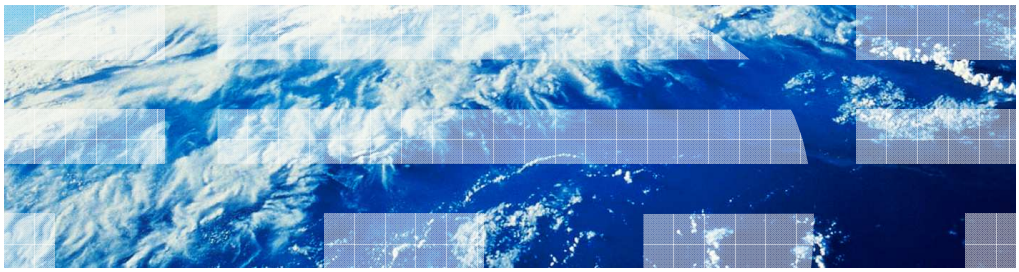
IBM Communication Service Enablers V7.2

Terminal location over Mobile Location Protocol



This presentation gives an introduction to Terminal Location over Mobile Location Protocol (MLP).

The agenda includes an introduction to Terminal Location over Mobile Location Protocol, which is used to get the location of a mobile station.

Also discussed are the new features added to the existing TL over MLP service implementation.

Other topics are installation and configuration for TL over MLP, how to troubleshoot, and reference links.

Introduction (1 of 2)

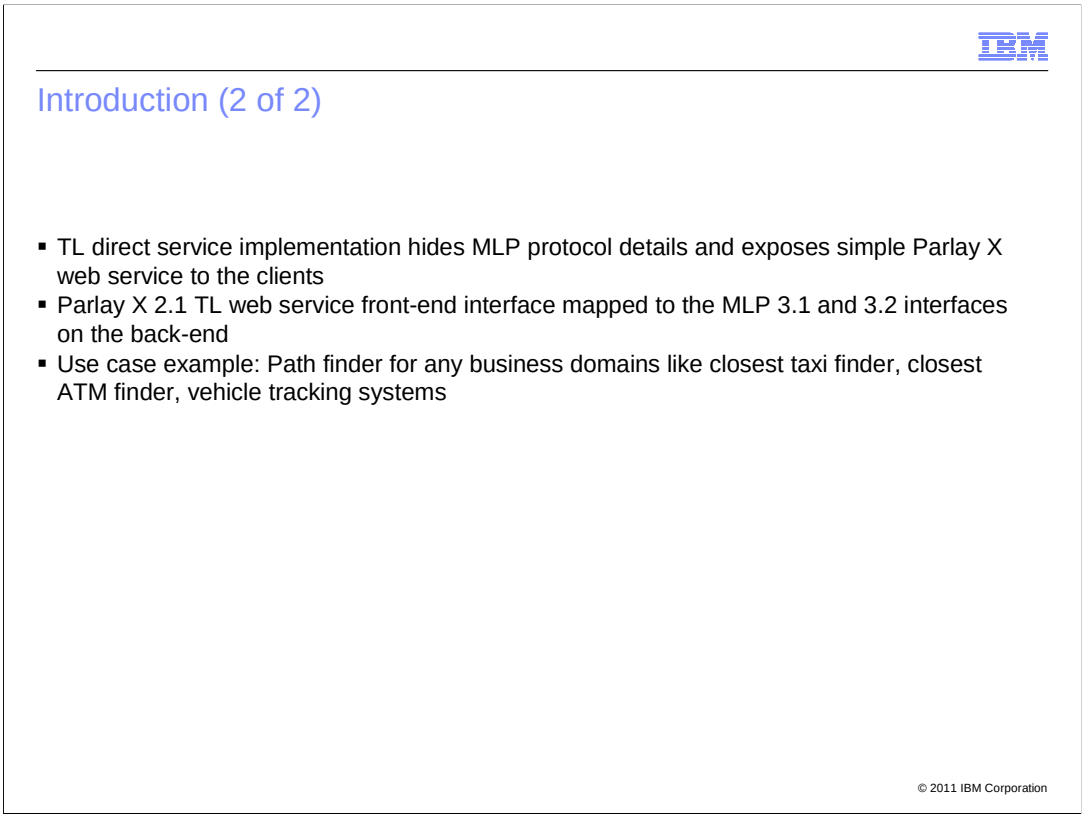
- TL Parlay X 2.1 operations are broadly classified into two categories:
  - Location-based synchronized operations
  - Notification-based triggered operations
- Parlay X 2.1 web service implementation actions are:
  - Request for the location of a terminal
  - Request for the location of a group of terminals
  - Retrieves the distance of a terminal from a particular point
  - Notification of a change in the location of a terminal
  - Notification of terminal location on a periodic basis
  - Location is expressed through a latitude, longitude and accuracy

TL over MLP supports two types of operations:

- Location-based synchronized operations
- Notification-based triggered operations.

The first type is used to get the location of a target address and the second type is used to receive notification when certain criteria are met.

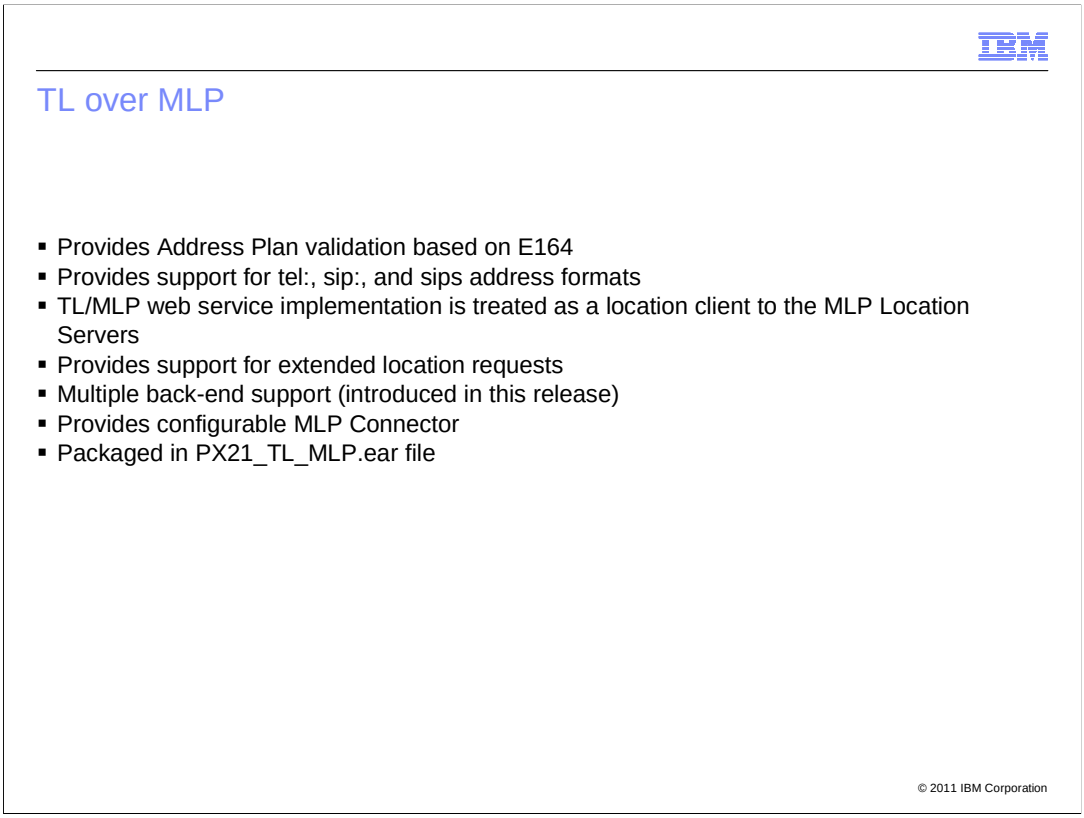
Some of the actions the web service takes to serve these two types of operations are mentioned in this slide.



Terminal Location is a direct service implementation that hides MLP protocol details and exposes simple Parlay X web service to the clients.

The front-end interface is mapped to the MLP 3.1 and 3.2 interfaces on the back-end.

A typical use case is to find the closest taxi finder, the closest ATM finder, or to be used in vehicle tracking systems.



Some of the other features of TL over MLP are:

- Address Plan validation based on E164

- Support for tel:, sip: and sips address formats

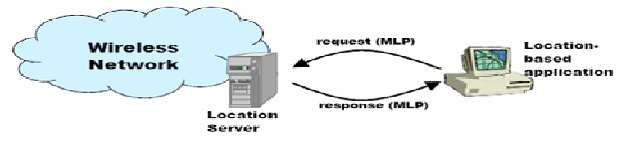
- The web service implementation is treated as a location client that sends request to the MLP Location Servers and processes the response from the server

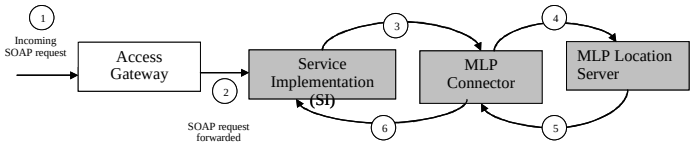
- Support for extended location requests - Typical features are accuracy, altitude, longitude, and latitude. Using extended location, different parameters can be specified for additional quality of service, such as maximum age and response time.

- Multiple back-end support is a new feature introduced in this release. This is discussed in detail in the next few slides.

- Provides configurable MLP Connector with which back-ends and aliases can be configured and packaged in the PX21_TL_MLP.ear file.

High-level architecture and message flow





This slide gives the high-level architecture and message flow in TL over MLP.

The diagram at the top indicates that MLP is an interface between the location client and the location server.

The location client sends the request to the location server and a response is received from the location server.

The message flow is shown in the diagram at the bottom.

The incoming SOAP request goes to the Access Gateway and the SOAP request is forwarded to the service implementation. The service implementation sends it to the MLP Connector. Actually the MLP Connector is a part of the Service Implementation, but it has been separated in the diagram for clarity.

The MLP Connector then forwards the request to the MLP Location Server and gets the response.

Interfaces and operations (1 of 2)

TerminalLocation		
getLocation	Retrieves the location of a single terminal	Synchronous
getLocationForGroup	Retrieves the location of group of terminals	Synchronous
getTerminalDistance	Determines the distance of the terminal from a given location	Synchronous
TerminalLocationNotificationManager		
startGeographicNotification	Registers a 3rd party web service application to receive a notification when a terminal enters or exits an area specified by a longitude, latitude, and radius	Synchronous
startPeriodicNotification	Registers a 3rd party web service application to receive a notification at some given interval	Synchronous
endNotification	De-registers a 3rd party web service application from receiving notifications	Synchronous

The first three operations are location-based operations.

getLocation - gets you the location of a single terminal

getLocationForGroup - is for multiple terminals

getTerminalDistance - essentially uses getLocation, but the information is converted into distance using the mathematical formula to calculate the distance between two points

The next set of operations are notification-based triggered operations.

startGeorgraphicNotification - is received when a terminal enters or exists a certain area specified by a longitude, latitude, and a radius

startPeriodicNotification - is received when you have requested the location information of a client for every 30 seconds or a specified interval

endNotificaiton - de-registers a third-party web service application from receiving notifications for a particular target

Interfaces and operations (2 of 2)

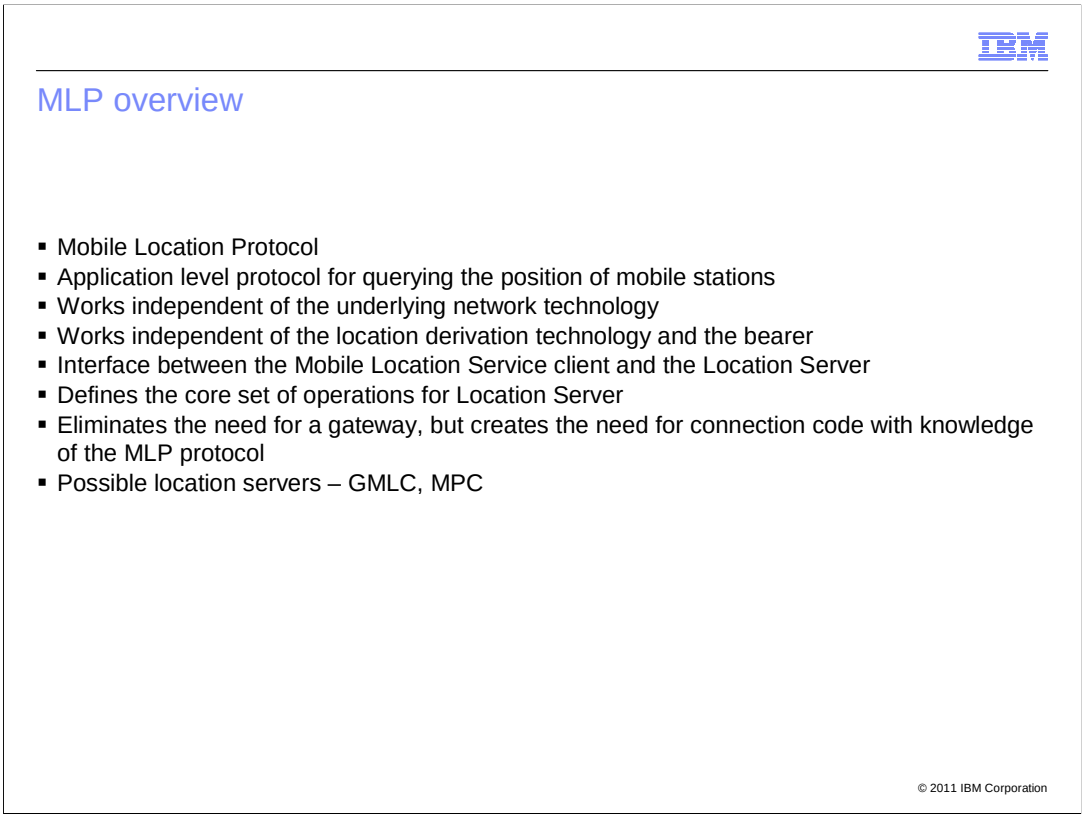
TerminalLocationNotification		
locationNotification	•Invoked to notify the application of the new location of a terminal •Used when startGeographical & startPeriodic notifications are opted	Asynchronous
locationError	•Invoked to notify the application the notification for a terminal or the whole notification is being cancelled by the web service •Used in startGeographical or startPeriodic notification operations	Asynchronous
locationEnd	•Invoked to notify the application when the notifications are completed for a particular correlator •Invoked when the duration or count for notifications have completed •Will not be delivered during end notification due to error or the deliberate ending of the notifications	Asynchronous

The rest of the notification-based triggered operations are given here:

locationNofication - is used to notify a client whenever the startGeorgraphical and startPeriodic notifications are opted

locationError - is used to notify the application for a terminal or the whole notification is being cancelled; it is used in startGeograhical or startPeriodic notification operations

locationEnd - is used when notifications are completed for a particular correlator or the duration or count has completed; it is not used when endNotification operation is used



This slide provides an overview of Mobile Location Protocol (MLP).

MLP is an application-level protocol for querying the position of mobile stations.

It works independent of the underlying network technology, the location derivation technology, and the bearer.

It also interfaces between the Mobile Location Service client and the Location Server to find the core set of operations for Location Server.

MLP eliminates the need for a gateway but creates the need for connection code with knowledge of the MLP protocol.

Possible location servers are GMLC (Gateway Mobile Location Protocol) and MPC (Mobile Positioning Center).

Supported MLP versions

- Common for all the operations
- In this service implementation (TWSS 7.2) based on both MLP 3.1 and MLP 3.2, the Parlay X Address URI is converted to MSID (Mobile Station Identifier) of type MSISDN
- Location-based synchronized operations
- While sending a request in MLP 3.2, a unique transaction ID can be sent and it is returned along with the response
- Notification-based triggered operations
- Only in MLP 3.2 are there provisions to send a triggered location reporting request with the criteria to ENTER and EXIT a particular area; this is why startGeographicalNotification is supported only in 3.2

The supported MLP versions are 3.1 and 3.2, and this slide gives a comparison between 3.1 and 3.2.

One feature is common to all the operations – The Parlay X Address URI is converted to MSID (Mobile Station Identifier) of type MSISDN.

Location-based synchronized operations – In MLP 3.2, while sending a request, a unique transaction ID can be sent and it is returned along with the response.

Notification-based triggered operations – Only in MLP 3.2 are there provisions to send a triggered location reporting request with the ENTER and EXIT criteria. This is why startGeographicalNotification is supported only in 3.2 and not in 3.1.

Supported MLP shapes

- Out of the MLP shapes, 'CircularArea' is the closest match to 'Parlay X LocationInfo'
- Support for shapes other than 'CircularArea' requires mathematical conversion
- Mathematical conversion might result in loss of precision
- The different shapes are:
  - Point
  - Circular area
  - Circular arc area
  - Elliptical area
  - LineString
  - LinearRing
  - Box
  - Polygon
  - MultiPoint

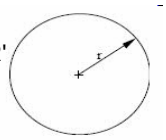
This slide gives the supported MLP shapes.

A number of MLP shapes are supported. But ‘CircularArea’ is the closest match to ‘Parlay X LocationInfo’. This is because 'CircularArea' has these parameters - the center (latitude and longitude) and the radius (which is the accuracy).

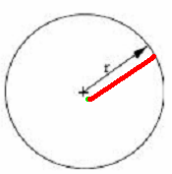
All other shapes can be converted to 'CircularArea' with the help of mathematical conversion. But this might result in the loss of precision.

Shapes – Circular area

- **Description:**
  - A set of points in the geography with an uncertainty radius of distance 'R'



- **Example value:**
- <CircularArea srsName="www.epsg.org#4326" gid="some_thing">
- <coord>
- <X>X1 </X>
- <Y>Y1</Y>
- </coord>
- <radius>R</radius>
- </CircularArea>



- **MLP to Parlay X Mapping:**
  - Circular area is mapped to a LocationInfo object where 'X1, Y1' is the center and 'R' is the radius

This slide has an example for Circular Area. The center is marked as the longitude and latitude. That is, a set of points in the geography with an uncertainty radius of distance “R”.

Multiple back-end support

- Back-end is a GMLC server
- Alias defines a mapping between the back-end server details and a range of target addresses configured as TL/MLP Service Policies
- There are two back-ends corresponding to each alias namely primary and secondary; secondary is used in a fail over scenario
- For every alias, at least one of the back-ends need to be configured with a valid value
- Two new service policies:
  - service.config.target.TLMLPAliases
  - service.config.target.TLMLPRanges

This slide discusses the multiple back-end support that has been added as a new feature.

A back-end is a GMLC server. And multiple back-end support is achieved by using aliases.

Each alias defines a mapping between the back-end server details and a range of target addresses. The target addresses are configured as TL over MLP Service Policies.

Back-end details of the aliases can be configured in the administration console.

The configuration is discussed in detail later.

There are two back-ends corresponding to each alias - namely primary and secondary. Secondary is used in a fail over scenario.

For every alias, at least one of the back-ends need to be configured with a valid value.

The two new service policies introduced to achieve multiple back-end support are:
- service.config.target.TLMLPAliases
- service.config.target.TLMLPRanges

Service policies for multiple back-end support

Policy	Operation	Parameter	Description
service.config.target.TLMLPAliases	All	Alias names	Semicolon-separated list of alias names
service.config.target.TLMLPRanges	All	Alias Range	Semicolon-separated list of corresponding range of numbers for each alias

Example:

<policy attribute="service.config.target.TLMLPAliases" value="north;south"/>
<policy attribute="service.config.target.TLMLPRanges" value="11111222330-11111222550;11111222660-11111222770"/>

This slide contains an example for the two new service policies introduced for multiple back-end support.

- TLMLPAliases policy - is a semicolon-separated list of alias names

- TLMLPRanges policy - is a semicolon-separated list of range of addresses corresponding to each alias

Installation and configuration (1 of 2)

- PX21_TL_MLP.ear
  - Package contains the Terminal Location enterprise application
- TWSS 7.2 administrative console
  - Contains the TWSS administration console installed on WebSphere® Application Server administrative console
- TWSS 7.2 Service Platform services
  - Admission control
  - Fault and alarm
  - Traffic shaping
  - Usage record
  - PxNotify
  - Network resources
  - Notification management

This slide discusses the installation and configuration for TL over MLP.

The PX21_TL_MLP.ear file contains the Terminal Location enterprise application.

TWSS 7.2 administration console and TWSS 7.2 Service Platform services are used by TL over MLP.

First Steps can be used for installing TL over MLP.

Installation and configuration (2 of 2)

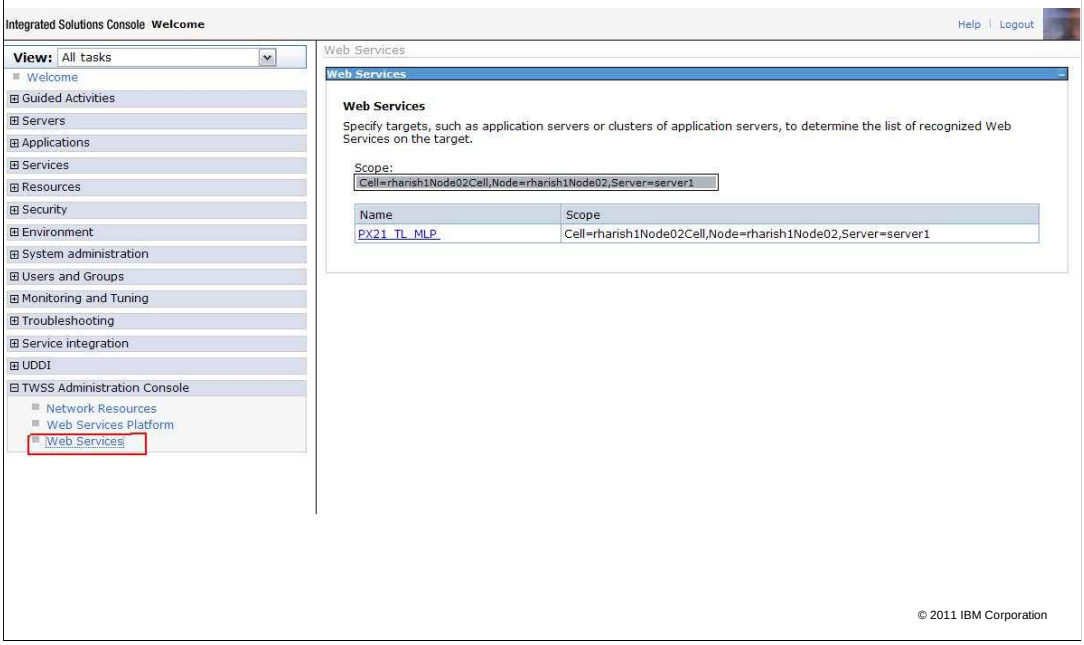
- Data source
- Network resource
- Admission control
- Common settings
- Service Platform common components
- Back-end and aliases
- Terminal Location web service settings
- Terminal Location notification web service settings

The different components that need to be configured are mentioned in this slide.

First Steps achieves most of the configuration.

The new feature – "Back-end and aliases" – is discussed in detail in the next few slides.

Back-end configuration (1 of 3)

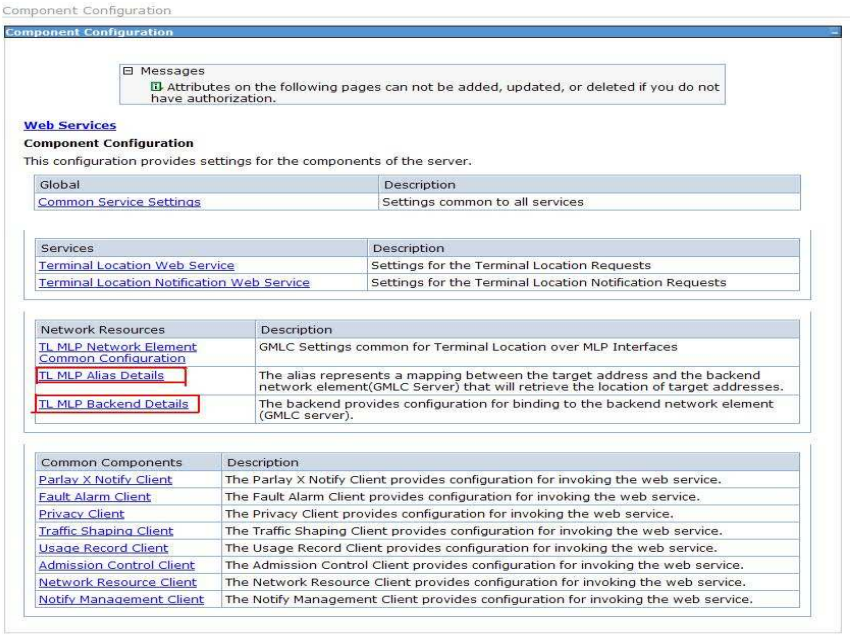


This slide shows how to configure the back-end.

To configure the back-end, log on to the administration console and click **TWSS Administration Console** and then click **Web Services**.

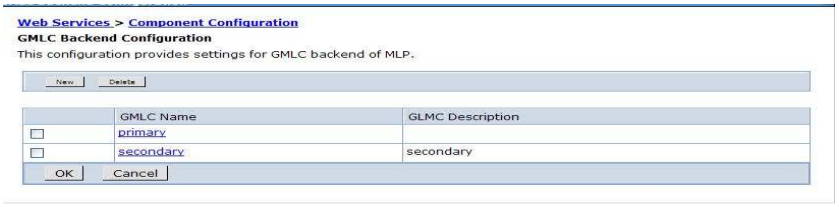
Select **PX21_TL_MLP** under **Web Services**.

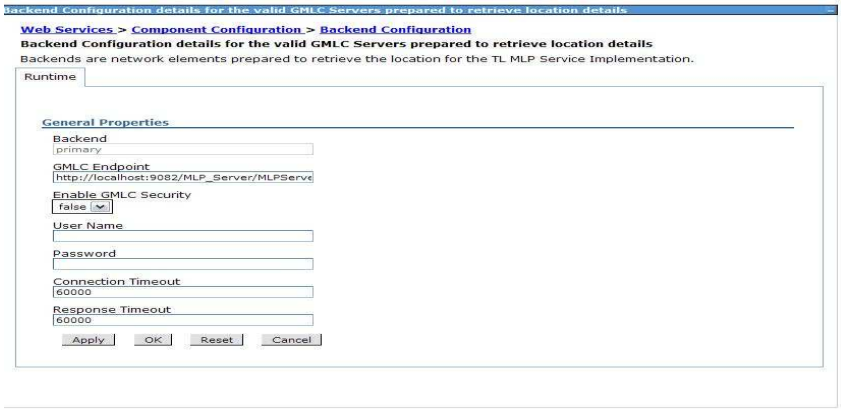
Back-end configuration (2 of 3)



This screen capture highlights two boxes in red - the alias and back-end details. Configure the alias and back-end in each of these sections.

Back-end configuration (3 of 3)



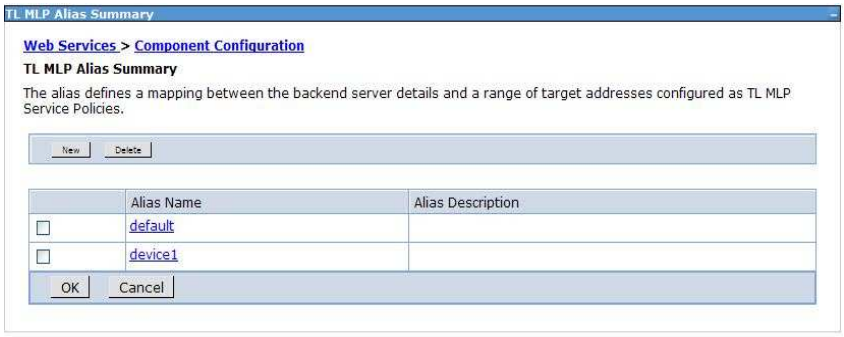


This slide shows two screen captures for the back-end.

Click **New** to specify the name and description for the back-end, as shown in the screen capture on top.

Click the new GMLC Name, to get the screen at the bottom. You can specify details such as Endpoint, Security, Connection Timeout, and Response Timeout.

Alias configuration (1 of 2)

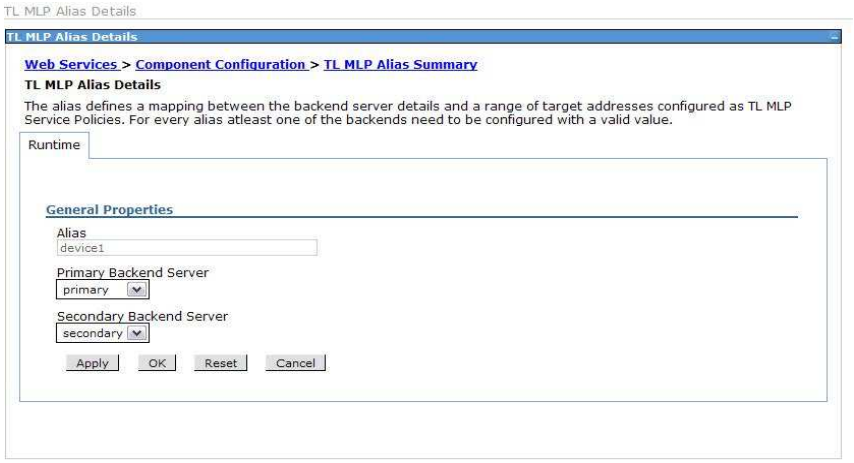


This slide shows the alias configuration.

This screen is displayed when you click **TL MLP Alias Details**.

Click **New** and create a new alias name and description.

Alias configuration (2 of 2)



You can click any of the alias names to configure the primary and secondary back-end servers for the alias, based on the back-ends created.

Example request (1 of 2)

```
<SOAP-ENV:Header>
      <policies>
        <policy attribute="service.config.target.Aliases" value="default;device1"/>
        <policy attribute="service.config.target.Ranges" value="111112223301-111112223305;111112223306-
  111112223310">
      </policies>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <getLocation xmlns="http://www.csapi.org/schema/parlayx/terminal_location/v2_2/local">
      <address>tel:+111112223301</address>
      <requestedAccuracy>100</requestedAccuracy>
      <acceptableAccuracy>110</acceptableAccuracy>
      <maximumAge>
        <metric xmlns="">Millisecond</metric>
        <units xmlns="">70000</units>
      </maximumAge>
      <responseTime>
        <metric xmlns="">Millisecond</metric>
        <units xmlns="">120000</units>
      </responseTime>
    </getLocation>
  </SOAP-ENV:Body>
```


Here is an example to demonstrate the new feature.

The two new policies have been highlighted, where two aliases - **default** and **device1** – are given and there are two corresponding ranges that ends in 01-05, 06-10.

The address is highlighted and it ends in 01.

When the request for this address is sent, it gets mapped to the alias - **default** - because it falls in the range configured for the **default** alias.

The **default** alias has two back-ends. This feature helps assign different addresses to different back-ends, which improves performance.

Example request (2 of 2)

```
<soapenv:Body>
    <p250:getLocationResponse xmlns:p250="http://www.csapi.org/schema/parlayx/terminal_location/v2_2/local">
      <p250:result>
        <latitude>60.274445</latitude>
        <longitude>75.25916</longitude>
        <altitude>0.0</altitude>
        <accuracy>250</accuracy>
        <timestamp>2007-10-12T13:44:53.000Z</timestamp>
      </p250:result>

    </p250:getLocationResponse>
 </soapenv:Body>
```

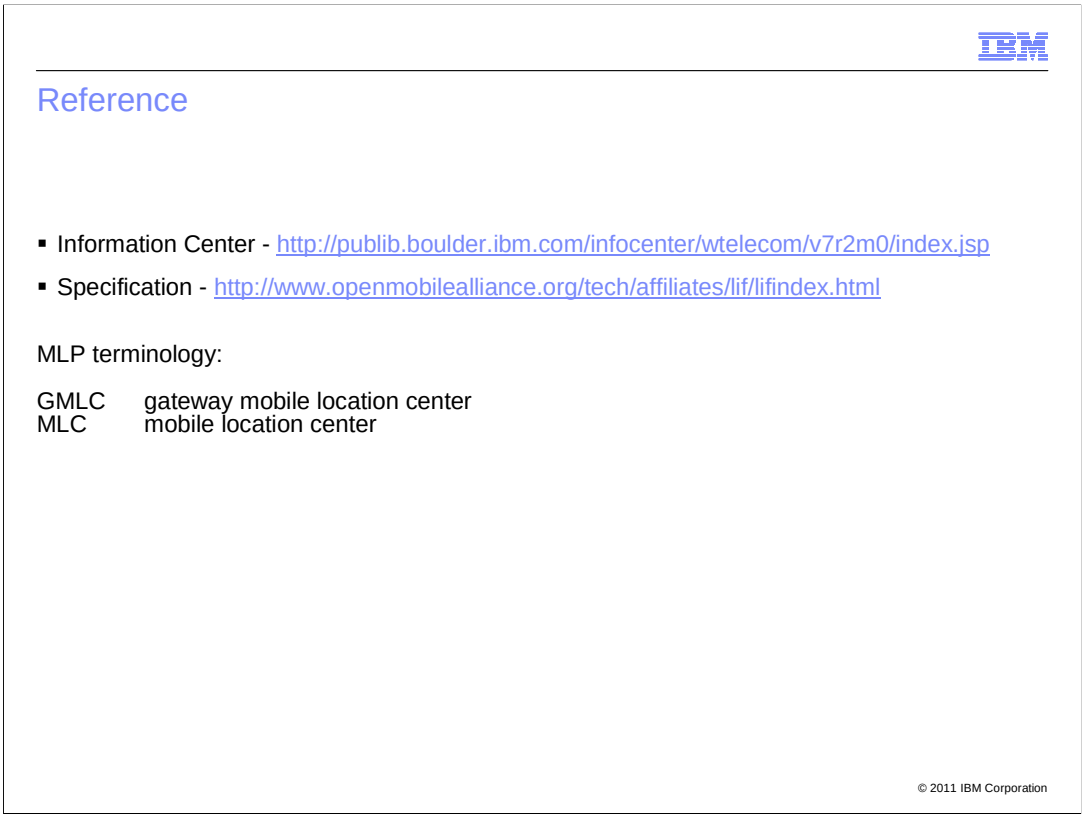

This is the response.

Troubleshooting

- **Log and trace files**
  - First check
- **Trace option for common framework**
  - com.ibm.mds.comm.*=all
- **Trace options for TL over MLP**
  - com.ibm.twss.parlayx21.tl.mlp.ctrl. * =all (Web Controller)
  - com.ibm.twss.parlayx21.tl.mlp.util.* =all (Utility)
- **Trace option for MLP connector**
  - com.ibm.twss.parlayx21.tl.mlp.conn.*=all

This slide gives information on troubleshooting.

Use the Log and Trace files to troubleshoot, and different trace options need to be given for different components.



For more information, see:

- Information Center (http://publib.boulder.ibm.com/infocenter/wtelecom/v7r2m0/index.jsp)
- Specification (http://www.openmobilealliance.org/tech/affiliates/lif/lifindex.html).

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_TL_over_MLP.ppt

This module is also available in PDF format at: ../TL_over_MLP.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, 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.
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.