Informatica Ultra Messaging (Version 6.7)

Dynamic Routing Guide

Informatica Ultra Messaging Dynamic Routing Guide

Version 6.7
April 2014

Part Number: UMDRO-UG-67000-0001

Table of Contents

Preface

This document explains design concepts and product implementation for the **Ultra Messaging® Dynamic Routing Option**.

Informatica Resources

Informatica My Support Portal

As an Informatica customer, you can access the Informatica My Support Portal at http://mysupport.informatica.com.

The site contains product information, user group information, newsletters, access to the Informatica customer support case management system (ATLAS), the Informatica How-To Library, the Informatica Knowledge Base, Informatica Product Documentation, and access to the Informatica user community.

Informatica Documentation

The Informatica Documentation team takes every effort to create accurate, usable documentation. If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com. We will use your feedback to improve our documentation. Let us know if we can contact you regarding your comments.

The Documentation team updates documentation as needed. To get the latest documentation for your product, navigate to Product Documentation from http://mysupport.informatica.com.

Informatica Web Site

You can access the Informatica corporate web site at http://www.informatica.com. The site contains information about Informatica, its background, upcoming events, and sales offices. You will also find product and partner information. The services area of the site includes important information about technical support, training and education, and implementation services.

Informatica How-To Library

As an Informatica customer, you can access the Informatica How-To Library at http://mysupport.informatica.com. The How-To Library is a collection of resources to help you learn more about Informatica products and features. It includes articles and interactive demonstrations that provide solutions to common problems, compare features and behaviors, and guide you through performing specific real-world tasks.

Informatica Knowledge Base

As an Informatica customer, you can access the Informatica Knowledge Base at http://mysupport.informatica.com. Use the Knowledge Base to search for documented solutions to known technical issues about Informatica products. You can also find answers to frequently asked questions,

technical white papers, and technical tips. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team through email at KB_Feedback@informatica.com.

Informatica Support YouTube Channel

You can access the Informatica Support YouTube channel at http://www.youtube.com/user/INFASupport. The Informatica Support YouTube channel includes videos about solutions that guide you through performing specific tasks. If you have questions, comments, or ideas about the Informatica Support YouTube channel, contact the Support YouTube team through email at supportvideos@informatica.com or send a tweet to @INFASupport.

Informatica Marketplace

The Informatica Marketplace is a forum where developers and partners can share solutions that augment, extend, or enhance data integration implementations. By leveraging any of the hundreds of solutions available on the Marketplace, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at http://www.informaticamarketplace.com.

Informatica Velocity

You can access Informatica Velocity at http://mysupport.informatica.com. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Global Customer Support

You can contact a Customer Support Center by telephone or through the Online Support.

Online Support requires a user name and password. You can request a user name and password at http://mysupport.informatica.com.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at http://www.informatica.com/us/services-and-training/support-services/global-support-centers/.

CHAPTER 1

Introduction

The **Ultra Messaging® Dynamic Routing Option** is a daemon that bridges disjoint Topic Resolution Domains (TRDs) by effectively forwarding control and user traffic between them. Thus, the UM Router facilitates WAN routing where multicast routing capability is absent, possibly due to technical obstacles or enterprise policies.

The UM Router transfers multicast and/or unicast topic resolution information, thus ensuring that receivers in disjoint topic resolution domains from the source can receive the topic messages to which they subscribe.

Features

The UM Router includes the following features:

- Full bidirectional forwarding
- Multi-hop forwarding
- Mesh, loop, or alternate path UM Router configurations
- Automatic rerouting around faults
- Support for wildcard receivers
- Support of Request/Response messages
- Traffic filtering on multiple criteria
- UM Router resilience
- UMP persistence support
- UM transport monitoring statistics
- Web Monitoring
- MIM and UIM forwarding
- UM JMS Pub/Sub model

The following features are not fully supported in this release of the UM Router:

- Queuing (including any JMS queuing functionality)
- UM JMS Point-to-Point model
- Ultra Load Balancing (ULB)
- Multitransport Threads (MTT)

Full UM Router support of these features is in development. If you desire any of these features or any configuration or topology not presented in this document, please contact Informatica Ultra Messaging Support for possible alternatives.

Restriction: The UM Router is not directly supported on the OpenVMS® platform. UM applications running on the OpenVMS® platform, however, can use a UM Router running on a different platform, such as Microsoft Windows® or Linux.

Known Issues

- When configuring a UM Router Peer Connection on a Microsoft® Windows® machine, the `single-tcp` configuration option can result in no message traffic across the link. We recommend using `tcp` instead of `single-tcp`.
- The UM Router cannot run as a standalone Windows service. This will be resolved in a future release.
- When using LBT-IPC transports, 64-bit applications cannot interoperate with 32-bit applications (with or without UM Routers)
- If using LBT-RDMA with the UM Router and you exit the UM Router with **Ctrl-C**, you may see a segmentation fault. Informatica is aware of this and has not observed any ill effects from this segmentation fault.

Comparison to UM Gateway

With the release of Ultra Messaging 6.0, the UM Gateway feature is being discontinued and replaced by the Ultra Messaging Dynamic Routing Option (also referred to as the UM Router).

The UM Router's primary improvement over the UM Gateway is its ability to intelligently select efficient traffic routes from multiple path choices on a dynamic topic-by-topic basis.

Note: This release of the UM Router is not backward compatible with earlier versions of the UM Gateway in the sense that you cannot have UM Routers and UM Gateways in the same network.

Added Features and Differences

In addition to routing functionality, the following are features of the UM Router that were not provided in the UM Gateway:

- Multi-path, ring, or mesh topologies
- Interoperability with MIM, UMP, and UM JMS (see "UM Feature Compatibility" on page 16 for complete feature interoperability information)
- Ability to restart the UM Router within a transport's activity timeout period
- Reduced topic resolution traffic via more efficient use of Use Queries and Use Query Responses
- The default value for the portal `<cost>` is `1` (one). `0` (zero) is not a valid cost value.
- The UM Router daemon (`tnwgd`) logs version information on startup.
- The following configuration options exist in the UM Router but not the UM Gateway. See Chapter 5, "XML Configuration Reference" on page 46 for more information on these options.

  - `<name>` (as a `<daemon>` child)

  - `<route-info>`

  - `<route-recalculation>`

  - `<source-deletion-delay>`

  - `<max-queue>`

  - `<remote-topic-interest>`
  - `<remote-pattern-interest>`
  - `<rate-limit>`
  - `<domain-route>`
  - `<remote-topic>`
  - `<remote-pattern>`
  - `<sourcemap>`
- The following UM Router configurations have been deprecated and will be removed in a later release:
  - `<propagation-delay>`
  - `<late-join>`
  - `<topic-purge>`
  - `<topic-interest-generate>`
  - `<topic-domain-activity>`
  - `<pattern-purge>`
  - `<pattern-interest-generate>`
  - `<pattern-domain-activity>`
  - `<topic-use-check>`
  - `<pattern-use-check>`
- You cannot configure two endpoint portals on a UM Router to have the same adjacent `domain id`
- You cannot configure two peer portals on a UM Router to connect to the same adjacent UM Router
- The UM Router ignores UM Spectrum configuration options, which results in UM Routers forwarding all channel data without applying any filtering. See the *UM Concepts Guide*, Chapter 5, *Spectrum* for more about channels.

UM Gateway Bug Fixes

The following are UM Router capabilities that address bugs in the last version of the UM Gateway:

- When disconnecting and reconnecting TCP links between peer portals, memory leaks no longer occur.
- The UM Router does not issue the following error log message when installed with a UMS or UMP installation: `[error] Gwd-6033-256: endpoint portal [LAN2] unable to send: datagram size mismatch. transport_XXX_datagram_max_size must be properly configured. This is a configuration error.`
- The 64-bit Windows UM Dynamic Routing Option installer places files directly into the `Win2k-x86_64` directory.

CHAPTER 2

Architecture

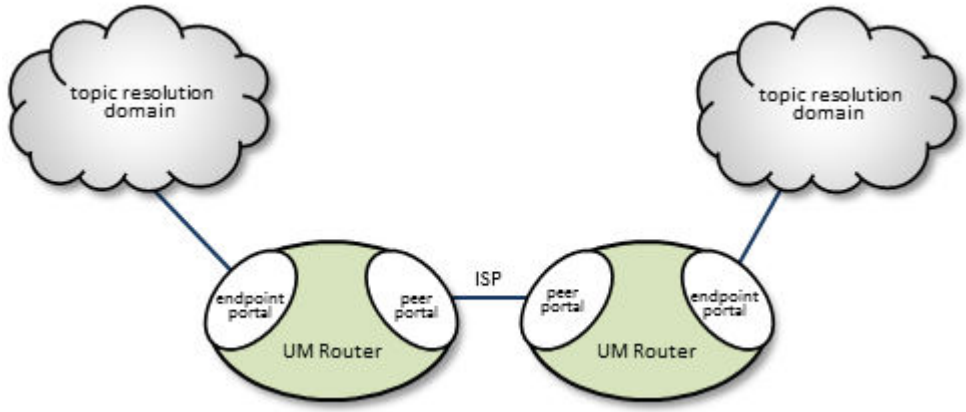
This chapter includes the following topics:

UM Router Portals

The UM Router uses interfaces, called *portals*, through which to pass data. A UM Router consists of two or more bidirectional portals that may be one of two types:

- An **endpoint** portal, which communicates directly to a UM topic resolution domain (TRD; see "Topic Resolution Domains" on page 5).
- A **peer** portal, which communicates via TCP with another peer portal (of another UM Router), allowing tunneling between UM Routers. Two peer portals connected to each other are referred to as *companion* peers, and utilize TCP connections for all data and control traffic (UDP is not supported for this). Between the peer portals, you can configure either a single TCP connection, or a dual TCP with one connection serving each direction.

The figure below shows a simple UM Router use case, where two UM Routers bridge an ISP to connect two



TRDs.

You configure portals in the UM Router's XML configuration file, specifying the portal's name, cost, UM Configuration, Access Control Lists and other attributes. See Chapter 5, "XML Configuration Reference" on page 46 for more.

Topic Resolution Domains

Since topic resolution uses UDP, sources and receivers must have UDP connectivity to each other. When they do, we consider them to be in the same topic resolution domain (TRD). More specifically, UM contexts must satisfy the following two requirements to belong to the same topic resolution domain.

- The contexts must use the same topic resolution UM configuration (i.e., `resolver_*` options are the same).
- Contexts can communicate using the protocols required for both message transport and topic resolution traffic.

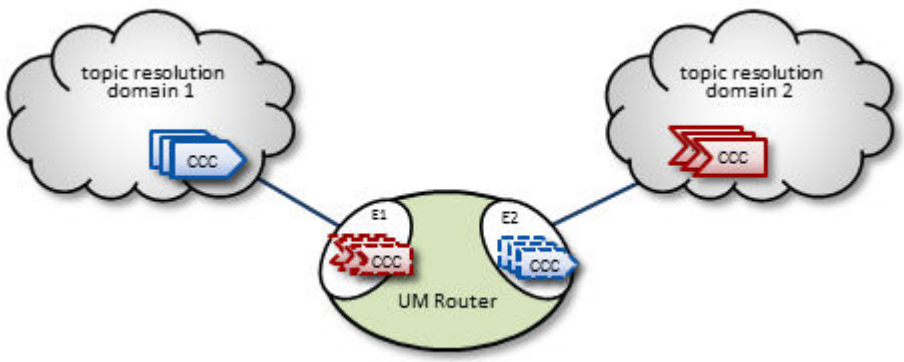
For example, two contexts on separate machines in the same LAN are not in the same topic resolution domain if they use different resolver addresses. See *`Multicast Resolver Network Options`* . A topic resolution domain can span a WAN if the UM contexts on each side of a firewall use the same UM configuration and the firewall allows UDP traffic (multicast or unicast) to pass.

Each endpoint portal must identify its associated topic resolution domain with a `domain-id` the UM Router's XML configuration file, as in the example below. All portals in the same TRD must have the same `domain-id`, and different TRDs networked together via UM Routers must have domain-ids unique to each other.

```
<portals>
  <endpoint>
    <name>LAN100</name>
    <domain-id>100</domain-id>
    <lbm-config>lan100.cfg</lbm-config>
  </endpoint>
  <endpoint>
    <name>LAN200</name>
    <domain-id>200</domain-id>
    <lbm-config>lan200.cfg</lbm-config>
  </endpoint>
</portals>
```

Proxy Sources and Proxy Receivers

To resolve a topic across a UM Router (described in Topic Resolution Across the UM Router), the UM Router creates, within portals, proxy sources and proxy receivers (shown in the figure below by their dashed lines). These proxies behave like their UM counterparts; they resolve topics on the TRDs like normal sources and receivers, and the UM Router internally passes data from one portal to the other. However unlike regular sources, proxy sources do not have retransmission retention buffers normally used for Late Join or OTR.



Portals exist while the UM Router is running, however, the UM Router creates proxy sources and receivers during topic resolution and deletes them when the topic is retired.

Routing

In multiple-UM Router environments where more than one UM Router can provide possible messaging pathways, the UM Routers are able to cooperatively determine and establish optimal routes. Also, the UM Router network is able to detect link or other UM Router outages and automatically reroute traffic as needed. UM Router Concepts, Routing Topologies, describes possible topologies in more detail.

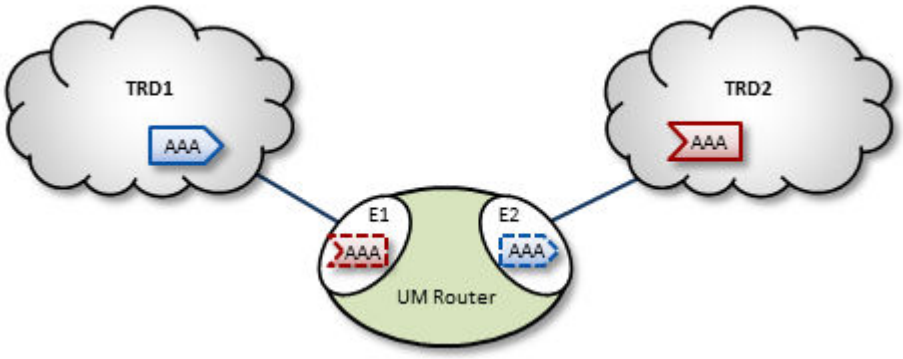
CHAPTER 3

UM Router Concepts

This chapter includes the following topics:

Basic UM Router Operation

The diagram below shows a UM Router bridging topic resolution domains TRD1 and TRD2, for topic AAA, in a *direct link* configuration. Endpoint E1 contains a proxy receiver for topic AAA and endpoint E2 has a proxy source for topic AAA.



Topic Resolution Across the UM Router

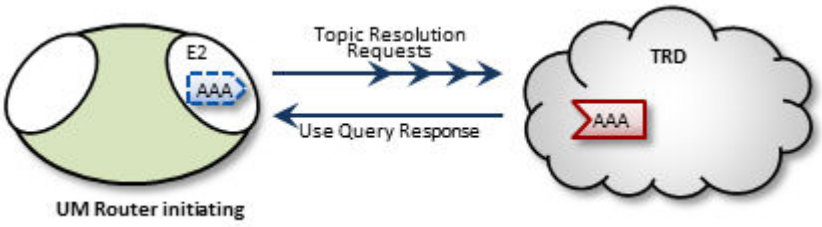
To establish topic resolution in an already-running UM Router, the following typically occurs in an example like the above figure.

1. A receiver in TRD2 issues a TQR (Topic Query Record) for topic AAA.

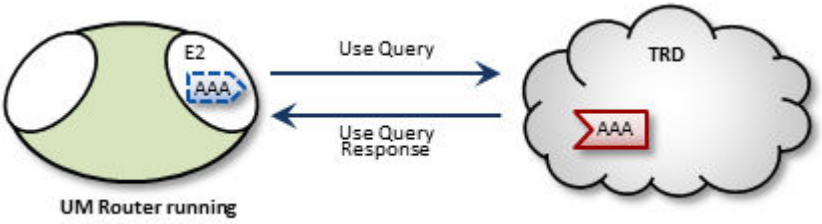
2. Portal E2 receives the TQR and passes information about topic AAA to all other portals in the UM Router. (In this case, E1 is the only other portal.)
3. E1 immediately responds with three actions: a) create a proxy receiver for topic AAA, b) the new proxy receiver sends a TQR for AAA into TRD1, and c) E1 issues a Topic Interest message into TRD1 for the benefit of any other UM Routers that may be connected to that domain.
4. A source for topic AAA in TRD1 sees the TQR and issues a TIR (Topic Information Record).
5. E2 creates proxy source AAA, which then issues a TIR to TRD2. The receiver in TRD2 joins the transport, thus completing topic resolution.
6. E1's AAA proxy receiver sees the TIR and requests that E2 (and any other interested portals in the UM Router, if there were any) create a proxy source for AAA.

Interest and Use Queries

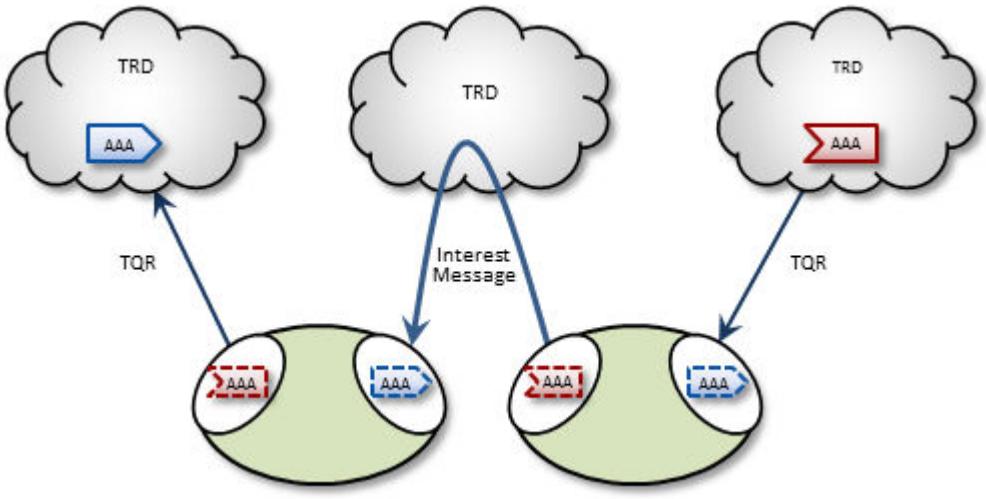
When a UM Router starts, its endpoint portals issue a brief series of Topic Resolution Request messages to their respective topic resolution domains. This provokes quiescent receivers (and wildcard receivers) into sending Use Query Responses, indicating interest in various topics. Each portal then records this interest.



After a UM Router has been running, endpoint portals issue periodic Topic Use Queries and Pattern Use Queries (collectively referred to as simply Use Queries). Use Query Responses from UM contexts confirm that the receivers for these topics indeed still exist, thus maintaining these topics on the interest list. Autonomous TQRs also refresh interest and have the effect of suppressing the generation of Use Queries.



In the case of multi-hop UM Router configurations, UM Routers cannot detect interest for remote contexts via Use Queries or TQRs. They do this instead via Interest Messages. An endpoint portal generates periodic interest messages, which are picked up by adjacent UM Routers (i.e., the next hop over), at which time interest is refreshed.



You can adjust intervals, limits, and durations for these topic resolution and interest mechanisms via UM Router configuration options (see Chapter 5, “XML Configuration Reference” on page 46).

UM Router Keepalive

To maintain a reliable connection, peer portals exchange *UM Router Keepalive* signals. Keepalive intervals and connection timeouts are configurable on a per-portal basis, and you can also set the UM Router to send keepalives only when traffic is idle (default). When both traffic and keepalives go silent at a portal's ingress, the connection is considered lost and the portal disconnects the TCP link (and attempts to reconnect). See “<gateway-keepalive/>” on page 92.

More About Proxy Sources and Receivers

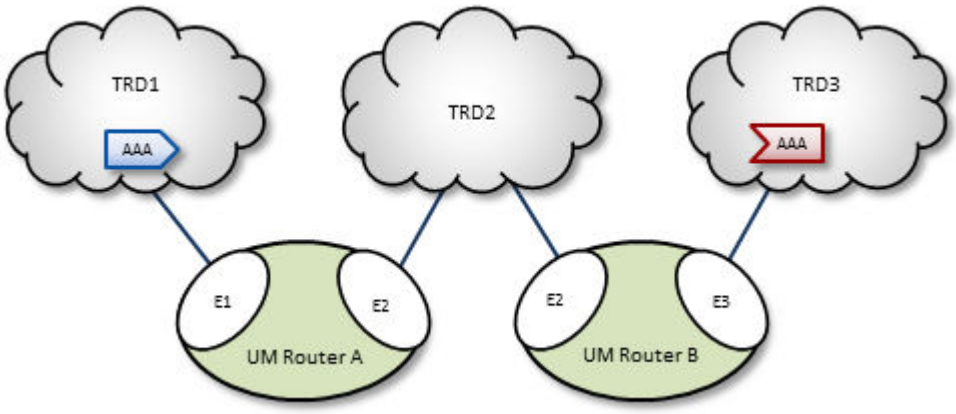
The `domain-id` is used by Interest Messages and other internal and UM Router-to-UM Router traffic to ensure forwarding of all messages (payload and topic resolution) to the correct recipients. This also has the effect of not creating proxy sources/receivers where they are not needed. Thus, UM Routers create proxy sources and receivers based solely on receiver interest.

If more than one source sends on a given topic, the receiving portal's single proxy receiver for that topic receives all messages sent on that topic. The sending portal, however creates a proxy source for every source sending on the topic. The UM Router maintains a table of proxy sources, each keyed by an Originating Transport ID (OTID), enabling the proxy receiver to forward each message to the correct proxy source. An OTID uniquely identifies a source's transport session, and is included in topic advertisements.

Note: It is important to keep maximum datagram sizes exactly the same across all TRDs and transports. For example, if the TRD on one side of a UM Router uses LBT-RM message transport and the TRD on the other side uses LBT-RDMA with a larger maximum datagram size configured, fragments from domain 1 will be too large for domain 2. See `transport_*_datagram_max_size` in the *UM Configuration Guide*.

Multi-Hop Forwarding

UM can resolve topics across a span of multiple UM Routers. Consider a simple example UM Router deployment, as shown in the following figure.



In this diagram, UM Router A has two endpoint portals connected to topic resolution domains TRD1 and TRD2. UM Router B also has two endpoint portals, which bridge TRD2 and TRD3. Endpoint portal names reflect the topic resolution domain to which they connect. For example, UM Router A endpoint E2 interfaces TRD2.

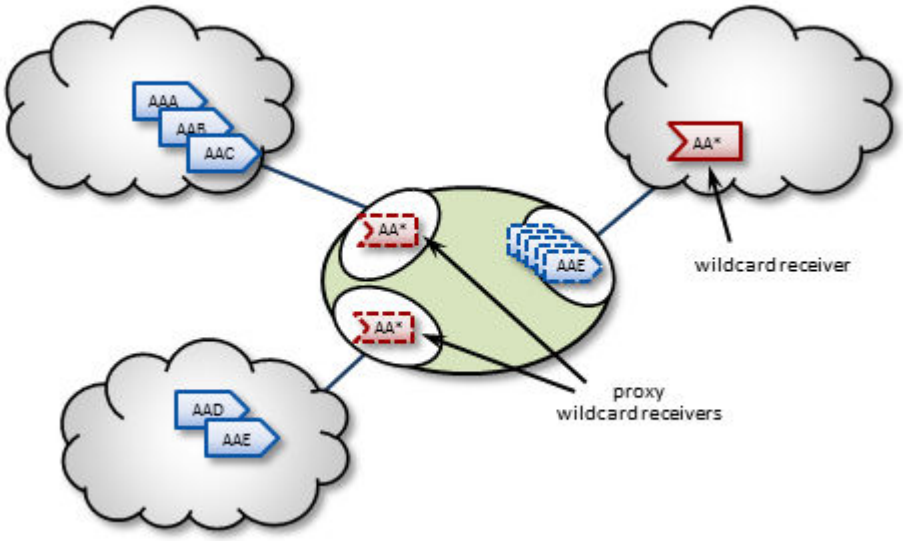
TRD1 has a source for topic AAA, and TRD3, an AAA receiver. The following sequence of events enables the forwarding of topic messages from source AAA to receiver AAA.

1. Receiver AAA queries (issues a TQR).
2. UM Router B, endpoint E3 (B-E3) receives the TQR and passes information about topic AAA to all other portals in the UM Router. In this case, B-E2 is the only other portal.
3. In response, B-E2 creates a proxy receiver for AAA and sends a Topic Interest message for AAA into TRD2. The proxy receiver also issues a TQR, which in this case is ignored.
4. UM Router A, endpoint E2 (A-E2) receives this Topic Interest message and passes information about topic AAA to all other portals in the UM Router. In this case, A-E1 is the only other portal.
5. In response, A-E1 creates a proxy receiver for AAA and sends a Topic Interest message and TQR for AAA into TRD1.
6. Source AAA responds to the TQR by sending a TIR for topic AAA. In this case, the Topic Interest message is ignored.
7. The AAA proxy receiver created by A-E1 receives this TIR and requests that all UM Router A portals with an interest in topic AAA create a proxy source for AAA.
8. In response, A-E2 creates a proxy source, which sends a TIR for topic AAA via TRD2.
9. The AAA proxy receiver at B-E2 receives this TIR and requests that all UM Router B portals with an interest in topic AAA create a proxy source for AAA.
10. In response, B-E3 creates a proxy source, which sends a TIR for topic AAA via TRD3. The receiver in TRD3 joins the transport.
11. Topic AAA has now been resolved across both UM Routers, which forward all topic messages sent by source AAA to receiver AAA.

Note: Access Control Lists can alter this sequence, for example, by accepting only certain transport types or forwarding only certain topics. For more, see "Access Control Lists (ACL)" on page 21. Also note that since this example uses a straight-line path, routing functionality does not apply here.

Wildcard Receivers

The UM Router supports topic resolution for wildcard receivers in a manner very similar to non-wildcard receivers. Wildcard receivers in a TRD issuing a WC-TQR cause corresponding proxy wildcard receivers to be created in portals, as shown in the following figure. The UM Router creates a single proxy source for pattern match.



Forwarding Costs

Forwarding a message through a UM Router incurs a cost in terms of latency, network bandwidth, and CPU utilization on the UM Router machine (which may in turn affect the latency of other forwarded messages). Transiting multiple UM Routers adds even more cumulative latency to a message. Other UM Router-related factors such as portal buffering, network bandwidth, switches, etc., can also add latency.

Factors other than latency contribute to the cost of forwarding a message. Consider a message that can be sent from one domain to its destination domain over one of two paths. A three-hop path over 1Gbps links may be faster than a single-hop path over a 100Mbps link. Further, it may be the case that the 100Mbps link is more expensive or less reliable.

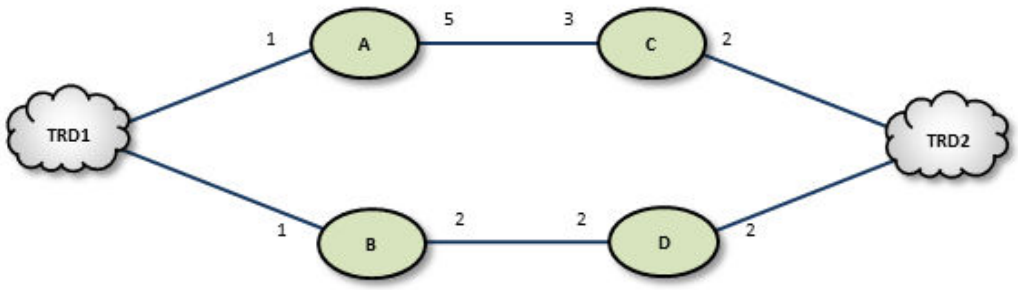
You assign forwarding cost values on a per-portal basis. When summed over a path, these values determine the cost of that entire path. A network of UM Routers uses forwarding cost as the criterion for determining the best path over which to resolve a topic.

UM Router Routing

UM Routers have an awareness of other UM Routers in their network and how they are linked. Thus, they each maintain a topology map, which is periodically confirmed and updated. This map also includes forwarding cost information.

Using this information, the UM Routers can cooperate during topic resolution to determine the best (lowest cost) path over which to resolve a topic or to route control information. They do this by totaling the costs of all portals along each candidate route, then comparing the totals.

For example, the following figure shows two possible paths from TRD1 to TRD2: A-C (total route cost of 11) and B-D (total route cost of 7). In this case, the UM Routers select path B-D.



If a UM Router or link along path B-D should fail, the UM Routers detect this and reroute over path A-C. Similarly, if an administrator revises cost values along path B-D to exceed a total of 12, the UM Routers reroute to A-C.

If the UM Routers find more than one path with the same lowest total cost value, i.e., a "tie", they select the path based on a node-ID selection algorithm. Since administrators do not have access to node IDs, this will appear to be a pseudo-random selection.

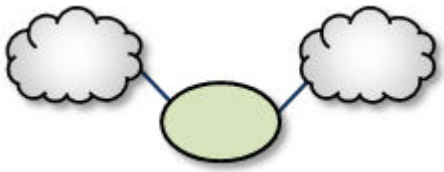
Note: You cannot configure parallel paths (such as for load balancing or Hot failover), as the UM Routers always select the lowest-cost path and only the lowest-cost path for all data between two points. However, you can devise an exception to this rule by configuring the destinations to be in different TRDs. For example, you can create an HFX Receiver bridging two receivers in different TRD contexts. The UM Routers route to both TRDs, and the HFX Receiver merges to a single stream for the application.

Routing Topologies

You can configure multiple UM Routers in a variety of topologies. Following are several examples.

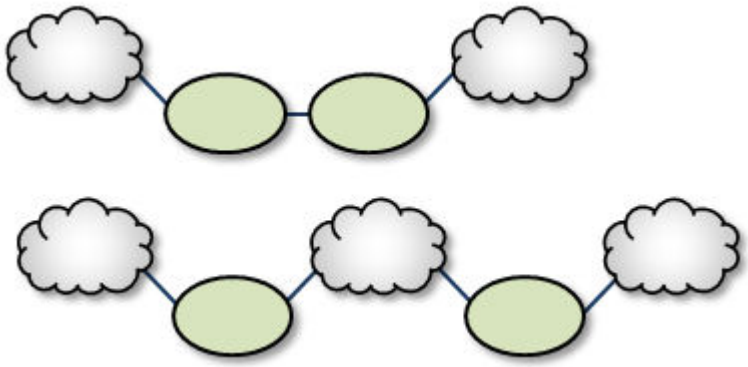
Direct Link

The *Direct Link* configuration uses a single UM Router to directly connect two TRDs. For a configuration example, see “Direct Link” on page 26.



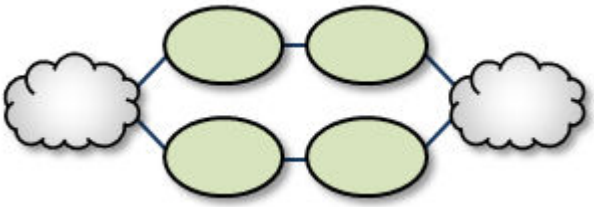
Single Link

A *Single Link* configuration connects two TRDs over an intermediate link or through an intermediate TRD. For configuration examples, see “Single Link with Intermediate LAN” on page 28 and “Single Link, Peer to Peer” on page 27.



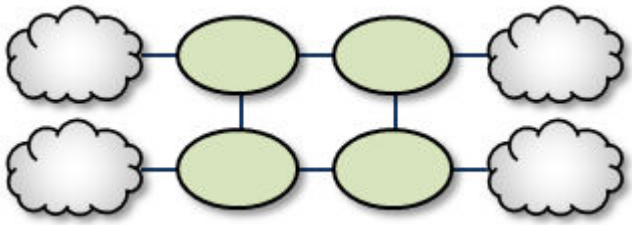
Parallel Links

Parallel Links offer multiple complete paths between two TRDs. You can use these in a failover configuration by setting the primary path for the lowest cost and standby paths at higher costs. For a configuration example, see “Parallel Links” on page 30.



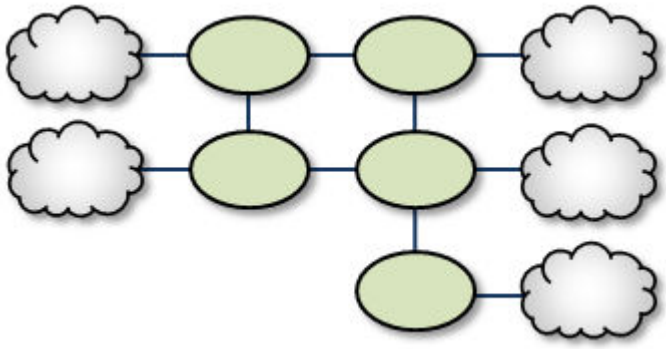
Loop

Loops let you route packets back to the originating UM Router without reusing any paths. Also, if any peer-peer links are interrupted, the looped UM Routers are able to find an alternate route between any two TRDs.



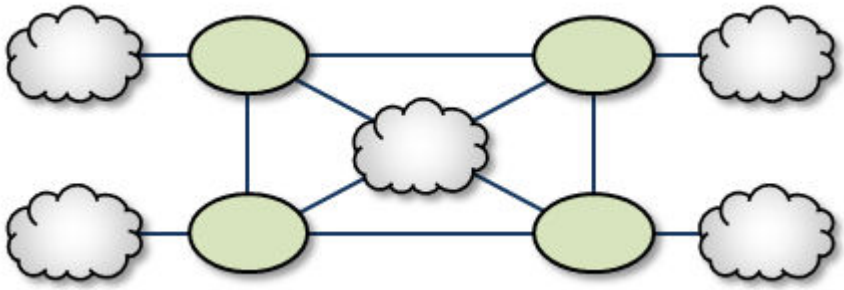
Loop and Spur

The *Loop and Spur* has a one or more UM Routers tangential to the loop and accessible only through a single UM Router participating in the loop. For a configuration example, see “Loop and Spur” on page 32.



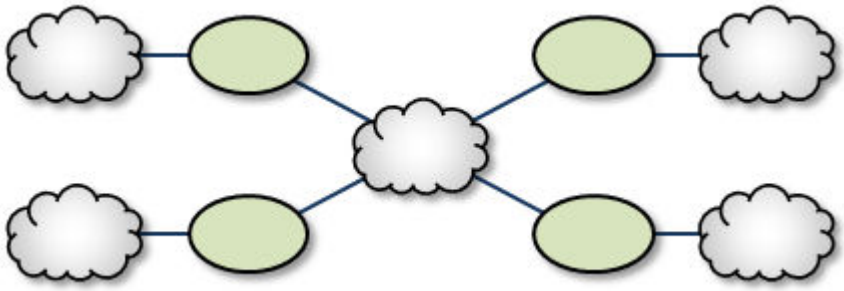
Loop with Centralized TRD

Adding a TRD to the center of a loop enhances its rerouting capabilities.



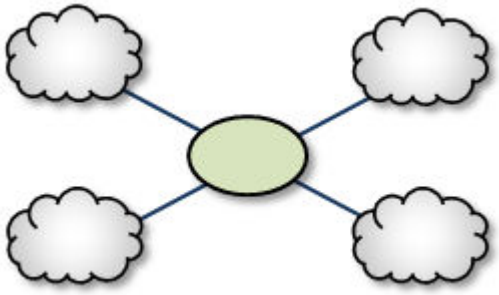
Star with centralized TRD

A Star with a centralized TRD does not offer rerouting capabilities but does provide an economical way to join multiple disparate TRDs.



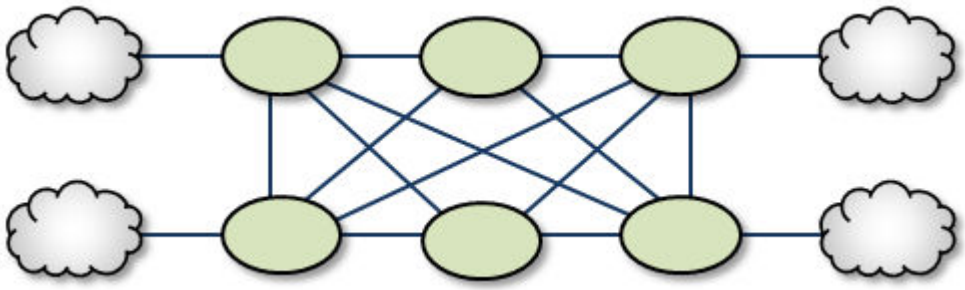
Star with Centralized UM Router

The Star with a centralized UM Router is the simplest way to bridge multiple TRDs. For a configuration example, see “Star with Centralized UM Router” on page 35.



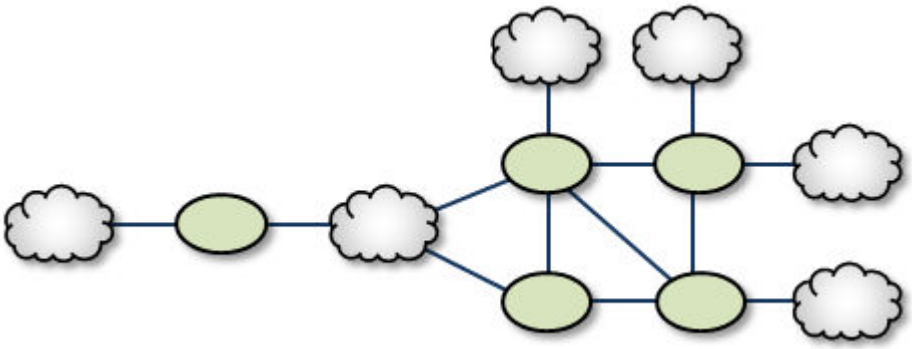
Mesh

The Mesh topology provides peer portal interconnects between many UM Routers, approaching an all-connected-to-all configuration. This provides multiple possible paths between any two TRDs in the mesh. Note that this diagram is illustrative of the UM Router's connectability, and not necessarily a practical or recommended application. For a configuration example, see “Mesh” on page 37.



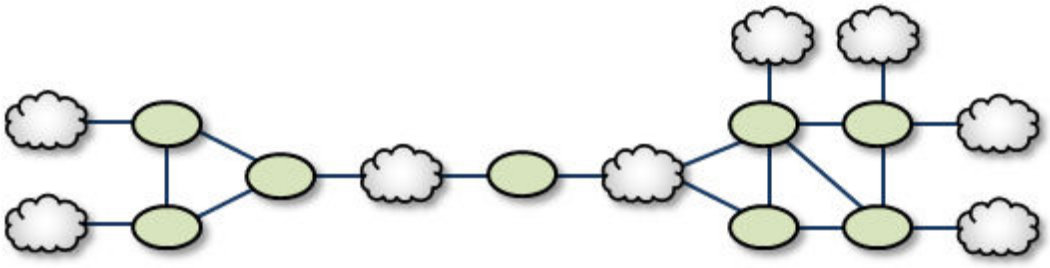
Palm Tree

The *Palm Tree* has a set of series-connected TRDs fanning out to a more richly meshed set of TRDs. This topology tends to pass more concentrated traffic over common links for part of its transit while supporting a loop, star, or mesh near its terminus.



Dumbbell

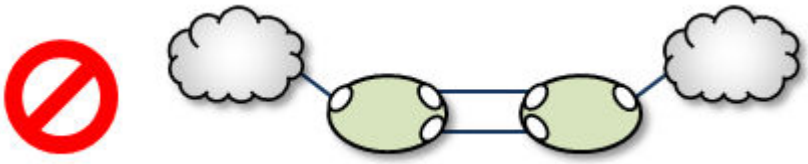
Similar to the Palm Tree, the *Dumbbell* has a funneled route with a loop, star, or mesh topology on each end.



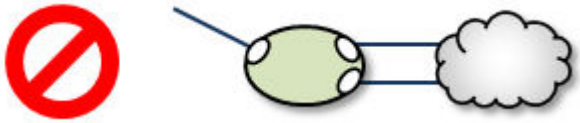
Unsupported Configurations

When designing UM Router networks, do not use any of the following topology constructs:

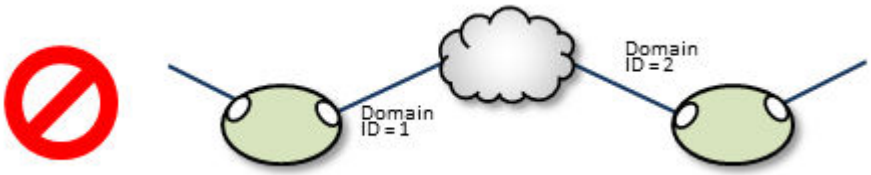
- Two peer-to-peer connections between the same two UM Routers



- Two endpoint connections from the same UM Router to the same TRD



- Assigning two different Domain ID values (from different UM Routers) to the same TRD



UM Feature Compatibility

You must install the UM Dynamic Routing Option with its companion Ultra Messaging UMS, UMP, or UMQ product, and versions must match. While most UM features are compatible with the UM Router, some are not. Following is a table of features and their compatibilities with the UM Router.

UM Feature	UM Router compatible?	Notes
Acceleration - DBL	Yes	
Acceleration - UD	Yes	

UM Feature	UM Router compatible?	Notes
Hot Failover (HF)	Yes	The UM Router supports paths between HF sources and receivers, however it does not provide proxy HF sources or HF receivers.
Hot Failover Across Contexts (HFX)	Yes	
Late Join	Yes	
Message Batching	Yes	
Monitoring/Statistics	Yes	
Multicast Immediate Messaging (MIM)	Yes	
Multi-Transport Threads	No	
Off-Transport Recovery (OTR)	Yes	
Ordered Delivery	Yes	
Pre-Defined Messaging (PDM)	Yes	
Request/Response	Yes	
Self-Describing Messaging (SDM)	Yes	
Source Side Filtering	Yes	The UM Router supports transport source side filtering. You can activate this either at the originating TRD source, or at a downstream proxy source.
Transport LBT-IPC	Yes	
Transport LBT-RDMA	Yes	
Transport LBT-RM	Yes	
Transport LBT-RU	Yes	
Transport LBT-SMX	Partial	The UM Router changes any proxy sources configured for LBT-SMX to TCP. The UM Router generates a log message recording the transport change. The UM Router accepts LBT-SMX ingress traffic to proxy receivers.
Transport TCP	Yes	
Transport TCP-LB	Yes	

UM Feature	UM Router compatible?	Notes
UM JMS, Publish/Subscribe Model	Yes	
UM JMS, Point-To-Point Model	No	
UM Spectrum	Yes	The UM Router supports UM Spectrum traffic, but you cannot implement Spectrum channels in UM Routerproxy sources or receivers.
UMP Implicit/Explicit Acknowledgements	Yes	
UMP Persistent Store	Yes	
UMP Proxy Sources	Yes	
UMP Quorum Consensus	Yes	
UMP Registration ID/Session Management	Yes	
UMP Receiver-Paced Persistence (RPP)	Yes	
UMP Round Robin	Yes	
UMP Store Failover	Yes	
UMQ Application Sets	No	
UMQ Parallel Queue Dissemination (PQD)	No	
UMQ Queue Browser	No	
UMQ Queue Failover	No	
UMQ Queue Redundancy	No	
UMQ Registration ID/Session Management	No	
UMQ Serial Queue Dissemination (SQD)	No	
UMQ Source Dissemination (SD)	No	
UMQ Ultra Load Balancing (ULB)	No	
Ultra Messaging Desktop Services (UMDS)	No	
Ultra Messaging Manager (UMM)	Yes	No UM Router management

UM Feature	UM Router compatible?	Notes
UM SNMP Agent	No	
UMCache	No	
Wildcard Receivers	Yes	
Zero Object Delivery (ZOD)	Yes	

CHAPTER 4

UM Router Implementation

This chapter includes the following topics:

Configuration Overview

When the UM Router daemon launches, it uses configuration option settings to determine its behavior and expectations. You specify option values in an XML configuration file, and reference the file from a command line argument.

Typically, you have a separate XML configuration file for each UM Router, which contains structured configuration elements that describe aspects of the UM Router. Within this XML configuration file, each endpoint portal definition points to a UM configuration file, which allow the portal to properly connect to its TRD.

Creating Applications for UM Router Compatibility

When developing messaging applications that use Ultra Messaging and, in particular, the UM Router, please observe the following guidelines.

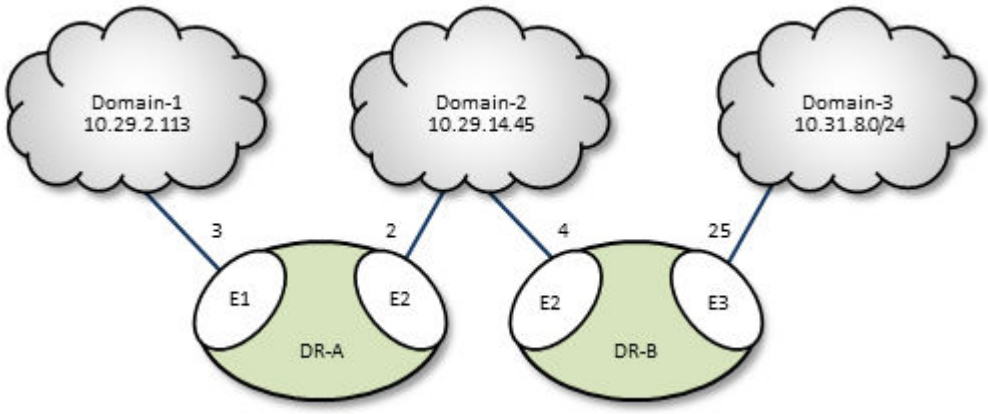
Naming and Identification

An important part to successfully implementing UM Routers is prudent and error-free naming of TRDs, UM Routers, portals, etc., as well as correct identification of IP addresses and ports. It is good practice to first design the UM Router network by defining all connections and uniquely naming all UM Routers, portals, and

TRDs. This works well as a diagram similar to some examples presented in this document. Include the following names and parameters in your design diagram:

- TRD names and IDs
- UM Router names
- Portal names
- Portal costs

For example, a well-prepared UM Router design could look like the following figure.



Portal Costs

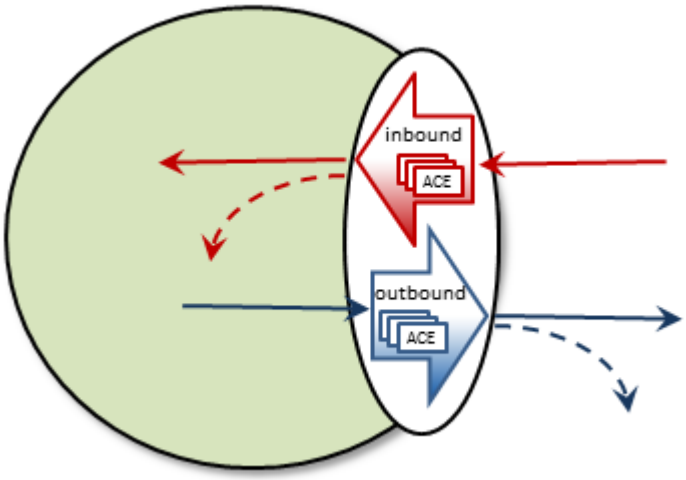
A network of UM Routers uses forwarding cost as the criterion for determining the best (lowest cost) path over which to resolve a topic and route data. Forwarding cost is simply the sum of all portal costs along a multi-UM Router path. Thus, total cost for the single path in the above example is 34. (Note that this is a non-real-world example, since costs are pointless without alternate routes to compare to.) You assign portal costs via the `<cost>` configuration option.

After the UM Router network calculates its paths, if a new lower-cost source becomes available, receivers switch to that path.

Access Control Lists (ACL)

You can apply Access Control Lists (ACL) to a UM Router's portals to filter traffic by certain topics, transports, topic patterns, multicast groups, etc. You configure ACLs in a UM Router's XML configuration file, as children of an `<endpoint>` or `<peer>` portal. As traffic arrives at the portal, the portal either forwards it or rejects it per ACL criteria.

Inbound ACLs determine what information to forward to other portals in the UM Router, while Outbound ACLs determine (by topic) what information from other portals that this portal can send out the UM Router. Each portal (endpoint or peer) can have up to one inbound ACL and one outbound ACL.



An ACL can contain one or more Access Control Entries (ACEs). ACEs are the filters that let you match (and accept or reject based on), criteria elements. For example, to accept only messages for topic ABC:

```
<acl>
  <inbound>
    <ace match="accept">
      <topic>ABC</topic>
    </ace>
  </inbound>
</acl>
```

Possible ACE condition elements are:

- `<multicast-group/>` *
- `<pcre-pattern>` (PCRE wildcard patterns)
- `<regex-pattern>` (Regex wildcard patterns)
- `<source-ip/>` *
- `<tcp-source-port/>` *
- `<topic>`
- `<transport/>` *
- `<udp-destination-port/>` *
- `<udp-source-port/>` *
- `<xport-id/>` * (for LBT-IPC traffic)

* These items apply to only inbound ACLs, and are ignored if used with an outbound ACL.

The above elements are all children of the `<ace>` element. When an ACL has multiple ACE entries, the UM Router goes down the list until it finds a match. It then accepts (forwards) or rejects, and is done with that ACL. An implicit "reject all" is at the end of every ACL, so the UM Router rejects any topic not matched. If you place multiple conditions within an ACE, the UM Router performs an "and" operation with them.

Note that the portal ignores a condition element if a) it is inbound-only and used in an outbound ACL, or b) it simply does not apply (such as a `<udp-source-port/>` if the transport is TCP).

Also note that ACLs can affect topic resolution traffic as well as user messages. They can, for example, block a topic (which prevents the creation of proxy receivers) and, thus, protect remote TRDs from unwanted queries and advertisements. This effect does not apply to wildcard receivers, however, because ACLs match

only on discrete topics. Thus, while ACLs can operate on specific topic traffic derived from wildcard topic resolution, they cannot prevent pattern interest from propagating throughout the network.

Consider the following example, where we configure a portal to forward on specific topics. This example also illustrates the parent/child hierarchy for ACLs, ACEs, and ACE conditions.

```
<endpoint>
  <name>LAN1</name>
  <lbm-config>lan1.cfg</lbm-config>
  <domain-id>1</domain-id>
    <acl>
      <inbound>
        <ace match="accept">
          <topic>ABC</topic>
        </ace>
        <ace match="accept">
          <topic>DEF</topic>
          <transport value=lbt-rm comparison=eq/>
        </ace>
        <ace match="accept">
          <topic>GHI</topic>
        </ace>
      </inbound>
    </acl>
</endpoint>
```

The above example shows each topic match in a separate ACE. When topic "GHI" arrives, the portal finds a match in the third ACE and forwards the topic. (Placing all three `<topic>`s in a single ACE would never match anything.) Also note that "DEF" is forwarded only if it uses an LBT-RM transport.

Since the behavior for multiple ACEs is "first match, then done", list ACEs in a specific-to-general order. For the example below, to forward topic "ABC123" but reject similar topics such as "ABCD123" or "ABCE123", list the ACE for "ABC123" first (as done below). If the ACE to reject "ABC.*123" was listed first, it would also (undesirably) match and reject "ABC123".

```
<endpoint>
  <name>LAN1</name>
  <lbm-config>lan1.cfg</lbm-config>
  <domain-id>1</domain-id>
    <acl>
      <inbound>
        <ace match="accept">
          <topic>ABC123</topic>
        </ace>
        <ace match="reject">
          <pcre-pattern>ABC.*123</pcre-pattern>
        </ace>
      </inbound>
    </acl>
</endpoint>
```

You can also filter on certain transport types to accept multicast traffic but reject tcp traffic, as shown below.

```
<endpoint>
  <name>LAN1</name>
  <lbm-config>lan1.cfg</lbm-config>
  <domain-id>1</domain-id>
    <acl>
      <inbound>
        <ace match="accept">
          <transport comparison="equal" value="lbtrm"/>
        </ace>
        <ace match="reject">
          <transport comparison="equal" value="tcp"/>
        </ace>
```

```
        </inbound>
      </acl>
</endpoint>
```

Timers and Intervals

The UM Router offers a wide choice of timer and interval options to fine tune its behavior and performance. There are interactions and dependencies between some of these, and if misconfigured, they may cause race or failure conditions.

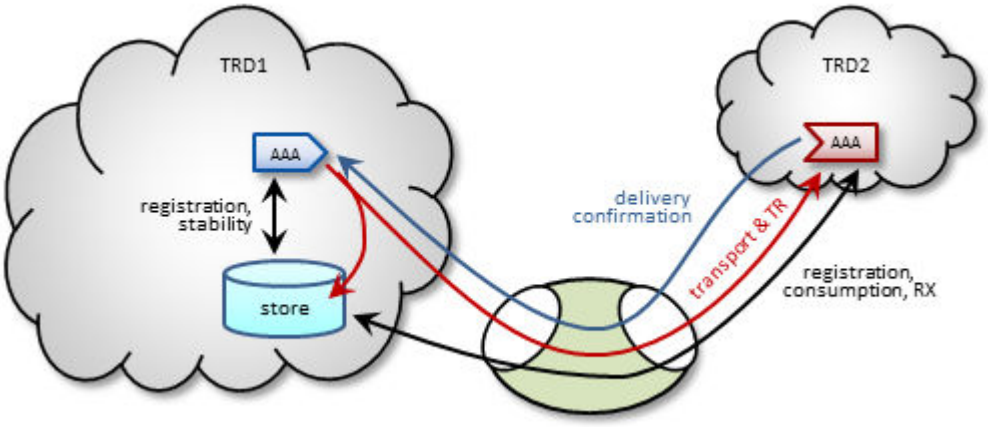
This manual's description of configuration options (see Chapter 5, “XML Configuration Reference” on page 46), includes identification of such relationships. Please heed them.

Multicast Immediate Messaging Considerations

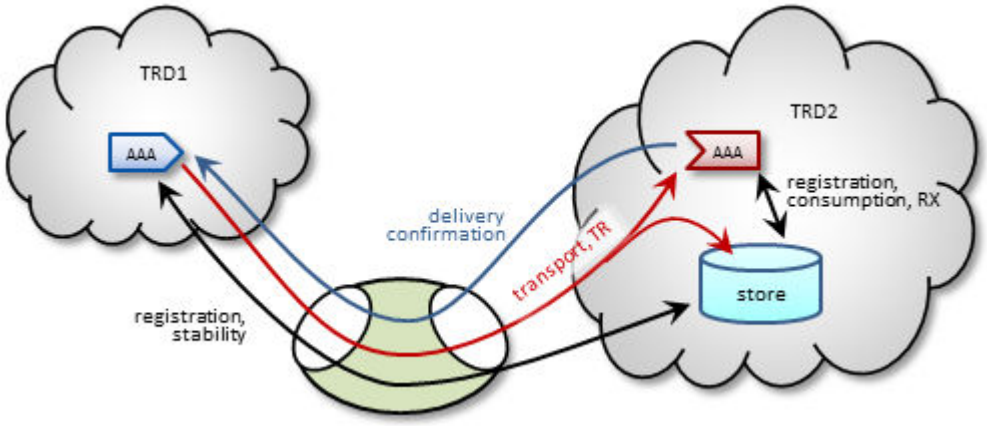
Multicast Immediate Messages (MIMs) may pass through the UM Router. You cannot filter MIMs with “Access Control Lists (ACL)” on page 21—MIMs are forwarded to all TRDs. Informatica does not recommend using MIM for messaging traffic across the UM Router. MIM is intended for short-lived topics and applications that cannot tolerate a delay between source creation and the sending of the first message. See also the *UM Concepts* document, *Multicast Immediate Messaging*.

Persistence Over the UM Router

The UM Router supports UMP persistence by routing all necessary control and retransmission channels along with transport and topic resolution traffic. A typical implementation places the UMP persistent store in the same TRD as its registered source, as shown in the following figure.



The UM Router also supports UMP implementations with the store located in a receiver's TRD, as shown in the following figure.



Note: For more reliable operation when using UMP with UM Routers, Informatica recommends enabling OTR.

UM JMS and the UM Router

The UM Router transparently supports endpoint applications that use UM JMS, Publish/Subscribe model. For durable subscriptions (i.e., when UMP is used with UM JMS), we recommend configuring a UM JMS store list by using the `ume_store_name` configuration option to connect to stores. This makes configuration files more flexible because a specific IP:port is not required. This is the default connection method that our sample configuration files use.

The UM Router does not currently support the UM JMS Point-to-Point model, because it relies on UMQ (which the UM Router also does not currently support).

Support for Late Join and Off-Transport Recovery

The UM Router supports sources and receivers configured for Late Join and/or Off-Transport Recovery (OTR). Retransmission requests and subsequent retransmissions are conducted across the entire path through the UM Router network. A UM Router's proxy sources do not have Late-Join/OTR retention buffers and hence, are not able to provide recovered messages.

Topic Resolution Reliability

Topic resolution can sometimes remain in a quiescent phase due to link interruption, preventing needed re-subscription topic resolution activity. Two ways you can address this are:

- For isolated incidents, call `lbm_context_topic_resolution_request()` (see example `lbmtrreq.c`). This restarts the sustaining phase.
- For more chronic problems, such as a UM Router link (especially an endpoint link) over a WAN of questionable reliability, consider configuring Topic resolution to stay in the sustaining phase (option `resolver_*_minimum_sustain_duration`).

BOS and EOS Behavior Over the UM Router

Through a network of UM Routers, a topic traverses a separate session for each link along its path. Thus, the UM Router reports BOS/EOSs based on the activity between the proxy source transport and its associated receiver. There is no end-to-end, application-to-application reporting of the data path state. Also, in the case of multiple topics being assigned to multiple sessions, topics may find themselves with different session

mates from hop to hop. Of course, this all influences when, and for which transport session, a topic's BOSs and EOSs are issued.

Topology Configuration Examples

Following are example configurations for a variety of UM Router topologies. These are the topology examples presented in Chapter 3.

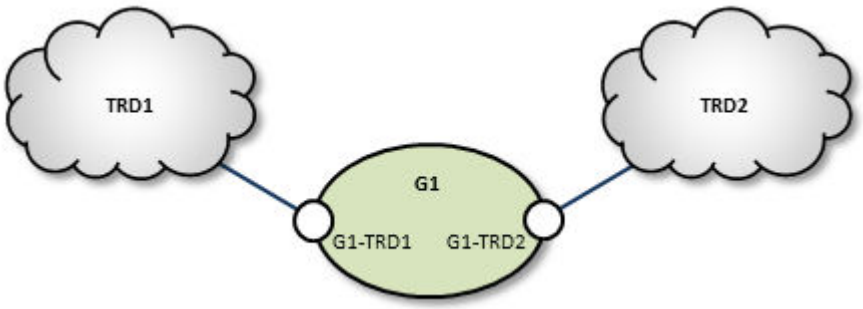
In a real-world situation, you would have UM Router XML configuration files with their portal interfaces referencing complete UM configuration files. However, for these examples, the referred domain configuration files are simplified to contain only information relevant to the applicable UM Router. As part of this simplification, domain configuration files show interfaces for only one or two transport types.

Also, IP addresses are provided in some cases and omitted in other cases. This is because peer portals need to know companion IP addresses (and port numbers) to establish connections, whereas endpoint portals communicate via topic resolution and thus, do not need to know IP addresses.

Note: Before designing any UM Router implementations based on configurations or examples other than the types presented in this document, please contact your technical support representative.

Direct Link

This example uses a UM Router to connect two topic resolution domain LANs.



TRD1 Configuration

This UM configuration file, trd1.cfg, describes TRD1 and is referenced in the UM Router configuration file.

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.1.37.85
```

G1 Configuration

This UM Router configuration file defines two endpoint portals. In the daemon section, we have turned on monitoring for the all endpoint portals in the UM Router. The configuration specifies that all statistics be collected every 5 seconds and uses the lbm transport module to send statistics to your monitoring application, which runs in TRD1. See also *UM Concepts, Monitoring UMS*. The Web Monitor has also been turned on (port 15304) to monitor the performance of the UM Router.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!—G1 xml file- 2 endpoint portals -->
<tnw-gateway version="1.0">
  <daemon>
```

```
    <log type="console"/>
    <lbm-license-file>lic0014.txt</lbm-license-file>
    <monitor interval="5">
      <transport-module module="lbm" options="config=trd1.cfg"/>
    </monitor>
    <web-monitor>*:15304</web-monitor>
  </daemon>
  <portals>
    <endpoint>
      <name>G1-TRD1</name>
      <domain-id>1</domain-id>
      <lbm-config>trd1.cfg</lbm-config>
    </endpoint>
    <endpoint>
      <name>G1-TRD2</name>
      <domain-id>2</domain-id>
      <lbm-config>trd2.cfg</lbm-config>
    </endpoint>
  </portals>
</tnw-gateway>
```

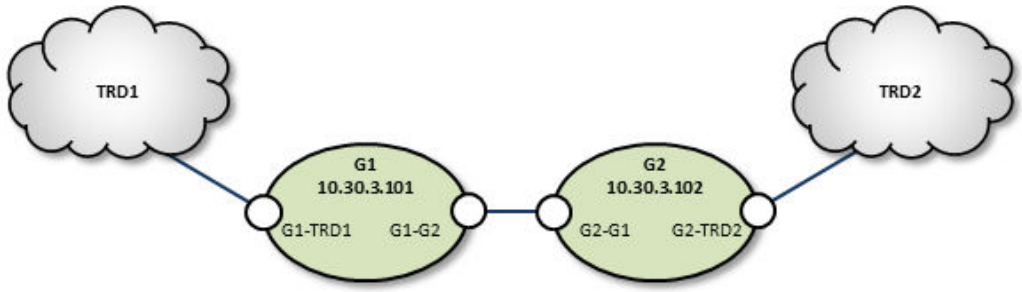
TRD2 Configuration

The configuration file `trd2.cfg` could look something like this.

```
# Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.2.37.85
```

Single Link, Peer to Peer

In cases where the UM Router connection between two TRDs must tunnel through a WAN or TCP/IP network, you can implement a UM Router at each end, as shown in the example below.



TRD1 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.1.37.85
```

G1 Configuration

Following is an example of two companion peer portals (on different UM Routers) configured via UM Router XML configuration file for a single TCP setup. Note that one must be an initiator and the other, an acceptor.

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
       <log type="console"/>
    </daemon>
    <portals>
```

```
        <endpoint>
           <name>G1-TRD1</name>
           <domain-id>1</domain-id>
           <lbm-config>TRD1.cfg</lbm-config>
        </endpoint>
        <peer>
           <name>G1-G2</name>
           <single-tcp>
              <interface>10.30.3.100</interface>
              <initiator>
                 <address>10.30.3.102</address>
                 <port>26123</port>
              </initiator>
           </single-tcp>
        </peer>
    </portals>
</tnw-gateway>
```

G2 Configuration

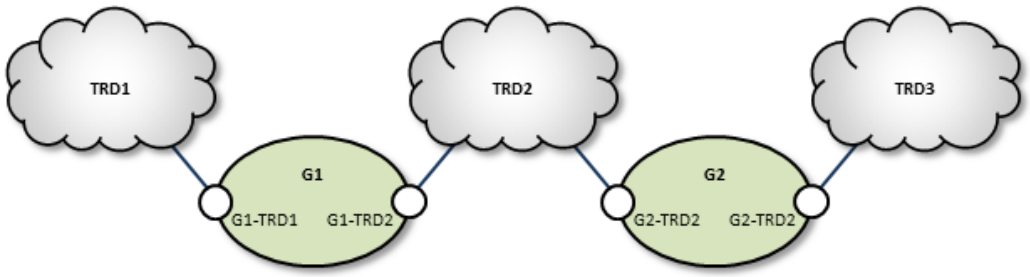
```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G2-G1</name>
         <single-tcp>
           <interface>10.30.3.102</interface>
           <acceptor>
             <listen-port>26123</listen-port>
           </acceptor>
         </single-tcp>
      </peer>
      <endpoint>
         <name>G2-TRD2</name>
         <domain-id>2</domain-id>
         <lbm-config>TRD2.cfg</lbm-config>
      </endpoint>
    </portals>
</tnw-gateway>
```

TRD2 Configuration

```
## LAN2 Configuration Options ##
context request_tcp_interface 10.33.3.0/24
context resolver_multicast_port 13965
```

Single Link with Intermediate LAN

This example, like the previous one, configures two localized UM Routers tunneling a connection between two TRDs, however, the UM Routers in this example are tunneling through an intermediate TRD. This has the added effect of connecting three TRDs.



TRD1 Configuration

```
## TRD1 Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.2.37.85
```

G1 Configuration

Following is an example of two companion peer portals (on different UM Routers) configured via UM Router XML configuration file for a single TCP setup. Note that one must be an initiator and the other, an acceptor.

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <endpoint>
         <name>G1-TRD1</name>
         <domain-id>1</domain-id>
         <lbm-config>TRD1.cfg</lbm-config>
      </endpoint>
      <endpoint>
         <name>G1-TRD2</name>
         <domain-id>2</domain-id>
         <lbm-config>TRD2.cfg</lbm-config>
      </endpoint>
    </portals>
</tnw-gateway>
```

TRD2 Configuration

```
## TRD2 Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.2.37.85
```

G2 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <endpoint>
         <name>G2-TRD2</name>
         <domain-id>2</domain-id>
         <lbm-config>TRD2.cfg</lbm-config>
      </endpoint>
      <endpoint>
         <name>G2-TRD3</name>
         <domain-id>3</domain-id>
         <lbm-config>TRD3.cfg</lbm-config>
      </endpoint>
    </portals>
</tnw-gateway>
```

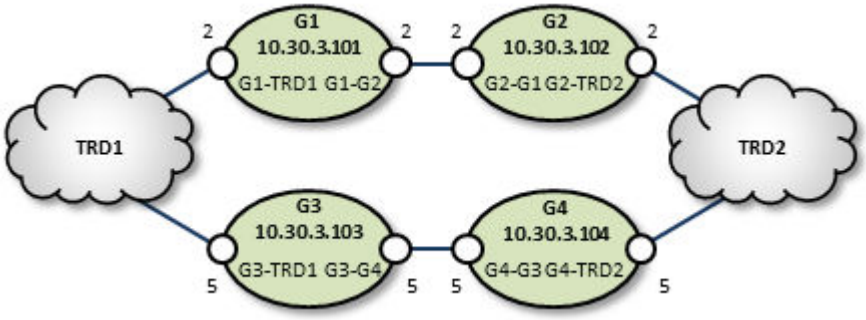
TRD3 Configuration

```
## TRD3 Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.3.37.85
```

Parallel Links

This example is similar in purpose to the single link, peer-to-peer example, except that a second pair of UM Routers is added as a backup route. You can set one of these as a secondary route by assigning a higher cost to portals along the path. In this case we set G3 and G4's portal costs to 5, forcing the lower route to be selected only if the upper (G1, G2) route fails.

Also note that we have configured the peer portals for the leftmost or odd-numbered UM Routers as initiators, and the rightmost or even-numbered UM Router peers as acceptors.



TRD1 Configuration

```
## TRD1 Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.1.37.85
```

G1 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <endpoint>
         <name>G1-TRD1</name>
         <domain-id>1</domain-id>
         <cost>2</cost>
         <lbm-config>TRD1.cfg</lbm-config>
      </endpoint>
      <peer>
         <name>G1-G2</name>
         <cost>2</cost>
         <single-tcp>
            <interface>10.30.3.101</interface>
            <initiator>
               <address>10.30.3.102</address>
               <port>23745</port>
            </initiator>
         </single-tcp>
      </peer>
    </portals>
</tnw-gateway>
```

G2 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
```

```
            <name>G2-G1</name>
            <cost>2</cost>
            <single-tcp>
               <interface>10.30.3.102</interface>
               <acceptor>
                  <listen-port>23745</listen-port>
               </acceptor>
            </single-tcp>
         </peer>
         <endpoint>
            <name>G2-TRD2</name>
            <domain-id>2</domain-id>
            <cost>2</cost>
            <lbm-config>TRD2.cfg</lbm-config>
         </endpoint>
      </portals>
   </tnw-gateway>
```

G3 Configuration

```
   <?xml version="1.0" encoding="UTF-8" ?>
   <tnw-gateway version="1.0">
       <daemon>
           <log type="console"/>
       </daemon>
       <portals>
         <endpoint>
            <name>G3-TRD1</name>
            <domain-id>1</domain-id>
            <cost>5</cost>
            <lbm-config>TRD1.cfg</lbm-config>
         </endpoint>
         <peer>
            <name>G3-G4</name>
            <cost>5</cost>
            <single-tcp>
               <interface>10.30.3.103</interface>
               <initiator>
                  <address>10.30.3.104</address>
                  <port>23746</port>
               </initiator>
            </single-tcp>
         </peer>

       </portals>
   </tnw-gateway>
```

G4 Configuration

```
   <?xml version="1.0" encoding="UTF-8" ?>
   <tnw-gateway version="1.0">
       <daemon>
           <log type="console"/>
       </daemon>
       <portals>
         <peer>
            <name>G4-G3</name>
            <cost>5</cost>
            <single-tcp>
               <interface>10.30.3.104</interface>
               <acceptor>
                  <listen-port>23746</listen-port>
               </acceptor>
            </single-tcp>
         </peer>
         <endpoint>
            <name>G4-TRD2</name>
            <domain-id>2</domain-id>
            <cost>5</cost>
            <lbm-config>TRD2.cfg</lbm-config>
         </endpoint>
```

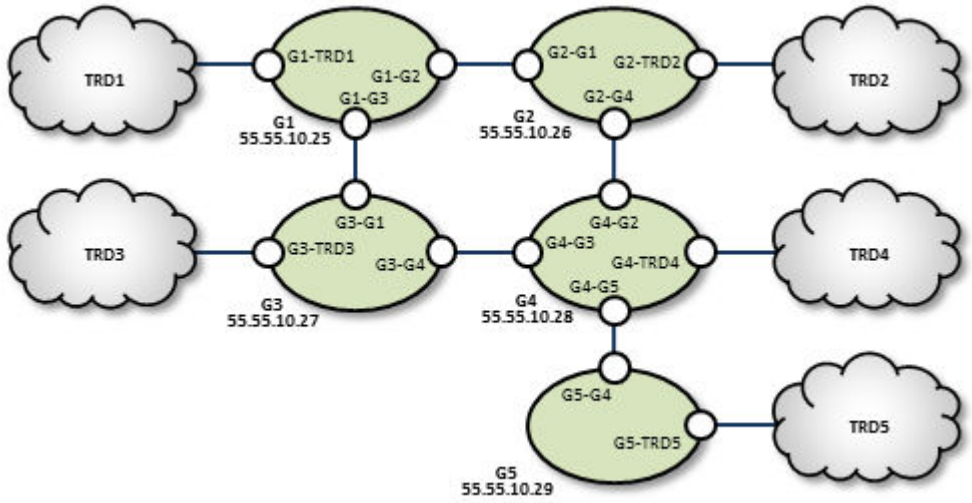
```
    </portals>
</tnw-gateway>
```

TRD2 Configuration

```
## TRD2 Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.2.37.85
```

Loop and Spur

Figure 1.



TRD1 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.1.37.85
```

G1 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G1_to_G3</name>
         <tcp>
            <listen-port>23587</listen-port>
            <companion>
               <address>55.55.10.27</address>
               <port>23801</port>
            </companion>
         </tcp>
      </peer>
      <peer>
         <name>G1_to_G2</name>
         <tcp>
            <listen-port>23967</listen-port>
            <companion>
               <address>55.55.10.26</address>
```

```
            <port>23745</port>
         </companion>
      </tcp>
    </peer>
    <endpoint>
       <name>G1_to_TRD1</name>
       <domain-id>1</domain-id>
       <lbm-config>TRD1.cfg</lbm-config>
    </endpoint>
  </portals>
</tnw-gateway>
```

G2 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G2_to_G4</name>
         <tcp>
            <listen-port>23709</listen-port>
            <companion>
               <address>55.55.10.28</address>
               <port>23632</port>
            </companion>
         </tcp>
      </peer>
      <peer>
         <name>G2_to_G1</name>
         <tcp>
            <listen-port>23745</listen-port>
            <companion>
               <address>55.55.10.25</address>
               <port>23967</port>
            </companion>
         </tcp>
      </peer>
      <endpoint>
         <name>G2_to_TRD2</name>
         <domain-id>2</domain-id>
         <lbm-config>TRD2.cfg</lbm-config>
      </endpoint>
    </portals>
</tnw-gateway>
```

TRD2 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.2.37.85
```

TRD3 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.3.37.85
```

G3 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
```

```
      <peer>
         <name>G3_to_G4</name>
         <tcp>
            <listen-port>23804</listen-port>
            <companion>
               <address>55.55.10.28</address>
               <port>23754</port>
            </companion>
         </tcp>
      </peer>
      <peer>
         <name>G3_to_G1</name>
         <tcp>
            <listen-port>23801</listen-port>
            <companion>
               <address>55.55.10.25</address>
               <port>23587</port>
            </companion>
         </tcp>
      </peer>
      <endpoint>
         <name>G3_to_TRD3</name>
         <domain-id>3</domain-id>
         <lbm-config>TRD3.cfg</lbm-config>
      </endpoint>
    </portals>
</tnw-gateway>
```

G4 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G4_to_G3</name>
         <tcp>
            <listen-port>23754</listen-port>
            <companion>
               <address>55.55.10.27</address>
               <port>23804</port>
            </companion>
         </tcp>
      </peer>
      <endpoint>
         <name>G4_to_TRD4</name>
         <domain-id>4</domain-id>
         <lbm-config>TRD4.cfg</lbm-config>
      </endpoint>
      <peer>
         <name>G4_to_G2</name>
         <tcp>
            <listen-port>23632</listen-port>
            <companion>
               <address>55.55.10.26</address>
               <port>23709</port>
            </companion>
         </tcp>
      </peer>
      <peer>
         <name>G4_to_G5</name>
         <tcp>
            <listen-port>23909</listen-port>
            <companion>
               <address>55.55.10.29</address>
               <port>23739</port>
            </companion>
         </tcp>
      </peer>
```

```
    </portals>
</tnw-gateway>
```

TRD4 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.4.37.85
```

G5 Configuration

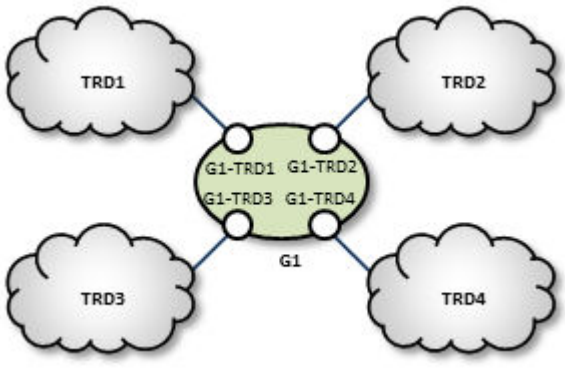
```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <endpoint>
         <name>G5_to_TRD5</name>
         <domain-id>5</domain-id>
         <lbm-config>TRD5.cfg</lbm-config>
      </endpoint>
      <peer>
         <name>G5_to_G4</name>
         <tcp>
            <listen-port>23739</listen-port>
            <companion>
               <address>55.55.10.28</address>
               <port>23909</port>
            </companion>
         </tcp>
      </peer>
    </portals>
</tnw-gateway>
```

TRD5 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.5.37.85
```

Star with Centralized UM Router

This network consists of four TRDs. Within each TRD, full multicast connectivity exists. However, no multicast connectivity exists between the four TRDs.



G1 Configuration

The configuration for this UM Router also has transport statistics monitoring and the WebMonitor turned on.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- UM GW xml file- 3 endpoint portals -->
<tnw-gateway version="1.0">
  <daemon>
    <log type="console"/>
    <lbm-license-file>lic0014.txt</lbm-license-file>
    <monitor interval="5">
      <transport-module module="lbm" options="config=trd1.cfg"/>
    </monitor>
    <web-monitor>*:15304</web-monitor>
  </daemon>
  <portals>
    <endpoint>
      <name>G1_to_TRD1</name>
      <domain-id>1</domain-id>
      <lbm-config>trd1.cfg</lbm-config>
    </endpoint>
    <endpoint>
      <name>G1_to_TRD2</name>
      <domain-id>2</domain-id>
      <lbm-config>trd2.cfg</lbm-config>
    </endpoint>
    <endpoint>
      <name>G1_to_TRD3</name>
      <domain-id>3</domain-id>
      <lbm-config>trd3.cfg</lbm-config>
    </endpoint>
    <endpoint>
      <name>G1_to_TRD4</name>
      <domain-id>4</domain-id>
      <lbm-config>trd4.cfg</lbm-config>
    </endpoint>
  </portals>
</tnw-gateway>
```

TRD1 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.1.37.85
```

TRD2 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.2.37.85
```

TRD3 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.3.37.85
```

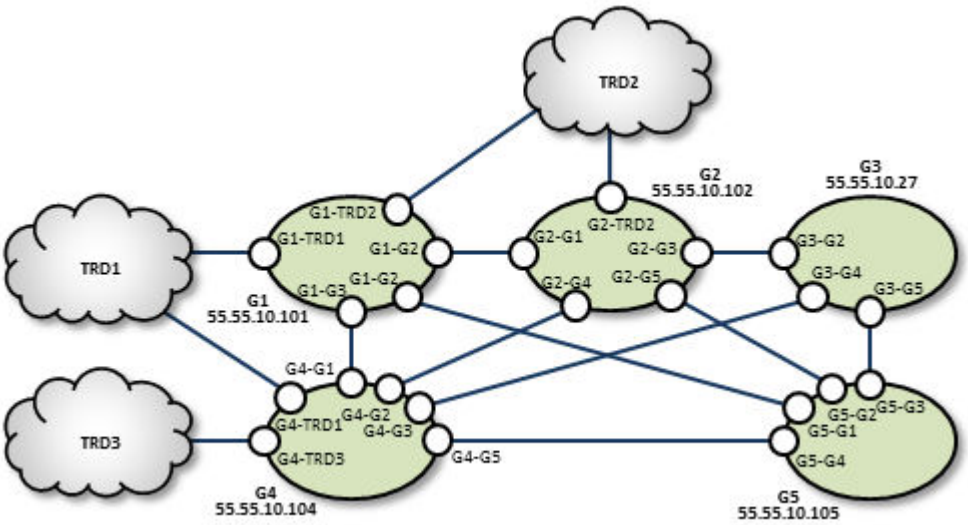
TRD4 Configuration

```
## Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.4.37.85
```

Mesh

The mesh topology utilizes many connections between many nodes, to provide a variety of alternate routes. However, meshes are not the best solution in many cases, as unneeded complexity can increase the chance for configuration errors or make it more difficult to trace problems.

In this example, the peer portal connections are using dual TCP configurations.



TRD1 Configuration

```
### Global Configuration Options ##
context request_tcp_interface 10.29.3.0/24
context resolver_multicast_port 13965
context resolver_multicast_interface 10.29.3.0/24
context resolver_multicast_address 225.1.37.85
```

G1 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
         <name>G1_to_G5</name>
         <tcp>
            <listen-port>23805</listen-port>
            <companion>
               <address>55.55.10.105</address>
               <port>23880</port>
            </companion>
         </tcp>
      </peer>
      <peer>
         <name>G1_to_G4</name>
         <tcp>
            <listen-port>23587</listen-port>
            <companion>
               <address>55.55.10.104</address>
               <port>23801</port>
            </companion>
         </tcp>
      </peer>
      <endpoint>
         <name>G1_to_TRD1</name>
         <domain-id>1</domain-id>
```

```
            <lbm-config>TRD1.cfg</lbm-config>
        </endpoint>
        <endpoint>
            <name>G1_to_TRD2</name>
            <domain-id>2</domain-id>
            <lbm-config>TRD2.cfg</lbm-config>
        </endpoint>
        <peer>
            <name>G1_to_G2</name>
            <tcp>
                <listen-port>23967</listen-port>
                <companion>
                    <address>55.55.10.102</address>
                    <port>23745</port>
                </companion>
            </tcp>
        </peer>
    </portals>
</tnw-gateway>
```

G2 Configuration

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
    <daemon>
        <log type="console"/>
    </daemon>
    <portals>
      <peer>
        <name>G2_to_G5</name>
        <tcp>
            <listen-port>23986</listen-port>
            <companion>
                <address>55.55.10.105</address>
                <port>23608</port>
            </companion>
        </tcp>
      </peer>
      <peer>
        <name>G2_to_G4</name>
        <tcp>
            <listen-port>23831</listen-port>
            <companion>
                <address>55.55.10.104</address>
                <port>23803</port>
            </companion>
        </tcp>
      </peer>
      <peer>
        <name>G2_to_G1</name>
        <tcp>
            <listen-port>23745</listen-port>
            <companion>
                <address>55.55.10.101</address>
                <port>23967</port>
            </companion>
        </tcp>
      </peer>
      <peer>
        <name>G2_to_G3</name>
        <tcp>
            <listen-port>23709</listen-port>
            <companion>
                <address>55.55.10.103</address>
                <port>23632</port>
            </companion>
        </tcp>
      </peer>
      <endpoint>
        <name>G2_to_TRD2</name>
        <domain-id>2</domain-id>
```

```
            <lbm-config>TRD2.cfg</lbm-config>
          </endpoint>
        </portals>
    </tnw-gateway>
```

G3 Configuration

```
    <?xml version="1.0" encoding="UTF-8" ?>
    <tnw-gateway version="1.0">
        <daemon>
            <log type="console"/>
        </daemon>
        <portals>
          <peer>
             <name>G3_to_G5</name>
             <tcp>
                <listen-port>23909</listen-port>
                <companion>
                   <address>55.55.10.105</address>
                   <port>23739</port>
                </companion>
             </tcp>
          </peer>
          <peer>
             <name>G3_to_G4</name>
             <tcp>
                <listen-port>23754</listen-port>
                <companion>
                   <address>55.55.10.104</address>
                   <port>23804</port>
                </companion>
             </tcp>
          </peer>
          <peer>
             <name>G3_to_G2</name>
             <tcp>
                <listen-port>23632</listen-port>
                <companion>
                   <address>55.55.10.102</address>
                   <port>23709</port>
                </companion>
             </tcp>
          </peer>
        </portals>
    </tnw-gateway>
```

TRD2 Configuration

```
    ## Global Configuration Options ##
    context request_tcp_interface 10.29.3.0/24
    context resolver_multicast_port 13965
    context resolver_multicast_interface 10.29.3.0/24
    context resolver_multicast_address 225.2.37.85
```

TRD3 Configuration

```
    ## Global Configuration Options ##
    context request_tcp_interface 10.29.3.0/24
    context resolver_multicast_port 13965
    context resolver_multicast_interface 10.29.3.0/24
    context resolver_multicast_address 225.3.37.85
```

G4 Configuration

```
    <?xml version="1.0" encoding="UTF-8" ?>
    <tnw-gateway version="1.0">
        <daemon>
            <log type="console"/>
        </daemon>
        <portals>
          <peer>
             <name>G4_to_G5</name>
```

```
            <tcp>
               <listen-port>23617</listen-port>
               <companion>
                  <address>55.55.10.105</address>
                  <port>23580</port>
               </companion>
            </tcp>
         </peer>
         <endpoint>
            <name>G4_to_TRD1</name>
            <domain-id>1</domain-id>
            <lbm-config>TRD1.cfg</lbm-config>
         </endpoint>
         <endpoint>
            <name>G4_to_TRD3</name>
            <domain-id>3</domain-id>
            <lbm-config>TRD3.cfg</lbm-config>
         </endpoint>
         <peer>
            <name>G4_to_G1</name>
            <tcp>
               <listen-port>23801</listen-port>
               <companion>
                  <address>55.55.10.101</address>
                  <port>23587</port>
               </companion>
            </tcp>
         </peer>
         <peer>
            <name>G4_to_G3</name>
            <tcp>
               <listen-port>23804</listen-port>
               <companion>
                  <address>55.55.10.103</address>
                  <port>23754</port>
               </companion>
            </tcp>
         </peer>
         <peer>
            <name>G4_to_G2</name>
            <tcp>
               <listen-port>23803</listen-port>
               <companion>
                  <address>55.55.10.102</address>
                  <port>23831</port>
               </companion>
            </tcp>
         </peer>
      </portals>
   </tnw-gateway>
```

G5 Configuration

```
   <?xml version="1.0" encoding="UTF-8" ?>
   <tnw-gateway version="1.0">
       <daemon>
           <log type="console"/>
       </daemon>
       <portals>
         <peer>
            <name>G5_to_G4</name>
            <tcp>
               <listen-port>23580</listen-port>
               <companion>
                  <address>55.55.10.104</address>
                  <port>23617</port>
               </companion>
            </tcp>
         </peer>
         <peer>
            <name>G5_to_G1</name>
```

```
            <tcp>
               <listen-port>23880</listen-port>
               <companion>
                  <address>55.55.10.101</address>
                  <port>23805</port>
               </companion>
            </tcp>
         </peer>
         <peer>
            <name>G5_to_G3</name>
            <tcp>
               <listen-port>23739</listen-port>
               <companion>
                  <address>55.55.10.103</address>
                  <port>23909</port>
               </companion>
            </tcp>
         </peer>
         <peer>
            <name>G5_to_G2</name>
            <tcp>
               <listen-port>23608</listen-port>
               <companion>
                  <address>55.55.10.102</address>
                  <port>23986</port>
               </companion>
            </tcp>
         </peer>
      </portals>
   </tnw-gateway>
```

Using UM Configuration Files with the UM Router

Within the UM Router configuration file, the endpoint portal's `<lbm-config>` element lets you import configurations from either a plain text or XML UM configuration file. However, using the XML type of UM configuration files provides the following advantages over plain text UM configuration files:

- You can apply UM attributes per topic and/or per context.
- You can apply attributes to all portals on a particular UM Router using a UM XML template (instead of individual portal settings).
- Using UM XML templates to set options for individual portals lets the UM Router process these settings in the `<daemon>` element instead of within each portal's configuration.

Note: You must have each UM Router endpoint's `request_tcp_interface` option set when the UM Router starts, thus, you must set it in each `<endpoint>` element of the UM Router XML configuration file and not in a UM XML configuration file, the latter of which is not implemented until the UM Router creates proxy sources and receivers.

```
<endpoint>
  <name>G1-TRD1</name>
  <domain-id>1</domain-id>
  <lbm-attributes>
    <option name="request_tcp_interface" scope="context" value="10.29.3.0/24"/>
  </lbm-attributes>
</endpoint>
```

Setting Individual Endpoint Options

When setting endpoint options, first name the context of each endpoint in the UM Router's XML configuration file.

```
<portals>
  <endpoint>
    <name>Endpoint_1</name>
    <domain-id>1</domain-id>
    <source-context-name>G1_E1</source-context-name>
    <lbm-attributes>
      <option name="request_tcp_interface" scope="context" value="10.29.4.0/24"/>
    </lbm-attributes>
  </endpoint>
  <endpoint>
    <name>G1-TRD2</name>
    <domain-id>2</domain-id>
    <receiver-context-name>G1_E2</source-context-name>
    <lbm-attributes>
      <option name="request_tcp_interface" scope="context" value="10.29.5.0/24" />
    </lbm-attributes>
  </endpoint>
</portals>
```

Then assign configuration templates to those contexts in the UM XML configuration file.

```
<application name="tnwgd" template="global">
  <contexts>
    <context name="G1_E1" template="G1-E1-options">
      <sources />
    </context>
    <context name="G1_E2" template="G1-E2-options">
      <sources />
    </context>
  </contexts>
</application>
```

You specify the unique options for each of this UM Router's two endpoints in the UM XML configuration `<templates>` section used for `G1-E1-options` and `G1-E2-options`.

UM Router and UM XML Configuration Use Cases

One advantage of using UM XML configuration files with the UM Router is the ability to assign unique UM attributes to the topics and contexts used for the proxy sources and receivers (which plain text UM configuration files cannot do). The following example shows how to assign a different LBTRM multicast address to a source based on its topic.

Create a new UM XML configuration template for the desired topic name.

```
<template name="AAA-template">
  <options type="source">
    <option name="transport_lbtrm_multicast_address"
    default-value="225.2.37.88"/>
  </options>
</template>
```

Then include this template in the `<application>` element associated with the UM Router.

```
<application name="tnwgd" template="global-options">
  <contexts>
    <context>
      <sources template="source-options">
```

```
        <topic topicname="AAA" template="AAA-template" />
      </sources>
    </context>
  </contexts>
</application>
```

It is also possible to assign UM attributes directly in the `<application>` tag. For example, the following specifies that a particular topic should use an LBT-RU transport.

```
<application name="tnwgd" template="tnwgd-common">
  <contexts>
    <context>
      <sources template="source-template">
        <topic topicname="LBTRU_TOPIC">
          <options type="source">
            <option name="transport" default-value="lbtru" />
            </options>
        </topic>
      </sources>
    </context>
  </contexts>
</application>
```

Sample Configuration

The following sample configuration incorporates many of the examples mentioned above. The UM Router applies options to all UM objects created. The UM XML configuration file overwrites these options for two specific topics. The first topic, LBTRM_TOPIC, uses a different template to change its transport from TCP to LBTRM, and to set an additional property. The second topic, LBTRU_TOPIC, also changes its transport from TCP to a new value. However, its new attributes are applied directly in its associated topic tag, instead of referencing a template. In addition, this sample configuration assigns the `rm-source` template to all sources and receivers associated with the context `endpt_1`.

XML UM Configuration File

```
<?xml version="1.0" encoding="UTF-8" ?>
<um-configuration version="1.0">
  <templates>
    <template name="tnwgd-common">
      <options type="source">
        <option name="transport" default-value="tcp" />
      </options>
      <options type="context">
        <option name="request_tcp_interface" default-value="10.29.5.6" />
        <option name="transport_tcp_port_low" default-value="4400" />
        <option name="transport_tcp_port_high" default-value="4500" />
        <option name="resolver_multicast_address" default-value="225.2.37.88"/>
      </options>
    </template>
    <template name="rm-source">
      <options type="source">
        <option name="transport" default-value="lbtrm" />
        <option name="transport_lbtrm_multicast_address" default-value="225.2.37.89"/>
      </options>
    </template>
  </templates>
  <applications>
    <application name="tnwgd" template="tnwgd-common">
     <contexts>
        <context>
          <sources>
            <topic topicname="LBTRM_TOPIC" template="rm-source" />
            <topic topicname="LBTRU_TOPIC">
```

```
              <options type="source">
                <option name="transport" default-value="lbtru" />
                <option name="resolver_unicast_daemon" default-value="10.29.5.1:1234" />
              </options>
            </topic>
          </sources>
        </context>
        <context name="endpt_1">
          <sources template="rm-source"/>
        </context>
      </contexts>
    </application>
  </applications>
</um-configuration>
```

XML UM Router Configuration File

This UM Router uses the above XML UM configuration file, `sample-config.xml`, to set its UM options. It has three endpoints, one of which has the context `endpt_1`.

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
  <daemon>
    <log type="console"/>
    <xml-config>sample-config.xml</xml-config>
  </daemon>
  <portals>
    <endpoint>
      <name>Endpoint_1</name>
      <domain-id>1</domain-id>
      <lbm-attributes>
        <option name="context_name" scope="context" value="endpt_1" />
        <option name="request_tcp_interface" scope="context"
        value="10.29.4.0/24"/>
      </lbm-attributes>
    </endpoint>
    <endpoint>
      <name>Endpoint_2</name>
      <domain-id>2</domain-id>
      <lbm-attributes>
        <option name="request_tcp_interface" scope="context"
        value="10.29.5.0/24"/>
      </lbm-attributes>
    </endpoint>
    <endpoint>
      <name>Endpoint_3</name>
      <domain-id>3</domain-id>
      <lbm-attributes>
        <option name="request_tcp_interface" scope="context"
        value="10.29.6.0/24"/>
      </lbm-attributes>
    </endpoint>
  </portals>
</tnw-gateway>
```

Running the UM Router Daemon

To run the UM Router, ensure the following:

- Library environment variable paths are set correctly (LD_LIBRARY_PATH)
- The license environment variable LBM_LICENSE_FILENAME points to a valid UM Router license file.
- The configuration file is error free.

Typically, you run the UM Router with one configuration file argument, for example:

```
tnwgd gw1-config.xml
```

The UM Router logs version information on startup. The following is an example of this information:

```
Version 6.0 Build: Sep 26 2012, 00:31:33 (UMS 6.0 [UMP-6.0] [UMQ-6.0] [64-bit] Build:
Sep 26 2012, 00:27:17 ( DEBUG license LBT-RM LBT-RU LBT-IPC LBT-RDMA ) WC[PCRE 7.4
2007-09-21, regex, appcb] HRT[gettimeofday()])
```

tnwgd

Name

tnwgd -- UM Router Daemon

Synopsis

```
tnwgd [-d] [--dump-dtd] [-h] [--help] [-f] [--detach] [-v] [--validate] configfile
```

Description

UM Router services are provided by tnwgd. A UM Router configuration file is required. The contents and format of the configuration file are documented separately.

The DTD used to validate a configuration file is dumped to standard output with -d or --dump-dtd. After dumping the DTD, tnwgd exits instead of providing UM Router services as usual.

To validate the configuration file, use either the -v or --validate options. After attempting validation, tnwgd exits instead of providing UM Router services as usual. The exit status will be 0 for a configuration file successfully validated by the DTD, and non-zero otherwise.

tnwgd normally remains attached to the controlling terminal and runs until interrupted. If the -f or --detach options are given, tnwgd instead forks, detaches the child from the controlling terminal, and the parent exits immediately.

Command line help is available with -h or --help.

Exit Status

The exit status from tnwgd is 0 for success and some non-zero value for failure.

CHAPTER 5

XML Configuration Reference

This chapter includes the following topics:

XML Configuration Reference Overview

For controlling/configuring each UM Router, you use a XML UM Router configuration file, which also contains references to UM configuration files to extract needed information about the TRDs interfaced by endpoint portals. This chapter includes a lookup reference for the XML UM Router configuration file's elements and DTD.

An XML UM Router configuration file follows standard XML conventions. Element declarations or a pointer to a DTD file are not needed, as these are handled by the UM Router.

File Structure

An XML UM Router configuration file generally comprises two primary elements: “<daemon>” on page 49 and “<portals>” on page 59. Organized and contained within these are option value assignments. `<daemon>` sub-containers let you set options global to the UM Router. Of course, `<portals>` sub-containers let you configure each portal in the UM Router individually.

XML UM Router configuration files use the high-level structure shown in the following example. This example includes only some container elements, and only some options.

```
<?xml version="1.0" encoding="UTF-8" ?>
<tnw-gateway version="1.0">
  <daemon>
    <log type="console"/>
    <uid>0</uid>
    <gid>0</gid>
    <pidfile>/path/file.pid</pidfile>
    <lbm-license-file>/path/file.lic</lbm-license-file>
    <topicmap/>
    <patternmap/>
    <monitor>
      <transport-module/>
```

```
    <format-module/>
  </monitor>
  <web-monitor>*:21000</web-monitor>
  <propagation-delay/>
  <xml-config>sample-config.xml</xml-config>
</daemon>
<portals>
  <endpoint>
    <name>Endpoint_1</name>
    <domain-id>1</domain-id>
    <cost>1</cost>
    <lbm-config>endpoint2.cfg</lbm-config>
    <lbm-attributes>
      <option name="context_name" scope="context" value="endpt_1" />
    </lbm-attributes>
    <acl>
      <inbound>
        <ace match="accept">
          <topic>ABC123</topic>
          <pcre-pattern >pattern</pcre-pattern >
          <regex-pattern >pattern</regex-pattern >
          <transport/>
          <source-ip/>
          <multicast-group/>
          <udp-source-port/>
          <udp-destination-port/>
          <tcp-source-port/>
          <xport-id/>
        </ace>
      </inbound>
      <outbound>
        <ace match="accept">
          <topic>ABC123</topic>
          <pcre-pattern >pattern</pcre-pattern >
          <regex-pattern >pattern</regex-pattern >
          <transport/>
          <source-ip/>
          <multicast-group/>
          <udp-source-port/>
          <udp-destination-port/>
          <tcp-source-port/>
          <xport-id/>
        </ace>
      </outbound>
    </acl>
  </endpoint>
  <peer>
    <name>Peer_1</name>
    <cost>1</cost>
    <tcp>
      <interface>
      <listen-port>
      <receive-buffer>
      <send-buffer>
      <keepalive>
      <nodelay>
      <companion>
        <address>
        <port>
      </companion>
    </tcp>
    <single-tcp>
      <interface>
      <receive-buffer>
      <send-buffer>
      <keepalive>
      <nodelay>
      <initiator>
        <address>
        <port>
      </initiator>
```

```
        <acceptor>
          <listen-port>
        </acceptor>
      </single-tcp>
      <max-queue>
      <max-datagram>
      <batching>
        <min-length>
        <batch-interval>
      </batching>
      <lbm-config>peer1.cfg</lbm-config>
      <lbm-attributes>
        <option name="name" scope="scope" value="value" />
      </lbm-attributes>
      <acl> (see above)
      <topic-purge>
      <topic-interest-generate>
      <topic-domain-activity>
      <pattern-purge>
      <pattern-interest-generate>
      <pattern-domain-activity>
      <topic-use-check/>
      <pattern-use-check>
      <source-context-name>
      <receiver-context-name>
      <sqn-window>
      <context-query>
      <gateway-keepalive>
    </peer>
  </portals>
</tnw-gateway>
```

Elements Reference

Following are descriptions of the XML UM Router configuration file elements. For the children listings, + designates 1 or more, * designates 0 or more, and ? designates 0 or 1. You must insert children in the order presented.

<tnw-gateway>

The `<tnw-gateway>` element is a required container for all options residing in the XML UM Router configuration file. This is the top-level element.

Cardinality. 1

Parents. None.

Children. *<daemon>? <portals>*

XML Attributes:

XML Attribute	Description	Default Value
`version`	The version of the DTD, which is currently `1.0`. (This is not the product version.)	none

Example:

```
<tnw-gateway version="1.0">
  <daemon>
      . . .
      . . .
  </daemon>
  <portals>
    <endpoint>
      . . .
      . . .
    </endpoint>
  </portals>
</tnw-gateway>
```

<daemon>

The `<daemon>` element is a container for options common to the entire UM Router.

Cardinality. 0 or 1

Parents. *<tnw-gateway>*

Children. *<log>?, <uid>?, <gid>?, <pidfile>?, <lbm-license-file>?, <topicmap>?, <patternmap>?, <monitor>?, <web-monitor>?, <propagation-delay>?, <xml-config>?*

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
  <daemon>
      . . .
      . . .
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<name>

The `<name>` element lets you set a name for this UM Router (do not duplicate for any other known UM Routers), or for the name of an endpoint or peer portal. Each portal name must be unique within the UM Router.

Cardinality. 1

Parents. *<daemon> <endpoint> <peer>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <name>UM Router1</name>
  </daemon>
  <portals>
    <endpoint>
      <name>endpoint1</name>
      . . .
      . . .
    </endpoint>
  </portals>      . . .
```

```
</tnw-gateway>
```

<log>

The <log> element specifies the destination for UM Router log messages. If you set the type for file, use this element to contain the full pathname/filename.

Cardinality. 0 or 1

Parents. <*daemon*> None.

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
type	file, syslog, console	console
xml:space	How whitespace is handled. default trims leading and trailing whitespace (e.g., tabs, spaces, linefeeds, etc.), and compresses multiple whitespace characters into a single space character. preserve preserves the whitespace exactly as read.	default

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <log type=”syslog”/>
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<uid>

The <uid> element specifies a User ID (UID) for the daemon process (if run as root).

Cardinality. 0 or 1

Parents. <*daemon*>

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <uid>5555</uid>
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<gid>

The `<gid>` element specifies a Group ID (GID) for daemon process (if run as root).

Cardinality. 0 or 1

Parents. *<daemon>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <gid>1234</gid>
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<pidfile>

The `<pidfile>` element contains the pathname for daemon process ID (PID) file.

Cardinality. 0 or 1

Parents. *<daemon>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`xml:space`	How whitespace is handled. `default` trims leading and trailing whitespace (e.g., tabs, spaces, linefeeds, etc.), and compresses multiple whitespace characters into a single space character. `preserve` preserves the whitespace exactly as read.	`default`

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <pidfile>/configs/stores/umestored1/umestored.pid</pidfile>  . . .
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<lbm-license-file>

The `<lbm-license-file>` element specifies the UM license file's pathname/filename.

Cardinality. 0 or 1

Parents. *<daemon>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`xml:space`	How whitespace is handled. `default` trims leading and trailing whitespace (e.g., tabs, spaces, linefeeds, etc.), and compresses multiple whitespace characters into a single space character. `preserve` preserves the whitespace exactly as read.	`default`

Example:

```
<tnw-gateway version="1.0">
  <daemon>
      . . .
    <lbm-license-file>lic0014.txt</lbm-license-file>
      . . .
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<topicmap/>

The `<topicmap>` element specifies characteristics of the internal topic maps.

Cardinality. 0 or 1

Parents. *<daemon>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`hash-function`	Topic map hash function to use. Choices are `classic` – A "classic" good string hash function. Works best when topic names have a constant prefix with a changing suffix. `djb2` – The Dan Bernstein algorithm from comp.lang.c. Works best when topic names have a changing prefix with a constant suffix. `sdbm` – sdbm database library (used in Berkeley DB). A useful alternative to djb2. `murmur2` – Good all-around hash function by Austin Appleby. Best for medium to long topic strings.	`murmur2`
`size`	Number of entries in the topic map.	`131111`

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    < topicmap hash-function=”djb2” size=”5000”>
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<patternmap/>

The `<patternmap>` element determines characteristics of the internal pattern maps.

Cardinality. 0 or 1

Parents. *<daemon>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`hash-function`	Pattern map hash function to use. Choices are `classic` – A "classic" good string hash function. Works best when topic names have a constant prefix with a changing suffix. `djb2` – The Dan Bernstein algorithm from comp.lang.c. Works best when topic names have a changing prefix with a constant suffix. `sdbm` – sdbm database library (used in Berkeley DB). A useful alternative to djb2. `murmur2` – Good all-around hash function by Austin Appleby. Best for medium to long topic strings.	`murmur2`
`size`	Number of entries in the pattern map.	`131111`

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <patternmap hash-function=”djb2” size=”5000”>
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<monitor>

The `<monitor>` element is a container for monitoring configuration elements.

Cardinality. 0 or 1

Parents. *<daemon>*

Children. *<transport-module>?, <format-module>*

XML Attributes:

XML Attribute	Description	Default Value
`interval`	Monitoring interval, in seconds. `0` disables monitoring.	`0`

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <monitor interval=30>
      <transport-module module="lbm" options="config=/cfgs/TD1.cfg;topic=stats"/>
      <format-module options="config=/cfgs/TD1.cfg;separator=|"/>
```

```
    </monitor>
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<transport-module/>

The `<transport-module>` element specifies characteristics about the monitoring transport module used.

Cardinality. 0 or 1

Parents. *<monitor>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`module`	Specifies the monitoring transport module to use. Choices are `lbm` (LBMMON UMS), `lbmsnmp` (LBMMON SNMP), or `udp` (LBMMON UDP).	`lbm`
`options`	Option string to be passed to the transport module. Available options are `config` (configuration path and filename), `topic` (the topic name to use for sending and receiving statistics; default `/29west/statistics`), and `wctopic` (for monitor receivers only, a wildcard pattern).	none

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <monitor interval=30>
      <transport-module module="lbm" options="config=/cfgs/TD1.cfg;topic=stats"/>
      <format-module options="config=/cfgs/TD1.cfg;separator=|"/>
    </monitor>
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<format-module/>

The `<format-module>` element provides specifics about the monitoring format module.

Cardinality. 0 or 1

Parents. *<monitor>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`module`	Currently, the one available choice is `csv` (comma-separated variables).	`csv`
`options`	Option string to be passed to the format module. The one available option is `separator` (field separator character).	none

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <monitor interval=30>
      <transport-module module="lbm" options="config=/cfgs/TD1.cfg;topic=stats"/>
      <format-module options="separator=|"/>
    </monitor>
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<web-monitor>

The `<web-monitor>` element identifies the address for the web monitor, in the form of `interface:port`. You can use "`*`" to specify the local host.

Cardinality. 0 or 1

Parents. *<daemon>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`xml:space`	How whitespace is handled. `default` trims leading and trailing whitespace (e.g., tabs, spaces, linefeeds, etc.), and compresses multiple whitespace characters into a single space character. `preserve` preserves the whitespace exactly as read.	`default`

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <web-monitor>*:21001</web-monitor>
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<xml-config>

The `<xml-config>` element specifies the UM XML configuration file.

Cardinality. 0 or 1

Parents. *<daemon>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`xml:space`	How whitespace is handled. `default` trims leading and trailing whitespace (e.g., tabs, spaces, linefeeds, etc.), and compresses multiple whitespace characters into a single space character. `preserve` preserves the whitespace exactly as read.	`default`

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <xml-config>configfile.xml</xml-config>
  </daemon>
      . . .
      . . .
</tnw-gateway>
<route-info propagation-interval=”1000” check-interval=”750” timeout=”4000” max-hop-
count=”100”/>
    <route-recalculation backoff-interval=”5000” warning-interval=”10000”/>
```

<route-info>

The `<route-info>` element lets you set control parameters for UM Router initial route setup (or reroute) behavior.

Cardinality. 0 or 1

Parents. *<daemon>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`propagation-interval`	The time interval between route information messages that the UM Router sends to other UM Routers	`1000`
`check-interval`	How often the UM Router checks to see if a route information message needs to be sent, a UM Router has timed out, and/or the routes need to be recalculated.	`750`
`timeout`	How long a UM Router waits after receiving no route information messages from another UM Router before determining that that UM Router is out of service or unreachable.	`4000`
`max-hop-count`	The maximum number of UM Routers a route information message can traverse before being discarded.	`100`
`xml:space`	How whitespace is handled. `default` trims leading and trailing whitespace (e.g., tabs, spaces, linefeeds, etc.), and compresses multiple whitespace characters into a single space character. `preserve` preserves the whitespace exactly as read.	`default`

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <route-info propagation-interval=”1000” check-interval=”750” timeout=”4000” max-hop-
count=”100”/>
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<route-recalculation>

The `<route-recalculation>` element lets you set timing parameters for UM Router rerouting route calculation behavior.

Cardinality. 0 or 1

Parents. *<daemon>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`backoff-interval`	How long, in milliseconds, the UM Router waits after the last detected change in topology before initiating a route recalculation	`5000`
`warning-interval`	How long, in milliseconds, the UM Router waits before warning that a route recalculation is being held up due to a non-converging topology.	`10000`
`xml:space`	How whitespace is handled. `default` trims leading and trailing whitespace (e.g., tabs, spaces, linefeeds, etc.), and compresses multiple whitespace characters into a single space character. `preserve` preserves the whitespace exactly as read.	`default`

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <route-recalculation backoff-interval=”5000” warning-interval=”10000”/>
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<portals>

The `<portals>` element is a container for all endpoint and peer portal configuration information.

Cardinality. 0 or 1

Parents. *<daemon>*

Children. *(<endpoint> | <peer>)+*

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      . . .
      . . .
    </endpoint>
  </portals>
</tnw-gateway>
```

<endpoint>

The <endpoint> element is a container element for all configuration options of a single endpoint portal.

Cardinality.0 or more

Parents. *<portals>*

Children. *<name>, <domain-id>, <cost>?, <source-deletion-delay>?, <max-queue>?, <lbm-config>?, <lbm-attributes>?, <acl>?, <topic-resolution>?, <late-join>?, <topic-purge>?, <topic-interest-generate>?, <topic-domain-activity>?, <pattern-purge>?, <pattern-interest-generate>?, <pattern-domain-activity>?, <remote-topic>?, <remote-pattern>?, <source-context-name?,> <receiver-context-name>?, <sqn-window>?, <context-query>?*

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
  <daemon>
      . . .
      . . .
  </daemon>
  <portals>
    <endpoint>
      <name>E1</name>
      <domain-id>1</domain-id>
      <cost>1</cost>
      . . .
      . . .
    </endpoint>
  </portals>
</tnw-gateway>
```

<domain-id>

The <domain-id> element identifies the TRD for this endpoint portal. It must be unique within the UM Router (which means that for any TRD, you can assign only one endpoint portal per UM Router). Also, all endpoints interfacing a given TRD must have the same <domain-id> value.

Cardinality.1

Parents. *<endpoint>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
  <daemon>
      . . .
      . . .
  </daemon>
  <portals>
    <endpoint>
      <name>E1</name>
      <domain-id>1</domain-id>
      <cost>1</cost>
      . . .
      . . .
    </endpoint>
  </portals>
</tnw-gateway>
```

<cost>

The `<cost>` element assigns a positive non-zero integer cost to the portal. The default value is `1`.

Cardinality. 0 or 1

Parents. *<endpoint> <peer>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
  <daemon>
      . . .
      . . .
  </daemon>
  <portals>
    <endpoint>
      <name>E1</name>
      <domain-id>1</domain-id>
      <cost>25</cost>
      . . .
      . . .
    </endpoint>
  </portals>
</tnw-gateway>
```

<source-deletion-delay>

The `<source-deletion-delay>` element sets the time in milliseconds to wait after a route map change occurs before deleting a proxy source. Such a route map change could be due to failure of a UM Router or link within a network..

Cardinality. 0 or 1

Parents. *<endpoint>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
  <daemon>
      . . .
      . . .
  </daemon>
  <portals>
    <endpoint>
      <name>E1</name>
      <domain-id>1</domain-id>
      <source-deletion-delay>500</source-deletion-delay>
      . . .
      . . .
    </endpoint>
  </portals>
</tnw-gateway>
```

<max-queue>

The `<max-queue>` element sets the maximum buffer size for blocking messages. If not specified, this defaults to 1000000 bytes.

Cardinality. 0 or 1

Parents. *<endpoint> <peer>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
  <daemon>
      . . .
      . . .
  </daemon>
  <portals>
    <endpoint>
      <name>E1</name>
      <domain-id>1</domain-id>
      <max-queue>500000</max-queue>
      . . .
      . . .
    </endpoint>
  </portals>
</tnw-gateway>
```

<lbm-config>

The `<lbm-config>` element specifies the UM configuration file that contains configuration options associated with this portal.

Cardinality. 0 or 1

Parents. *<endpoint> <peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`xml:space`	How whitespace is handled. `default` trims leading and trailing whitespace (e.g., tabs, spaces, linefeeds, etc.), and compresses multiple whitespace characters into a single space character. `preserve` preserves the whitespace exactly as read.	`default`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <name>E2</name>
      <domain-id>1</domain-id>
```

```
      <lbm-config>/path/endpoint2.cfg</lbm-config>
      . . .
      . . .
    </endpoint>
  </portals>
</tnw-gateway>
```

<lbm-attributes>

The `<lbm-attributes>` element is a container for individual UM-option-setting elements. It lets you set individual UM attributes without referencing a UM configuration file. These values override any values set via files referenced by `<lbm-config>`.

Cardinality. 0 or 1

Parents. *<endpoint> <peer>*

Children. *<option>+*

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <name>E2</name>
      <domain-id>1</domain-id>
      <lbm-attributes>
        <option scope="context" name="request_tcp_interface" value="10.28.5.5" />
        <option scope="context" name="response_tcp_interface" value="127.0.0.1" />
      </lbm-attributes>      . . .
      . . .
    </endpoint>
  </portals>
</tnw-gateway>
```

<option/>

The `<option>` element lets you set an individual UM configuration option without referencing a UM configuration file. This value overrides any values set via files referenced by `<lbm-config>`.

Cardinality. 1 or more

Parents. *<lbm-attributes>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`scope`	The type of object to which an option can apply. Possible scopes are `context`, `source`, `receiver`, `wildcard_receiver`, `event_queue`, and `hfx`.	none
`name`	The name of the option.	none
`value`	The value of the option.	none

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <name>E2</name>
      <domain-id>1</domain-id>
      <lbm-attributes>
        <option scope="context" name="request_tcp_interface" value="127.0.0.1" />
        <option scope="context" name="response_tcp_interface" value="10.28.5.5" />
      </lbm-attributes>      . . .
      . . .
    </endpoint>
  </portals>
</tnw-gateway>
```

<acl>

The `<acl>` element contains elements (inbound and outbound ACEs) that describe how an ACL (Access Control List) filters messages.

Cardinality. 0 or 1

Parents. *<endpoint> <peer>*

Children. *<inbound>?, <outbound>?*

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <acl>
        <inbound>
          <ace>
            <topic>AAA</topic>
          </ace>
        </inbound>
      </acl>
    </endpoint>
  </portals>
</tnw-gateway>
```

<inbound>

The `<inbound>` element is a container for ACE elements, to separate inbound ACEs from outbound ACEs.

Cardinality. 0 or 1

Parents. *<acl>*

Children. *<ace>+*

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <acl>
        <inbound>
          <ace>
            <topic>AAA</topic>
          </ace>
        </inbound>
      </acl>
    </endpoint>
  </portals>
</tnw-gateway>
```

<outbound>

The `<outbound>` element is a container for ACE elements, to separate outbound ACEs from inbound ACEs.

Cardinality. 0 or 1

Parents. *<acl>*

Children. *<ace>+*

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <acl>
        <outbound>
          <ace>
            <pcre-pattern>A[AB]A</pcre-pattern>
          </ace>
        </outbound>
      </acl>
    </endpoint>
  </portals>
</tnw-gateway>
```

<ace>

Within an inbound or outbound ACL, you can have one or more `<ace>` elements. Each ACE (Access Control Entry). lets you match and accept or reject messages based on ACC (Access Control Condition) elements, which are the elements contained within an `<ace>` element.

Cardinality. 0 or 1

Parents. *<inbound> <outbound>*

Children. *(topic | pcre-pattern | regex-pattern | transport | source-ip | multicast-group | udp-source-port | udp-destination-port | tcp-source-port |xport-id)+*

XML Attributes:

XML Attribute	Description	Default Value
`match`	This required attribute determines what to do with matched messages. Possible values are `accept` or `reject`.	none

Example:

```
<tnw-gateway version="1.0">
  <portals>
    <endpoint>
      <name>LAN1</name>
      <lbm-config>lan1.cfg</lbm-config>
      <domain-id>1</domain-id>
        <acl>
          <inbound>
            <ace match="accept">
              <topic>ABC</topic>
            </ace>
            <ace match="accept">
              <topic>DEF</topic>
              <transport value=lbt-rm comparison=eq/>
            </ace>
            <ace match="accept">
              <topic>GHI</topic>
            </ace>
          </inbound>
        </acl>
    </endpoint>
      . . .
      . . .
```

<topic>

The `<topic>` element defines a condition used in an ACE. Specifically, this is a match pattern for a topic name.

Cardinality. 0 or more

Parents. *<ace>* (inbound only)

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`xml:space`	How whitespace is handled. `default` trims leading and trailing whitespace (e.g., tabs, spaces, linefeeds, etc.), and compresses multiple whitespace characters into a single space character. `preserve` preserves the whitespace exactly as read.	`default`

Example:

```
<ace match="accept">
  <topic>ABC</topic>
</ace>
```

<pcre-pattern>

The `<pcre-pattern>` element defines a condition used in an ACE. Specifically, this is a match pattern for a PCRE regular expression matched in the message.

Cardinality. 0 or more

Parents. *<ace>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`xml:space`	How whitespace is handled. `default` trims leading and trailing whitespace (e.g., tabs, spaces, linefeeds, etc.), and compresses multiple whitespace characters into a single space character. `preserve` preserves the whitespace exactly as read.	`default`

Example:

```
<ace match="accept">
  <pcre-pattern>ABC</pcre-pattern>
</ace>
```

<regex-pattern>

The `<regex-pattern>` element defines a condition used in an ACE. Specifically, this is a match pattern for a RegEx regular expression matched in the message.

Cardinality. 0 or more

Parents. *<ace>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`xml:space`	How whitespace is handled. `default` trims leading and trailing whitespace (e.g., tabs, spaces, linefeeds, etc.), and compresses multiple whitespace characters into a single space character. `preserve` preserves the whitespace exactly as read.	`default`

Example:

```
<ace match="accept">
  <regex-pattern>ABC</regex-pattern>
</ace>
```

<transport/>

The `<transport>` element defines a condition used in an ACE. Specifically, this is a match pattern for a UM transport type.

Cardinality. 0 or more

Parents. <*ace*> (inbound only)

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`comparison`	Defines a match condition. Choices are equal (`eq` or `equal`) or not equal (`ne` or `notequal`).	none
`value`	The source IP address as a dotted-decimal value to be matched via the comparison.	none

Example:

```
<ace match="accept">
  <transport comparison=”equal” value=”lbtrm”/>
</ace>
```

<source-ip/>

The `<source-ip>` element defines a condition used in an ACE. Specifically, this is a match pattern for the message source IP address. This applies only to TCP, LBT-RM, and LBT-RU transports.

Cardinality. 0 or more

Parents. <*ace*> (inbound only)

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`comparison`	Defines a match condition. Choices are equal (`eq` or `equal`) or not equal (`ne` or `notequal`), less than (`lt` or `lessthan`), less than or equal (`le` or `lessthanequal`), greater than (`gt` or `greaterthan`), or greater than or equal (`ge` or `greaterthanequal`	none
`value`	The value to be matched via the comparison. Choices are: `tcp`, `lbt-rm`, `lbtrm`, `lbt-ru`, `lbtru`, `lbt-ipc`, or `lbtipc`.	none

Example:

```
<ace match="accept">
  <source-ip comparison=”equal” value=”127.0.0.1”/>
</ace>
```

<multicast-group/>

The `<multicast-group>` element defines a condition used in an ACE. Specifically, this is a match pattern for the message's multicast group address. This applies only to LBT-RM transports.

Cardinality. 0 or more

Parents. <*ace*> (inbound only)

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`comparison`	Defines a match condition. Choices are equal (`eq` or `equal`) or not equal (`ne` or `notequal`), less than (`lt` or `lessthan`), less than or equal (`le` or `lessthanequal`), greater than (`gt` or `greaterthan`), or greater than or equal (`ge` or `greaterthanequal`	none
`value`	The multicast group IP address as a dotted-decimal value to be matched via the comparison.	none

Example:

```
<ace match="accept">
  <multicast-group comparison=”equal” value=”127.0.0.1”/>
</ace>
```

<udp-source-port/>

The `<udp-source-port>` element defines a condition used in an ACE. Specifically, this is a match pattern for the message's UDP source port number. This applies only to LBT-RM and LBT-RU transports.

Cardinality. 0 or more

Parents. <*ace*> (inbound only)

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`comparison`	Defines a match condition. Choices are equal (`eq` or `equal`) or not equal (`ne` or `notequal`), less than (`lt` or `lessthan`), less than or equal (`le` or `lessthanequal`), greater than (`gt` or `greaterthan`), or greater than or equal (`ge` or `greaterthanequal`	none
`value`	The UDP source port number.	none

Example:

```
<ace match="accept">
  <udp-source-port comparison=”equal” value=”1234”/>
</ace>
```

<udp-destination-port/>

The `<udp-destination-port>` element defines a condition used in an ACE. Specifically, this is a match pattern for the message's UDP destination port number. This applies only to LBT-RM transports.

Cardinality. 0 or more

Parents. <*ace*> (inbound only)

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`comparison`	Defines a match condition. Choices are equal (`eq` or `equal`) or not equal (`ne` or `notequal`), less than (`lt` or `lessthan`), less than or equal (`le` or `lessthanequal`), greater than (`gt` or `greaterthan`), or greater than or equal (`ge` or `greaterthanequal`	none
`value`	The UDP destination port number.	none

Example:

```
<ace match="accept">
  <udp-destination-port comparison=”equal” value=”1234”/>
</ace>
```

<tcp-source-port/>

The `<tcp-source-port>` element defines a condition used in an ACE. Specifically, this is a match pattern for the message's TCP source port number. This applies only to TCP transports.

Cardinality. 0 or more

Parents. *<ace>* (inbound only)

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`comparison`	Defines a match condition. Choices are equal (`eq` or `equal`) or not equal (`ne` or `notequal`), less than (`lt` or `lessthan`), less than or equal (`le` or `lessthanequal`), greater than (`gt` or `greaterthan`), or greater than or equal (`ge` or `greaterthanequal`	none
`value`	The TCP source port number.	none

Example:

```
<ace match="accept">
  <tcp-source-port comparison=”equal” value=”1234”/>
</ace>
```

<xport-id/>

The `<xport-id>` element defines a condition used in an ACE. Specifically, this is a match pattern for the message's xport ID number. This applies only to LBT-IPC transports.

Cardinality. 0 or more

Parents. *<ace>* (inbound only)

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`comparison`	Defines a match condition. Choices are equal (`eq` or `equal`) or not equal (`ne` or `notequal`), less than (`lt` or `lessthan`), less than or equal (`le` or `lessthanequal`), greater than (`gt` or `greaterthan`), or greater than or equal (`ge` or `greaterthanequal`	none
`value`	The xport ID number.	none

Example:

```
<ace match="accept">
  <xport-id comparison=”equal” value=”1234”/>
</ace>
```

<topic-resolution>

The `<topic-resolution>` element is a container for UM Router topic resolution behavior options.

Cardinality. 0 or 1

Parents. *<endpoint>*

Children. *<topic-use-query>?, <pattern-use-query>?, <remote-topic-interest>?, <remote-pattern-interest>?, <domain-route>?, <initial-request>?*

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <topic-resolution>
        <initial-request>
          <rate-limit/>
        </initial-request>
      </topic-resolution>
    </endpoint>
  </portals>
</tnw-gateway>
```

<initial-request/>

The `<initial-request>` element sets interval and duration for initial topic resolution requests.

Cardinality. 0 or 1

Parents. *<topic-resolution>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`duration`	The minimum duration for which the initial topic resolution requests are sent. Before changing the value of this option, please contact Informatica Support.	`10`
`periodic-interval`	The interval at which the initial topic resolution requests are sent. Before changing the value of this option, please contact Informatica Support	`1000`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <topic-resolution>
        <initial-request duration=”15” periodic-interval=”800”/>
        </topic-use-query>
      </topic-resolution>
    </endpoint>
  </portals>
</tnw-gateway>
```

<topic-use-query>

The `<topic-use-query>` element sets parameters for when and how often this endpoint portal sends topic use queries.

Cardinality. 0 or 1

Parents. *<topic-resolution>*

Children. *<rate-limit>*

XML Attributes:

XML Attribute	Description	Default Value
max	Maximum number of topic use queries to send for a given topic, each separated by the timeout value before giving up and removing the topic from the topic list. Before changing the value of this option, please contact Informatica Support.	5
periodic-interval	The interval, in milliseconds, between periodic topic use queries being sent for each topic the portal has interest in. Before changing the value of this option, please contact Informatica Support.	300000
timeout	The maximum time, in milliseconds, to wait for a topic use response. Before changing the value of this option, please contact Informatica Support.	3000

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <topic-resolution>
        <topic-use-query max=”6” periodic-interval=”250000” timeout=”4000”>
          <rate-limit/>
        </topic-use-query>
      </topic-resolution>
    </endpoint>
  </portals>
</tnw-gateway>
```

<rate-limit/>

The <rate-limit> element sets rate limits for topic resolution data sent over the network.

You can set rate limits individually for each of the following topic resolution message types:

- topic use query
- pattern use query
- topic interest
- pattern interest
- domain route

Cardinality. 0 or 1

Parents. *<topic-use-query> <pattern-use-query> <remote-topic-interest> <remote-pattern-interest > <domain-route>*

Children. None.

XML Attributes: You can set a limit in bps, objects per second, or both. The UM Router begins limiting when the lower of these attributes is reached.

XML Attribute	Description	Default Value
`bps`	The limit in Bits per Second that data will be sent on the network. Before changing the value of this option, please contact Informatica Support.	For use queries and interest messages: `500000` For domain route messages: `0`
`objects-per-second`	The limit in Objects per Second that data will be sent on the network. Before changing the value of this option, please contact Informatica Support.	For use queries:`500` For interest messages: `0` For domain route messages: `50`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <topic-resolution>
        <topic-use-query max=”6” periodic-interval=”250000” timeout=”4000”/>
          <rate-limit bps="550000" objects-per-second="0"/>
        </topic-use-query>
      </topic-resolution>
    </endpoint>
  </portals>
</tnw-gateway>
```

<pattern-use-query>

The `<pattern-use-query>` element sets parameters for when and how often this endpoint portal sends pattern use queries.

Cardinality. 0 or 1

Parents. *<topic-resolution>*

Children. *<rate-limit>*

XML Attributes:

XML Attribute	Description	Default Value
max	Maximum number of pattern use queries to send for a given pattern, each separated by the timeout value before giving up and removing the topic from the topic list. Before changing the value of this option, please contact Informatica Support.	5
periodic-interval	The interval, in milliseconds, between periodic pattern use queries being sent for each pattern the portal has interest in. Before changing the value of this option, please contact Informatica Support.	300000
timeout	The maximum time, in milliseconds, to wait for a pattern use response. Before changing the value of this option, please contact Informatica Support.	3000

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <topic-resolution>
        <pattern-use-query max=”6” periodic-interval=”250000” timeout=”4000”>
          <rate-limit/>
        </topic-use-query>
      </topic-resolution>
    </endpoint>
  </portals>
</tnw-gateway>
```

<remote-topic-interest>

The <remote-topic-interest> element sets parameters for when and how often this endpoint portal sends topic interest messages.

Cardinality. 0 or 1

Parents. *<topic-resolution>*

Children. *<rate-limit>*

XML Attributes:

XML Attribute	Description	Default Value
`min-interval`	The minimum interval, in milliseconds, between topic interest messages being sent for each topic the portal has interest in.	`1000`
`max-interval`	The maximum interval, in milliseconds, between topic interest messages being sent for each topic the portal has interest in.	`60000`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <topic-resolution>
        <remote-topic-interest min-interval=”1000” max-interval=”90000”>
          <rate-limit/>
        </remote-topic-interest>
      </topic-resolution>
    </endpoint>
  </portals>
</tnw-gateway>
```

<remote-pattern-interest>

The `<remote-pattern-interest>` element sets parameters for when and how often this endpoint portal sends pattern interest messages.

Cardinality. 0 or 1

Parents. *<topic-resolution>*

Children. *<rate-limit>*

XML Attributes:

XML Attribute	Description	Default Value
`min-interval`	The minimum interval, in milliseconds, between pattern interest messages being sent for each pattern the portal has interest in.	`1000`
`max-interval`	The maximum interval, in milliseconds, between pattern interest messages being sent for each pattern the portal has interest in.	`60000`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <topic-resolution>
        <remote-pattern-interest min-interval=”1000” max-interval=”90000”>
          <rate-limit/>
        </remote-pattern-interest>
      </topic-resolution>
    </endpoint>
  </portals>
</tnw-gateway>
```

<domain-route>

The `<domain-route>` element sets maximum and minimum limits for the interval between periodic domain route messages being sent for each remote domain that the portal is servicing.

Cardinality. 0 or 1

Parents. *<topic-resolution>*

Children. *<rate-limit>*

XML Attributes:

XML Attribute	Description	Default Value
`min-interval`	The minimum interval, in milliseconds, between domain route messages being sent for each domain.	`100`
`max-interval`	The maximum interval, in milliseconds, between domain route messages being sent for each domain.	`1000`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <topic-resolution>
        <domain-route min-interval=”100” max-interval=”1000”>
          <rate-limit bps="0" objects-per-second="50"/>
        </domain-route>
      </topic-resolution>
    </endpoint>
  </portals>
</tnw-gateway>
```

<remote-topic/>

The `<remote-topic>` element determines timings and limits for determination of continued topic interest at this portal.

Cardinality. 0 or 1

Parents. *<endpoint>* *<peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`check-interval`	Interval (in milliseconds) between checking individual topics for continued interest. Before changing the value of this option, please contact Informatica Support.	`90000`
`max-topics`	Maximum number of topics to check at a time. Before changing the value of this option, please contact Informatica Support.	`100`
`timeout`	Minimum time (in milliseconds) remote interest for a topic must be unrefreshed before interest is removed for that domain. Before changing the value of this option, please contact Informatica Support.	`300000`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <remote-topic check-interval=”80000” max-topics=”80” timeout=”250000”/>
    </endpoint>
  </portals>
</tnw-gateway>
```

<remote-pattern/>

The `<remote-pattern>` element determines timings and limits for determination of continued pattern interest at this portal.

Cardinality. 0 or 1

Parents. *<endpoint>* *<peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`check-interval`	Interval (in milliseconds) between checking individual patterns for continued interest. Before changing the value of this option, please contact Informatica Support.	`90000`
`max-topics`	Maximum number of patterns to check at a time. Before changing the value of this option, please contact Informatica Support.	`100`
`timeout`	Minimum time (in milliseconds) remote interest for a pattern must be unrefreshed before interest is removed for that domain. Before changing the value of this option, please contact Informatica Support.	`300000`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <remote-pattern check-interval=”80000” max-topics=”80” timeout=”250000”/>
    </endpoint>
  </portals>
</tnw-gateway>
```

<source-context-name>

The `<source-context-name>` element specifies the portal source context name.

Cardinality. 0 or 1

Parents. *<endpoint>* *<peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`xml:space`	How whitespace is handled. `default` trims leading and trailing whitespace (e.g., tabs, spaces, linefeeds, etc.), and compresses multiple whitespace characters into a single space character. `preserve` preserves the whitespace exactly as read.	`default`

Example:

```
<tnw-gateway version="1.0">
      . . .
```

```
      . . .
  <portals>
    <endpoint>
      <source-context-name>SourceContext01</source-context-name>
    </endpoint>
  </portals>
</tnw-gateway>
```

<receiver-context-name>

The `<receiver-context-name>` element specifies the portal receiver context name.

Cardinality. 0 or 1

Parents. *<endpoint>* *<peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`xml:space`	How whitespace is handled. `default` trims leading and trailing whitespace (e.g., tabs, spaces, linefeeds, etc.), and compresses multiple whitespace characters into a single space character. `preserve` preserves the whitespace exactly as read.	`default`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <receiver-context-name>RcvrContext01</source-context-name>
    </endpoint>
  </portals>
</tnw-gateway>
```

<sqn-window/>

The `<sqn-window>` (sequence number window) element specifies the portal's awareness of received message sequence numbers, for the purpose of detecting duplicates.

Cardinality. 0 or 1

Parents. *<endpoint>* *<peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`increment`	Determines the minimum increment, in topic (fragment) sequence numbers, by which the sequence number window is moved when the window size (below) is exceeded. Must be a multiple of 8, an even divisor of the window size, and less the window size. Before changing the value of this option, please contact Informatica Support.	`2048`
`size`	Determines the maximum number of topic (fragment) sequence numbers maintained in the window, for any given source. Must be a multiple of 8. Before changing the value of this option, please contact Informatica Support.	`16384`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <sqn-window size="1024" increment="512"/>
    </endpoint>
  </portals>
</tnw-gateway>
```

<context-query/>

The `<context-query>` element determines timing characteristics for context name queries generated at this portal.

Cardinality. 0 or 1

Parents. *<endpoint> <peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`interval`	Interval (in milliseconds) between groups of context queries. Before changing the value of this option, please contact Informatica Support.	`200`
`max-contexts`	Maximum number of contexts for which queries are generated at one time. Before changing the value of this option, please contact Informatica Support.	`20`
`periodic-interval`	Interval (in milliseconds) at which context queries are generated. Before changing the value of this option, please contact Informatica Support.	`300000`
`timeout`	Minimum time (in seconds) a context query must be unanswered before it is removed for the portal. Before changing the value of this option, please contact Informatica Support.	`900`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <context-query periodic-interval="25000" max-contexts="15" interval="180"
timeout="875"/>
    </endpoint>
  </portals>
</tnw-gateway>
```

<peer>

The `<peer>` element is a container element for all configuration options of a single peer portal.

Cardinality.0 or more

Parents. *<portals>*

Children. *<name>, <cost>?, <sourcemap>?, (<tcp> | <single-tcp>), <source-deletion-delay>?, <max-queue>?, <max-datagram>?, <batching>?, <lbm-config>?, <lbm-attributes>?, <acl>?, <topic-purge>?, <topic-interest-generate>?, <topic-domain-activity>?, <pattern-purge>?, <pattern-interest-generate>?, <pattern-domain-activity>?, <topic-use-check>?, <pattern-use-check>?, <source-context-name>?, <receiver-context-name>?, <sqn-window>?, <context-query>?, <gateway-keepalive>?*

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
  <daemon>
      . . .
      . . .
```

```
  </daemon>
  <portals>
    <peer>
      <name>P1</name>
      <cost>1</cost>
      . . .
      . . .
    </peer>
  </portals>
</tnw-gateway>
```

<sourcemap/>

The `<sourcemap>` element sets the size of the peer portal's source map.

Cardinality. 0 or 1

Parents. *<peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`size`	Number of entries in the source map. Must be able to be factorized by all 2s (e.g., 1024, 2048, etc.).	`131072`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <sourcemap size=”131072”/>
      . . .
      . . .
    </peer>
  </portals>
</tnw-gateway>
```

<tcp>

The `<tcp>` element contains elements for a peer portal's tcp settings, when configuring the peer for dual-tcp operation.

Cardinality. 0 or 1

Parents. *<peer>*

Children. *<interface>?, <listen-port>, <receive-buffer>?, <send-buffer>?, <keepalive>?, <nodelay>?, <companion>*

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
```

```
      <tcp>
        <interface>10.28.5.5/24</interface>
        <listen-port>
      . . .
      . . .
      </tcp>
    </peer>
  </portals>
</tnw-gateway>
```

<interface>

The `<interface>` element contains the IP host or network address for this peer portal, specified in dotted-decimal or CIDR format.

Cardinality. 0 or 1

Parents. *<tcp>* *<single-tcp>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <tcp>
        <interface>10.28.5.5/24</interface>
      . . .
      . . .
      </tcp>
    </peer>
  </portals>
</tnw-gateway>
```

<listen-port>

The `<listen-port>` element contains port number on which this peer portal listens for connections from the companion peer portal. The companion peer portal configuration must specify this port as its companion port.

Cardinality.1

Parents. *<tcp>* *<acceptor>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <tcp>
        <interface>10.28.5.5/24</interface>
        <listen-port>46000</listen-port>
      . . .
      . . .
      </tcp>
    </peer>
```

```
  </portals>
</tnw-gateway>
```

<receive-buffer>

The `<receive-buffer>` element contains the size of the TCP receive buffer. If not specified, the UM Router uses the system default size.

Cardinality. 0 or 1

Parents. *<tcp>* *<single-tcp>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <single-tcp>
        <send-buffer>128000</receive-buffer>
      . . .
      . . .
      </tcp>
    </peer>
  </portals>
</tnw-gateway>
```

<send-buffer>

The `<send-buffer>` element contains the size of the TCP send buffer. If not specified, the UM Router uses the system default size.

Cardinality. 0 or 1

Parents. *<tcp>* *<single-tcp>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <single-tcp>
        <send-buffer>128000</send-buffer>
      . . .
      . . .
      </tcp>
    </peer>
  </portals>
</tnw-gateway>
```

<keepalive/>

The `<keepalive>` element, when present, enables a TCP keepalive signal transmission, which is disabled by default.

Cardinality. 0 or 1

Parents. *<tcp>* *<single-tcp>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <single-tcp>
        <keepalive/>
      . . .
      . . .
      </tcp>
    </peer>
  </portals>
</tnw-gateway>
```

<nodelay/>

The `<nodelay>` element, when present, allows immediate sending of messages without waiting for the batching send buffer to fill. This is disabled by default.

Cardinality. 0 or 1

Parents. *<tcp>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <single-tcp>
        <nodelay/>
      . . .
      . . .
      </tcp>
    </peer>
  </portals>
</tnw-gateway>
```

<companion>

The `<companion>` element contains the IP address and the port of the companion peer portal on another UM Router, to which this peer is connected.

Cardinality. 0 or 1

Parents. *<tcp>*

Children. *<address>, <port>*

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <tcp>
        <companion>
          <address>10.28.3.91</address>
          <port>25000</port>
        </companion>
      . . .
      . . .
      </tcp>
    </peer>
  </portals>
</tnw-gateway>
```

<address>

The `<address>` element contains the IP address of the companion peer portal on another UM Router, to which this peer is connected.

Cardinality. 0 or 1

Parents. *<companion> <initiator>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <tcp>
        <companion>
          <address>10.28.3.91</address>
          <port>25000</port>
        </companion>
      . . .
      . . .
      </tcp>
    </peer>
  </portals>
</tnw-gateway>
```

<port>

The `<port>` element contains the IP port of the companion peer portal on another UM Router, to which this peer is connected.

Cardinality. 0 or 1

Parents. *<companion> <initiator>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <tcp>
        <companion>
          <address>10.28.3.91</address>
          <port>25000</port>
        </companion>
      . . .
      . . .
      </tcp>
    </peer>
  </portals>
</tnw-gateway>
```

<single-tcp>

The `<single-tcp>` element contains elements for a peer portal's tcp settings, when configuring the peer for single-tcp operation.

Cardinality. 0 or 1

Parents. *<peer>*

Children. *<interface>?, <receive-buffer>?, <send-buffer>?, <keepalive>?, <nodelay>?, (<initiator>* | *<acceptor>)*

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <single-tcp>
        <interface>10.28.5.5/24</interface>
        <acceptor>
          <listen-port>23746</listen-port>
        </acceptor>
        . . .
        . . .
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

<initiator>

The `<initiator>` element contains the IP address and the port of the companion peer portal on another UM Router, to which this peer is connected. This element is used in single-tcp configurations.

Cardinality. 0 or 1

Parents. *<tcp>*

Children. *<address>, <port>*

XML Attributes:

XML Attribute	Description	Default Value
`reconnect-interval`	The time interval, in milliseconds, to wait before reconnecting to the companion portal if this connection is interrupted.	`5000`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <single-tcp>
        <initiator>
          <address>10.28.3.91</address>
          <port>25000</port>
        </initiator>
      . . .
      . . .
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

<acceptor>

The `<acceptor>` element contains the listen port address of the companion peer portal on another UM Router, to which this peer is connected. This element is used in single-tcp configurations.

Cardinality. 0 or 1

Parents. *<single-tcp>*

Children. *<listen-port>*

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <single-tcp>
        <acceptor>
          <listen-port>25000</port>
        </acceptor>
      . . .
      . . .
      </single-tcp>
    </peer>
  </portals>
</tnw-gateway>
```

<max-datagram>

The `<max-datagram>` element specifies the maximum size a peer portal will allow an outgoing datagram to be before fragmenting it.

Cardinality. 0 or 1

Parents. *<peer>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <max-datagram>50000</max-datagram>
    </peer>
  </portals>
</tnw-gateway>
```

<batching>

The `<batching>` element contains batching size and timing parameters. This applies to data messages only: the UM Router sends control messages immediately (flushing any batched data messages).

Cardinality. 0 or 1

Parents. *<peer>*

Children. *<min-length> ?, <batch-interval>?*

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <batching>
        <min-length>4096</min-length>
        <batch-interval>500</batch-interval>
      </batching>
    </peer>
  </portals>
</tnw-gateway>
```

<min-length>

The `<min-length>` element specifies the minimum length of a set of batched messages. When the total length of the batched messages reaches or exceeds this value, the batch is sent. If not specified, it defaults to 8192 bytes.

Cardinality. 0 or 1

Parents. *<batching>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <batching>
        <min-length>4096</min-length>
        <batch-interval>500</batch-interval>
      </batching>
    </peer>
  </portals>
</tnw-gateway>
```

<batch-interval>

The `<batch-interval>` element specifies the maximum interval (in milliseconds) between when the first message of a batch is queued until the batch is sent. A message stays in the batch queue until this value or <min-length> is met or exceeded (whichever occurs first). If not specified, it defaults to 200 milliseconds. The minimum allowed value is 3 milliseconds.

Cardinality. 0 or 1

Parents. *<peer>*

Children. None.

XML Attributes: None.

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <batching>
        <min-length>4096</min-length>
        <batch-interval>500</batch-interval>
      </batching>
    </peer>
  </portals>
</tnw-gateway>
```

<gateway-keepalive/>

The `<gateway-keepalive>` element contains parameters for the keepalive signals sent from this peer portal. This is a UM Router-level keepalive, not to be confused with the TCP-level <keepalive> element.

Cardinality. 0 or 1

Parents. *<peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`idle`	Determines if UM Router keepalives should be sent only if no traffic has been sent or received in the last interval. Possible values are `yes` or `no`.	`yes`
`interval`	Minimum interval, in milliseconds, between keepalive messages sent. We recommend setting this to `2000` or greater. A value of `0` (zero) disables keepalives.	`5000`
`timeout`	Maximum time, in milliseconds, a peer can receive nothing from the companion before determining the connection is dead and disconnecting. We recommend setting this to 3 times the `interval` value.	`15000`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <gateway-keepalive idle=”no” interval="1000"/>
    </peer>
  </portals>
</tnw-gateway>
```

Deprecated Elements

The following option elements are deprecated, and provided only for UM Gateway configuration file backward compatibility. Values assigned to them have no effect on the operation of the UM Router.

<propagation-delay/>

The `<propagation-delay>` element specifies the difference between the shortest and longest propagation delays in the network. **This element is deprecated.**

Cardinality. 0 or 1

Parents. *<daemon>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`delta`	The difference, in milliseconds, between the shortest and longest propagation delays in the network.	`0`

Example:

```
<tnw-gateway version="1.0">
  <daemon>
    <propagation-delay delta=50/>
  </daemon>
      . . .
      . . .
</tnw-gateway>
```

<late-join/>

The `<late-join>` element determines how Late Join is handled by this endpoint portal. **This element is deprecated.**

Cardinality. 0 or 1

Parents. <*endpoint*>

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`forward`	If late join retransmission requests to a source on the portal are unable to be filled locally, determines if requests are forwarded to the original source. Choices are yes or no . This only applies to sources created on the portal with late join support.	`yes`
`provide`	Determines whether sources created on a portal should provide late join to receivers. Allowable values are `source` – Provide late join only if the original source provides late join. `always` – Always provide late join, even if the original source does not. `never` – Never provide late join, even if the original source does. The UM configuration specified for the portal determines the late join configuration.	`source`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <late-join provide="source" forward="yes"/>
    </endpoint>
  </portals>
</tnw-gateway>
```

<topic-purge/>

The `<topic-purge>` element determines when this portal's proxy receivers can purge topics. **This element is deprecated.**

Cardinality. 0 or 1

Parents. *<endpoint> <peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`periodic-interval`	Interval (in milliseconds) at which receiver topics are checked to determine if they can be purged.	`6000`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <topic-purge periodic-interval="5000"/>
    </endpoint>
  </portals>
</tnw-gateway>
```

<topic-interest-generate/>

The `<topic-interest-generate>` element determines timing characteristics for interest message generation at this portal. **This element is deprecated.**

Cardinality. 0 or 1

Parents. *<endpoint> <peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`interval`	Interval (in milliseconds) between groups of topics.	`200`
`max-topics`	Maximum topics for which interest is generated at one time.	`20`
`periodic-interval`	Interval (in milliseconds) at which topic interest is generated.	`300000`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <topic-interest-generate periodic-interval="250000"/>
    </endpoint>
  </portals>
</tnw-gateway>
```

<topic-domain-activity/>

The `<topic-domain-activity>` element determines how long a domain remains quiescent until it is determined inactive. **This element is deprecated.**

Cardinality. 0 or 1

Parents. *<endpoint> <peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`timeout`	Minimum time (in seconds) domain interest for a topic must be unrefreshed before interest is removed for that domain.	`900`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <topic-domain-activity timeout="800"/>
    </endpoint>
  </portals>
</tnw-gateway>
```

<pattern-purge/>

The <pattern-purge> element determines when this portal's proxy receivers can purge pattern. **This element is deprecated.**

Cardinality. 0 or 1

Parents. *<endpoint> <peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
periodic-interval	Interval (in milliseconds) at which receiver patterns are checked to determine if they can be purged.	6000

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <pattern-purge periodic-interval="5000"/>
    </endpoint>
  </portals>
</tnw-gateway>
```

<pattern-interest-generate/>

The <pattern-interest-generate> element determines timing characteristics for interest message generation at this portal. **This element is deprecated.**

Cardinality. 0 or 1

Parents. *<endpoint> <peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
interval	Interval (in milliseconds) between groups of patterns.	200
max-patterns	Maximum patterns for which interest is generated at one time.	300000
periodic-interval	Interval (in milliseconds) at which pattern interest is generated.	300000

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
```

```
    <endpoint>
      <pattern-interest-generate periodic-interval="250000"/>
    </endpoint>
  </portals>
</tnw-gateway>
```

<pattern-domain-activity/>

The `<pattern-domain-activity>` element determines how long a domain remains quiescent until it is determined inactive. **This element is deprecated.**

Cardinality. 0 or 1

Parents. *<endpoint> <peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`timeout`	Minimum time (in seconds) domain interest for a pattern must be unrefreshed before interest is removed for that domain.	`900`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <endpoint>
      <pattern-domain-activity timeout="800"/>
    </endpoint>
  </portals>
</tnw-gateway>
```

<topic-use-check/>

The `<topic-use-check>` element checks for interest in topics at periodic intervals. **This element is deprecated.**

Cardinality. 0 or 1

Parents. *<peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`periodic-interval`	The interval (in milliseconds) at which source topics are checked to determine if there is no more interest. Before changing the value of this option, please contact Informatica Support.	`300000`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <topic-use-check periodic-interval="1000"/>
    </peer>
  </portals>
</tnw-gateway>
```

<pattern-use-check/>

The `<pattern-use-check>` element checks for interest in patterns at periodic intervals. **This element is deprecated.**

Cardinality. 0 or 1

Parents. *<peer>*

Children. None.

XML Attributes:

XML Attribute	Description	Default Value
`periodic-interval`	The interval (in milliseconds) at which source pattern are checked to determine if there is no more interest. Before changing the value of this option, please contact Informatica Support.	`300000`

Example:

```
<tnw-gateway version="1.0">
      . . .
      . . .
  <portals>
    <peer>
      <topic-use-check periodic-interval="1000"/>
    </peer>
  </portals>
</tnw-gateway>
```

CHAPTER 6

UM Router Monitoring

This chapter includes the following topics:

UM Router Web Monitor

The built-in web monitor (configured in the `tnwgd` XML configuration file; see Chapter 5, "XML Configuration Reference" on page 46) provides valuable statistics about the UM Router and its portals, for which, the Web Monitor separates into receive statistics and send statistics. The Web Monitor provides a page for each endpoint and peer portal.

Main Page

This page displays general information about the UM Router, and also provides the following links to more detailed statistical and configuration information.

UM Router configuration

Displays the UM Router XML configuration file used by this UM Router.

Portals

Displays portal statistics and information, one portal per page. The Portals page allows you to link to any of the Peer or Endpoint portals configured for the UM Router.

Topology Info

This links to a page that displays UM Router network connectivity information from the perspective of this UM Router.

Path Info

This lets you query and display a hop path that messages will take between any two TRDs.

On some platforms, the Main page may include a link (`GNU malloc info`) to a memory allocation display page that displays the following:

arena

Non-mmapped space allocated (bytes)

ordblks

Number of free chunks

hblks

Number of mmapped regions

hblkhd

Space allocated in mmapped regions (bytes)

uordblks

Total allocated space (bytes)

fordblks

Total free space (bytes)

Endpoint Portal Page

The Endpoint Portal Page displays Receive and Send statistics for the selected endpoint portal. Receive statistics pertain to messages entering the portal from its connected TRD. Send statistics pertain to messages sent out to the TRD.

Click on any of the links at the top of the page to review configuration option values for the portal's UM topic resolution domain. The two columns provide different units of measure for a given statistic type, where the first column is typically in fragments or messages (depending on the statistic type), and the second column is in bytes.

Endpoint Portal *name*

Domain ID

The ID for the Topic Resolution Domain (TRD) to which this portal is connected.

Portal Cost

The cost value assigned to this portal.

Local Interest

Totals (listed below) for topics and patterns in this portal's interest list that originated from receivers in the immediately adjacent TRD.

Topics

Of the local interest total, the number of topics.

PCRE patterns

Of the local interest total, the number of wildcard patterns, using PCRE pattern matching.

REGEX patterns

Of the local interest total, the number of wildcard patterns, using REGEX pattern matching.

Remote Interest

Totals (listed below) for topics and patterns in this portal's interest list that originated from receivers beyond and downstream from the immediately adjacent TRD.

Topics

Of the remote interest total, the number of topics.

PCRE patterns

Of the remote interest total, the number of wildcard patterns, using PCRE pattern matching.

REGEX patterns

Of the remote interest total, the number of wildcard patterns, using REGEX pattern matching.

Proxy Receivers

The number of proxy receivers active in this portal.

Receiver Topics

The number of topics in which the other portals in the UM Router have detected current interest and summarily propagated to this portal.

Receiver PCRE patterns

The number of wildcard patterns, using PCRE pattern matching, in which the other portals in the UM Router have detected current interest and summarily propagated to this portal.

Receiver REGEX patterns

The number of wildcard patterns, using REGEX pattern matching, in which the other portals in the UM Router have detected current interest and summarily propagated to this portal.

Proxy Sources

The number of proxy sources active in this portal.

Endpoint Receive Statistics

Transport topic fragments/bytes received

The total transport-based topic-related traffic of messages containing user data received by this portal from a TRD. The first column counts the number of fragments (or whole messages for messages that were not fragmented).

Transport topic request fragments/bytes received

Topic messages received that are request messages, i.e., messages send via `lbm_send_request*()` rather than `lbm_src_send*()`.

Transport topic control msgs/bytes received

The total transport-based topic-related traffic received by this portal from a TRD. These are supervisory messages, which include TSNIs, SRIs., etc. The first column counts the number of messages.

Immediate topic fragments/bytes received

The total number of Multicast Immediate Messaging (MIM) messages or message fragments, and bytes (second column), that have a topic, received at this portal.

Immediate topic request fragments/bytes received

Of the MIM topic messages received, this is the amount of those that are requests.

Immediate topicless fragments/bytes received

The total number of MIM messages or message fragments, and bytes (second column), with null topics, received by his portal.

Immediate topicless request fragments/bytes received

Of the MIM topicless messages received, this is the amount of those that are requests.

Unicast data messages/bytes received

The total number of Unicast Immediate Messaging (UIM) messages (and bytes, second column) containing user data, received by this portal.

Duplicate unicast data messages/bytes dropped

UIM data messages discarded because they were duplicates of messages already received.

Unicast data messages/bytes received with no stream info

UIM data messages discarded because they were from an earlier, incompatible version of UM. This tally should stay at 0; otherwise, contact Informatica Support.

Unicast data messages/bytes received with no route to destination

UIM data messages that are on a wrong path, possibly due to a route recalculation. This tally should stay at 0, though it may increment a few messages at the time of a topology change.

Unicast control messages/bytes received

The total number of Unicast Immediate Messaging (UIM) supervisory (non-data) messages (and bytes, second column) received by this portal.

Duplicate unicast control messages/bytes dropped

Supervisory UIMs dropped because they were duplicates of messages already received.

Unicast control messages/bytes received with no stream info

Supervisory UIMs dropped because they were from an earlier, incompatible version of UM. This tally should stay at 0; otherwise, contact Informatica Support.

Unicast control messages/bytes received with no route to destination

Supervisory UIM messages that are on a wrong path, possibly due to a route recalculation. This tally should stay at 0, though it may increment a few messages at the time of a topology change.

Endpoint Send Statistics

Transport topic fragments/bytes forwarded

The total transport-based topic-related traffic forwarded to this portal from other portals in this UM Router. This could include user messages, TSNIs, SRIs, etc. The first column counts the number of fragments (or whole messages for messages that were not fragmented).

Transport topic fragments/bytes sent

Of the transport topic traffic forwarded, this is the amount of traffic sent out to the TRD.

Transport topic request fragments/bytes sent

Of the messages sent, this is the amount of those that are requests.

Duplicate transport topic fragments/bytes dropped

Of the messages forwarded to this portal, this is the total of those that were discarded because they were duplicates of messages already received.

Transport topic fragments/bytes dropped due to blocking

Of the messages forwarded to this portal, this is the amount of those that were discarded because they were blocked from sending, and were unable to be buffered. Message rates on other portals probably exceeded the rate controller limit on this portal.

Transport topic fragments/bytes dropped due to error

Of the messages forwarded to this portal, this is the total of those that were discarded due to an application or network connection failure.

Transport topic fragments/bytes dropped due to fragment size error

Of the messages forwarded to this portal, this is the total of those that were discarded possibly because of a configuration error. If this count is not at or near 0, verify that maximum datagram size for all transports is the same throughout the network.

Immediate topic fragments/bytes forwarded

The total number of Multicast Immediate Messaging (MIM) messages or message fragments, and bytes (second column), forwarded to this portal from other portals in this UM Router.

Immediate topic fragments/bytes sent

Of the MIM topic messages forwarded to this portal, this is the amount of traffic sent out to the TRD.

Immediate topic request fragments sent

Of the MIM topic messages sent, this is the amount of those that are requests.

Immediate topic fragments/bytes dropped due to blocking

Of the MIM topic messages forwarded to this portal, this is the amount of those that were discarded because they were blocked from sending, and were unable to be buffered. Message rates on other portals probably exceeded the rate controller limit on this portal.

Immediate topic fragments/bytes dropped due to error

Of the MIM topic messages forwarded to this portal, those that were discarded due to an application or network connection failure.

Immediate topic fragments/bytes dropped due to fragment size error

Of the MIM topic messages forwarded to this portal, those that were dropped possibly because of a configuration error. If this count is not at or near 0, verify that maximum datagram size for all transports is the same throughout the network.

Immediate topicless fragments/bytes forwarded

The total number of Multicast Immediate Messaging (MIM) messages or message fragments, and bytes (second column), with null topics, forwarded to this portal from other portals in this UM Router.

Immediate topicless fragments/bytes sent

Of the MIM topicless messages forwarded to this portal, this is the amount of traffic sent out to the TRD.

Immediate topicless request fragments sent

Of the MIM topicless messages sent, this is the amount of those that are requests.

Immediate topicless fragments/bytes dropped due to blocking

Of the MIM topicless messages forwarded to this portal, this is the amount of those that were discarded because they were blocked from sending, and were unable to be buffered. Message rates on other portals probably exceeded the rate controller limit on this portal.

Immediate topicless fragments/bytes dropped due to error

Of the MIM topicless messages forwarded to this portal, those that were discarded due to an application or network connection failure.

Immediate topicless fragments/bytes dropped due to fragment size error

Of the MIM topicless messages forwarded to this portal, those that were dropped possibly because of a configuration error. If this count is not at or near 0, verify that maximum datagram size for all transports is the same throughout the network.

Unicast messages/bytes forwarded

The total number of Unicast Immediate Messaging (UIM) messages (and bytes, second column), both control and containing user data, forwarded to this portal.

Unicast messages/bytes sent

Of the UIM data messages forwarded to this portal, this is the amount of traffic sent out to the TRD.

Unicast messages/bytes dropped due to error

Of the UIM data messages forwarded to this portal, those that were discarded due to an application or network connection failure.

Current/maximum data bytes enqueued (limit: *n*)

For bytes in this portal's send buffer (due to a blocking send), the first column is a snapshot of the current amount, and the second column is a high-water mark. The displayed limit (*n*) is the configuration value for option `<max-queue>`.

Peer Portal Page

This page allows you to see Receive and Send statistics for the selected peer portal. Click on any of the links at the top of the page to review configuration option values for the portal's UM topic resolution domain.

The peer portal page displays the following statistics:

Peer Portal *name*

Portal Cost

The cost value assigned to this portal.

Interest

Totals (listed below) for topics and patterns in this portal's interest list that originated from receivers beyond and downstream from the immediately adjacent UM Router.

Topics

Of the interest total, the number of topics.

PCRE patterns

Of the interest total, the number of wildcard patterns, using PCRE pattern matching.

REGEX patterns

Of the interest total, the number of wildcard patterns, using REGEX pattern matching.

Proxy Receivers

The number of proxy receivers active in this portal.

Receiver topics

All topics in which the other portals in the UM Router have detected current interest and summarily propagated to this portal.

Receiver PCRE patterns

All wildcard patterns, using PCRE pattern matching, in which the other portals in the UM Router have detected current interest and summarily propagated to this portal.

Receiver REGEX patterns

All wildcard patterns, using REGEX pattern matching, in which the other portals in the UM Router have detected current interest and summarily propagated to this portal.

Proxy Sources

The number of proxy sources active in this portal.

Peer Receive Statistics

Data messages/bytes received

The total of messages containing data received at this portal. The first column counts the number of fragments (or whole messages for messages that were not fragmented).

Transport topic fragment data messages/bytes received

The total of user-data messages received on any topic resolved through this portal. The first column counts the number of fragments (or whole messages for messages that were not fragmented).

Transport topic fragment data messages/bytes received with unknown source

Topic messages received whose source this UM Router has not seen before.

Transport topic request fragment data messages/bytes received

These are topic messages received that are request messages, i.e., messages send via `lbm_send_request*()` rather than `lbm_src_send*()`.

Transport topic request fragment data messages/bytes received with unknown source

Of the request messages received, the topic messages received whose source this UM Router has not seen before.

Immediate topic fragments/bytes received

The total number of Multicast Immediate Messaging (MIM) messages or message fragments, and bytes (second column), that have a topic, received by all proxy receivers at this portal.

Immediate topic request fragments/bytes received

Of the MIM topic messages received, this is the total of those that are requests.

Immediate topicless fragments/bytes received

The total number of MIM messages or message fragments, and bytes (second column), with null topics, received by all proxy receivers at this portal.

Immediate topicless request fragments/bytes received

Of the MIM topicless messages received, this is the total of those that are requests.

Unicast data messages/bytes received

The total number of Unicast Immediate Messaging (UIM) messages (and bytes, second column) containing user data, received by this portal.

Unicast data messages/bytes received with no stream information

UIM data messages discarded because they were from an earlier, incompatible version of UM. This tally should stay at 0; otherwise, contact Informatica Support.

Unicast data messages/bytes received with no route to destination

UIM data messages that are on a wrong path, possibly due to a route recalculation. This tally should stay at 0, though it may increment a few messages at the time of a topology change.

Control messages/bytes received

The total of supervisory messages (containing no data) received at this portal.

Transport topic control messages/bytes received

Of the control messages received, those that are transport/topic based (such as TSNIs, SRIs., etc.).

Transport topic control messages/bytes received with unknown source

Of the transport/topic control messages received whose source this UM Router has not seen before.

Unicast control messages/bytes received

The total number of Unicast Immediate Messaging (UIM) supervisory (non-data) messages (and bytes, second column) received by this portal.

Retransmission requests/bytes received

Supervisory UIMs that are requests for retransmission of lost (or Late Join) messages.

Control messages/bytes received with no stream info

Supervisory UIMs discarded because they were from an earlier, incompatible version of UM. This tally should stay at 0; otherwise, contact Informatica Support.

Control messages/bytes received with no route to destination

Supervisory UIM messages that are on a wrong path, possibly due to a route recalculation.

Gateway control messages/bytes received

The total of UM Router-only, peer-to-peer supervisory messages received at this portal.

Unhandled control messages/bytes received

Supervisory UIMs discarded because, though they are well-formed, they have no valid action request. This tally should stay at 0; otherwise, contact Informatica Support.

Peer Send Statistics

Transport topic fragments/bytes forwarded

The total transport-based topic-related traffic forwarded to this portal from other portals in this UM Router. This could include user messages, TSNIs, SRIs., etc. The first column counts the number of fragments (or whole messages for messages that were not fragmented).

Transport topic fragments/bytes sent

Of transport topic messages forwarded to this portal, the amount of traffic sent to the adjacent UM Router.

Transport topic request fragments/bytes sent

Of transport topic messages sent, those that were request messages.

Transport topic fragments/bytes dropped (duplicate)

Of transport topic messages forwarded to this portal, messages discarded because they were duplicates of messages already received.

Transport topic fragments/bytes dropped (blocking)

Of transport topic messages forwarded to this portal, this is the amount of those that were discarded because they were blocked from sending, probably due to TCP flow control, and were unable to be buffered. The UM Router's XML configuration file may need to be adjusted.

Transport topic fragments/bytes dropped (not operational)

Of transport topic messages forwarded to this portal, messages discarded because the peer link is down.

Transport topic fragments/bytes dropped (queue failure)

Of transport topic messages forwarded to this portal, messages discarded due to a memory allocation failure.

Unicast messages/bytes forwarded

The total number of supervisory (no data payloads) Unicast Immediate Messaging (UIM) messages (and bytes, second column) forwarded to this portal from other portals in this UM Router. These messages can be either control (supervisory) messages or contain user data.

Unicast messages/bytes sent

Of the UIMs forwarded to this portal, the amount of traffic sent to the adjacent UM Router.

Unicast messages/bytes dropped (blocking)

Of the UIMs forwarded to this portal, this is the amount of those that were discarded because they were blocked from sending, probably due to TCP flow control, and were unable to be buffered. The UM Router's XML configuration file may need to be adjusted.

Unicast messages/bytes dropped (not operational)

Of the UIMs forwarded to this portal, messages discarded because the peer link is down.

Unicast messages/bytes dropped (queue failure)

Of the UIMs forwarded to this portal, messages discarded due to a memory allocation failure.

Gateway control messages/bytes sent

The total number of UM Router supervisory messages (and bytes, second column), generated at this portal.

Gateway control messages/bytes sent

Of the UM Router supervisory messages generated, the number sent to the adjacent UM Router.

Gateway control messages/bytes dropped (blocking)

The amount of UM Router supervisory messages that were discarded because they were blocked from sending, probably due to TCP flow control, and were unable to be buffered. The UM Router's XML configuration file may need to be adjusted.

Gateway control messages/bytes dropped (not operational)

The amount of UM Router supervisory messages that were discarded because the peer link was down.

Gateway control messages/bytes dropped (queue failure)

The amount of UM Router supervisory messages that were discarded due to a memory allocation failure.

Batches

The number of times messages were batched.

Minimum messages/bytes per batch

The lowest recorded number of messages in a batch, and the number of bytes in that batch.

Average messages/bytes per batch

The average number of messages in a batch, and the number of bytes in that average batch.

Maximum messages/bytes per batch

The highest recorded number of messages in a batch, and the number of bytes in that batch.

Current/maximum data bytes enqueued

For bytes in this portal's send buffer (due to a blocking send), the first column is a snapshot of the current amount, and the second column is a high-water mark. The displayed limit is the configuration value for option “<max-queue>” on page 62.

Keepalive/RTT samples

The number of keepalive messages that have been set to the other UM Router's portal and responded to.

Minimum RTT (microseconds)

Of the keepalives sent and responded to, the lowest recorded round-trip time.

Mean RTT (microseconds)

Of the keepalives sent and responded to, the mean recorded round-trip time.

Maximum RTT (microseconds)

Of the keepalives sent and responded to, the highest recorded round-trip time.

Last keepalive responded to

The send timestamp (date and time) of the last sent keepalive message that was responded to.

Topology Info Page

This page allows you to see UM Router network connectivity information from the perspective of this UM Router. The **Other UM Routers** section provides information in the same format as is used for the local UM Router.

Local UM Router Name

The UM Router name as assigned via configuration.

Local UM Router ID

A unique value that the UM Router assigns to itself automatically.

Self Version

A configuration version for this UM Router, as seen collectively by the UM Router network.

Topology Signature

An identifier for the "map" of this UM Router network's routes. This value should be the same for all UM Routers.

Last recalc duration

The amount of time in seconds that it took this UM Router to perform its most recent route recalculation.

Graph Version

The number of times this UM Router has updated its view of the topology.

UM Router Count

The number of UM Routers in this UM Router network.

Topic Resolution Domain Count

The number of TRDs in this UM Router network.

Portal (endpoint or peer)

This display is repeated for each portal of this UM Router.

Portal Name

The portal's name as assigned via configuration.

Adjacent Domain/UM Router ID

For an endpoint portal, this is the configured `<domain-id>` for the connected TRD. For a peer portal, this is an automatically assigned unique identifier for the connected UM Router.

Cost

This portal's configured cost.

Last interest recalc duration

The amount of time in seconds that it took this UM Router to perform a recalculation that resulted in an update to the interest status for this portal.

Last proxy receiver recalc duration

The amount of time in seconds that it took this UM Router to perform recalculation that resulted in an update to the status of proxy receivers (create, maintain, or destroy) for this portal.

Other UM Routers

This display is repeated for each other UM Router in this UM Router's network.

UM Router Name

The UM Router name as assigned via configuration.

UM Router ID

A unique value that the UM Router assigns to itself automatically.

Version

A configuration version for the UM Router, as seen collectively by the UM Router network.

Topology Signature

An identifier for the "map" of this UM Router network's routes. This value should be the same for all UM Routers.

Last Activity *n* seconds ago

How long since the last time this local UM Router received a route info packet from the designated "other" UM Router.

Adjacent Domain ID

The configured ID of one of this "other" UM Router's connected TRD, plus the cost assigned to the associate endpoint portal. If there are more than one endpoint portals in the UM Router, this line is repeated for each.

Adjacent UM Router ID

The automatically assigned ID of one of this "other" UM Router's connected UM Router, plus the cost assigned to the associate peer portal. If there are more than one peer portals in the UM Router, this line is repeated for each.

Path Info

The Path Info page lets you query and display a hop path that messages will take between any two TRDs that you enter into the Domain ID 1 and Domain ID 2 text boxes. Fill in the boxes and click the **Calculate Shortest Path** button, and you see the following fields:

Hop Count

The number of hops from none node to the next along the displayed route, where a node can be either a UM Router or a TRD.

Aggregate Cost

A sum of the cost values of all portals along the displayed path.

Path

A display of the UM Router and TRD hops listed in route order from the starting TRD to the ending TRD.

UM Router Log Messages

The UM Router daemon can generate error or informational log messages. You can configure these to be directed to standard output (console, default), syslog, or a specified log file, via the `<log>` configuration element. The UM Router does not over-write log files on startup, but instead appends them.

The following are important UM Router log messages.

Connection Failure Messages

```
peer portal [name] failed to connect to peer at [IP:port] via [interface] [err]: reason
```

```
peer portal [name] failed to accept connection (accept) [err]: reason
```

Lost Connection Messages

```
peer portal [name] lost connection to peer at [IP:port] via [interface]
```

```
peer portal [name] connection destroyed due to socket failure
```

```
peer portal [name] detected dropped inbound connection (read) [err]: reason
```

```
peer portal [name] detected dropped inbound connection (zero-len read)
```

Endpoint Messages

If a UMP store is adjacent to the UM Router, and the UM Router has been restarted, you typically see messages of the form

```
endpoint portal [name] has no forwarding entry for destination ctxinst [string], dropping
msg (lbmc cntl ume)
```

These messages are normal, and cease when the UM Router has established the forwarding information for the given context.

Peer Messages

Dual TCP:

```
peer portal [name] received connection from [IP:port]
```

```
peer portal [name] connected to [IP:port]
```

Single TCP:

```
Acceptor: peer portal [name] received connection from [IP:port]
```

```
Initiator: peer portal [name] connected to [IP:port]
```

UM Router Transport Statistics

Using the `<monitor>` element in a UM Router's XML configuration file and the UMS Monitoring feature, you can monitor the transport activity between the UM Router and its Topic Resolution Domain. The configuration also provides Context and Event Queue statistics. The statistics output identifies individual portals by name.

APPENDIX A

Glossary

Access Control List (ACL)

A portal configuration you can use to filter out messages based on a variety of criteria.

forwarding cost

A value assigned to a portal to help determine best-path routing selection.

UM Router keepalive

Messages exchanged between UM Routers to confirm that UM Routers are still running.

Interest Message

Messages exchanged between UM Routers to confirm that UM Routers are still running.

Originating Transport ID (OTID)

Unique identifier of a message's transport session at the originating source.

portal

A TCP/IP interface (socket) on a UM Router through which the UM Router passes data. Endpoint portals interface TRDs, and peer portals interface peer portals of other UM Routers.

Topic Resolution Domain (TRD)

The realm of UDP multicast or unicast connectivity that allows UM topic resolution to occur. Blocking of this UDP connectivity (for example, by a firewall or a restrictive WAN link) defines a TRD's boundaries. Contexts within a TRD must have the same topic resolution configuration option settings (multicast group IP address/port and resolver interface full or CIDR address).

Use Query

A periodic message multicasted out to a TRD to verify the continued presence of receivers for a given topic or pattern.

web monitor

A web-based realtime UM Router statistics and configuration display.