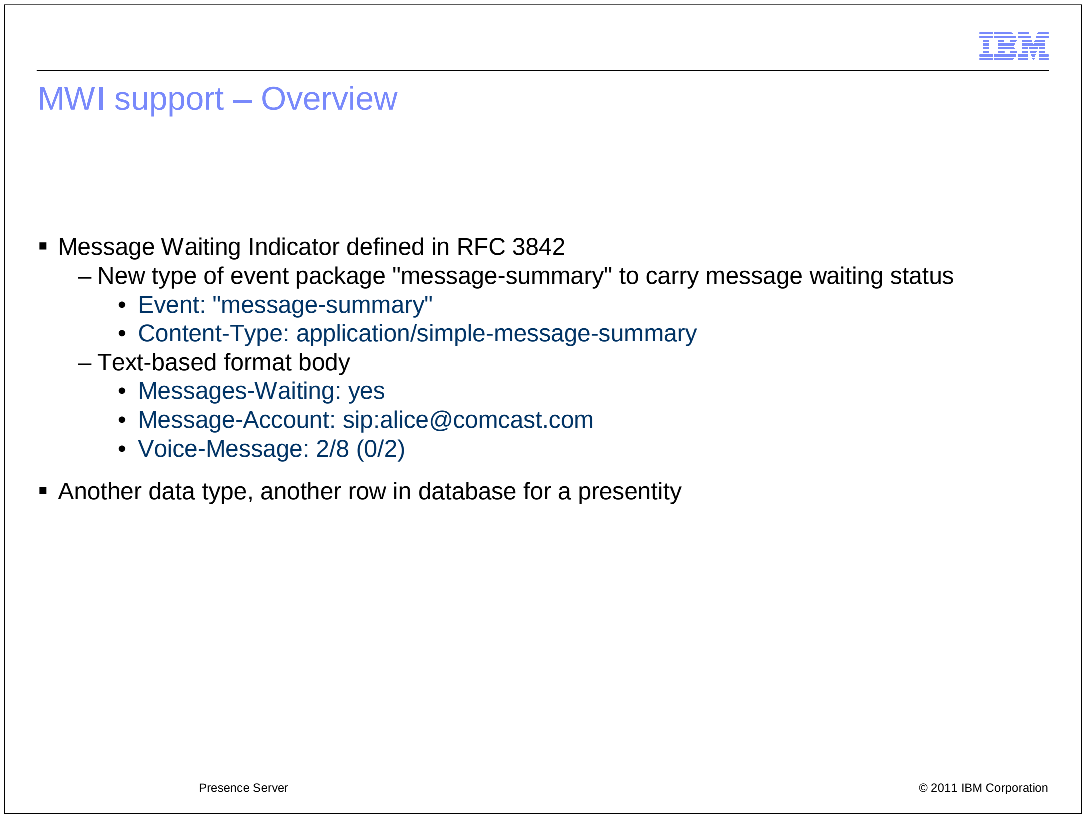
-- TAI component dependency

-- Configuration

- **Recovery from database outage**:

-- Overview

-- Configuration



MWI support – Overview

- Message Waiting Indicator defined in RFC 3842
  - New type of event package "message-summary" to carry message waiting status
    - Event: "message-summary"
    - Content-Type: application/simple-message-summary
  - Text-based format body
    - Messages-Waiting: yes
    - Message-Account: sip:alice@comcast.com
    - Voice-Message: 2/8 (0/2)
- Another data type, another row in database for a presentity

Presence Server

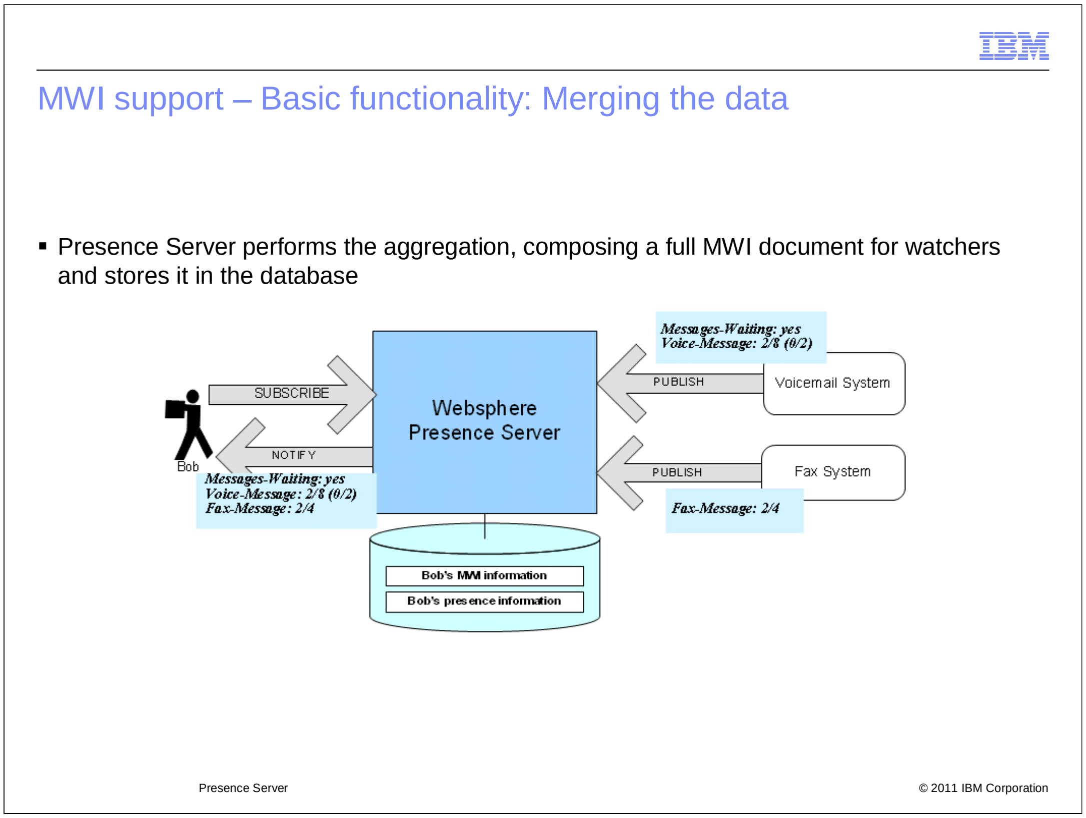
This slide gives an overview of the Message Waiting Indicator (MWI) support.

Presence Server 7.2 implements the RFC 3842, which defines a new type of event package "message-summary" to carry message waiting status.

The user can now subscribe to the “message-summary” events and for that they have to specify “message-summary” in the event package and the supported content type for MWI.

Then they are notified of the messages waiting for them on voice mail or other systems.

MWI is just another type of data for which a separate row is created in the database.



MWI support – Basic functionality: Merging the data

- Presence Server performs the aggregation, composing a full MWI document for watchers and stores it in the database

Presence Server

This slide deals with the merging data support functionality of MWI.

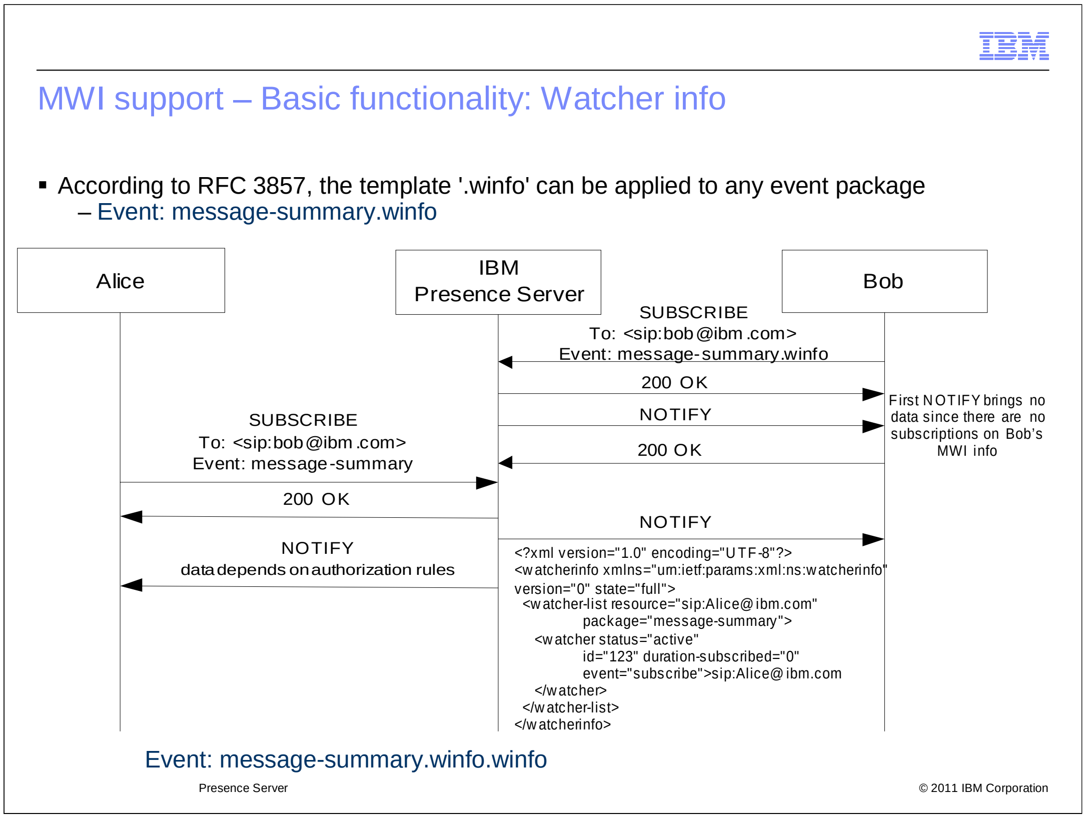
Presence Server performs the aggregation, composing a full MWI document composed of MWI segments received from different providers. It stores the composed document in the database exactly as it does for the presence events.

The diagram shows the user, Bob, subscribing to a message summary event package.

On the right side of the diagram, are two providers that carry two different MWI documents.

As the documents are related to the same user, Presence Server aggregates them into one MWI document. The aggregation process for message summary data is to put the lines received from different providers one after the other and finally the composed document is stored in a database in a separate row.

Once both the subscriptions are done, the notification with the data is sent from the voice mail and fax system as updates to Bob.



MWI support – Basic functionality: Watcher info

- According to RFC 3857, the template '.winfo' can be applied to any event package
  - Event: message-summary.winfo

Alice

IBM
Presence Server

Bob

SUBSCRIBE
To: <sip:bob@ibm.com>
Event: message-summary.winfo

200 OK

NOTIFY

First NOTIFY brings no data since there are no subscriptions on Bob's MWI info

200 OK

SUBSCRIBE
To: <sip:bob@ibm.com>
Event: message-summary

200 OK

NOTIFY

NOTIFY
data depends on authorization rules

```
<?xml version="1.0" encoding="UTF-8"?>
<watcherinfo xmlns="urn:ietf:params:xml:ns:watcherinfo"
version="0" state="full">
  <watcher-list resource="sip:Alice@ibm.com"
              package="message-summary">
    <watcher status="active"
              id="123" duration-subscribed="0"
              event="subscribe">sip:Alice@ibm.com
    </watcher>
  </watcher-list>
</watcherinfo>
```

Event: message-summary.winfo.winfo

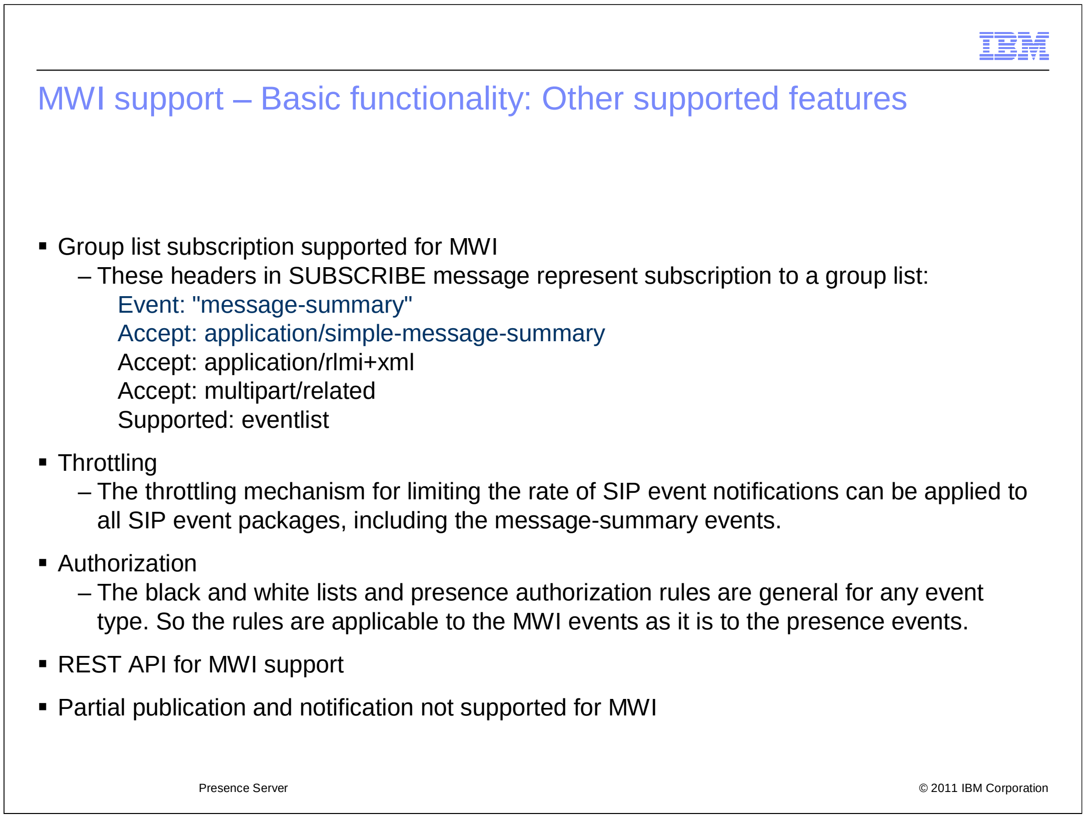
Presence Server

This slide deals with the watcher info support for MWI.

According to RFC 3857, which defines the watcher info behavior, the template '.winfo' can be applied to any event package. Presence Server applies the ".winfo" extension to the message summary event package that allows users to obtain information about their MWI event watcher.

The flow diagram shows Bob subscribing to his MWI watcher info event to learn about people who have subscribed to his message summary event.

Another user, Alice, subscribes to Bob's MWI data. As a result, Bob is notified about a new watcher, Alice, being interested in his MWI data.



MWI support – Basic functionality: Other supported features

- Group list subscription supported for MWI
  - These headers in SUBSCRIBE message represent subscription to a group list:

    Event: "message-summary"
    Accept: application/simple-message-summary
    Accept: application/rlmi+xml
    Accept: multipart/related
    Supported: eventlist
- Throttling
  - The throttling mechanism for limiting the rate of SIP event notifications can be applied to all SIP event packages, including the message-summary events.
- Authorization
  - The black and white lists and presence authorization rules are general for any event type. So the rules are applicable to the MWI events as it is to the presence events.
- REST API for MWI support
- Partial publication and notification not supported for MWI

Presence Server

This slide deals with other supported features for MWI.

Presence Server supports Group List subscriptions for message summary events. A group subscriber interested in the message summary status of a group member can send a SIP subscribe message, in which he specifies that he accepts the MWI document.

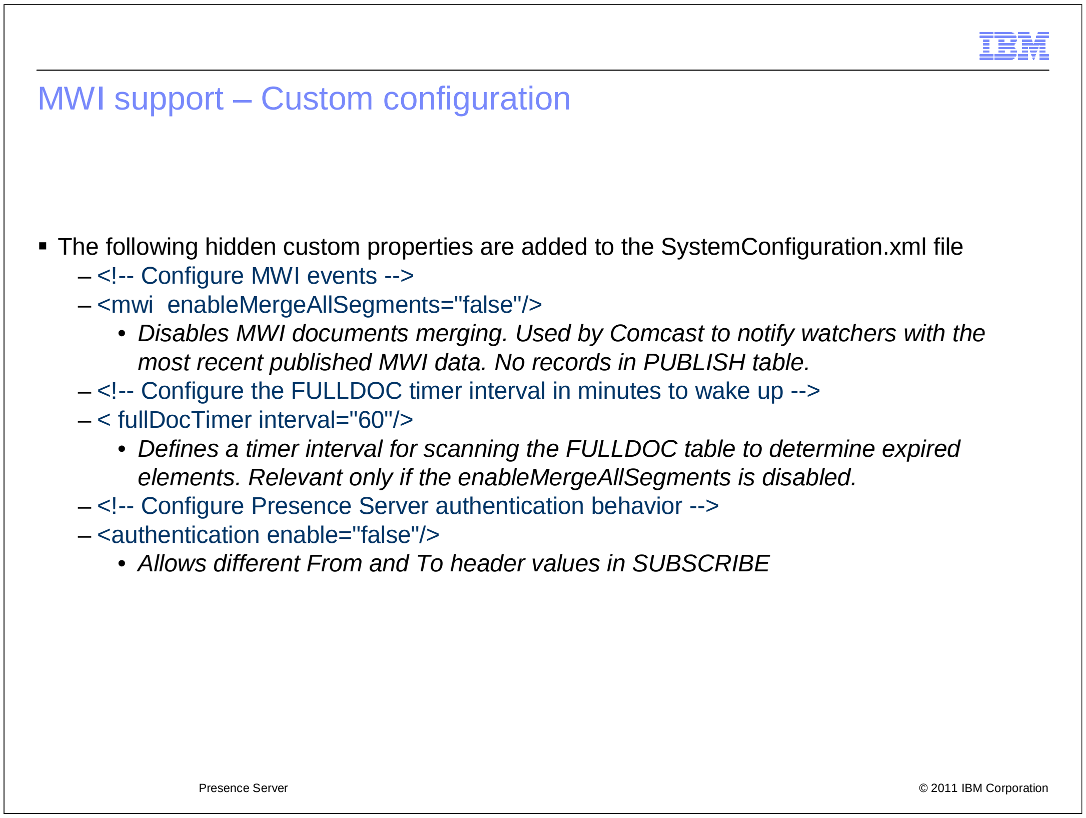
As for presence events, the Presence Server goes to XDMS component to obtain the group content. When the group content is obtained, Presence Server generates a multi-part related document for watcher, which includes message summary information for each resource of the list.

Throttling - The throttling mechanism is used to limit the rate of SIP notifications. A configuration can be applied to throttle SIP notification for message summary event packages.

Authorization rules are also supported for MWI. In general, the black and white lists and presence authorization rules are defined for any event type. So the rules are applicable to the MWI events exactly as for the presence events.

The REST API for MWI support functionality has been added in Presence Server 7.2 release.

There is no specification for partial publication and notification for MWI events. So this functionality is not there in Presence Server. There is no partial publication and no partial notification for MWI messages.



MWI support – Custom configuration

- The following hidden custom properties are added to the SystemConfiguration.xml file
  - <!-- Configure MWI events -->
  - <mwi enableMergeAllSegments="false"/>
    - *Disables MWI documents merging. Used by Comcast to notify watchers with the most recent published MWI data. No records in PUBLISH table.*
  - <!-- Configure the FULLDOC timer interval in minutes to wake up -->
  - < fullDocTimer interval="60"/>
    - *Defines a timer interval for scanning the FULLDOC table to determine expired elements. Relevant only if the enableMergeAllSegments is disabled.*
  - <!-- Configure Presence Server authentication behavior -->
  - <authentication enable="false"/>
    - *Allows different From and To header values in SUBSCRIBE*

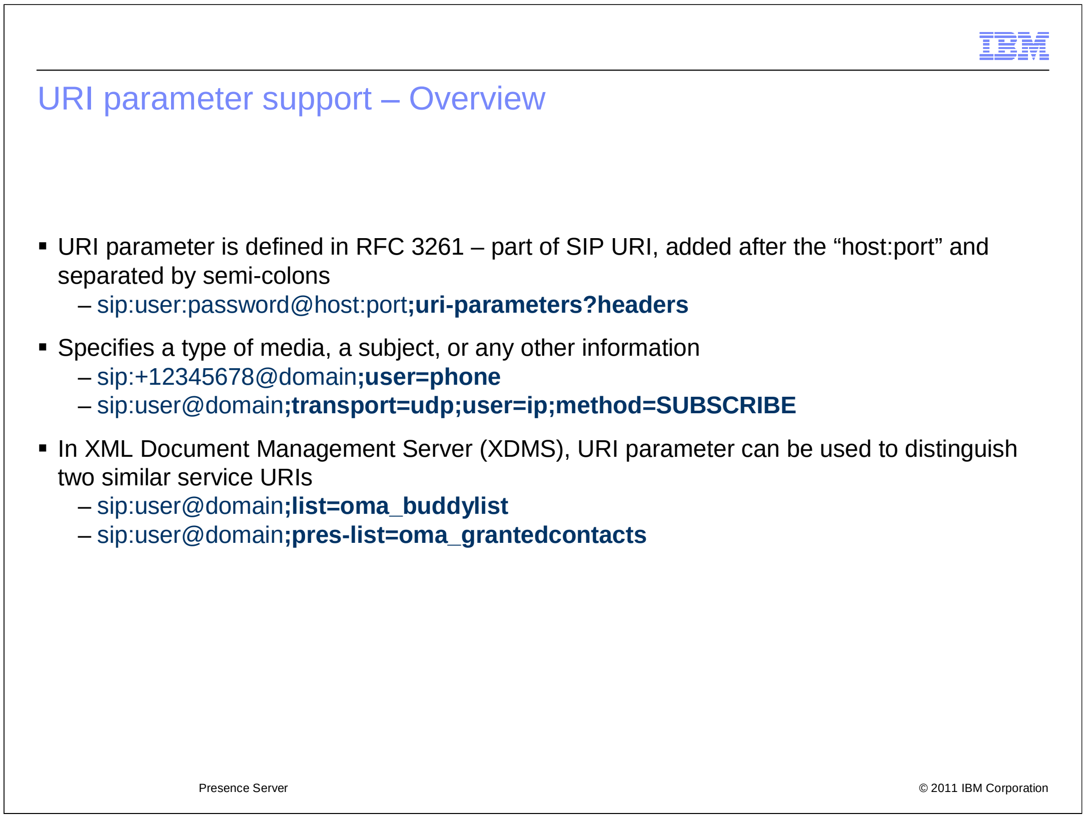
Presence Server

There are a few custom properties that are added in the SystemConfiguration.xml file.

The first property is the "enableMergeAllSegments". This allows to disable aggregation for MWI documents. If this is set to be “false”, there are no publish records in the database. In this case, Presence Server always uses the most recent published data, as the latest complete document to supply to the watcher.

The next property is the "fullDocTimer interval", which allows to configure a time interval for scanning the FULL DOC table. This property is required to determine the total number of expired documents. It is relevant only if the “enableMergeAllSegments” property is disabled. This is because the publish records are not available now, and so the only way to find the expired elements is to scan the FULL DOC table. It is a new functionality for Presence Server, but it is specific to the customer.

The last property is the "authentication" property. It is exposed to the customer to allow them to disable the authentication property so that they can use different From and To headers in subscribe messages.



URI parameter support – Overview

- URI parameter is defined in RFC 3261 – part of SIP URI, added after the “host:port” and separated by semi-colons
  - sip:user:password@host:port**;uri-parameters?headers**
- Specifies a type of media, a subject, or any other information
  - sip:+12345678@domain**;user=phone**
  - sip:user@domain**;transport=udp;user=ip;method=SUBSCRIBE**
- In XML Document Management Server (XDMS), URI parameter can be used to distinguish two similar service URIs
  - sip:user@domain**;list=oma_buddylist**
  - sip:user@domain**;pres-list=oma_grantedcontacts**

Presence Server

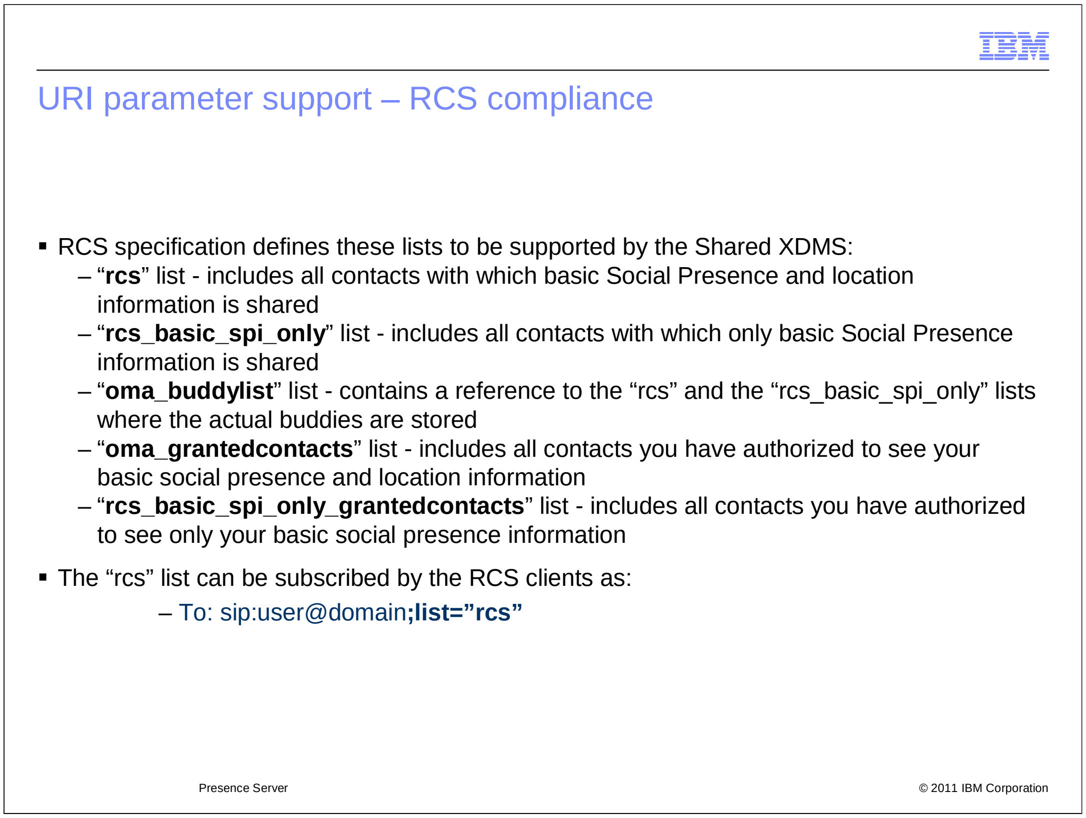
© 2011 IBM Corporation

URI parameter is defined in RFC 3261. It is a part of SIP URI, which is added after the “host:port” and separated by semi-colons.

An example of the URI format is given in this slide.

URI parameter makes it possible to specify a type of media or subject or any other information.

From the XDMS perspective, URI parameter can be used to distinguish between two similar service URIs.



URI parameter support – RCS compliance

- RCS specification defines these lists to be supported by the Shared XDMS:
  - “**rcs**” list - includes all contacts with which basic Social Presence and location information is shared
  - “**rcs_basic_spi_only**” list - includes all contacts with which only basic Social Presence information is shared
  - “**oma_buddylist**” list - contains a reference to the “rcs” and the “rcs_basic_spi_only” lists where the actual buddies are stored
  - “**oma_grantedcontacts**” list - includes all contacts you have authorized to see your basic social presence and location information
  - “**rcs_basic_spi_only_grantedcontacts**” list - includes all contacts you have authorized to see only your basic social presence information
- The “rcs” list can be subscribed by the RCS clients as:
  - To: sip:user@domain**;list=”rcs”**

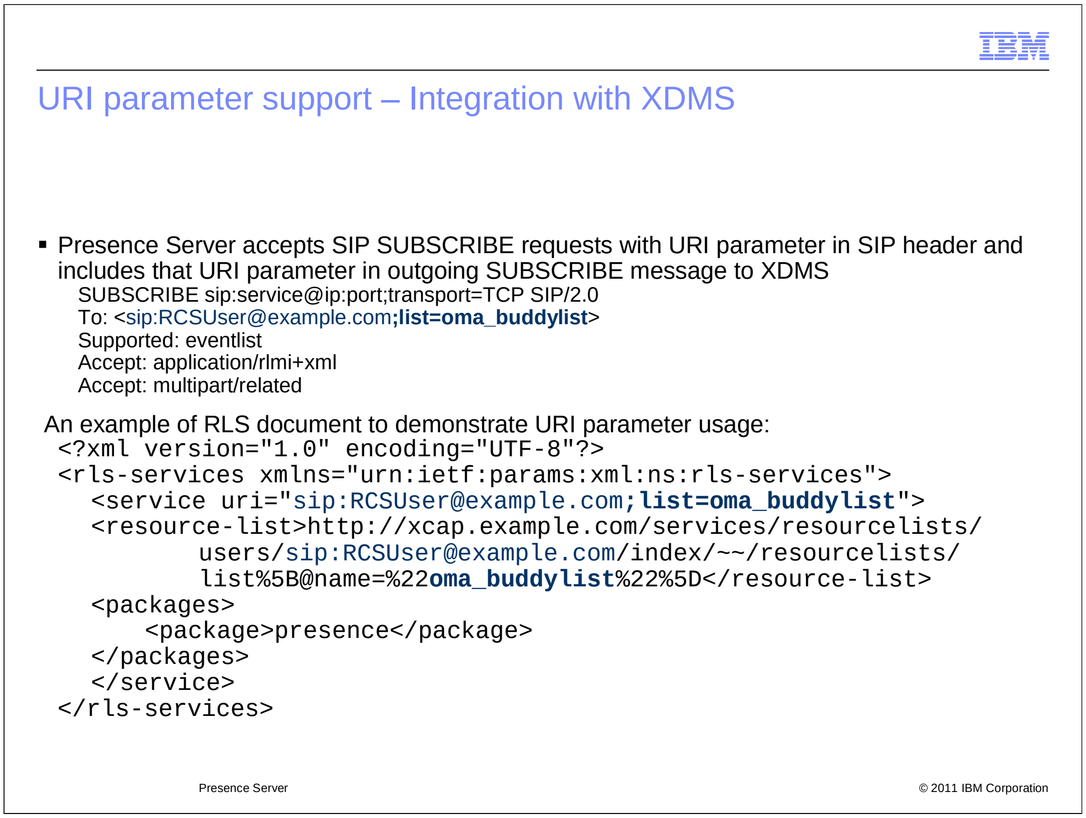
This slide deals with the RCS standard specification for the URI parameter.

The RCS specification defines some names to be supported by the Shared XDMS component. It is a list of values to be used in URI parameter. This slide provides a list of the names along with their definition.

For example:

“rcs” list - includes all contacts with whom basic Social Presence and location information is shared. RCS list can be used by RCS clients when subscribed to the Presence Server. The subscriber should provide the list name as the URI parameter in the To header as shown in the example at the bottom of the slide.

To process such requests, Presence Server interacts with the Shared XDMS component. Presence Server sends the exact URI received in the To header to XDMS.



URI parameter support – Integration with XDMS

- Presence Server accepts SIP SUBSCRIBE requests with URI parameter in SIP header and includes that URI parameter in outgoing SUBSCRIBE message to XDMS

```
SUBSCRIBE sip:service@ip:port;transport=TCP SIP/2.0
To: <sip:RCSUser@example.com;list=oma_buddylist>
Supported: eventlist
Accept: application/rlmi+xml
Accept: multipart/related
```

An example of RLS document to demonstrate URI parameter usage:

```
<?xml version="1.0" encoding="UTF-8"?>
<rls-services xmlns="urn:ietf:params:xml:ns:rls-services">
   <service uri="sip:RCSUser@example.com;list=oma_buddylist">
   <resource-list>http://xcap.example.com/services/resourcelists/
           users/sip:RCSUser@example.com/index/~~/resourcelists/
           list%5B@name=%22oma_buddylist%22%5D</resource-list>
   <packages>
       <package>presence</package>
   </packages>
   </service>
</rls-services>
```

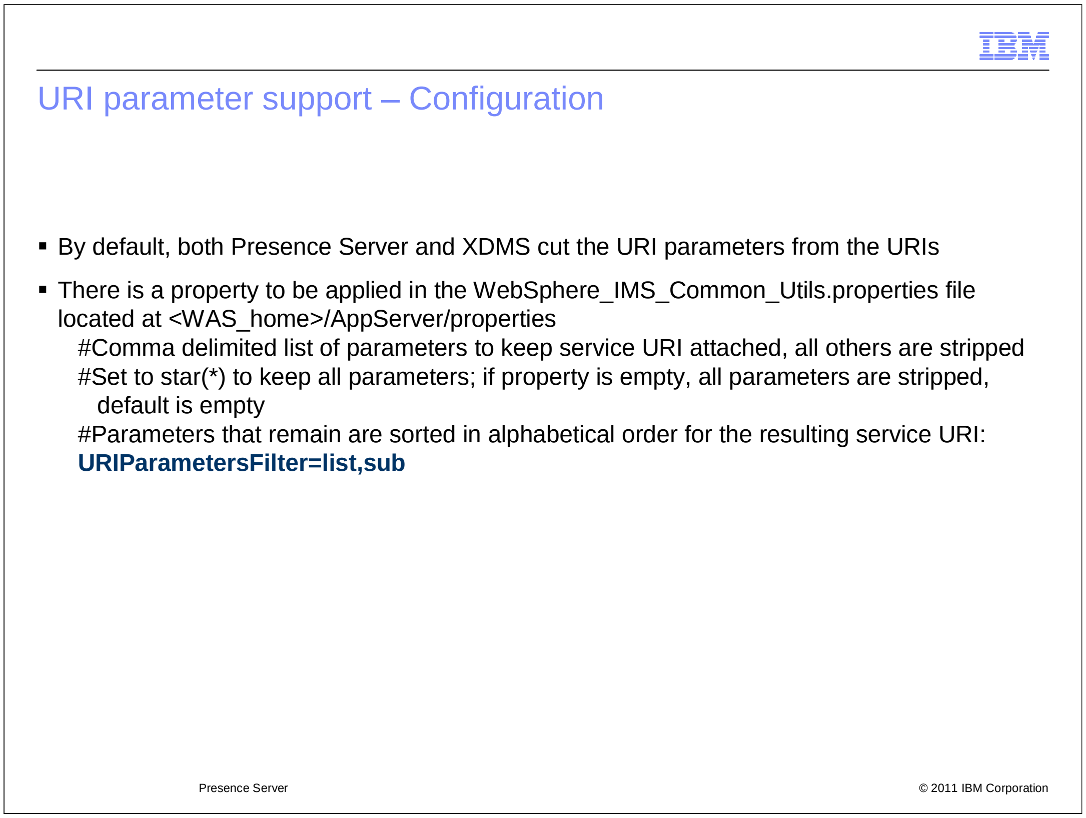
This slide discusses the URI Parameter support integration with XDMS.

Presence Server accepts SIP SUBSCRIBE requests with the URI parameter in To header.

In previous Presence Server releases, the URI was stripped from the URI parameter. But in Presence Server 7.2, the URI is kept unchanged and is used in the outgoing SUBSCRIBE message to XDMS.

XDMS checks the document and compares the received URI with the service URI from the RLS document.

If there is a match, then the XDMS component supplies the resource list of the document to the Presence Server.



URI parameter support – Configuration

- By default, both Presence Server and XDMS cut the URI parameters from the URIs
- There is a property to be applied in the WebSphere_IMS_Common_Utils.properties file located at <WAS_home>/AppServer/properties

```
#Comma delimited list of parameters to keep service URI attached, all others are stripped
#Set to star(*) to keep all parameters; if property is empty, all parameters are stripped,
   default is empty
#Parameters that remain are sorted in alphabetical order for the resulting service URI:
URIParametersFilter=list,sub
```

Presence Server

This slide deals with the URI parameter configuration.

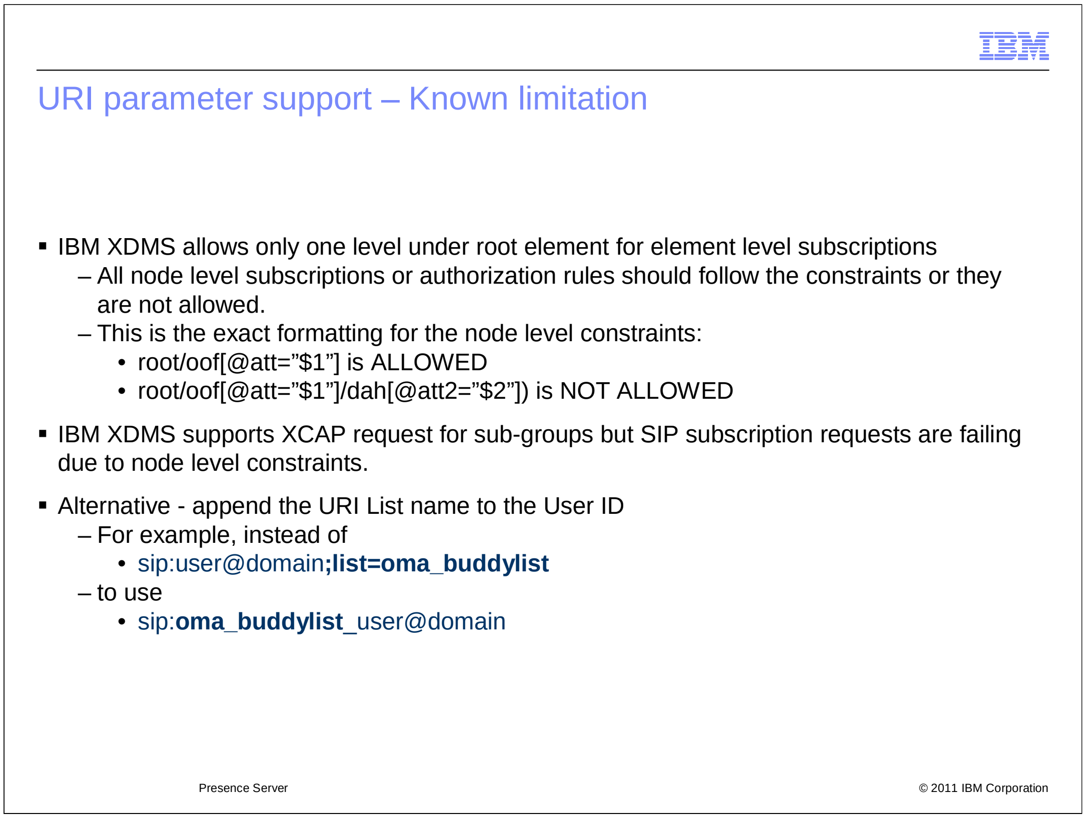
By default, both Presence Server and XDMS cut the URI parameters from the URIs.

To keep the URI unchanged, a property needs to be applied in the WebSphere_IMS_Common_Utils.properties file, which is located in the **<*WAS_home*>**/**AppServer**/**properties** folder.

The URI parameter filter is empty by default.

To keep specific URI as a part of the provided SIP URI, the name of the URI parameter should be defined in this attribute.

After applying the property, the application server and the node should be restarted.



URI parameter support – Known limitation

- IBM XDMS allows only one level under root element for element level subscriptions
  - All node level subscriptions or authorization rules should follow the constraints or they are not allowed.
  - This is the exact formatting for the node level constraints:
    - root/oof[@att="$1"] is ALLOWED
    - root/oof[@att="$1"]/dah[@att2="$2"]) is NOT ALLOWED
- IBM XDMS supports XCAP request for sub-groups but SIP subscription requests are failing due to node level constraints.
- Alternative - append the URI List name to the User ID
  - For example, instead of
    - sip:user@domain**;list=oma_buddylist**
  - to use
    - sip:**oma_buddylist**_user@domain

Presence Server

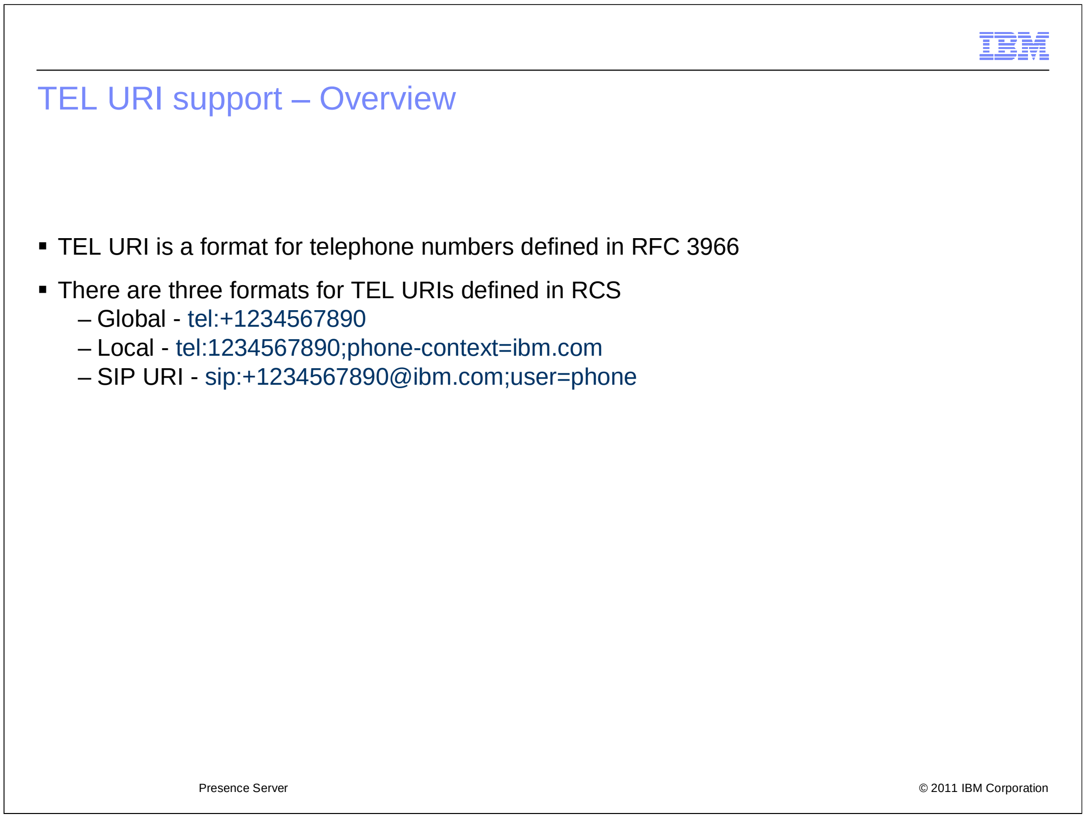
There is a known limitation in the integration with IBM XDMS component.

IBM XDMS allows only one level of subscriptions. The exact formatting for the node level constraints is given here.

XDMS supports XCAP requests; but due to the node level constraints limitation, XDMS cannot support SIP subscription on URI parameter.

Another approach is to append the URI List name instead of the URI parameter and to put it as part of the User ID.

Example of two URIs are given here. The first URI is defined by RCS and the second is the alternative.



TEL URI support – Overview

- TEL URI is a format for telephone numbers defined in RFC 3966
- There are three formats for TEL URIs defined in RCS
  - Global - tel:+1234567890
  - Local - tel:1234567890;phone-context=ibm.com
  - SIP URI - sip:+1234567890@ibm.com;user=phone

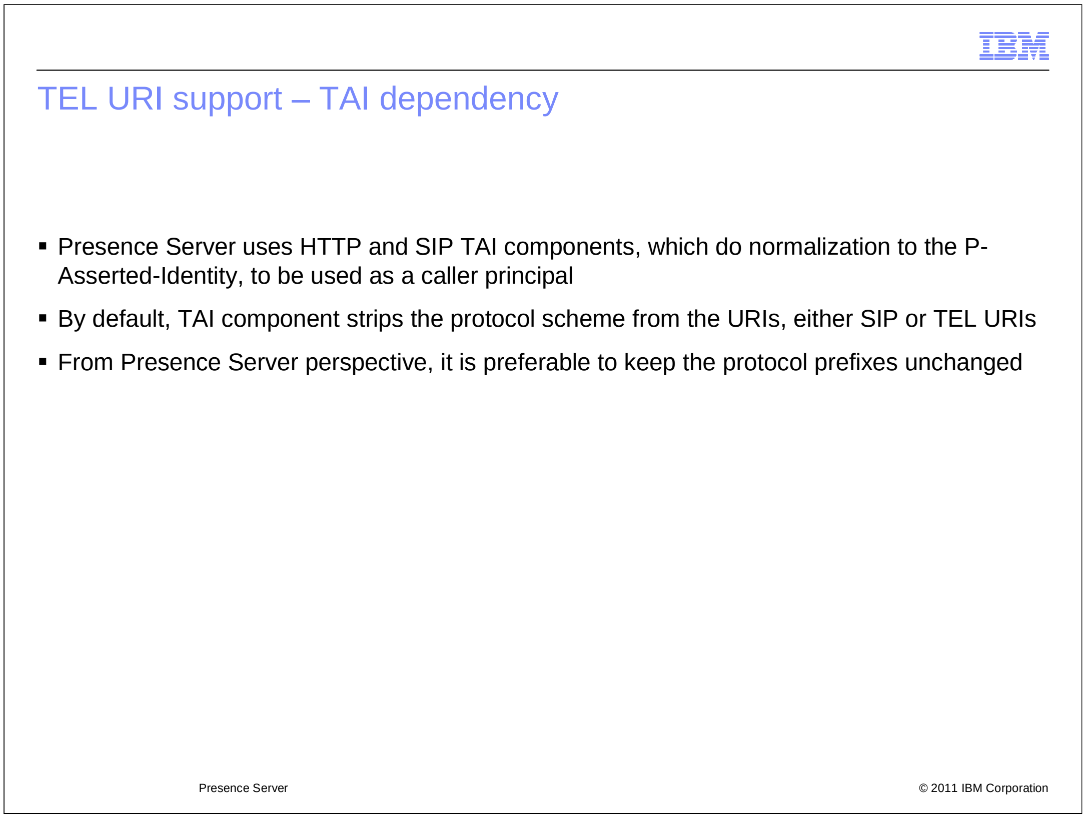
Presence Server

TEL URI is another format of URI that is used for telephone numbers. The framework is defined in RFC 3966.

There are three formats that can be used to define telephone numbers.

Users can work with Global tel uri, Local tel uri, or work with SIP URI using URI parameter showing that the user is over telephone.

Presence Server 7.2 supports all the three formats. Also, support is given for telephone numbers in any SIP header where URI is allowed, such as From, To, Contact, or Request URI.



TEL URI support – TAI dependency

- Presence Server uses HTTP and SIP TAI components, which do normalization to the P-Asserted-Identity, to be used as a caller principal
- By default, TAI component strips the protocol scheme from the URIs, either SIP or TEL URIs
- From Presence Server perspective, it is preferable to keep the protocol prefixes unchanged

Presence Server

Support is given for two different URIs - SIP and TEL..

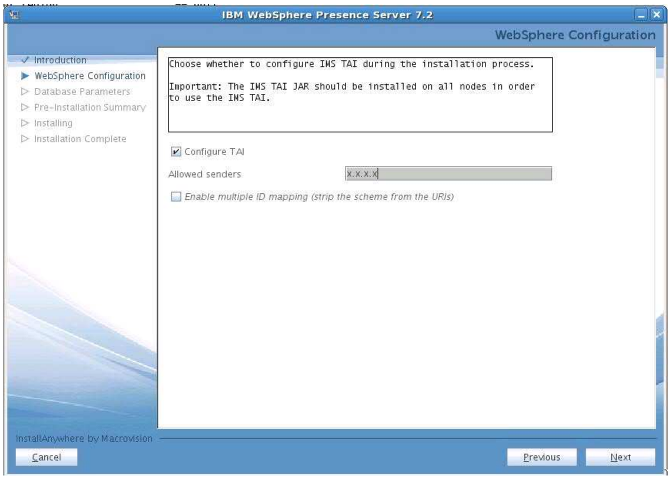
As the TAI (Trust Association Interceptor) component and Presence Sever strips the protocol schema from the URIs, it creates a conflict in Presence Server to identify the user.

So it is preferable to keep the URI unchanged.

Additions to the Presence Server configuration enable a property to define a TAI behavior for URI schema. This is shown in the next slide.

TEL URI support – Configuration: TAI property (1 of 2)

- TAI property <enableMultipleIDMapping> changes the TAI behavior
- Presence Server 7.2 installer allows changing the default value of the property during installation process


IBM WebSphere Presence Server 7.2
WebSphere Configuration
✓ Introduction
▶ WebSphere Configuration
▷ Database Parameters
▷ Pre-Installation Summary
▷ Installing
▷ Installation Complete
InstallAnywhere by Macrovision
Cancel
Previous
Next

```
Choose whether to configure IMS TAI during the installation process.

Important: The IMS TAI JAR should be installed on all nodes in order
to use the IMS TAI.
```

☑ Configure TAI

Allowed senders: x.x.x.x

□ Enable multiple ID mapping (strip the scheme from the URIs)

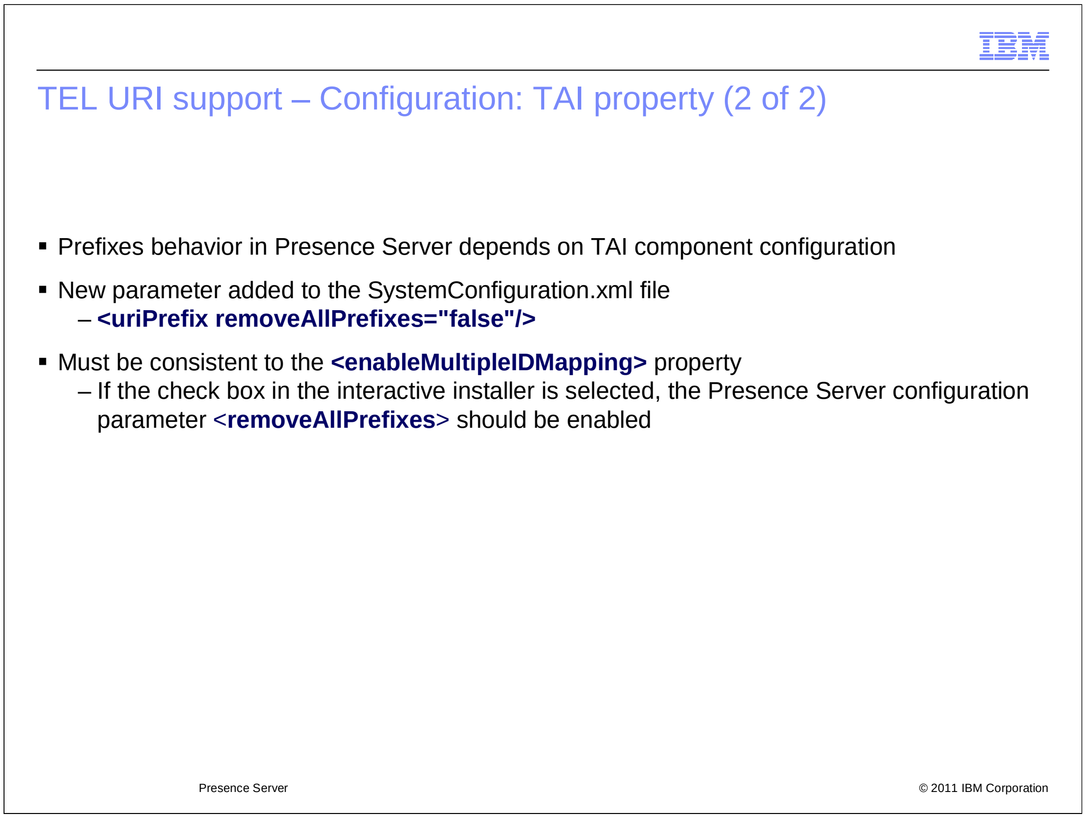
Presence Server

This slide gives the TAI property configuration.

There is a new TAI property "enableMultipleIDMapping" that is used to define the TAI behavior. The Presence Server installer helps to change the value of this property.

In the installation process, a check box is added to change the default value of the TAI component.

Do not select this check box if you want to keep it compatible with the Presence Server default behavior.



TEL URI support – Configuration: TAI property (2 of 2)

- Prefixes behavior in Presence Server depends on TAI component configuration
- New parameter added to the SystemConfiguration.xml file
  - **<uriPrefix removeAllPrefixes="false"/>**
- Must be consistent to the **<enableMultipleIDMapping>** property
  - If the check box in the interactive installer is selected, the Presence Server configuration parameter <**removeAllPrefixes**> should be enabled

Presence Server

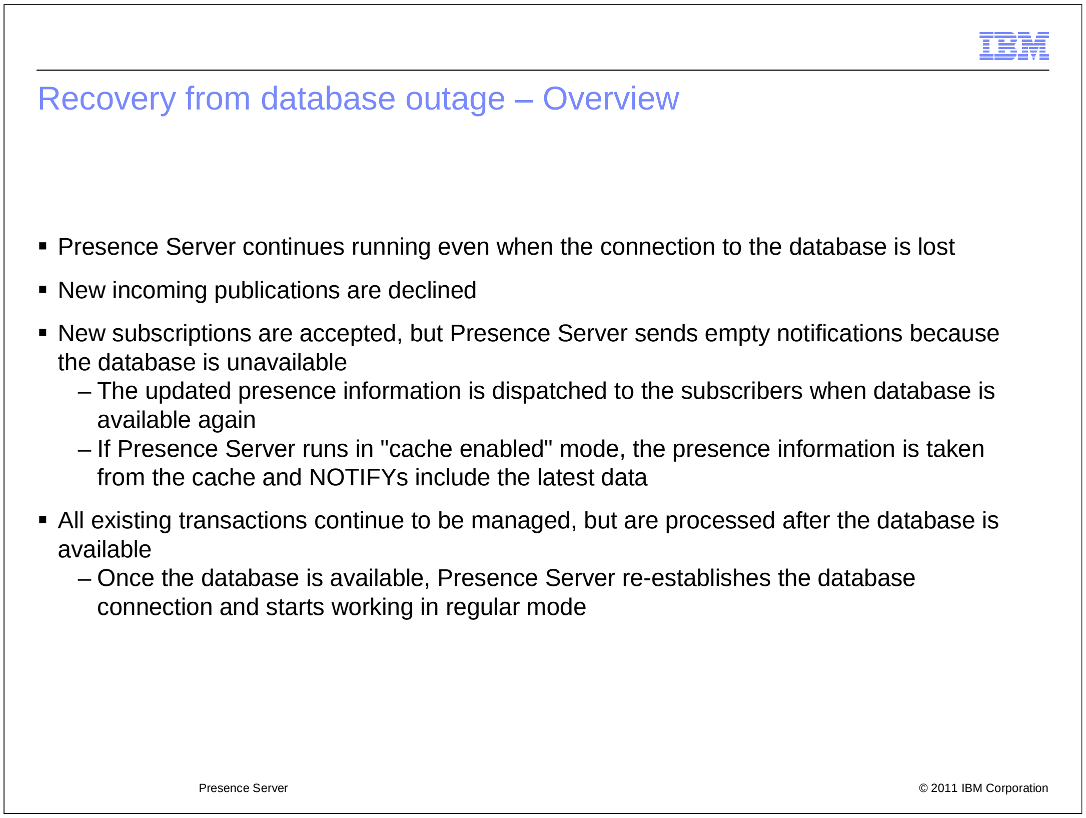
© 2011 IBM Corporation

There is also a new configuration attribute added to the SystemConfiguration.xml of Presence Server.

Presence Server attribute is a URI prefix called "removeAllPrefixes". The default value is “false” as all the prefixes are kept by default.

It is important to keep the configuration parameter consistent with the TAI property applied during Presence Server installation.

If the check box in the interactive installer is selected, the Presence Server configuration parameter <removeAllPrefixes> should be enabled.



Recovery from database outage – Overview

- Presence Server continues running even when the connection to the database is lost
- New incoming publications are declined
- New subscriptions are accepted, but Presence Server sends empty notifications because the database is unavailable
  - The updated presence information is dispatched to the subscribers when database is available again
  - If Presence Server runs in "cache enabled" mode, the presence information is taken from the cache and NOTIFYs include the latest data
- All existing transactions continue to be managed, but are processed after the database is available
  - Once the database is available, Presence Server re-establishes the database connection and starts working in regular mode

Presence Server

© 2011 IBM Corporation

When the connection to the database is lost, Presence Server 7.2 continues running in idle mode. It receives new incoming requests, but all the new publications are declined. This is because the database access required to keep the published data is unavailable.

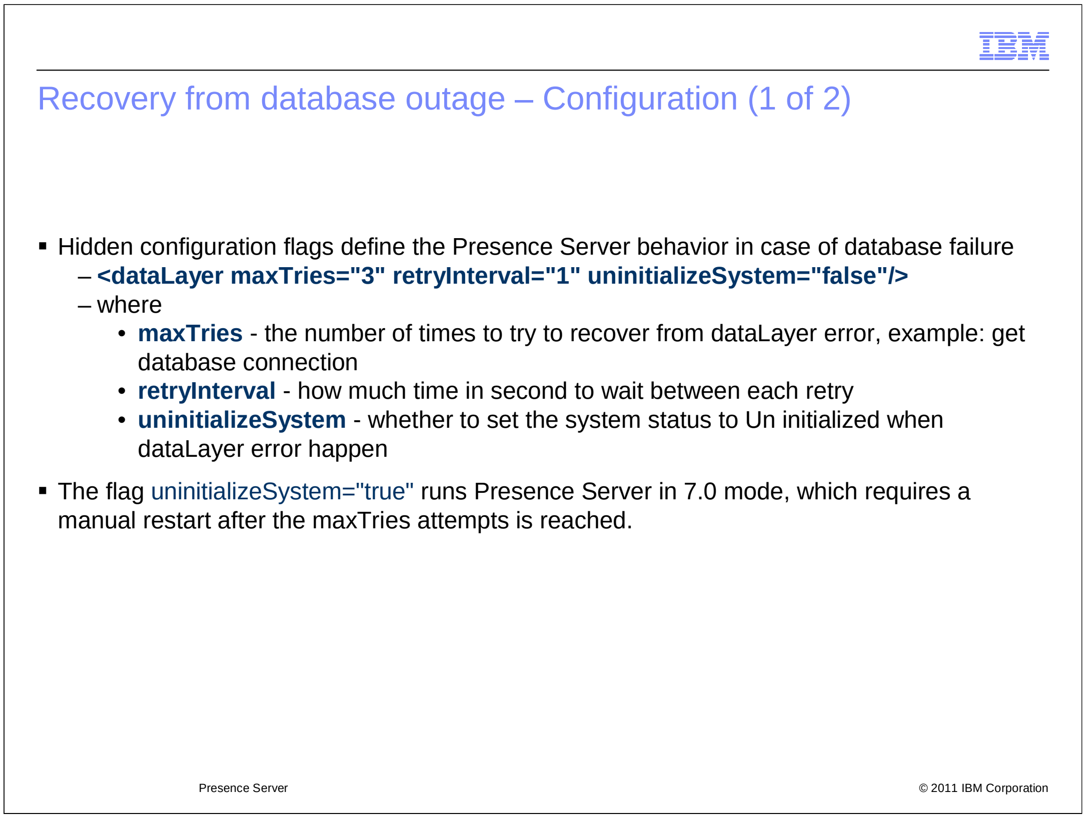
New subscriptions are accepted, but again because the database is not available, Presence Server sends empty notifications to the watcher.

The updated presence information is dispatched later to the subscribers when the database is available again.

If Presence Server runs in "cache enabled" mode, the presence information is taken from the cache and in this case NOTIFYs include the latest data from the cache.

Presence Server manages the existing transactions; but to process operations like remove expired documents, Presence Server waits for the database connection.

Once the database is available, Presence Server re-establishes the database connection and starts working in the regular mode.



Recovery from database outage – Configuration (1 of 2)

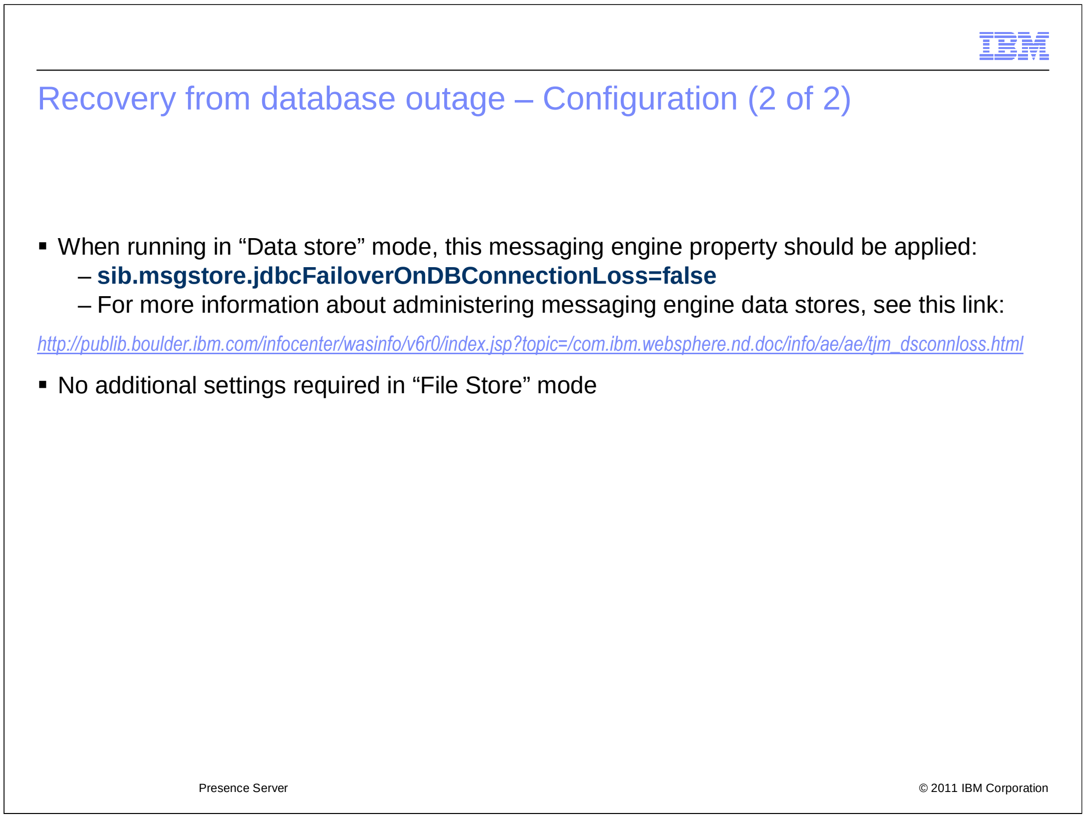
- Hidden configuration flags define the Presence Server behavior in case of database failure
  - **<dataLayer maxTries="3" retryInterval="1" uninitializeSystem="false"/>**
  - where
    - **maxTries** - the number of times to try to recover from dataLayer error, example: get database connection
    - **retryInterval** - how much time in second to wait between each retry
    - **uninitializeSystem** - whether to set the system status to Un initialized when dataLayer error happen
- The flag uninitializeSystem="true" runs Presence Server in 7.0 mode, which requires a manual restart after the maxTries attempts is reached.

This slide shows several configuration attributes added to the SystemConfiguration.xml file to define Presence Server behavior in case of database failure.

The new parameter "uninitializeSystem" was enabled in Presence Server 7.0. Because of this, Presence Server stops working when the database connection is lost and Presence Server requires a manual restart later.

In Presence Server 7.2, this flag is disabled to allow Presence Server to stay running and it continues to try to reconnect to the database.

According to the configuration, you can put any value in the **maxTries** and **retryInterval** parameters. All possible values are supported and this helps to keep Presence Server running as long as required.



Recovery from database outage – Configuration (2 of 2)

- When running in “Data store” mode, this messaging engine property should be applied:
  - **sib.msgstore.jdbcFailoverOnDBConnectionLoss=false**
  - For more information about administering messaging engine data stores, see this link:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tjm_dsconnloss.html

- No additional settings required in “File Store” mode

Presence Server

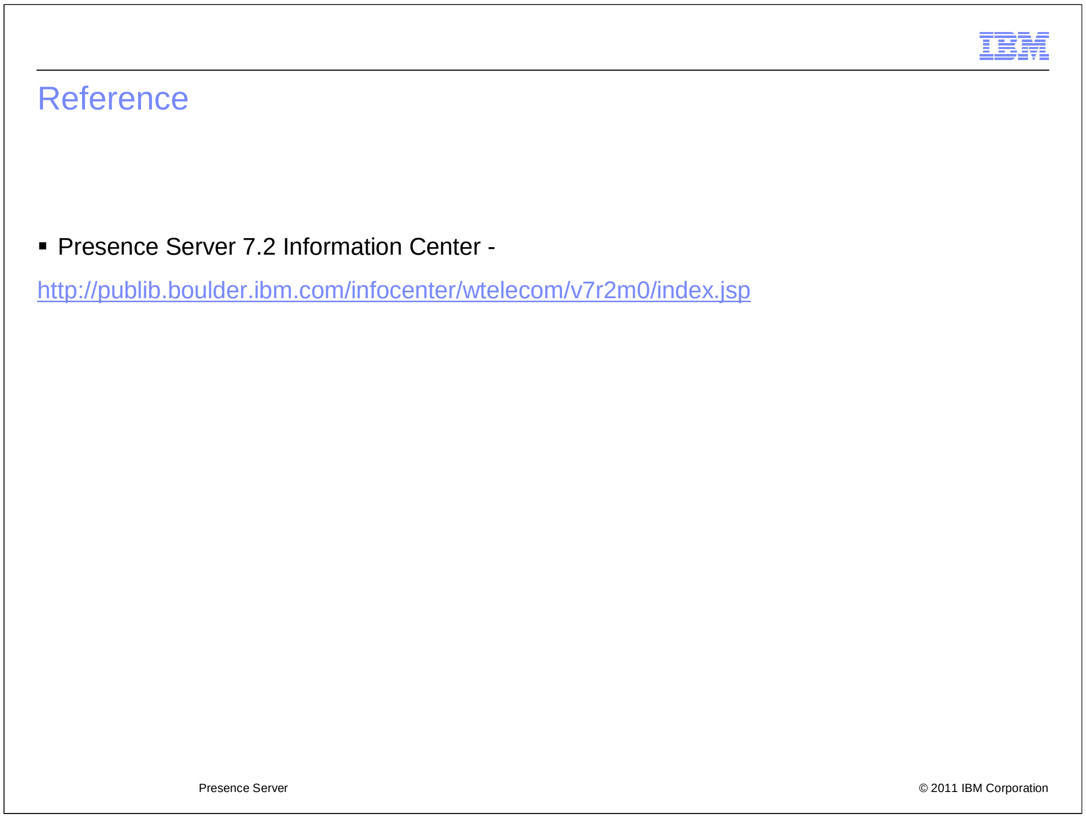
© 2011 IBM Corporation

While running Presence Server in “Data store” mode, the message engine might try to recover by itself and the application server might be automatically restarted.

To avoid such situations, a messaging engine property should be applied.

For more details on this property and on message engine administration, see the information center (http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/tjm_dsconnloss.html).

If Presence Server runs in “File Store” mode, no additional setting is required.



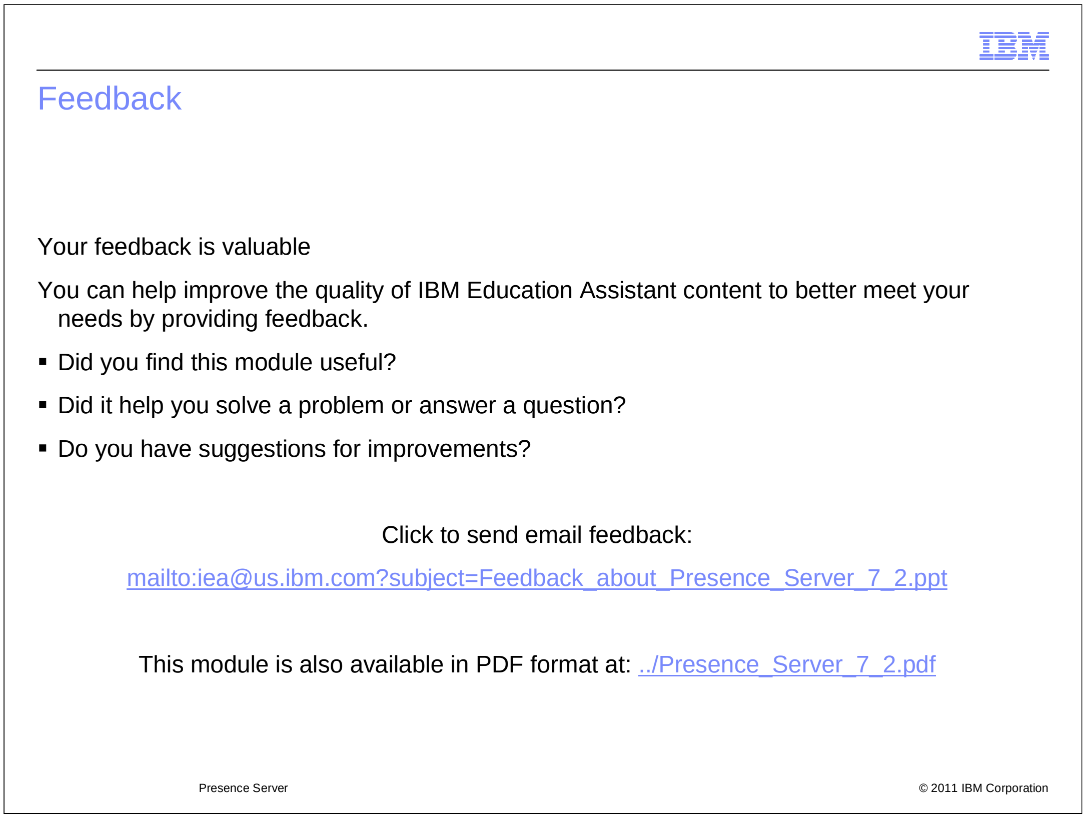
Reference

- Presence Server 7.2 Information Center -

http://publib.boulder.ibm.com/infocenter/wtelecom/v7r2m0/index.jsp

Presence Server

For more information on Presence Server 7.2, see the 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?

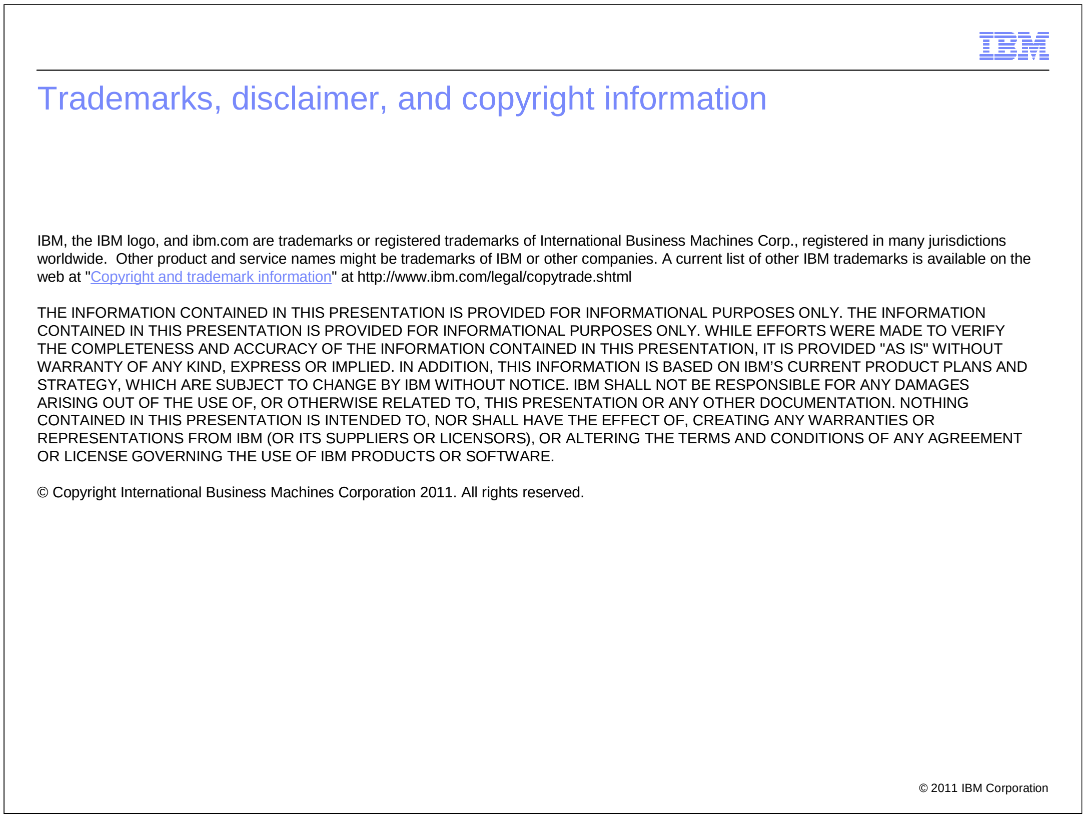
Click to send email feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_Presence_Server_7_2.ppt

This module is also available in PDF format at: ../Presence_Server_7_2.pdf

Presence Server

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



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, and ibm.com 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. 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.