Solace messaging concepts

**Core Concepts**

In this section we'll discuss several concepts at the core of messaging technology and Solace message routers:

* [Messaging](#_Messaging)
  + [Message Delivery Modes](#_Message_Delivery_Modes)
  + [Message Models](#_Message_Models)
* [Solace Message Router Concepts](#_Solace_Message_Router)
* [Solace API Concepts](#_Solace_API_Concepts)

Also, you can find discussions about the operational advantages of using the Solace message router's multi-protocol and cloud native capabilities here:

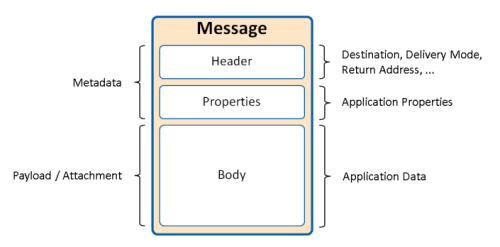
* [Multi-Protocol](http://dev.solace.com/tech/#multi-service)
* [Cloud Native](http://dev.solace.com/clouds/)

## **Messaging**

In application development terms, messaging (also commonly known as message-oriented middleware or just middleware) refers to technology that lets computer systems share information without requiring direct connections or awareness of one another’s location.

In many ways messaging is like postal/shipping services—just like you hand your letter or many packages to the carrier and trust that it will get where you want it to go, in messaging exchanges applications hand off information to the messaging system that routes it to whatever applications, analytics engines or user interfaces you’ve said you want it to get to. The analogy holds up when you consider different kinds of shipping and qualities of service (first class vs. bulk mail, ground vs. overnight, local vs. overseas, and so on) as different kinds of messaging are used for different kinds of interactions, such as publish/subscribe, request/reply… more on that below.

Clients interact by sending and receiving Messages, but what does a message look like? Fundamentally, the Message has three parts: Header, Properties and Body.



### Header

The Header is the part of the message used by the Solace Message Router for routing messages through the system. Some header fields, such as Destination and Delivery Mode, are required, while others are optional. For example, the reply-to topic is optional and is only needed for Request-Reply messaging where the Replier needs to know where to send its responses.

### Properties

In addition to the header fields, application-specific properties can also be included as part of the message. These application-specified properties are unmodified by the broker and can be used to facilitate communication between applications. Some APIs define their own standard properties. For instance, JMS has defined properties such as JMSXUserID and JMSXAppID which can be used to identify the user or application sending the message. Besides API-defined standard properties, applications can also specify properties. An example of an application-specified property might be an Order ID an Order Management System uses as a unique identifier to track all messages related to a given purchase.

### Body

The body of a message is often called the payload or the attachment. It contains data in an application specified format and is transported unmodified by the Solace message router. The body is made up of raw binary data or structured data such as text, streams, maps, or otherwise. Regardless, Producers and Consumer must agree upon the payload format so that the data can be properly interpreted.

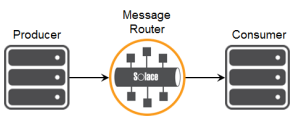
## **Message Delivery Modes**

Solace message routers support the following different message delivery modes:

* At most once delivery—Solace message routers provide two options: Direct and Non-Persistent message delivery. See details below for the differences.
* At least once delivery—Solace message routers provide this through Persistent messaging. At Solace this is often called Guaranteed delivery.
* Transacted delivery—Solace message routers support session based and XA transactions.

### Direct

Direct messaging is meant for high-speed applications that can tolerate occasional message loss. Producers can publish messages to a topic, and when these messages are received by the Solace Message Router, they are delivered to Consumers with matching topic subscriptions. Direct messages are neither spooled on the Solace Message Router (that is, they are not persisted) for consuming clients nor acknowledged when received.



**Learn More:**

* [Working With Direct Messages](#_Working_With_Direct)

### Non-Persistent

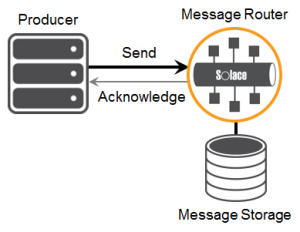
The Non-Persistent messaging delivery mode in Solace is used mostly to fulfill the requirements of the JMS specification. It is also used in message promotion and demotion, which is explained below. In general, it is not used directly by client applications when publishing.

### Persistent (Guaranteed)

Persistent or Guaranteed messaging is combination of sending persistent messages with proper publish confirmation handling and means at least once delivery. Guaranteed messages are never lost. Guaranteed messaging can be looked at from two angles: the publish side and the receive side.

### Guaranteed Message Publishing

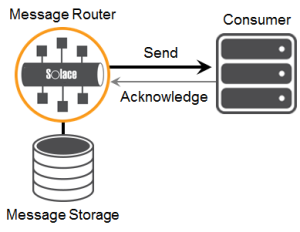
When Persistent Messaging is used to send messages to a Solace message router, clients can publish to either a Solace queue or topic.



When these messages are received by the Solace message router, they are saved in the Solace message router message spool (that is, they are persisted) before they are acknowledged back to the producers. The acknowledgment is the confirmation to the publisher that the Solace message router accepts the message and it will not be lost.

### Guaranteed Message Receiving

The messages are delivered to consuming clients that bind to the Solace endpoints containing the received messages. Messages are persisted on the Solace message router until they expire or the consuming client acknowledges the message indicating it has been consumed.



**Learn More:**

* [Working With Guaranteed Messages](#_Working_With_Guaranteed)

### Message Promotion / Demotion and Message Delivery Modes

In the previous explanations of Direct and Persistent messages, both the producer and consumer in the examples wanted the same message delivery mode. This is the simplest scenario, but it is not a requirement. The Solace message router matches messages based on topics or queue names and routes these messages to interested consumers. So it is possible to use different message delivery modes for the producer and consumer and end up with a hybrid of Direct messaging and Persistent messaging.

Message promotion is the scenario where the producer sends Direct messages and the consumer receives these message from a Guaranteed Messaging endpoint. In this case the consumer will receive Non-Persistent Solace messages. This is a typical scenario in applications where there is a real-time publisher sending in events, and it is critical that this publisher never be back pressured. However, there are some consumers that would like to receive the data in the most fault-tolerant Persistent way. In this scenario they can receive the events using a queue endpoint with mapped topics. If the consumer application is ever offline, then messages will accumulate in the Queue Endpoint. The consumer application will not miss messages if it's offline or slow.

Message demotion is the scenario where the producer sends Persistent messages and there are consumers that want to receive these messages but can tolerate lost messages. These consumers can add a Topic Subscription matching these messages and receive them in real time as Direct messages. These consumers will never back pressure the publisher application, and if they go offline those consumers will simply miss messages. In summary, the producer application is sending mission critical data but some consumers of this data are not mission critical and should never be allowed to affect the publishers. Message demotion achieves this requirement.

From a consumer’s point of view, the only way to ensure fully guaranteed messaging is if the message is received from an Endpoint with a delivery mode of Persistent. That means it was sent persistently and received on a Guaranteed Messaging Endpoint. All other combinations are examples of at-most-once messaging and the Consumer will receive them as either Solace Direct or Non-Persistent messages.

| **Delivery Mode of Published Message** | **Received by Endpoint as…** | **Received by a Client with a Matching Topic Subscription as…** |
| --- | --- | --- |
| **Direct** | Non-Persistent | Direct |
| **Non-Persistent** | Non-Persistent | Direct |
| **Persistent** | Persistent | Direct |

The following is a summary table outlining message promotion and demotion in Solace message routers.

**Learn More:**

* [Topic Matching and Message Delivery Modes](#_Topic_Matching_and)

### Message Delivery Modes Terminology

Different protocols and APIs use different terminology to describe delivery modes. The following tables provides a summary across the various protocols supported by the Solace message router.

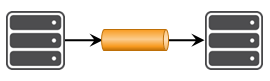
| **Message QoS** | **Solace APIs** | **JMS** | **MQTT** | **AMQP** | **Solace REST** |
| --- | --- | --- | --- | --- | --- |
| **At most once delivery** | Direct (default) | Non-Persistent | QoS 0 (default) | Direct for publishers only  (Direct subscribers not supported) | Direct |
| **At least once delivery** | Persistent | Persistent | QoS 1 | Persisted | Persistent (default) |
| **Transacted** | Session Based | Session + XA | N/A | N/A | N/A |

## **Message Models**

Most messaging applications can be reduced to a series of interactions that adhere to one of the following core messaging models. The Point-to-Point, Publish-Subscribe, and Request-Reply Message Exchange Patterns are the building blocks for describing the manner in which applications interact.

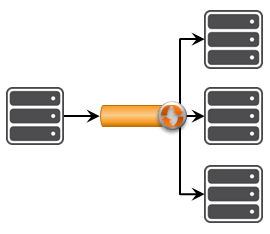
### Point-to-Point

With Point-to-Point messaging, messages sent by the Producer are to be processed by a single Consumer.



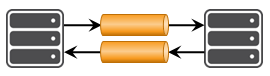
### Non-Exclusive Consumption

An extension of the traditional Point-to-Point messaging is the addition of multiple consumers consuming from a shared channel or queue. The scale of the overall receiving application is increased by having multiple consumers, but each message is still only delivered to a single consumer.



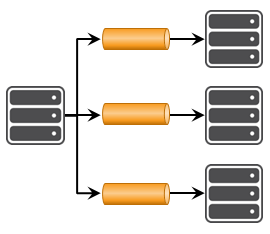
### Request-Reply

With request-reply messaging, applications achieve two-way communication using separate point-to-point channels: one for the requests and another for the replies.



### Publish – Subscribe

With publish-subscribe messaging, messages sent by the Producer are processed multiple times by different consumers. Each consumer receives its own copy of the message for processing.



## **Solace Message Router Concepts**

Solace message routers unify many kinds of data movement so companies can efficiently and cost-effectively move all of the information associated with better serving customers and making smarter decisions. Basically, they are multi-protocol message brokers and come in several form factors.

### Message Router Appliances

Solace message router appliances offer greater capacity and performance with lower TCO and complexity than any other messaging middleware technology. [Learn more here](http://dev.solace.com/tech/message-router-appliances/).

### Solace Virtual Message Router

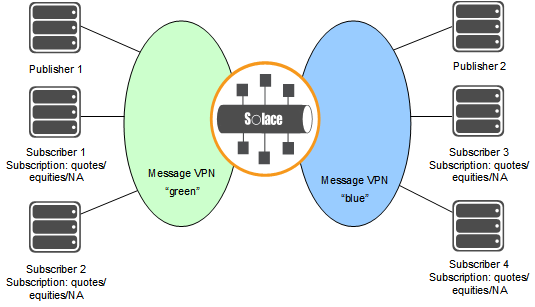
The Solace Virtual Message Router (VMR) is SolOS messaging technology that runs on general purpose processors instead of Solace purpose-built hardware. [Learn more here](http://dev.solace.com/tech/virtual-message-router/) or [download the Solace VMR here](http://dev.solace.com/downloads/).

### Message VPNs

The Message VPN is a core feature of the Solace message router. It allows for many separate applications to share a single message router while still remaining independent and separated. In essence it enables virtualization of the Solace message router into many individual virtual message brokers.

Message VPNs allow for the segregation of topic space and messaging space by creating fully separate messaging domains. Message VPNs also group clients connecting to a network of Solace message routers, such that messages published within a particular group are only visible to clients that belong to that group. Each client connection is associated with a single Message VPN.

Message VPNs also allow scoping of resources such as message spool, queues, topics, connections.



**Learn More:**

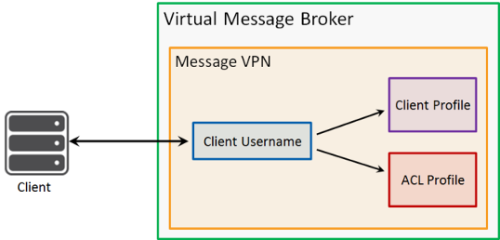
* [Working with Message VPNs](https://docs.solace.com/Features/Working-with-Message-VPNs.htm)

### Clients

Applications or devices that connect to Solace message routers are represented as clients. A client on the router represents a single logical connection to the router that is capable of sending and/or receiving messages. To establish a client connection to a Solace message router, client applications or devices connect using a specific Client Username account.

As part of the connection process, a unique Client Name is required to identify the Client’s session. That is, the Client Name serves as a session name. If the Client application or device does not provide its own Client Name when creating its session, a unique Client Name is automatically generated by the Solace API. Note that while Client Names are unique within a given Message VPN, the same name may exist in multiple Message VPNs.

Once connected, Clients inherit a number of characteristics, behaviors, and permissions based on a Client Profile and an ACL Profile associated with that Client Username.



### Client Username

Client Usernames are contained within a specific Message VPN. They are used by Client applications to authenticate with the Solace message router and may be used to make multiple Client connections. This lets applications horizontally scale without additional configuration on the Solace message router.

Each Client Username is associated with a Client Profile and an ACL profile. These profiles control properties and permissions of the connected Client and will be discussed later in this lesson.

**Learn More:**

* [Client Authentication/Authorization](https://docs.solace.com/Features/Client-authentication-and-authorization.htm)

### Client Profile

Client Profiles are contained within a specific Message VPN and can be applied Client Usernames. Being able to share Client Profiles gives administrators the ability to manage large groups of Clients without having to make individual changes to each one.

Client Profiles control a number of behaviors and capabilities. For example, resource allocation such as the maximum number of Client connections per Client Username and the per-client transport queues are both controlled by the Client Profile. Other characteristics controlled by the Client Profile include tuning TCP connection parameters, enabling persistent messaging capabilities, and adjusting the point at which certain events will be triggered.

**Learn More:**

* [Controlling Client Access with ACL Profiles](https://docs.solace.com/Features/Granting-Clients-Access.htm)

### ACL Profiles

Similar to Client Profiles, ACL Profiles are contained within a specific Message VPN and can be applied to Client Usernames. This gives administrators the ability to control the entitlements for large groups of clients.

ACL Profiles consist of a few Access Control Lists which serve to either allow or restrict Client actions. The ACL configuration defines the Client’s ability to connect based on its IP address. It also defines to which topics the Client is allowed to publish or subscribe.

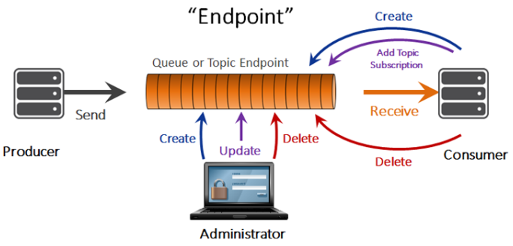
All of the ACL definitions have the same format in that they have a default behavior of either allow or disallow, and they have an exception list.

**Learn More:**

* [Configuring Clients with Client Profiles](https://docs.solace.com/Features/Assigning-Client-Profiles.htm)

### Endpoints / Queues

Guaranteed messages in Solace are stored in endpoints on the Solace message router. There are two types of endpoints; queues and topic endpoints. In most scenarios the queue endpoint is a superset of the topic endpoint and is the endpoint most commonly used. Topic endpoints are used by JMS (that is, durable topic subscriptions).



Queues on the Solace message router work very much like queues in all message queuing systems. Producers send guaranteed messages to the Solace message router. These messages are saved in the queue and delivered to consuming clients if they are online and connected. The consumer acknowledges the message once it has completed processing of the message, and it is then removed from the queue on the Solace message router. Queue objects are highly configurable with a variety of options to tailor their behavior to application needs.

Queues' lifecycles are determined by whether they are durable or non-durable. Durable queues remain around permanently until removed through configuration. They accumulate messages when clients are online or offline. When offline clients connect again, they receive all the messages that were accumulated while they were offline. Temporary queues follow the lifecycle of the client connection and are useful for ensuring message persistence while clients are online. If a client disconnects for a long period of time, the Solace message router will automatically remove the temporary queues. So messages will not accumulate while clients are offline.

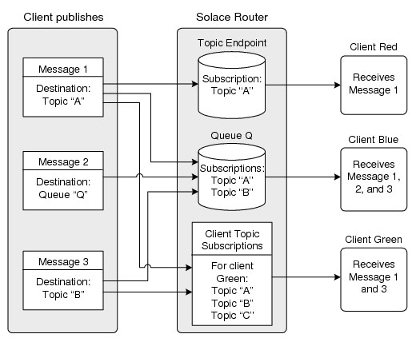
**Learn More:**

* [Working With Guaranteed Messages](https://docs.solace.com/Features/Working-With-Guaranteed-Messages.htm)

### Topic to Queue Mapping

In addition to spooling published Guaranteed messages that have a matching queue destination, it is possible to add one or more topic subscriptions to a durable queue so that Guaranteed messages published to those topics are also delivered to and spooled by the queue. This feature enables queues to participate equally in point-to-point and publish/subscribe messaging models, opening up a large number of interesting use cases.

As shown in the following figure, this allows a single message published to a topic to be delivered to a combination of topic endpoints, one or more queues, or even clients with matching Direct Messaging topic subscriptions.



Any Guaranteed messages published to topics that match subscriptions associated with queues are delivered to those queues. Error indications are returned to the publisher if the message cannot be delivered to one or more queues for any reason (that is, the feedback to the publisher is identical to that provided when the messages are published directly to the queues).

* Topic subscription to queue mappings are applicable to both durable and non‑durable queues. Deletion of a queue for any reason results in the deletion of all topic subscription to queue mappings for that queue.

**Learn More:**

* [Adding Topic Subscriptions to Queues](https://docs.solace.com/Features/Endpoints.htm#add-topics-to-queues)

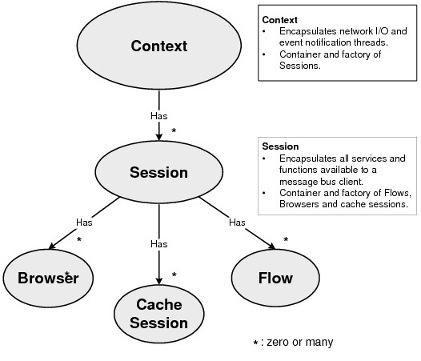
## Solace API Concepts

The following section provides an overview of Solace API core concepts.

### Contexts

The messaging APIs use processing Contexts for organizing communication between an application and a Solace appliance. Contexts act as containers in which Sessions are created and Session-related events can be handled.

A Context encapsulates threads that drive network I/O and message delivery notification for the Sessions and Session components associated with that Context. For the Java API, one thread is used for I/O and another for notification. For the Java RTO, C, and .NET APIs, a single thread is used for both I/O and for notification. The life cycle of a Context‑owned thread is bound to the life cycle of the Context.

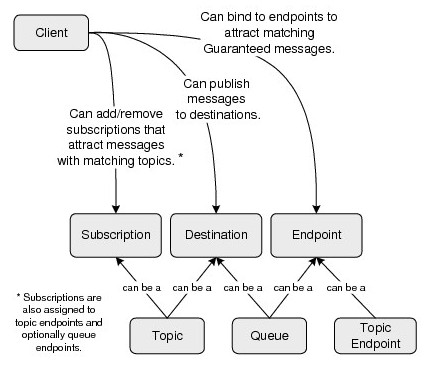


**Learn More:**

[Contexts](https://docs.solace.com/Solace-Messaging-APIs/Contexts.htm)

### Sessions

A Session creates a single, client connection to a Solace message router for sending and receiving messages. Sessions also allow applications to add and remove subscriptions.



**Learn More:**

[Sessions](https://docs.solace.com/Solace-Messaging-APIs/Sessions.htm)

### Flows

A Consumer Flow is an API concept that allows applications to receive Guaranteed messages from an endpoint, such as a Queue. It is created by a Solace session.

<https://docs.solace.com/Solace-Messaging-APIs/Core-Messaging-API-Concepts.htm>

# Working with Direct Messages

Direct messaging provides a reliable—but not guaranteed—delivery of messages from the Solace message bus to consuming clients.

Direct Messaging is the default message delivery system for the Solace Messaging Platform and does not require any configuration beyond that required to configure and start other features. Direct Messaging is always available by default to all clients connecting to the Solace router.

Direct messages have the following characteristics:

* they are delivered to subscribing clients in the order in which the publishers publish them
* they do not require subscribing clients to acknowledge receiving them
* they are not spooled on the message bus for consuming clients
* they are not retained for a client when it is not connected to a Solace router (that is, they are not spooled on the message bus for consuming clients)
* they can be discarded when congestion or system failures are encountered
* they can be reordered in the event of network topology changes when a multi‑node message routing is used

Direct Messaging is ideally suited for applications where:

* extremely high message rates and extremely low latency are required
* consuming clients can tolerate message loss in the event of network congestion
* messages do not need to be persisted for later delivery to slow or offline consumers
* messages must be efficiently published to a large number of clients with matching subscriptions

Direct messages are not delivered if:

* the client disconnects from the router while messages are being published
* the client fails to consume messages at the published rate and egress message buffer overflow results

## Load Balancing Messages with Deliver-To-One

Through the use of Solace messaging APIs, client applications can use the Deliver-To-One (DTO) feature of the Solace Messaging Platform to send a Direct delivery message to a single client even though there may be a number of clients with appropriate subscriptions that are capable of receiving the message.

The use of DTO allows for the load-balancing of received messages because only one connected client can consume a DTO message. (Note that, as a Direct message, if there are no connected clients, the router will not spool the message.)

**Note:**The DTO priority and behaviors are only relevant to messages published with the DTO flag set. Otherwise normal delivery rules apply (that is, all messages are delivered to all subscribers).

To control which client can receive a Deliver-To-One message, subscribing clients are assigned a local subscriber priority through a session property. The subscriber priority levels range from 1 (the highest) to 4 (the lowest).

A local priority is the priority that a client’s subscriptions have for receiving Deliver‑To-One messages published on the Solace router that the client is directly connected to.

When a Deliver-To-One message is published, a single local client with the highest subscriber priority level will receive the message. In a situation where the highest local subscriber priority level is shared by multiple clients, a single client is chosen in a round-robin fashion.

**Note:**When possible, a published DTO message is always delivered to a locally connected client. However, if there are no clients with assigned local subscriber priorities, the message will be delivered to a neighboring Solace router if one is available. Note that no attempt is made to evenly distribute DTO messages amongst remotely-connected routers should there be multiple options. This implies that a network of Solace routers will ensure single delivery of DTO messages amongst the appropriate clients connected to a network of Solace routers. However, load balancing can only be expected amongst clients connected to the same Solace router.

Clients can also override DTO delivery for specific topic subscriptions by applying the Deliver Always (DA) property to those subscriptions when they are added to the router. Topic subscriptions with the DA flag always receive all messages with topics that match the subscription—regardless of set priorities. Using the DA flag override can be useful if an application wants to take a copy of the message for auditing, replaying, or monitoring purposes. For example, a monitoring application should use the DA flag for every subscription it has so that it does not accidentally become a member of a DTO group and receive messages that should be delivered to other applications relying in DTO.

**Note:**As a special case, the subscription “>” (which matches all topics) is always treated as a DA subscription.

Subscriptions (with priority 1 to 4) and DA subscriptions that happen to share the same topic string are unique, meaning that a client may have both a DTO and DA subscription to the same topic. However, should a client be eligible to receive a given message due to both priority and DA subscriptions only a single copy of the message will be delivered.

The DTO priority of a subscription is shown by the **show smrp subscriptions** User EXEC command as either a value of P1 to P4 or the value DA to indicate a Deliver Always subscription.

[**[Open](javascript:void(0);) Example:**](javascript:void(0);)

solace> show smrp subscriptions  
  
Flags Legend:  
T - Destination Type (C=local-client, Q=local queue  
       R=remote-router)  
P - Subscription Persistence (P=persistent, N=non-persistent)  
R - Redundancy Type for Local Destinations (P=primary, B=backup  
       S=static -=not-applicable)  
  
Message VPN : default (exported: Yes; 100% complete)  
Destination Name       Flags BlkID DTO  Subscription  
                       T P R       Prio  
---------------------- - - - ----- ---- ------------------------------------  
#client                C N S     0   P1 #MCAST/>  
#client                C N S     0   P1 #SEMP/lab-129-4/>  
#client                C N S     0   P1 #P2P/lab-129-4/#client/>  
#client                C N S     0   P1 #P2P/v:lab-129-4/#client/>  
#client                C N S     0   P1 #SEMP/v:lab-129-4/>  
#client                C N S     0   P1 #P2P/lab-129-4/WnAHbdbR/#client/>  
dev2-180/316/#00000001 C N P    34   P1 #P2P/v:lab-129-4/27vgoQH9/dev2-180/316/#00000001/>  
dev2-180/316/#00000001 C N P    33   P1 a/b  
test000001             C N P    33   P3 #P2P/v:lab-129-4/YC4EzyB4/test000001/>  
testInstance           C N P    33   P1 #P2P/v:lab-129-4/0N7rz7R0/testInstance/>  
testInstance           C N P    33   DA a/b  
testInstance           C N P    33   P1 #P2P/CACHEINST/testInstance/>  
testInstance           C N P    34   P1 #P2P/CACHEINST/testCluster/>  
testInstance           C N P    35   P1 #P2P/CACHEINST/testCache/>  
  
Message VPN : testVpn (exported: No; 100% complete)  
Destination Name       Flags BlkID DTO  Subscription  
                       T P R       Prio  
---------------------- - - - ----- ---- ------------------------------------  
#client                C N S     -   P1 #MCAST/>  
#client                C N S     -   P1 #SEMP/lab-129-4/>  
#client                C N S     -   P1 #P2P/lab-129-4/#client/>  
#client                C N S     -   P1 #P2P/v:lab-129-4/#client/>  
#client                C N S     -   P1 #SEMP/v:lab-129-4/>  
#client                C N S     -   P1 #P2P/lab-129-4/WnAHbdbR/#client/>

## 

## Message Eliding

The message eliding feature enables client applications to receive only the most current Direct messages published to topics that they subscribe to, at a rate that they can manage, rather than queue up outdated messages. Message eliding can be useful in situations where there are slow consumers or where a slower message rate is required. Only Direct messages can be elided.

**Note:**Message eliding is not supported on Solace appliances that use a Network Acceleration Blade-0401EM (NAB-0401EM)

To use message eliding:

* Client applications must flag published message as eligible for message eliding.

**Note:**

* + Although only a Direct message can be elided, it is possible for a publishing client to flag a persistent or non-persistent message as eliding‑eligible. However, the message will not be elided unless the delivery mode of the message is changed to Direct. This could happen if the persistent or non-persistent message is published to a topic that matches a client’s topic subscription. In that case, to accommodate the client topic subscription, the message is changed to Direct; it can then be elided.
  + All messages published by MQTT client applications are treated as non‑eliding eligible.
* A receiving client application must be assigned a client profile through its client username that permits it to:
  + use message eliding
  + after the first message, receive subsequent messages with a time delay interval. This delay interval (configured in the client profile) controls on a topic-by-topic basis the rate of message updates sent to a client (for example, five messages per second per topic).
  + receive up to the maximum number of topics the router can track for performing the eliding function on each client connection (up to 32,000 per client as configured in the client profile, with a total of 2,000,000 per router; default is 256 per client). Once the maximum number of topics number is reached, the router ages out the elided topics for the client to prevent the consumption of more eliding resources than have been allocated for the connection. Eliding behavior then continues as if this were a new client connection, and a one-time Syslog event is generated on a per client basis.

**Note:**

* + Solace routers track the number of topics on a client connection dynamically. Whenever this number is below the set maximum, the eliding function is applied to the new incoming messages.
  + For subscribers with wild card subscriptions, each topic that matches the wildcard subscription is elided, up to the maximum number of subscriptions specified by the client profile.
  + It is recommended that consuming clients do not use discard indications when using message eliding. In a situation where a router’s egress priority queue for a client fills up with received messages, the oldest messages on the egress queue are discarded to make room for newly arriving messages, and the message at the head of the queue is flagged with the discard indication. However, if eliding is enabled, that message could be elided, and the client would not receive the discard indication.

### Functional Description

The following example explains how message eliding works:

1. Client P is publishing messages on topic T at a rate of one message per millisecond; each message is flagged as eliding-eligible.
2. Consuming Client S is assigned a client profile that has eliding enabled and has an eliding delay of 200 ms, which imposes a cap of five updates per second per topic.
3. When the first message, M1, for topic T arrives at the router for client S, it is sent to the client without any delay, assuming the TCP window is open.
4. When message M2 for topic T arrives a moment later at the router for subscriber S, it is held by the router but not sent to the client since there has already been a message (that is, M1) for this same topic sent within the 200ms delay time.
5. When subsequent messages (for example, M3, M4, etc.) arrive at the router for client S, each newer message replaces the previous message and is continuously held by the router.
6. Then 200ms after the previous message was sent for Client S, the held message is then sent to that client, assuming the TCP window is open.

### Application Use Cases

Use cases for this feature include:

* Eliding for congestion management.

A client application wants to receive every message provided it is able to keep up with the message flow. If the client cannot keep up, then any queued messages can be elided to provide only the most recent message for each topic. For this use case a delay interval time of 0 is set.

* Eliding for Message Rate Control.

A client application wants to receive at most five messages per second per topic. In this case, the Solace router rate controls the output of messages to it. For this use case a delay interval time of 200 ms (that is, five messages per second per topic) is set.

* Some example uses are:
* Streaming market data to human traders.

Even though market data updates might be published at a very high rate, humans can only deal with a few updates per second. The client always wants the latest information, but at a slower rate than the total feed rate. Message eliding can be used to limit the output stream to a few updates per topic per second.

* Controlling update rates to subscribers over a Wide Area Network (WAN).

Sometimes, WAN bandwidth versus receiver processing capacity require that only a subset of the entire feed rate be provided over the WAN. Message eliding can be used to control the message update rates to a client.

## Managing Client Subscriptions

Once clients are authenticated on the Solace router they can add and remove topic subscriptions for Direct messages published to the Message VPN to which the have connected. These topic subscriptions are not durable—the clients’ subscriptions are not maintained for them after they disconnect from the router.

A client typically performs the following steps for managing their own subscriptions:

1. The client connects to the Solace router and authenticates itself. At this point, the router has no subscriptions for that client.
2. The client adds subscriptions to the Solace router.
3. The client receives messages which matches the requested subscriptions.
4. When the client disconnects from router the router immediately removes all subscriptions associated with that client.

**Note:**The topics that a client is permitted to subscribe to can be limited by the ACL assigned to the client username account used by the client. For more information, refer to [Configuring Access Control Lists](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-Access-Control-Lists.htm).

### Managing Subscriptions on Behalf of Other Clients

A client application can manage subscriptions on behalf of other clients within a Message VPN when its client username is configured to be a Subscription Manager (see [Configuring Subscription Managers](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-Client-Usernames.htm#Configuring-Sub-Managers) ), which is useful for centralizing the assignment of the Message VPN's clients and services to direct messaging subscriptions. However, it should be noted that a Subscription Manager has no control over guaranteed messaging.

When a client is configured as Subscription Manager, the subscriptions that the subscription manager client is managing are then subject to the access control permissions configured on that client’s associated Access Control List (ACL) profile—the ACLs of destination clients are not used. Thus, the client configured as Subscription Manager can not add or remove subscriptions that its own ACL rules would deny the addition of. This prevents it from inadvertently adding or removing subscriptions that it is not entitled to.

When a client is configured as Subscription Manager, other clients within the Message VPN typically perform the following steps for managing their subscriptions:

1. The client connects to the Solace router.
2. The client notifies the Subscription Manager that it is ready to receive messages.
3. The Subscription Manager authenticates the client, determines the subscription set needed by the client, and adds subscriptions to the router on behalf of the client.

**Note:**Subscriptions added by the Subscription Manager on behalf of the client have the same subscription rate (subscriptions per second) as those added by the client directly for itself.

1. The client receives messages which match the associated subscription set.
2. Once the client is done, it disconnects and its associated subscription set is deleted from the router.

Once subscriptions have been added by a Subscription Manager on behalf of another client, they behave like any other subscriptions (for example, the subscriptions are removed if the client is disconnected). Disconnecting the Subscription Manager has no effect on the subscriptions already added by it.

For information on how clients using Solace APIs can act as Subscription Managers to add and remove subscriptions on behalf of others, see [Managing Topic Subscriptions on Behalf of Other Clients](https://docs.solace.com/Solace-Messaging-APIs/Managing-Subscriptions.htm#Manage-On-Behalf).

# Working with Guaranteed Messages

Guaranteed Messaging is a message delivery mode offered by the Solace Messaging Platform that guarantees the delivery of a message between two applications even in cases where the receiving application is offline, or there is a failure of a piece of network equipment. Once a Solace router has acknowledged receiving a Guaranteed message from a publisher, it is committed to delivering that message.

Guaranteed Messaging ensures that each published message is reliably delivered only once to a consuming client when its receipt is acknowledged by that consumer, and that messages are delivered in the order they are published. The platform manages transient network outages, router outages, and client disconnections so that publishers can be assured that once a message is accepted by the platform, it is delivered to recipients.

Guaranteed Messaging:

* keeps messages that are tagged as persistent or non-persistent (rather than Direct) across router restarts by spooling (that is, writing) them to persistent storage

**Note:**Although a persistent delivery mode is typically used for Guaranteed messages, a non-persistent delivery mode is provided to offer compatibility with Java Message Service (JMS) and to allow the delivery mode of the messages to be modified to accommodate the persistence requirements of an endpoint or a client subscription when there is a topic match (refer to [Topic Matching and Message Delivery Modes](https://docs.solace.com/Features/Topic-Matching-and-Delivery-Modes.htm)).

* keeps a copy of the message until successful delivery to all clients and downstream routers has been verified

Messages accepted by the platform through Guaranteed Messaging for delivery to clients are never lost, but might not get accepted if system resource limits are exceeded.

If an ingress message cannot be received by the platform (for example, if the spool quota is exceeded), the publisher is not acknowledged and the appropriate router statistic is incremented.

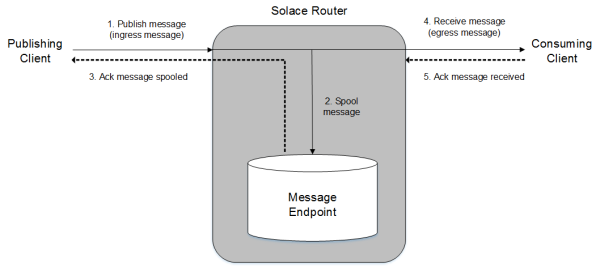
**Note:**

* To support Guaranteed Messaging, a Solace router must have message spooling enabled. A Solace appliance must also have an Assured Delivery Blade (ADB) installed.
* Guaranteed messages are not routed between separate Solace routers unless Message VPN bridges are specifically configured to link the Messages VPNs on those routers. Refer to [Working with Message VPNs](https://docs.solace.com/Features/Working-with-Message-VPNs.htm#working_with_message_vpns_1929913038_251090).

# Basic Guaranteed Messaging Operation

The figure below shows the basic steps of the Guaranteed Messaging process.

**Guaranteed Message Steps**



1. A client application that has connected to a Message VPN on a Solace router publishes a Guaranteed message (that is, a message with a delivery mode of persistent or non-persistent) to a topic or queue destination.
2. This ingress message received by the router is spooled to a queue or topic endpoint.
3. An acknowledgment that the message has been spooled successfully is sent back to the publishing client.
4. The router can deliver the spooled message to a consuming client if:
   * that client has successfully established a connection to the same Message VPN;
   * is authorized to receive Guaranteed messages;
   * has created a consumer flow in that session to bind to the endpoint;
   * and its flow is chosen to be the active flow for the endpoint (that is, it is the flow out of all of flows currently bound to that endpoint that can deliver messages)
5. After the router delivers the message to the consumer, an acknowledgment that the message has been delivered successfully is sent back to the router by the consuming client, and then the router deletes the message from the message spool.

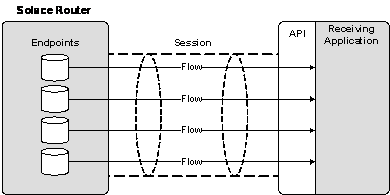
For more information on the message spooling and acknowledgment process mechanism, refer to [Message Spooling](https://docs.solace.com/Features/Message-Spooling.htm#Message_Spooling).

## Sessions and Consumer Flows

When applications are using Solace APIs, a session is used to provide a client connection between an application and a Solace router. A session uses a single TCP connection to the Solace router. A session is required for all client messaging operations: for example, a session is required to publish Guaranteed messages, and a session is required to create a consumer flow through which Guaranteed messages can be received.

A consumer flow can be created in a session to establish a one‑to‑one mapping of a consumer to a queue or topic endpoint. Consumer flows provide a means of transport for Guaranteed messages from the router to the consumer. One or more consumer flows can be created in a session.

**Consumer Flows**



**Note:**Consumer flows are not used to receive Direct messages. Instead, clients receive Direct messages published to a topic that matches the clients’ subscriptions directly from the router’s message bus through sessions.

**Publishing Guaranteed Messages**

When clients publish Guaranteed messages (that is, messages with a persistent or non-persistent delivery mode), they can either:

* publish the messages to a topic (used for publish and subscribe models)
* publish the messages to a queue (used for point-to-point models)

A Guaranteed message published to a topic is acknowledged by the Solace message bus to indicate that the system has received the message. A message that is published to a topic can be received by no clients, a single client, or multiple clients. If there is no subscriber for a message, that message is discarded because in this paradigm the producer and subscriber is decoupled.

The Solace message bus also acknowledges a Guaranteed message published to a queue to indicate that the system has received the message. However, messages published to a queue are more tightly coupled than those published to a topic. Therefore, if the queue does not exist or is not able to spool the message (perhaps if it is shutdown or over quota), then the Solace message bus rejects the message, and the publishing client application has this immediate feedback. A message that is published to a queue can be consumed by a single consumer.

When a client publishes Guaranteed messages in a session, a publisher flow to the host Solace router is automatically established. The published messages have persistence because they are spooled (that is, they are written to persistent storage) to one of the following types of endpoints provisioned on the router:

* a queue that matches the messages’ queue destination
* a topic endpoint that is assigned a topic subscription that matches the message’s topic destination

**Note:** It is also possible for messages published with a Direct delivery mode to be spooled to an endpoint when there is a match between the message’s topic destination and the topic subscription set for the endpoint. In this case, when the message is spooled to the endpoint, it is given a non-persistent delivery mode so that a client can consume the message over its established Guaranteed message Flow. For more information, refer to [Topic Matching and Message Delivery Modes](https://docs.solace.com/Features/Topic-Matching-and-Delivery-Modes.htm).

# Receiving Guaranteed Messages

To receive published Guaranteed messages, a client must create a consumer flow to bind to a queue endpoint or a topic endpoint provisioned on the router that the published messages spooled to.

Once the consumer is bound to the endpoint, and it has an active consumer flow (that is, a flow through which messages are being delivered), it can consume Guaranteed messages spooled to that endpoint.

When a consumer binds to a queue, it can be a exclusive or non-exclusive consumer of messages spooled to that endpoint (for information, refer to [Queue Access Types](https://docs.solace.com/Features/Endpoints.htm#Queue_Access_Types)). When a client binds to a topic endpoint, it provides the topic subscription that it wants the endpoint to spool messages for (the topic subscription that a topic endpoint will spool messages for is not specified when a topic endpoint is provisioned).

When a consuming client successfully receives a Guaranteed message, a corresponding application acknowledgment is required to indicate to the router that the client application received the message, and then the message can be removed from the endpoint.

**Transactions**

Transactions can be used to group a set of Guaranteed messages to be published and/or be consumed as an atomic unit of work.

Combining the messaging operations in an atomic unit ensures the integrity of messaging operations that depend on one another—if any single failure occurs, the whole transaction can be rolled back so that any published messages are deleted and any messages to be consumed remain on the endpoints they were spooled to. After a rollback, an administrator can then take corrective action to address the failure.

A Solace router can support transactions in both local transacted sessions or in transaction branches within an XA session that can be used in global distributed transactions:

* transacted sessions—Transacted sessions enable client applications to encapsulate multiple message send and/or receive operations together in single atomic transactions. Transacted sessions can be considered as local sessions, in that they only involve a single resource—the Solace router. Unlike transaction branches, they cannot be included in a global XA transaction that spans multiple network resources.

Transacted sessions are supported by the Solace Java, Java RTO, C, .NET, and JMS messaging APIs.

* XA transaction branches—Transaction branches are similar to transactions contained within a local transacted session in that they encapsulate multiple message send and/or receive operations as single atomic units, but transaction branches are to be used in a larger distributed transaction.
* Transaction branches are created within an XA session that the Solace router, which acts as a resource manager, handles. Each of these transaction branches can be included in distributed XA transactions that involve other transaction branches from separate, networked Java resources. It is the responsibility of remote Transaction Manager software to manage the transaction branches used in distributed transaction in a coordinated way.

XA transaction branches are only supported by the Solace implementation of the JMS API.

**Note:**Solace does not offer a Transaction Manager. However, to allow administrators to integrate the Solace Messaging Platform with Java Platform Enterprise Edition (Java EE) application server environments, Solace provides a Java Connector Architecture (JCA) v1.5 resource adapter for Red Hat JBoss EAP 6.2, WebSphere V7 and V8, as well as WebLogic 11gR1 PS5 and newer. The Solace JCA resource adapter is provided as standalone resource adapter archive (RAR) file that includes embedded versions of the Solace JMS API libraries. For more information, contact Solace support.

**Cut-Through Messaging**

Cut-Through Messaging allows for the delivery of Guaranteed messages with very low latency from the Solace Messaging Platform to consumers. This is done by using the low-latency, Direct Messaging data path for the bulk of the message flow, while also relying on the Guaranteed Messaging data path for message recovery in the event of a message loss.

Cut-Through Messaging is a controlled availability feature. For more information on using Guaranteed Messaging with a cut-through message mode, contact Solace support.

# Endpoints

The Solace router supports two types of endpoints for receiving, storing, and delivering Guaranteed messages:

* [Queues](https://docs.solace.com/Features/Endpoints.htm#Queues)
* [Topic Endpoints](https://docs.solace.com/Features/Endpoints.htm#Topic_Enpdoints)

**Note:**Endpoints are of local router significance only and are not a network‑wide concept insofar as a consumer must connect to the router that has the endpoint. There is no propagation through router-to-router control/routing protocols of the existence of endpoints. Thus, there is no enforcement of endpoint name uniqueness throughout a network by the routers.

## Queues

A queue acts as both a destination that clients can publish messages to and as an endpoint that clients can bind consumers to and consume messages from. A queue is typically used in a point-to-point (PTP) messaging environment.

Although many consumers can bind to a queue, an individual message spooled to a queue can only be consumed by a single consumer.

**Note:**It is also possible to add topic subscriptions to a queue so messages published to matching topics are delivered to the queue. (For more information, refer to [Adding Topic Subscriptions to Queues](https://docs.solace.com/Features/Endpoints.htm#add-topics-to-queues).) Therefore, it is also possible to use queues in a publish and subscribe (Pub/Sub) model.

Queues are significantly more flexible than topic endpoints and are the recommended approach for most applications. The use of topic endpoints should be restricted to JMS applications.

### Queue Access Types

A queue has an access type, which determines how messages are delivered when multiple consumer flows are bound to it. Queues can be assigned one of the following access types:

* exclusive—Only one consumer can receive a message at any one time, while additional consumers may be connected as standby. That is, only one Flow can be active. Only the first consumer to bind can receive messages. If the first consumer disconnects, the second consumer receives data, and so on. An exclusive queue always delivers messages in the order they are received.
* non‑exclusive—Multiple consumers can bind to a non-exclusive queue. Each consumer is serviced in round‑robin fashion. This provides load-balancing; however, if a connection fails, then unacknowledged messages are delivered to another consumer with the re-delivered flag set. In this way, messages can be delivered to consumers out of order.

**Note:**

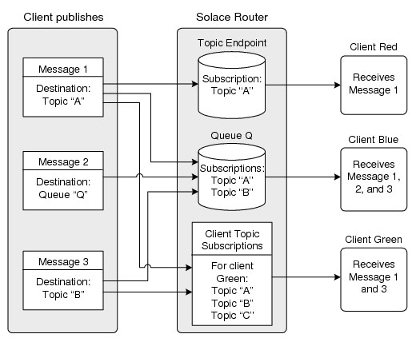
* The access type can only be changed for a durable queue and only when client access to that durable queue is disabled through the **shutdown** Queue CONFIG command.
* The access type assigned to a queue does not affect the ability of clients to browse the message on a queue with a queue browser.

### Adding Topic Subscriptions to Queues

You can add one or more topic subscriptions to a durable queue so that Guaranteed messages published to matching topics are also delivered to and spooled by the queue.

As shown below, this allows a single message published to a topic to be delivered to a combination of topic endpoints, one or more queues, or even clients with matching Direct Messaging topic subscriptions. (For information on how a message’s delivery mode can be changed to deliver a message to a client’s topic subscription, refer to [Topic Matching and Message Delivery Modes](https://docs.solace.com/Features/Topic-Matching-and-Delivery-Modes.htm).)

**Possible Routing of a Message According to its Topic**



Any Guaranteed messages published to topics that match subscriptions associated with queues are delivered to those queues. Error indications are returned to the publisher if the message cannot be delivered to one or more queues for any reason (that is, the feedback to the publisher is identical to that provided when the messages are published directly to the queues).

Topic subscription to queue mappings are applicable to both durable and non‑durable queues. Deletion of a queue for any reason results in the deletion of all topic subscription to queue mappings for that queue.

**Note:**

* It is also possible for published Guaranteed messages to be spooled to a queue if it is assigned a topic subscription that matches the message’s topic destination. For more information, refer to [Adding Topic Subscriptions to Queues](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-Queues.htm#Adding)
* A client cannot remove topic subscriptions from a queue that an administrator has added through the CLI or SEMP.

### Adding Topic Subscription Exceptions to Queues

In addition to topic subscriptions, you can add one or more topic subscription exceptions to a durable queue so that Guaranteed messages published to matching topics are discarded and not delivered to the queue.

Using topic subscription exceptions allows you to easily exclude specific topics from the set of topics delivered to a queue by adding a leading "!" character to the topic subscription. For example, if you add a topic subscription "animals/f\*" and a topic subscription exception "!animals/fox" to a queue, guaranteed messages published to "animals/frog" are delivered to the queue; however, messages published to "animals/fox" are discarded.

Topic subscription exceptions are enabled by default, and always take precedence over subscriptions regardless of the order in which the subscriptions and the exceptions are configured. To enable or disable this functionality on Solace routers refer to [System-Level Subscription Exception Configuration](https://docs.solace.com/Configuring-and-Managing-Routers/System-Level-Subscription-Exception-Config.htm).

For information describing how to add subscription exceptions to queues refer to [Adding Topic Subscription Exceptions to Queues](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-Queues.htm#Adding2).

## Topic Endpoints

A topic endpoint attracts messages published to a topic for which the topic endpoint has a matching topic subscription. The topic subscription for the topic endpoint is specified in the client request to bind a Flow to that topic endpoint.

Topic endpoints can be used in a pub/sub model. They are equivalent to durable topic subscriptions in Java Message Service (JMS).

## Endpoint Durability

Queues and topic endpoints can be either durable or temporary.

### Durable Endpoints

Durable queues and topic endpoints are provisioned objects on the router that have a life span independent of a particular client session. They also survive a router restart and are preserved as part of the router configuration for backup and restoration purposes. Administrators can provision durable queues and topic endpoints through the Solace CLI, SEMP, or SolAdmin; client applications can dynamically provision durable endpoints through the Solace messaging APIs.

### Durable Endpoint Access Types

A durable endpoint has an associated access type which determines how messages are delivered to bound consumer flows. Durable endpoints can be assigned one of the following access types:

* exclusive— For a queue, multiple subscriber flows established by clients can be bound, but only one flow out of all the bound flows is able to receive messages at a time. The first flow to bind to the queue can consume messages from it, and when that flow disconnects, the next oldest flow to bind becomes active and can begin receiving messages, and so on .For a topic endpoint, only one consumer can bind and receive messages; if other consumers attempt to bind, they will be rejected. An exclusive durable topic endpoint always delivers messages in the order they are received.
* non‑exclusive—For either queues or topic endpoints, multiple consumers can bind to a non-exclusive durable topic endpoint, and each is serviced in round‑robin fashion.

### Temporary Endpoints

Temporary queues and topic endpoints are dynamically created and destroyed by client applications. Temporary queues and topic endpoints are typically used as temporary destinations for service requests.

When a client application has a connected session, it can provision a temporary queue or topic endpoint and create a consumer flow to bind to it. These temporary queues and topic endpoints are non-durable because they only last as long as the client’s session is connected. Only a single consumer may bind to a non-durable endpoint, and there is no support for multiple consumers or non-exclusive access.

The naming of the temporary destinations is controlled by the application. On a BIND request from the client, a name is provided. It the destination does not exist, one is created.

When a client dynamically creates a queue, it is configured by those endpoint properties and provision flags that the client may provide with a create API function or method. Any other endpoint parameters are then configured with the values used for endpoints provisioned by an administrator through the Solace CLI. By default, the system defaults are used. However, it is also possible to use CLI-provisioned queues and topic endpoints with custom values, and those values will be applied to any new client-created queues and topic endpoints.

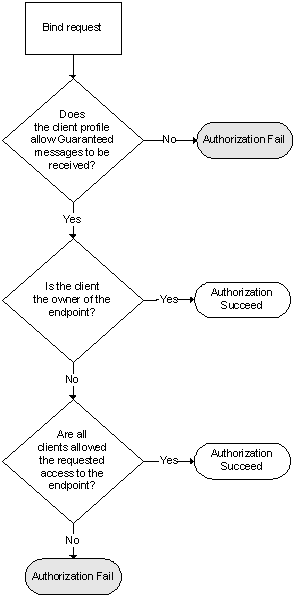
A non-durable queue or topic endpoint can be dynamically created if:

* The client is associated with a profile that permits clients to create endpoints. See [Allowing Guaranteed Endpoint Creates](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-Client-Profiles.htm#Allow-G-Msg-Creates).
* The maximum number of endpoints available for the Message VPN that the client is connected has not been reached. See [Configuring the Max Number of Endpoints](https://docs.solace.com/Configuring-and-Managing-Routers/VPN-Level-Msg-Spool-Config.htm#Configur5).
* The maximum number of endpoints permitted per client username within a client profile has not been exceeded. See [Configuring the Max Endpoints Permitted Per Client Username](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-Client-Profiles.htm#Config-Max-Endpoints).
* The number of endpoints does not exceed the system-wide limit.

## Endpoint Permissions and Access Control

The flow chart below shows the layers of access control that are used to determine whether clients may bind to and browse or consume messages from endpoints.

**Access Control Layers Used for Consuming Clients**



The first layer is a configuration in the associated client profile that determines whether the client is permitted to receive Guaranteed messages. If this is allowed, then the endpoint permissions are checked.

The endpoint permission rules are:

* The owner has full access to an endpoint. The owner of an endpoint is configurable. In the case of dynamically‑created endpoints, the default owner is the client username of the client that created the endpoint.
* CLI-created queues and topic endpoints can only be deleted through the CLI.
* Each endpoint has a “Permission All” parameter that determines what access control clients other than the owner of an endpoint have.

**Note:**By default, the “Permission All” parameter has a permission level of none. However, permissions can be explicitly specified when the endpoint is initially created through the CLI or dynamically through a messaging API. The permission level can also be changed through the CLI.

The permissions available at the following levels are:

* none—no access.
* read-only—allows the messages only to be read; they cannot be removed or consumed from the message spool.
* consume—allows messages to be browsed and consumed from the message spool.
* modify-topic—allows the topics assigned to endpoints to be modified. Modify-topic also implicitly includes same permissions as consume.
* delete—allows deleting queues or topic endpoints. Delete also implicitly includes same permissions as modify-topic.

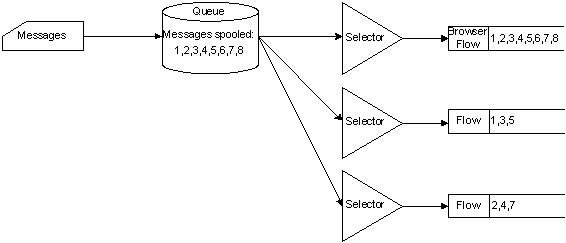
## Browsing Queues

Client applications using Solace enterprise APIs can create a queue browser in a session to look at messages spooled on a queue in order of oldest to newest without consuming them. That is, browsing messages returns the full content of messages, complete with all message headers and payloads, but those browsed messages are not removed from the message spool.

**Note:**When you browse a queue that has an active consumer, it is possible that the browser will not receive all messages published to the queue because the consumer can receive and acknowledge messages before they are delivered to the browser.

The figure below shows how a Browser Flow returns messages from the queue, without consuming them so that clients with established Flows can still consume them. In this example, a selector string is also used so that the client application only browses messages that match that selector. (Refer to [Selectors](https://docs.solace.com/Features/Endpoints.htm#working_with_guaranteed_messages_2286334743_467289) for more information.)

**Queue Browsing**



## Selectors

Selectors are filters that clients can apply when they bind to endpoints. They enable clients to filter which messages they are interested in receiving, as determined by the messages’ header field and property values. A selector is a string up to a maximum of 1,023 bytes that uses a conditional expression syntax that is a subset of SQL92.

As shown in the figure below, selectors can filter messages from queues and topic endpoints.

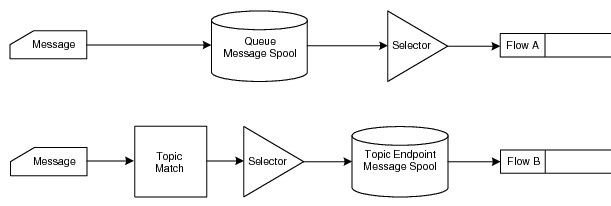
* For queues, any messages that do not match the client’s selector string are not delivered to the requesting client, but remain in the queue.
* For topic endpoints, any messages that do not match the selector are not delivered to the requesting client, but are removed from the topic endpoint.

Selectors on topic endpoints are a background process to avoid slowing down publishing pipelines. It is possible to see the number of messages in a topic endpoint shrink as the selector is applied. For example, if 100 messages are published which match the associated topic of a topic endpoint, then these 100 messages are spooled into the topic endpoint. The associated selector starts testing these 100 new messages, and any messages that the selector rejects are removed from the topic endpoint. If there is a selector associated with a topic endpoint, then the selector must filter the messages prior to delivery to the flow.

**Note:**If either the topic or selector changes, then all messages are removed from the topic endpoint before applying the new topic and selector.

Client applications can also specify a selector string when browsing a queue so that the client only browses messages that match the selector.

**Guaranteed Message Selectors**

****

## Dead Message Queues

By default, Guaranteed messages are removed from a durable endpoint's message spool and discarded when:

* the number of redelivery attempts for a message exceeds the Max Redelivery value for the original destination endpoint;
* or a message’s TTL value has been exceeded and the endpoint is configured to respect message TTL expiry times.

However, messages that are flagged as DMQ-eligible by the publishing client can be sent to a DMQ assigned to the endpoint rather than be discarded.

Any durable queue on the same Message VPN as the endpoint that the messages were spooled to can be assigned as that endpoint's DMQ. Although durable endpoints are assigned a default DMQ (#DEAD\_MSG\_QUEUE), every durable endpoint can be assigned a specific DMQ. Therefore, there can be multiple DMQs per Message VPN.

**Note:**If an endpoint's assigned DMQ does not exist, the Guaranteed messages will be discarded even if they are DMQ-eligible. In addition, although it is the default DMQ, a management user must manually create the #DEAD\_MSG\_QUEUE.

The following are use-cases for how using different endpoints within a Message VPN associated with different DMQs might be advantageous:

* When messages are delivered to an endpoint through topic-to-queue mapping, and that message is subsequently moved to the default DMQ used by multiple endpoints, there is no way for an application servicing the DMQ to know which endpoint the message came from. Having a DMQ that is used only by a single endpoint would solve this problem.
* It is possible to deliberately use message TTLs in conjunction with the DMQs to achieve a delayed delivery of messages. If client applications bind to a DMQ rather than a "regular" endpoint, they will only consume expired messages that were moved a DMQ. In this use-case, applications don’t want the messages from multiple endpoints converging on a single DMQ.

For information, see [Setting Dead Message Queues](https://docs.solace.com/Configuring-and-Managing-Routers/Setting-Dead-Msg-Queues.htm).

## Last Value Queues

If a queue is assigned a max-spool-usage value of 0, that queue can only spool the last message that it received. With this spooling behavior, the queue is acting as a Last Value Queue.

A client publishing Guaranteed messages can apply a topic subscription to a Last Value Queue so that it attracts all the messages that the client publishes. This allows that client to use the Last Value Queue to accurately determine the very last Guaranteed message that it successfully published.

This could be beneficial, for example, if an application failure occurs after a message has been published. In this case, if the client application does not receive an acknowledgment from the router for the message, it does not know if the published message was lost entirely, or if the message was received but just the acknowledgment was lost. If a Last Value Queue is used, when the client reconnects and rebinds to the Last Value Queue, it can determine what was the last successfully published message, and it can continue publishing from where it left off without creating duplicate messages or losing messages.

To configure a queue can be configured to behave as a Last Value Queue, you can set the max spool usage value for the queue to 0 (refer to [Configuring Queues](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-Queues.htm)).

**Note:**Solace recommends as a best practice to NOT use this feature with topic endpoints.

When there is more than one publisher for a given topic, the publisher should be identified in published topic, which can by wildcarded by subscribing applications. For example, assign the Last Value Queue for each publishing client a topic subscription of the form uniquePubId/>(so Pub1/> for the first publisher, Pub2/> for the second publisher, and so on). The clients can then publish messages to topics of the form uniquePubId/some/hierarchical/topic/. For example, Pub1 might publish to Pub1/price/equities/apple, while Pub2 might publish to Pub2/price/equities/apple, and so on.

This allows clients to specifically subscribe to their own Last Value Queue, but other clients, who also want to receive messages on those topics, can use a topic subscription of the form \*/some/hierarchical/topic. For example, \*/price/equities/apple.

**Note:**

* Message selectors are not supported with Last Value Queues.
* When the **show queue** User EXEC command is used to display the configuration and status information for a Last Value Queue, and no clients are bound to that queue, the output might show that multiple messages are spooled even though, in fact, only the most current message is spooled.

# Topic Matching and Message Delivery Modes

When the Solace Messaging Platform using Guaranteed Messaging receives messages published to a topic, it can automatically change the delivery mode of the messages to accommodate the persistence requirements of an endpoint or a client subscription when there is a topic match. This changing of messagesʼ delivery modes is also known as message promotion and demotion.

To show how a messageʼs delivery mode might be changed, consider when a client publishes a Direct message to a topic that matches the topic subscription set for an endpoint (either a queue that has a topic subscription added to it or a topic endpoint). Because an endpoint can only accept non‑persistent and persistent messages, the router will change the messageʼs delivery mode to non-persistent so that the endpoint can receive it. Note that when a client publishes a message as Direct, the message is not acknowledged back to the client. However, when the message’s delivery mode is changed to non‑persistent and is spooled to the endpoint, the message then acts as a Guaranteed message, and an acknowledgment from the receiving client is provided to the router.

For an illustration of how the delivery mode of a message published as Direct can be changed if it matches a Topic assigned to an endpoint, refer to [Publishing Direct Messages](https://docs.solace.com/Solace-Messaging-APIs/Publishing-Direct-Messages.htm#publishing_direct_messages_2102779381_656733).

Also, if the router receives a persistent message published to a topic, it delivers the message to any endpoints that have matching topic subscriptions. However, because topic subscriptions registered for individual clients are for Direct messages, the router can change the message’s delivery mode to Direct and send it to those clients with matching topic subscriptions. In this scenario, because the client published the message as persistent, the message is acknowledged back to the publishing client.

The only way for a client to receive Persistent messages is to either bind to a durable topic endpoint or queue, or request that a non-durable queue or topic endpoint be created for it.

For an illustration of how the delivery mode of a message published as Non‑Persistent or Persistent can be changed if it matches a client’s Topic subscription, refer to [Publishing Guaranteed Messages](https://docs.solace.com/Solace-Messaging-APIs/Publishing-Guaranteed-Messages.htm#publishing_guaranteed_messages_3926332939_612041).

The table below lists how messages’ delivery modes can be changed for endpoints with matching topics.

| **Possible Message Delivery Mode Changes on Topic Match** | |
| --- | --- |
| **Delivery Mode of Published Message** | **Received by Endpoint as ...** |
| Direct | Non-Persistent |
| Non-Persistent | Non-Persistent |
| Persistent | Persistent |

# Message Spooling

As a Solace router receives Guaranteed messages (that is, messages with persistent or non-persistent delivery modes), it processes that message to determine if there are any registered topic subscriptions or queues that match the destination the message was published to. If there is a topic subscription match or a matching queue on the router, the message and all the matches are spooled and then the router acknowledges receipt of the message. After acknowledging the message, the router attempts to deliver it to all the matching clients/routers. As each client/router acknowledges receipt of the message, the associated match is deleted from the match list of the message. Once there are no matches left associated with a message, the message itself is deleted from the spool (that is, it is discarded).

If one or more of the clients are offline or have fallen behind, the message is held in the ADB until the message can be delivered. If there are too many messages to hold in the ADB’s memory, messages are written to the disk in large blocks. This means only messages for slow/offline clients need to be written to the disk and they can be written in an efficient manner.

**Note:**If all of the clients acknowledge the message quickly enough, the message does not need to be written to disk.

It is possible for a message to not be spooled because of resource or operating limitations. A variety of checks are performed before a message is spooled. These include:

* Would spooling the message exceed the router-wide message spool quota?
* Would spooling the message exceed the Message VPN’s message spool quota?
* Would spooling the message exceed the endpoint’s message spool quota?
* Would spooling the message exceed the endpoint’s maximum permitted message size?
* Would the message exceed the endpoint’s maximum message size?
* Is the destination endpoint shutdown?
* If the message is a low-priority message, would spooling the message exceed the endpoint’s reject low‑priority message limit?

Depending on the reason for why the message was not spooled to the endpoint, either no acknowledgment is returned to the publisher or a negative acknowledgment (that is, a ‘nack’) is returned, and it is up to the publisher to handle these possibilities. A statistic is then incremented on the router.

**Note:**

* If a subscription is deleted when a message is spooled for that subscription, the message will still be delivered.
* If an endpoint is deleted (for example, through the Solace CLI) while a message is spooled to that endpoint, the message will not be delivered.

## Spool Files

Guaranteed messages are spooled to a Solace router through the use of spool files that can each hold approximately 8 MB worth of messages. A router can support one million or more spool files—the maximum number of message spool files available depends on the Solace router model. If the router’s spool file usage reaches this limit, it cannot receive any more messages until some spooled messages are acknowledged, which could free some spool files.

If a router reaches its maximum spool file usage, negative acknowledgments (that is, ‘nacks’) are returned to all publishing clients. However, spool file thresholds can be configured so that events are generated when the system-level message spool usage gets too high but is not exceeded, or when it gets abnormally low. Refer to [System-Level Message Spool Configuration](https://docs.solace.com/Configuring-and-Managing-Routers/System-Level-Msg-Spool-Config.htm).

By default, an event is generated when more than 80% or less than 60% of spool files are in use.

## Windowed Acknowledgment

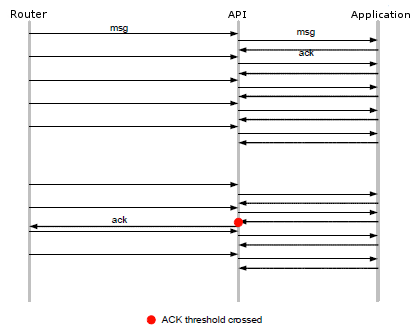
A windowed acknowledgment mechanism is used at the transport level between the router and individual clients publishing and receiving Guaranteed messages.

A windowed acknowledgment prevents the round-trip acknowledgment time from becoming the limiting factor for message throughput. It does this by allowing a configurable number of messages to be in transit between a Solace router and a publishing or subscribing client before an acknowledgment is required. The window size can be configured through a Solace messaging API flow property.

Solace APIs also batch acknowledgments from the application to the router. The figure below shows a client application sending multiple acknowledgments for Guaranteed messages, which the API consolidate into a single acknowledgment on the wire that is returned to the router. The size of the batch is configured through an API flow property.

**Note:**With any windowed acknowledgment scheme there is the possibility of failure in the time between a message being received by the client, and the time at which the router receives the acknowledgment. A failure at this time requires all non-acknowledged messages in transit to be sent again. Thus, the number of messages redelivered increases and is directly proportional to the combination of window size and acknowledgment threshold.

**Cumulative ACK and Acknowledgment Thresholds**



## Delivered-But-Unacknowledged Messages

There is a hard limit for the number of Guaranteed messages that can be delivered through a consumer flow to a bound client without that client returning an acknowledgment for those messages. On reaching this limit (plus one window size of messages), the Solace router stops delivering messages to the client until the client acknowledges messages that have already been delivered.

**Note:**You can configure the maximum number of delivered‑but‑unacknowledged messages limit for a queue or topic endpoint provisioned on a Message VPN (refer to [Message VPN-Level Guaranteed Messaging Configuration](https://docs.solace.com/Configuring-and-Managing-Routers/VPN-Level-Msg-Spool-Config.htm)).

A Solace router has a system-level limit for the maximum number delivered‑but‑unacknowledged messages for all clients at a given time. On reaching this maximum, no more messages are delivered to clients until some clients return acknowledgments back to the router, or they are disconnected. The maximum number delivered‑but‑unacknowledged messages supported is dependent on the Solace router model.

By default, an event is generated when the number of outstanding messages that have not been acknowledged by their receiving clients reaches 80 percent of the system maximum (the set value), and this is followed by a further event the number of client-unacknowledged messages returns below 60 percent of the system maximum (the clear value). The thresholds for when these events are generated are configurable (refer to [Delivered Unacked Thresholds](https://docs.solace.com/System-and-Software-Maintenance/Configuring-System-Event-Thresholds.htm#montoring_and_management_1462237600_255180)).

## Message Expiry

To set a limit on the amount of time that published messages have to be consumed and an acknowledgment of receipt is not returned to the router, you can assign Time‑to-Live (TTL) expiry values to the messages.

* If your application is using the Solace messaging APIs or REST service to published Guaranteed messages, you can set a Time‑to-Live (TTL) expiry value on each published Guaranteed message to indicate how long the message should be considered valid. A publisher TTL expiration starts when a message is published and counts down as the message passes through the network.
* You can configure a Max Time to Live (TTL) value for a durable endpoint so that received messages are provided with expiration value to limit how long they can remain on that durable endpoint when a Max TTL is used. The Max TTL only applies when a message is on the endpoint.

When TTL values are applied to messages in either or both of these ways, messages that are not consumed and acknowledged before their expiration times are reached are discarded or moved to a Dead Message Queue (DMQ) . If a message has both a publisher-assigned TTL and an endpoint-assigned Max TTL, the router will use the minimum of the two TTL values when the message is on the endpoint.

**Note:**If a Message VPN bridge is used so that published messages that match topic subscriptions can be delivered from a remote Message VPN on one router to a local Message VPN on another router, the amount of time the message spends on each router is counted. That is, the amount of time a message spends on the remote router is counted, and its remaining time to live is updated when it is sent to the local Message VPN. For example, with a publisher-provided TTL of eight seconds, if a message spends two seconds on the first router, before it reaches the local Message VPN on the second router, it will have a TTL of six seconds.

Using TTLs to expire messages that have not been processed quickly can help prevent stale messages from being delivered to consumers. However, it should be noted that monitoring and processing messages using TTLs can affect the system-level limits for Guaranteed message delivery (for more information, refer to [Guaranteed Message Queuing Limits](https://docs.solace.com/Features/G-Msg-Queueing-Limits.htm#queueing_limits)).

# Service Outage Protection

When Guaranteed Messaging is enabled, the following features can be used to protect against disruptions to messaging service should a router or an entire data center go out of service:

* [Appliance Redundancy](https://docs.solace.com/Features/Service-Outage-Protection.htm#router_redundancy)
* [Replication](https://docs.solace.com/Features/Service-Outage-Protection.htm#replication)
* [Config-Sync](https://docs.solace.com/Features/Service-Outage-Protection.htm#config_sync)

## Appliance Redundancy

High-availability (HA) appliance redundancy is a feature that allows two Solace appliances to be configured as a redundant pair. When two appliances are configured as an HA pair, if one of the appliances in the pair is taken out of service or fails, its mate automatically takes over responsibility for the clients associated with the unavailable appliance.

Router redundancy is supported for Solace appliances that have installed ADBs and HBAs. It also requires a customer-supplied external disk storage array. For more information, refer to [Appliance Redundancy and Fault Tolerance](https://docs.solace.com/Features/Redundancy-and-Fault-Tolerance.htm).

## Replication

The Replication facility provides a data center redundancy and disaster recovery solution for the Solace Message platform. It uses corresponding Message VPNs at separate Replication sites to ensure that Guaranteed Messaging clients can continue to have service through one of the Replicated Message VPNs should one data center become unavailable.

If a service switch-over to one Replication site occurs, clients can reconnect to the same Message VPN at a different Replication site to continue to obtain service, and any messages that were received, but not consumed, before the service interruption can be delivered to them.

For more information, refer to [Data Center Replication](https://docs.solace.com/Features/Data-Center-Replication.htm).

## Config-Sync

Config-Sync is a configuration synchronization feature that is used for:

* HA redundant pairs—Config-Sync should be enabled for both routers in an HA pair to automatically synchronize initial configuration information and subsequent changes between both redundant Solace routers. When Config‑Sync is enabled for both routers in a HA pair, it can also increase the speed at which Guaranteed Messaging information is handled if a redundancy switchover occurs.
* Replicated Message VPNs—Config-Sync must enabled on all routers in each Replication site to automatically synchronize configuration parameters between Replication‑enabled Message VPNs of the same names that exist on separate Replication sites.

**Guaranteed Message Queuing Limits**

The maximum number of Guaranteed messages that Solace routers can queue at one time for delivery will be cumulatively reduced by the following factors:

* the number of messages that provisioned endpoints receive—If a single published message is queued for delivery on *n* endpoints, that counts as *n* messages queued for delivery, even though only one copy of the message is stored in either the ADB or external disk storage array.
* whether data from the router is being replicated—When the Replication facility is used, a copy of a message in the Replication queue counts as another message queued for delivery. That is, if a message is queued for delivery on n endpoints, and it is to be replicated, that counts as n +1 messages queued for delivery.
* whether the router is processing transactions—When transactions are used, the router has to maintain extra information for each message in a transaction until the transaction is committed or rolled back.
* whether there are provisioned endpoints configured to respect published messages’ TTL expiry times—When an endpoint respects a message’s TTL, the TTL is recorded and maintained until the message is consumed or discarded.

# Guaranteed Messaging Use Cases

This section provides examples of scenarios where Guaranteed Messaging functionality can be used:

* [Request/Reply with Guaranteed Messages](https://docs.solace.com/Features/G-Msg-Use-Cases.htm#request_reply)
* [Message Request Queues](https://docs.solace.com/Features/G-Msg-Use-Cases.htm#message_request_queues)
* [Slow Consumers](https://docs.solace.com/Features/G-Msg-Use-Cases.htm#Slow_Consumers)

## Request/Reply with Guaranteed Messages

From a high-level point-of-view, the following steps occur:

1. The router is provisioned to support Guaranteed messaging.
2. The customer application uses a client to create a non-durable queue on the router, and then creates a consumer flow to bind to that queue.
3. The client publishes a (possibly Guaranteed) message with the request to a well-known topic, specifying that the reply to the request be sent to the newly created temporary queue.
4. The server listens to the topic (or binds to the queue if it has a topic subscription added to it) and responds to the request by publishing to the specified non‑durable queue.
5. The client consumes the reply message from the non-durable queue. Once it has finished processing the response(s), it closes the flow, which deletes the non-durable queue.

## Message Request Queues

If the queue is created with non-exclusive access mode, consumers can load balance the processing of the message requests. Alternatively, the queue can be made exclusive so that only the first consumer will process requests and the other customers that bind act as backup should the first consumer disconnect. The producers to the message queue are decoupled from knowing how the message items are processed—they simply publish request messages to the queue.

The consumers can also configure their maximum window sizes and maximum unacked messages to set how many message items are pipelined to them in their flows. As each message item is processed, the consumer application triggers an acknowledgment back to the router to pull another message into its flow pipeline.

If you need to ensure that messages persist even if all consumers disconnect from the queue, make use of a durable queue instead of one that's temporary.

As well, If you need to constrain access to the message queue so that not just anyone can access request messages, set a specific user (for example, client user name messageItemProcessor) as the owner of the queue, and set the others permission to none.

## Slow Consumers

Guaranteed messaging can be used to buffer messages for slow consumers. For example, a client application that uses a topic subscription to receive Direct messages and then writes them to a database may not be able to keep up with a large burst of messages from a publisher.

To handle such a scenario, the client could create a durable queue and add a topic subscription to it. The topic subscription would attract the messages, which would be changed to non-persistent and spooled to the queue. The client application can then consume and acknowledge without worrying about message rates that overwhelm its processing capabilities. The publisher is also able to publish as quickly as possible and does not need to account for potential slow receiving applications.

# Working with Message VPNs

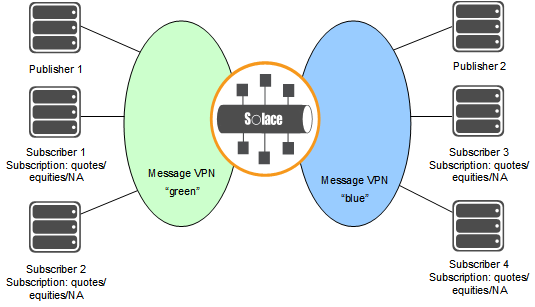
Message Virtual Private Networks (VPNs) are managed objects on Solace routers that allow for the segregation of topic space and clients. Message VPNs also group clients connecting to a network of Solace routers, such that messages published within a particular group are only visible to clients that belong to that group. Each client connection is associated with a single Message VPN.

As shown in the figure below, Message VPNs can be used to effectively separate which clients can receive messages from which publishers. In this example, clients in different Message VPNs are permitted to subscribe to identical topics, and two clients in different Message VPNs are permitted to publish messages to topics that matches those client subscriptions. Yet due to Message VPN membership, only the clients that are connected to the same Message VPN as the publisher receive the messages from that publisher.

**Note:**Published messages cannot cross Message VPN boundaries, even in the presence of identical subscriptions in each Message VPN. For messages published to one Message VPN to be transferred to another Message VPN, a Message VPN bridge must be explicitly configured between them.

In this example, all the subscriber clients have subscribed to the same topic: “quotes/equities/NA”. However, because the clients are connected to separate Message VPNs, when Publisher 1 publishes a message to topic “quotes/equities/NA”, the message is only delivered to Subscriber 1 and Subscriber 2. Similarly, if Publisher 2 publishes a message to topic “quotes/equities/NA”, the message is only delivered to Subscriber 3 and Subscriber 4.

**Message VPN Publishing and Subscribing Example**



Each Message VPN can be administratively enabled and disabled through the **shutdown** Message VPN CONFIG command. When disabled, all client connections belonging to that Message VPN are disconnected and new client connections to it are rejected. Message VPNs are disabled by default (that is, not running) on Solace routers. Each client must identify the Message VPN which the client wishes to connect to. If the client username is not configured within the requested Message VPN, then the client connection is denied.

## Connecting to Message VPNs

Each client connection is associated with a single Message VPN. When a client sends its initial login connection request to a Solace router, the client typically includes a Message VPN name parameter in the login request. The router then verifies that the client username has been configured in the requested Message VPN and that the client username is authorized to connect to the requested Message VPN. A global, per-Message VPN, and per‑client statistic is incremented for every denied connection attempt.

**Note:**A client connection cannot change its assigned Message VPN once it has been established by the initial login request without disconnecting from the router first.

However, if the client does not provide the name of a Message VPN name to connect to, the default Message VPN named default (when enabled) is automatically assigned to the client.

Each Solace router has a Message VPN named default. The Message VPN named default cannot be deleted, but it can be configured like any other Message VPN object on the router.

**Client Authentication/Authorization**

To send messages to and receive messages from the Solace Messaging Platform, a customer application or device must establish a TCP client connection to a specific Message VPN on a Solace router. These client connections are created and deleted dynamically as they connect to and disconnect from a router.

Client applications using Solace messaging Application Programming Interfaces (APIs) create sessions to establish client connections to Message VPNs. Client applications can also use the standard client connection approaches offered through other technologies supported by Solace. These technologies include Open Middleware Agnostic Messaging API (OpenMAMA), Representational State Transfer (REST) messaging service, and Message Queuing Telemetry Transport (MQTT) protocol.

* [Client Authentication](https://docs.solace.com/Features/Client-Authentication.htm)—For an application to connect to a Message VPN as a client, the router must authenticate that client.
* [Client Authorization](https://docs.solace.com/Features/Client-Authorization.htm)—A successfully authenticated client then goes through an authorization process to give it access to specific router resources and messaging capabilities on that Message VPN.

Clients that are established on a Message VPN are uniquely identified by a client name. The messaging API or Solace router may automatically generate this client name based on the username used, or, optionally, a client application may explicitly provide a client name that is a valid non‑wildcard topic up to a maximum length of 160 characters.

**Note:**In addition to messaging applications, management applications that use SEMP Request Over Message Bus service may establish client connections to Solace routers perform monitoring and management operations. For more information, refer to [Legacy SEMP](https://docs.solace.com/SEMP/Using-Legacy-SEMP.htm).

# Client Authentication

When an application or device connects to a particular Message VPN, the resulting client connection must be authenticated before any client requests can be processed. A connecting client is authenticated on a per-Message VPN‑basis through one of the following client authentication schemes:

* [Basic Authentication](https://docs.solace.com/Features/Client-Authentication.htm#Basic-Authentication)
* [Client Certificate Authentication](https://docs.solace.com/Features/Client-Authentication.htm#Client-Certificate)
* [Kerberos Authentication](https://docs.solace.com/Features/Client-Authentication.htm#Kerberos)

**Note:**More than one type of authentication scheme can be configured and enabled for a Message VPN, but a client can only be configured to use one type of authentication scheme.

## Basic Authentication

A basic authentication scheme allows a connecting client to authenticate with a router by providing a valid client username and password as its credentials.

Basic authentication is the default client authentication scheme for a Message VPN. It is available for client applications using any Solace messaging API. It is also available for client applications using the OpenMAMA API, REST, or MQTT.

Clients can use Basic authentication for either a plain-text or a Transport Layer Security (TLS) / Secure Sockets Layer (SSL)-encrypted client connection to the router.

This authentication scheme uses one of the following authentication types:

* Internal—The client username and password provided by the client are verified against an internal Solace router database.
* RADIUS—The client username and password are sent to an external RADIUS server for authentication.
* LDAP—The client username and password are sent to an external LDAP server for authentication.
* None—No client authentication is performed for the client. Solace strongly recommends against using no client authentication.

**Related Provisioning and Configuration Information**

To use basic authentication to authenticate connecting clients, the following configuration is required for the following areas:

* client configuration
  + For clients using Solace messaging APIs, their client username and password are provided as configurable session properties. See the following sections for information on how to set session properties and create sessions: [Creating Client Sessions](https://docs.solace.com/Solace-Messaging-APIs/Creating-Client-Sessions.htm) for Solace enterprise APIs, [Creating JMS Connections](https://docs.solace.com/Solace-JMS-API/Creating-JMS-Connections.htm) for Solace JMS API; [Creating Client Sessions](https://docs.solace.com/Solace-Messaging-APIs/Creating-Client-Sessions-1.htm) for Solace Web messaging APIs.
  + For OpenMAMA clients, basic authentication parameters are configured for the Solace Middleware Bridge that is used to establish a connection to the Solace router. See [Configuring Solace OpenMAMA Bridges](https://docs.solace.com/Solace-OpenMama/Configuring-Solace-OpenMAMA-Bridges.htm).
  + For REST clients, the client username and password are provided as a string in an HTTP standard header. See [Client Authentication](https://docs.solace.com/RESTMessagingPrtl/Solace-Router-Interactions.htm#Client).
  + For MQTT clients, the connect packet contains username and password fields. The CONNECT Packet contains Username and Password fields. These are mapped to a Solace client username and password. For more information, see [3.1.3.4 User Name](https://docs.solace.com/MQTT-311-Prtl-Conformance-Spec/MQTT%20Control%20Packets.htm#_Toc430865055) in the Solace MQTT 3.1.1 Messaging Protocol Conformance section.
* Solace router configuration
  + A client username and password combination must be configured and enabled for internal authentication.

For information on the system and Message VPN-level configurations that are required on a Solace router to implement a basic authentication scheme, refer [Managing Client Authentication](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-Client-Authentication.htm).

**Note:**For REST and MQTT clients, there are some additional Message VPN configurations, such as enabling the appropriate listen ports, are required. For more information, see [Using REST](https://docs.solace.com/Open-APIs-Protocols/Using-REST.htm) and [Using MQTT](https://docs.solace.com/Open-APIs-Protocols/Using-MQTT.htm).

## Client Certificate Authentication

A client certificate authentication scheme allows a client to prove its identity to the Solace router by providing a valid X509v3 client certificate from a recognized Certificate Authority (CA).

For this authentication scheme, the common name (CN) of the certificate provided to the router is mapped to the client’s assigned client username, which can be used for subsequent client authorization.

Client certificate authentication is available for clients using Solace enterprise messaging APIs. It is also available for client applications using the OpenMAMA API, REST, or MQTT.

**Note:**Solace VMRs do not support client certificate authentication for REST clients.

**Related Provisioning and Configuration Information**

To use client certificates to authenticate connecting clients, the following configuration is required for the following areas:

* client configuration
  + For clients using Solace messaging APIs, secure sessions must be used to establish TLS/SSL-encrypted client connections to the Solace router. To create a secure Session, a client certificate authentication scheme and a client certificate and a private key (depending on the API used, these could be separate files or be contained in a single keystore file) must be specified.

For information on creating a secure client sessions and setting session properties using Solace APIs, see [Creating Client Sessions](https://docs.solace.com/Solace-Messaging-APIs/Creating-Client-Sessions.htm)for Solace enterprise APIs or [Establishing Connections](https://docs.solace.com/Solace-JMS-API/Establishing-Connections.htm) for the Solace JMS API.

* + For OpenMAMA clients, client certificate authentication parameters are configured for the Solace Middleware Bridge that is used to establish a connection to the Solace router. For more information, see [Configuring Solace OpenMAMA Bridges](https://docs.solace.com/Solace-OpenMama/Configuring-Solace-OpenMAMA-Bridges.htm).
  + TLS/SSL authentication is supported for REST clients. For information, see [Client Authentication](https://docs.solace.com/RESTMessagingPrtl/Solace-Router-Interactions.htm#Client).
  + TLS/SSL authentication is supported for MQTT clients. For information, see [Contact Us ׀ Support ׀ Blog ׀ solace.com](https://docs.solace.com/MQTT-311-Prtl-Conformance-Spec/MQTT_311_Prtl_Conformance_Spec.htm).
* Solace router configuration
  + SolOS 6.1 or greater must be used.
  + Trusted root certificates must be loaded onto the Solace router. Client certificate authentication must be configured and enabled for any Message VPNs that the clients will connect to.
  + To enable the required secure client connections, TLS/SSL service must be configured and enabled.

**Note:**

* + For REST and MQTT clients, some additional Message VPN configurations, such as enabling the appropriate listen ports, are required. For more information, see [Using REST](https://docs.solace.com/Open-APIs-Protocols/Using-REST.htm) and [Using MQTT](https://docs.solace.com/Open-APIs-Protocols/Using-MQTT.htm).
  + Client certificate authentication can also be used on Message VPN bridges, Message VPN replication bridges, and Replication Config-Sync bridges. For more information, see [Managing TLS/SSL Service](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-TLS-SSL-Service.htm).

## Kerberos Authentication

A Kerberos authentication scheme allows clients that have been granted a valid Kerberos ticket to connect to a Solace router.

Kerberos authentication is only available for clients using Solace enterprise messaging APIs or the OpenMAMA API.

When a Kerberos authentication scheme is used for client authentication, a client must first authenticate with a Kerberos Authentication Server (AS) which grants the client a Ticket Granting Ticket (TGT) for a specified Kerberos User Principal. The TGT is typically obtained as part of a Single Sign-on procedure, such as logging into a Windows domain. With a valid TGT, a client can attempt to log onto a router using a service ticket that is in the client’s local ticket cache or has been obtained from the Ticket Granting Service (TGS). The AS and TGS (components of a Key Distribution Center (KDC)) are hosted on an external server or servers—not on a Solace router.

The client then provides this time-stamped "Kerberos" ticket to the Solace router. If the ticket is successfully validated, the client’s connection to the Message VPN is granted.

For this authentication scheme, the client’s assigned client username, which is used for subsequent client authorization, is the user principal name in the ticket provided to the router.

**Related Provisioning and Configuration Information**

To use Kerberos to authenticate clients connecting to a Solace router, the following configurations are required:

* client-side configuration
  + For clients using Solace messaging APIs, the appropriate Java distribution must be used or the appropriate Kerberos libraries must be installed for the Solace messaging API used, and the client session must use a Kerberos authentication scheme.

For information on setting a Kerberos authentication scheme using Solace APIs, see [Creating Client Sessions](https://docs.solace.com/Solace-Messaging-APIs/Creating-Client-Sessions.htm) for Solace enterprise APIs or [Establishing Connections](https://docs.solace.com/Solace-JMS-API/Establishing-Connections.htm) for the Solace JMS API. For information on development requirements for developing Kerberos-compatible applications, see [Quick Start](https://docs.solace.com/Solace-Messaging-APIs/Quick-Start.htm) for Solace enterprise APIs or [Establishing Connections](https://docs.solace.com/Solace-JMS-API/Establishing-Connections.htm) for the Solace JMS API

* + For OpenMAMA clients, Kerberos authentication parameters are configured for the Solace Middleware Bridge used to establish a connection to the Solace router. For more information, see [Configuring Solace OpenMAMA Bridges](https://docs.solace.com/Solace-OpenMama/Configuring-Solace-OpenMAMA-Bridges.htm).
* Solace router configuration
  + SolOS 7.0 or greater must be used.
  + A Kerberos Keytab must be loaded on the router.
  + Kerberos authentication must be configured and enabled for any Message VPNs that Kerberos-authenticated clients will connect to.
  + Optionally, a Kerberos Service Principal Name (SPN) can be assigned to the IP address for the message backbone VRF that will be used for Kerberos‑authenticated clients. For information, see [Configuring Kerberos Authentication](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-Client-Authentication.htm#Config-Kerberos).

# Client Authorization

If a client connection to a Message VPN is successfully authenticated, access to the router resources and messaging capabilities within that Message VPN must be authorized for the client.

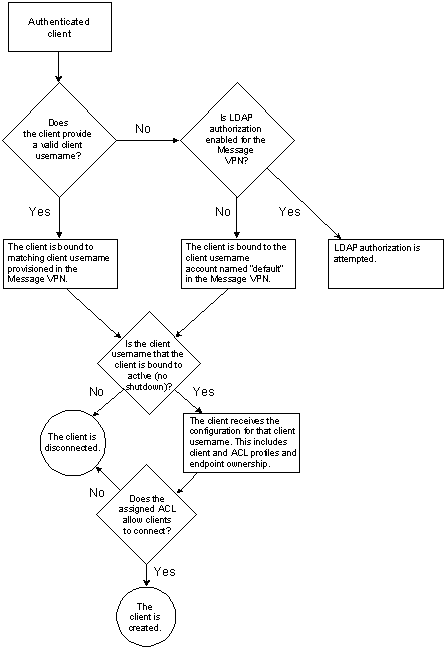
For a client to be authorized, it must provide the host router with a client username that matches one that is provisioned on the Message VPN to which the connection has been made. (These client usernames can either be provided by the connected client or be automatically generated for the client based on an LDAP group that it is a member of.) If the client provides a valid client username and password, the client’s connection is authorized.

Once authorized, the following two types of profiles that are assigned to the provisioned client username are then used to provide the client with its access permissions and messaging capabilities:

* Access Control List (ACL) profiles—ACL profiles define whether the client is permitted to connect to the Message VPN, and, if it is, permissions are assigned to the client that set whether it can publish messages to topics and whether it can subscribe to topics. They also set whether those its publish and subscribe rights are limited to an explicit range of topics. For more information, refer to [Controlling Client Access with ACL Profiles](https://docs.solace.com/Features/Granting-Clients-Access.htm).
* Client profiles—Client profiles are sets of common configuration parameters that can be applied to groups of clients, which allows consistent configurations to be readily defined for many clients. For more information, refer to [Configuring Clients with Client Profiles](https://docs.solace.com/Features/Assigning-Client-Profiles.htm).

The following figure shows the basic process for authorizing an authenticated client according to the authorization properties assigned to a client username provisioned on the Message VPN.

**Authorization Process Using Provisioned Client Usernames**



## Authorizing Clients Through the Internal Database

When internal authorization is used, rather than LDAP group authorization, client usernames provisioned on the Message VPN will determine a client’s authorization. If the client provides a client username that matches a client username provisioned in the Message VPN that the client has connected to, the client and ACL profiles configured for that client username are applied to the client. If the client does not provide a client username, the router will attempt to apply the client username named default and the client and ACL profiles configured for that client username.

Internal authorization is the default authorization mode for Message VPNs.

**Note:**The default client username account always exists on the Solace router and cannot be deleted. However, by default, this account is not enabled.

After the client is bound to a client username account in the specified Message VPN, the router checks whether that client username account is enabled or not. If the client username account is not enabled, the client is disconnected. (The response “403 Client Username Is Shutdown” is sent before disconnecting.)

If the bound client username account is enabled, the client is then created with the properties of the client profile and ACL profile configured for the bound client username account object.

**Related Provisioning and Configuration Information**

* For information on how to configure client authentication and client profiles, refer to [Managing Client Authentication](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-Client-Authentication.htm).
* For information on how to configure ACLs, refer to [Configuring Access Control Lists](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-Access-Control-Lists.htm).

## Authorizing Clients Through LDAP Groups

Clients can also receive their authorizations based on whether they belong to specific LDAP authorization groups. Using LDAP authorization groups to authorize clients can assist network administrators that deal with large numbers of clients, especially when those clients are already configured in a corporate server, and churn frequently as employees join and leave an organization.

When LDAP authorization is enabled for the Message VPN that an authenticated client is attempting to connect to, an LDAP attribute (typically MEMBEROF) is retrieved for the client, and an LDAP lookup is made to an external LDAP server to determine the LDAP groups that the client belongs to.

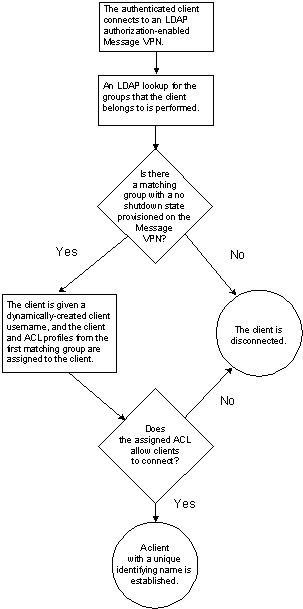
Any LDAP groups that the lookup returns are compared against the LDAP authorization groups configured on the router, and the client is assigned an matching authorization group with a no shutdown state that has the highest priority.

**Note:**The maximum number of authorization groups that may be retrieved for a client from an external LDAP server is 128. The client will not be authorized if more authorization groups are returned.

A client username is then automatically generated for the client, and the client and ACL profiles configured for the matching group are applied to the client username that the client is bound to. These profiles provide the client with its authorizations.

The following figure shows the basic process for authorizing an authenticated client that belongs to a configured LDAP authorization group.

**Authorization Process Using LDAP Groups**

****

**Related Provisioning and Configuration Information**

* For information on how to configure client authentication and client profiles, refer to [Managing Client Authentication](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-Client-Authentication.htm).
* For information on how to configure ACLs, refer to [Configuring Access Control Lists](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-Access-Control-Lists.htm).

# Configuring Clients with Client Profiles

Client profiles are objects provisioned on Message VPNs that are used to assign a common set of configuration properties to clients that have been successfully authorized.

Client profiles control a number of client behaviors and capabilities. For example, client profiles control the allocation of resources such as the maximum number of subscriptions permitted for a single client and the per-client transport queues. Other characteristics controlled by client profiles include tuning TCP connection parameters, enabling persistent messaging capabilities, and adjusting the point at which certain events are triggered.

Client profiles can be applied to multiple client usernames or LDAP authorization groups in a Message VPN. This enables administrators to manage large groups of clients by making a configuration change once and having it apply to many clients rather than having to make individual changes to each client.

If specific client profiles are not assigned to a client username account, then the client profile default is assigned automatically to each client username account. You can customize the configuration of the default client profile, but it cannot be deleted from the router.

**Related Provisioning and Configuration Information**

For information on how to use the Solace CLI to configure client profiles and the parameters that are associated with client profiles, refer to [Configuring Client Profiles](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-Client-Profiles.htm).

# Controlling Client Access with ACL Profiles

After a client is successfully authenticated, the ACL profile assigned to the client username used by the client, or the LDAP the authorization groups that the client belongs to (when LDAP authorization is used), is checked. This ACL profile controls whether an authenticated client is:

* permitted to establish a connection with the router (refer to [Client Connection Access Controls](https://docs.solace.com/Features/Granting-Clients-Access.htm#Client-Connect-Access-Controls))
* permitted to publish and subscribe to specific topics (refer to [Topic Access Controls](https://docs.solace.com/Features/Granting-Clients-Access.htm#client_authentication_and_authorization_2866343001_215561))

**Note:**

* If an administrator does not assign a specific ACL profile to a client username, it is automatically assigned the ACL profile named default. This default ACL profile is preconfigured for each Message VPN on a Solace router. Although you can modify the configuration of the default ACL profile, this ACL profile cannot be deleted.
* All internal clients that are automatically created by the system for internal features (for example, the Config-Sync client #config-sync) are assigned to the ACL profile #acl-profile.

**Related Provisioning and Configuration Information**

For information on how to use the Solace CLI to configure and manage ACLs on the Solace router, refer to [Configuring Access Control Lists](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-Access-Control-Lists.htm).

## Client Connection Access Controls

ACL profiles have a client connection access control that is used to determine whether clients are allowed to connect to a Solace router.

Client connection attempts can either be allow (permit client connections) or disallow (deny client connections). The default action for user-defined ACLs is disallow. The default actions for the default and #acl-profile ACL profiles is allow.

After you have set the default client connection action, you can create a list of clients that should be exceptions to the default action. Clients to be excepted must be expressed in Classless Inter‑Domain Routing (CIDR) address form (nnn.nnn.nnn.nnn/nn). Any client whose address falls into any of the IP/mask in this list gets the opposite behavior to the configured default action. There is a system-level limit of 10,000 ACL profile connect exceptions.

For example, if the client-connect access control for an ACL uses a default action of allow, but 10.1.1.0/24 is listed as an exception, clients on the 10.1.1.0/24 network are denied access to the Message VPN. Similarly, if the default client connection action is disallow, and a client on a network on the exceptions list attempts to connect, that client is connected with no restrictions.

**Note:**Changing the default client connect action, or removing clients from the exceptions list, does not immediately affect clients that already have an established connection to the Solace router. They remain connected.

A global statistic and a per-message VPN statistic is incremented for every denied connection attempt. In addition, a circular log is also maintained capturing:

* the current timestamp
* the IP/port of the denied client
* the username of the denied client
* the Message VPN of the denied client
* the client name of the denied client (shown only when the **show log acl** User EXEC command is used with the wide parameter option)
* the ACL profile name that triggered the denial (shown only when the **show log acl** User EXEC command is used with the wide parameter option)

## Topic Access Controls

ACL profiles provide publish-topic and subscribe-topic access controls, which enable you to set which topics Solace Message Format (SMF) and Message Queuing Telemetry Transport (MQTT) clients may publish to and which topics they may subscribe to.

An ACL profileʼs publish-topic access control will have:

* A default action of allow or disallow, where allow sets the default action to allow the publishing of messages to topics, and disallow sets the default action to disallow the publishing of messages to topics. The configured default action applies to both SMF and MQTT topics.
* An optional list of specific topics that you want to be excepted from the default publish-topic action. Topic exceptions must be made for SMF and MQTT topics separately.

An ACL profileʼs subscribe-topic access control will have:

* A default action of allow or disallow, where allow sets the default subscribe action to allow clients to subscribe to topics, and disallow sets the default subscribe action to disallow clients from subscribing to topics. The configured default action applies to both SMF and MQTT topics.
* An optional list of specific topics that you want to be excepted from the default subscribe-topic action. Topic exceptions must be made for SMF and MQTT topics separately.

There is no limit to the number of publish or subscription topic exceptions per ACL profile. However, there is a maximum of 40,000 exceptions (publish and subscription topic exceptions combined) allowed among all profiles. Also keep in mind that the more exceptions there are, the more difficult it is to comprehend and manage your topic access control configuration.

The topics that these access controls affect can be regular topic subscriptions, as well as reserved Solace topics that are used for special messaging features. For example, there are unique topics allow clients to publish and subscribe to topics bound to durable queues (#P2P/QUE/<queue‑name>) and topics that are only for the consumption of one specific client (#P2P/v:routerName/clientUsernameHash/clientName/>). For a complete list of special topics and their usage, see [Special Solace Topics](https://docs.solace.com/Features/Special-Solace-Topics.htm).

### Using Substitution Variables in Topic Exceptions

Substitution variables are a feature of ACLs (supported by Solace VMR Version 8.3.0+ and Solace appliance Version 8.3.0+), useable in conjunction with any client protocol, that can be helpful for deployments that have a lot of connecting clients (for example, IoT deployments may have thousands or hundreds of thousands of clients) where you may want to configure ACL rules so that those clients can only:

* publish to topics containing their own client username or MQTT client ID (to prevent a client from impersonating another client)
* subscribe to topics containing their own client username or MQTT client ID (to prevent a client from receiving data that is intended for another client)
* listen to their own #P2P topic (to prevent a client from receiving a reply message that is intended for another client)

To manually set up these types of restrictions for large numbers of clients is not practical. So, instead you can configure ACL topic exceptions with special substitution variables for one of the following:

* client username ($client-username)—When $client-username is encountered in an ACL topic exception, the router will substitute the client username provided by the client or retrieved from the TLS client-certificate during client login.
* client username hash ($client-username-hash)—When a client attempts to establish a connection, the router will automatically create a #P2P topic (#P2P/v:routerName/client-username-hash/clientname/>) for the client so that it can use Direct Request/Reply messaging. The router creates this topic based on the client username provided by the client or retrieved from the TLS client-certificate during client login. The variable $client-username-hash represents the unique 8-byte long “client-username-hash” that the router generates for clients using that client username.
* client’s MQTT ID ($client-id)— This configuration only applies to MQTT sessions. When $client-id is encountered in an ACL topic exception, the router will substitute the MQTT client-ID that the client provided when it established its session.

The strings $client-id, $client-username, and $client-username-hash have special meaning only for ACL exception topics. If they are used in regular topic subscriptions, they will be treated as literal strings and have no special meaning.

If substitution variables are used in ACL topic exceptions, then when the router applies the ACL profile for a client, it will replace the variables in the topic exceptions with the corresponding client username or MQTT client ID that the client provided when it established its connection to the router or the client-username-hash generated by the router.

Substitution variables must appear alone at any level in the topic exception, but they may appear multiple times in a single exception. For example, FRUIT/$client-id/$client-username/APPLES are valid uses of substitution variables. While FRUIT/$client-id\* and FRUIT/$client-id$client-id/\* are not valid because $client-id does not stand alone at a level.

https://docs.solace.com/Resources/Images/banner_alert2.gif

If you use substitution variables in ACL topic exceptions, you must ensure that client usernames or MQTT client IDs do not contain the “/” character. Although “/” is a valid character in MQTT client IDs or client usernames, because the router uses “/” as a delimiter between levels in a topic, any MQTT client IDs or client usernames that contain “/” characters will never be able to match the $client-id or $client-username elements of an ACL rule. Note that the router does not enforce this restriction.

### Wildcard Handling

Subscriptions are either fully accepted or fully rejected depending on whether they match the configured topic access controls. Special rules are employed when handling subscriptions containing wildcards to ensure configured ACLs are effective in blocking the traffic they have been configured to disallow. Wildcard subscriptions that match an ACL profile’s exceptions are disallowed if the ACL profile’s default rule is to allow all subscriptions. For example:

* If an ACL profile has been configured to allow all subscriptions except for the SMF subscription ANIMALS/CATS, a subscription to ANIMALS/> (covering ANIMALS/CATS) is disallowed. If ANIMALS/> were accepted, then messages published to ANIMALS/CATS would match ANIMALS/> and be delivered to the client. This would contradict the intention of the ACL.
* If the ACL profile’s default rule disallows all subscriptions, wildcard characters in the subscription are not given any special treatment when establishing matching exception rules. For example, if an ACL profile has been configured to disallow all subscriptions except ANIMALS/DOGS, an SMF subscription request to ANIMALS/> would be disallowed given that the “>” would not be treated as a wildcard character and therefore not cover the exception rule of ANIMALS/DOG. In suppressing the subscription, which requested everything below ANIMALS, the ACL profile’s intention of only allowing access to ANIMALS/DOG is enforced.

### Subscribe Topic Access Controls with Guaranteed Messaging

ACLs can be used with Guaranteed Messaging, but the following limitations should be considered:

* Subscribe-topic access controls are not considered when a consuming client binds directly to a durable endpoint (either a queue or topic endpoint). Therefore, a client could bypass subscribe-topic access control restrictions that have been set for it.
* Subscribe-topic access controls can prevent a client from adding a specific topic subscription to a queue if the client is not permitted to subscribe to that topic. However, a client can bind to an existing queue that has been mapped to a topic that the client is not permitted to subscribe to because the client’s subscribe-topic access controls are not checked when it binds to a queue.

### Topic Access Control Statistics

A global, per-Message VPN, and per-client statistic is incremented for every denied publish or subscribe topic attempt. In addition, a circular log is also maintained capturing:

* the current timestamp
* the username of the denied client
* the Message VPN the client was a member of
* the topic that was denied
* the client name of the denied client (shown only when the **show log acl** User EXEC command is entered with the wide parameter option)
* the ACL profile name that triggered the denial (shown only when the **show log acl** User EXEC command is entered with the wide parameter option)

# Topic Support and Syntax

This section describes the topic syntax supported by the Solace Messaging Platform and provides examples of valid topics for publishing and subscribing.

* publishing—Topics are used as subjects/destinations that clients can publish messages to.
* subscribing—Topics are used as subscriptions to attract published messages of interest for consumer clients. That is, when messages are published to a topic, matching topic subscriptions will attract those messages. If that topic subscription is registered for a client or for an endpoint that the client is bound to, a client with the proper authorization may receive that message.

**Note:**The SMF and MQTT protocols use similar topic syntax, so SMF and MQTT messaging applications can be used together in the same messaging network. However, there are some differences in topic syntax and usage between SMF and MQTT, and some planning considerations must be made if you are going to deploy MQTT applications along with SMF applications. See [MQTT Topics](https://docs.solace.com/Open-APIs-Protocols/MQTT-Topics.htm) for more information.

# SMF Topics

Solace Message Format (SMF) is the Solace Messaging Platform’s underlying messaging protocol.

Topics consist of NULL-terminated UTF-8 strings composed of one or more levels, in the format of “a/b/c”. The “/” character is used as a separator between “levels” (also called elements) of the string. A maximum of 128 topic levels are supported.

The maximum length of the string, excluding the NULL terminator, is 250 bytes.

Characters in topic strings are case-sensitive. Therefore, for example, a topic subscription of “animals/domestic/cats” is not equal to “Animals/Domestic/Cats”.

All UTF-8 characters are supported. However, the following characters have special meaning:

* “/” is a level separator for both publish and subscribe topics.
* <NULL> is an invalid character within a string for both publish and subscribe topics.
* “\*” is a single-level wildcard for subscribe topics only. For publish topics, “\*” is treated as a literal character.
* “>” is a multi-level wildcard when used at the last level of a topic subscription. “>” can only be used as a wildcard for subscribe topics only—for publish topics, “>” is treated as a literal character.
* A leading "!" (for example "!a/b/c") in queue topic subscriptions indicates a subscription exception. For more information, refer to [System-Level Subscription Exception Configuration](https://docs.solace.com/Configuring-and-Managing-Routers/System-Level-Subscription-Exception-Config.htm). For publish topics, a leading "!" is treated as a literal character.

To allow compatibility with MQTT topics, which permit empty topic levels (for example, “/a/b”, “a//b”, or “a/b/”), Solace routers Version 7.1.1 or greater also allow empty topic levels in SMF topics. However, there are some limitations to using empty topic levels with SMF topics:

* Client applications using Solace messaging APIs can publish to topics with empty levels, but they cannot use topic subscriptions with empty levels.
* ACL profiles do not allow SMF publishing or subscribing topic exceptions that have empty topic levels.

## Wildcard Characters in SMF Topic Subscriptions

When the “\*” and “>” characters are used in topic subscriptions, they function as wildcards, which affect the topic matches that may occur.

**Note:**The “\*” and “>” characters do not have any special meaning when they are used in topics that messages are published to. They are treated as literal characters—not as wildcards—and will not affect topic matches.

The use of the “\*” and “>” wildcard characters in topic subscriptions may affect the topic matches in the following ways:

* The “\*” character, when it appears by itself at a level within a subscription topic (for example, “animals/\*/cats”, or “animals/domestic/\*”), indicates a wildcard match at that level. A “\*” wildcard at the end of the subscription topic only performs a wildcard match at that level. For example, “animals/domestic/\*” matches the topics “animals/domestic/cats” and “animals/domestic/dogs”, but it does not match the topic “animals/domestic/dogs/beagles”.
* The “\*” wildcard character is allowed in conjunction with topic prefixes at a level within a subscription. For example, “animals/red\*/wild”. In this case, the “\*” wildcard provides a “0 or more” match—the topics “animals/red/wild” and “animals/reddish/wild” are both matches.

Further, use of the “\*” wildcard character within a level supports prefix matching, but not general sub-string matching. For example, while “animals/\*/brown” and “animals/domestic/white\*” are allowed, but “animals/\*bro” and “animals/br\*wn” are invalid as wildcard subscriptions and the subscription topic is not permitted.

* The “\*” wildcard may be used at multiple levels within a subscription topic, with or without topic prefixes. For example, “animals/\*/cats/\*” is valid and matches the topics “animals/domestic/cats/persian” and “animals/wild/cats/leopard”, but does not match the topic “animals/domestic/cats/persian/grey”, nor does it match the topic “animals/domestic/dogs/beagles”.
* The “>” character, when it appears by itself at the last level of a subscription topic (for example, “animals/domestic/>”), provides a “one or more” wildcard match for any topics with an identical prefix to the subscription. For example, “animals/domestic/>” does not match the topic “animals/domestic”, but it does match the topics “animals/domestic/cats”, “animals/domestic/dogs”, “animals/domestic/dogs/beagles”, and “animals/domestic/dogs/beagles/long-eared”.
* A “>” character that appears anywhere else other than by itself at the last level of a subscription topic in the string is treated as the “>” character rather than a wildcard. For example, “animals>” and “animals/domestic>” are literal subscriptions and do not match "animals/domestic/dogs/beagles".
* The “>” and “\*” characters may be used together within a subscription topic. For example, “animals/\*/cats/>” matches the topics “animals/domestic/cats/persian”, “animals/wild/cats/leopard”, and “animals/domestic/cats/persian/grey”, but it does not match the topic “animals/domestic/dogs/beagles”.

### Wildcard Restrictions

There are some restrictions on wildcard behavior:

* A Solace router automatically creates a topic subscription beginning with the prefix “#P2P” for each client, which enables messages to be sent directly to that client (for example, in request/reply scenarios). Therefore, to ensure that wildcards cannot be used to receive messages that are intended for the inboxes of other clients, the “\*” or “>” wildcards will never match the string “#P2P”, regardless of where the wildcard or “#P2P” string is located within the topic subscription string.
* Messages published to topics that begin with a “$” character will never be matched by standalone wildcards (“\*/...” or “>”) at the first level of a topic subscription. This is to ensure that system and event log messages beginning with a “$” character are not included in topic subscriptions for which they were not intended.

### Wildcard Usage Examples

The following table provides some examples of topic subscription matches when using wildcards.

| Topic Subject Matches for Subscriptions Using “\*” and “>” Wildcards | | |
| --- | --- | --- |
| **Wildcard Subject** | **Matches Messages With Subjects Like:** | **Does Not Match Messages With Subjects Like:** |
| animals/domestic/\* | animals/domestic/cats animals/domestic/dogs | animals/domestic/dogs/beagles |
| animals/\*/cats/\* | animals/domestic/cats/persian animals/wild/cats/leopard | animals/domestic/cats/persian/grey animals/domestic/dogs/beagles |
| animals/domestic/dog\* | animals/domestic/dog animals/domestic/doggy | animals/domestic/dog/beagle animals/domestic/cat |
| animals/domestic/> | animals/domestic/cats animals/domestic/dogs/beagles | animals animals/domestic animals/Domestic |
| animals/\*/cats/> | animals/domestic/cats/tabby/grey animals/wild/cats/leopard | animals/domestic/dogs/beagles |
| my/test/\* | my/test/topic | My/Test/Topic my/test |

# Special Solace Topics

The Solace messaging platform uses some reserved topics to implement some specific messaging features. Each of these special topic subscriptions begin with the “#” character, which is then followed by a specific, reserved sequence of characters. Some special topics only use a reserved string for the first hierarchical level of the topic subscription, others may reserve more than one level.

Solace Special Topic Subscriptions

| **Topic Subscription** | **Published Topics That Match the Subscription** | **Description** |
| --- | --- | --- |
| #LOG/... | #LOG/level/SYSTEM/hostName/event/Name #LOG/level/VPN/hostName/eventName/vpnName[/suffix] #LOG/level/CLIENT/hostName/eventName/vpnName/clientName #LOG/level/APPLICATION/hostName/eventName/vpnName[/suffix] #LOG/INFO/SUB\_ADD/subscribedTopic #LOG/INFO/SUB\_DEL/subscribedTopic | Used by the router to publish events to the message bus.  Published by: router Subscribed to by: management client applications |
| #MCAST/> | #MCAST/CACHEMGR | Used by SolCache instances to broadcast initial messages to find their Cache Manager. Published by: SolCache Instances Subscribed to by: router |
| #P2P/CACHEINST/distibutedCacheName | #P2P/CACHEINST/distibutedCacheName | Used to direct SolCache requests to a distributed cache. Published by: client applications Subscribed to by: SolCache Distributed Cache |
| #P2P/CACHEINST/cacheClusterName | #P2P/CACHEINST/cacheClusterName | Used to direct SolCache requests to a cluster in a distributed cache. Published by: client applications Subscribed to by: SolCache Cache Cluster |
| #P2P/CACHEINST/cacheInstanceName | #P2P/CACHEINST/cacheInstanceName | Used to direct SolCache requests to a specific cache instance in a distributed cache. Published by: client applications Subscribed to by: SolCache |
| #P2P/routerName/clientUsernameHash/#client/> | #P2P/routerName/clientUsernameHash/#client/CACHEMGR | Used by SolCache instances to communicate with their Cache Manager.  Published by: SolCache Subscribed to by: router (VMR Version 8.2.x or greater) |
| #P2P/routerName/#client/> | #P2P/routerName/#client/CACHEMGR | Used by SolCache instances to communicate with their Cache Manager.  Published by: SolCache Instances Subscribed to by: router (Solace VMRs prior to Version 8.2.x and on Solace appliances) |
| #P2P/v:routerName/clientUsernameHash/clientName/> | #P2P/v:routerName/clientUsernameHash/clientName/... | Used for Direct Messaging request/reply. The name space is controlled by the client API, but router sets the prefix. This topic is used by Solace VMRs Version 8.2.x or greater.  Published by: client applications Subscribed to by: client API |
| #P2P/v:routerName/clientName/> | #P2P/v:routerName/clientName/... | Used for Direct Messaging request/reply. The name space is controlled by the client API, but router sets the prefix. This topic is used for Solace VMRs prior to Version 8.2.x and for Solace appliances.  Published by: client applications Subscribed to by: client API |
| #P2P/QUE/queueName | #P2P/QUE/queueName | Subscriptions that allow clients to publish messages to and/or consume messages from the specified durable queues. Published by: client applications Subscribed to by: client applications |
| #P2P/QTMP/v:routerName/queueName | #P2P/QTMP/v:routerName/queueName | Subscriptions that allow clients to publish messages to and/or consume messages from the specified non-durable queues. The client App/Mgmt controls the name space, but the router sets the prefix. Published by: client applications Subscribed to by: client applications |
| #P2P/TTMP/v:routerName/tempTopic | #P2P/TTMP/v:routerName/tempTopic | Used for temporary topics. Name space is controlled by the client API, but router sets the prefix. Published by: client applications Subscribed to by: client applications |
| #P2P/v:routerName/> | #P2P/v:routerName/... | Subscription created when a virtual router is discovered, to attract #P2P traffic for the router.  This subscription is not visible in CLI. Published by: client applications Subscribed to by: router |
| #P2P/routerName/> | #P2P/routerName/... | Subscription always created when a physical router is discovered, to attract #P2P traffic  for the router. This subscription is not visible in CLI. Published by: client applications Subscribed to by: router |
| #SEMP/routerName/> | #SEMP/routerName/SHOW #SEMP/routerName//ADMIN/CLIENT #SEMP/routerName/ADMIN/DISTRIBUTED-CACHE | Used to direct SEMP commands to a physical router name. Published by: client applications Subscribed to by: router |
| #SEMP/v:routerName/> | #SEMP/v:routerName/SHOW #SEMP/v:routerName//ADMIN/CLIENT #SEMP/v:routerName/ADMIN/DISTRIBUTED-CACHE | Used to direct SEMP commands to a virtual router name. Subscription is only present when a given router is active for the virtual router name. Published by: client applications Subscribed to by: router |

# TLS/SSL Message Encryption

By default, the clients use single TCP connections instead of plain text over TCP to exchange data with Solace routers, and the data that is exchanged is not compressed.

However, clients can use Transport Layer Security (TLS)/ Secure Sockets Layer (SSL)‑encrypted connections to a Solace router. This allows an exchange of uncompressed Solace Message Format (SMF) data or Solace Element Management Protocol (SEMP) data with the Solace router using TLS/SSL over single TCP connections instead of plain text over TCP.

Using TLS/SSL-encrypted connections to a router offers:

* confidentiality—messages can only be received by the intended recipient
* message integrity—messages cannot be modified after being sent
* server authentication—the connected application is able to verify the server it is communicating with is the server it believes it is
* client authentication—optional client certificates can be used as a form of client authentication (for more information, refer to [Client Authentication/Authorization](https://docs.solace.com/Features/Client-authentication-and-authorization.htm))

The following types of applications can establish TLS/SSL-encrypted connections to a Solace router:

* Client applications that publish and/or receive messages.

By default, when a client application connects to a Solace router and creates a session, that session is unsecured. However, a client application can optionally create a secure session that requires a trusted server certificate to establish a TLS/SSL-encrypted client connection to a Solace router.

Client applications can use the Solace messaging APIs to establish secure connections to a Solace router.

* Network management applications that manage and monitor the Solace router through SEMP requests and replies.
  + If the network management application uses SEMP Request Over Message Bus service, which uses a Solace messaging API to establish a client connection through a session, SEMP requests and replies can be sent over the router message bus. A secure session can be established in the same manner as that used by a client application.
  + If the network management application uses SEMP Request over HTTP service, a TCP connection to a port on the router allows the application to send SEMP requests wrapped in HTTP to a management interface on the router and receive replies to them. For SEMP Request over HTTP service, a secure connection can be created when a server certificate is set on the router, a TLS/SSL-secure port is specified (443 by default), and an HTTPS URL scheme is used.

SSL encryption can also be used on Message VPN bridges and bridges established by the Message VPN replication and replication Config-Sync facilities. Using encryption on bridge links helps to ensure that communications between Solace routers over bridges are secure.

**Note:**Because encrypting and decrypting data requires processing power, enabling TLS/SSL encryption may cause a reduction in performance.

## Cipher Suites

The following cryptographic algorithms are used throughout the life of a TLS/SSL‑encrypted connection:

1. Key establishment—This algorithm is used to exchange or agree on the symmetric keys to be used for encrypting and decrypting the data payload during the session. Examples: RSA, Diffie-Hellman (DH), Ephemeral Diffie-Hellman (DHE), and Elliptic Curve Diffie-Hellman (ECDH).
2. Authentication—This algorithm is the digital signature used by the certificates passed between the client application and server. Examples: RSA and Digital Signature Standard (DSS).
3. Encryption—This algorithm encrypts and decrypts payload passed on the secure session. Examples: RC4, 3DES CBC, and Advanced Encryption Standard (AES).
4. Digest—This algorithm is used to maintain message integrity. Tampering with the message would render the digest invalid. Examples: SHA1, MD5.

The combination of these four cryptographic algorithms is known as a cipher suite.

For the full list of cipher suites Solace routers support, refer to [Supported Cipher Suites](https://docs.solace.com/Features/TLS-SSL-Message-Encryption.htm#transport_security_and_compression_440573439_213360)

### Cipher Suites for Inbound Connections

Solace routers maintain a separate cipher suite list for each of the following inbound TLS connection types:

* Network management applications using SEMP to manage the Solace router (management connections)
* Client applications connecting to the Solace router to publish or receive messages (message backbone connections)

Cipher suites for all inbound management and message backbone connections are selected according to the priority clients associate with each cipher suite.

### Cipher Suites for Outbound Connections

Solace routers maintain a separate prioritized cipher suite list for each of the following outbound TLS connection types:

* Message VPN and bridge connections
* Replication bridge connections
* Config-sync bridge connections
* Multi-node routing links

In addition, the router uses the same cipher suite list for outbound LDAP connections that it uses for inbound management connections.

Cipher suites for all outbound connections (including LDAP connections) are selected according to the priority configured in each corresponding cipher suite list.

### Ciphers for SSH Connections

Solace routers maintain a list of ciphers for all SSH, SCP, and SFTP connections. For purposes of encrypted connections, the cipher list has a similar function to a cipher suite list; however, key establishment, authentication, and digest algorithms are not used. For the full list of ciphers Solace routers support, refer to [Supported Ciphers](https://docs.solace.com/Features/TLS-SSL-Message-Encryption.htm#Supporte).

Ciphers for all SSH connections are selected according to the priority clients associate with each cipher.

## Supported Cipher Suites

The following table shows the full list of cipher suites Solace routers support for inbound and outbound connections.

| Supported Cipher Suites | | | | |
| --- | --- | --- | --- | --- |
| Cipher Suites  (strongest to weakest) | Key Establishment | Authentication | Encryption | Digest |
| ECDHE-RSA-AES256-GCM-SHA384 | ECDH | RSA | AES GCM (256) | SHA2 |
| ECDHE-RSA-AES256-SHA384 | ECDH | RSA | AES CBC (256) | SHA2 |
| ECDHE-RSA-AES256-SHA | ECDH | RSA | AES CBC (256) | SHA1 |
| AES256-GCM-SHA384 | RSA | RSA | AES GCM (256) | SHA2 |
| AES256-SHA256 | RSA | RSA | AES CBC (256) | SHA2 |
| AES256-SHA | RSA | RSA | AES CBC (256) | SHA1 |
| ECDHE-RSA-DES-CB3-SHA | ECDH | RSA | 3DES CBC (168) | SHA1 |
| DES-CBC3-SHA | RSA | RSA | 3DES CBC (168) | SHA1 |
| ECDHE-RSA-AES128-GCM-SHA256 | ECDH | RSA | AES GCM (128) | SHA2 |
| ECDHE-RSA-AES128-SHA256 | ECDH | RSA | AES CBC (128) | SHA2 |
| ECDHE-RSA-AES128-SHA | ECDH | RSA | AES CBC (128) | SHA1 |
| AES128-GCM-SHA256 | RSA | RSA | AES GCM (128) | SHA2 |
| AES128-SHA256 | RSA | RSA | AES CBC (128) | SHA2 |
| AES128-SHA | RSA | RSA | AES CBC (128) | SHA1 |

**Note:**When using TLS 1.2, certificate keys must be at least 1024 bits in length. When using keys shorter than 1024 bits, you may receive a “digest too big for RSA key” error.

## Supported Ciphers

The following table shows the full list of ciphers Solace routers support for SSH connections.

| Supported Ciphers |
| --- |
| **Ciphers** |
| aes128-ctr |
| aes192-ctr |
| aes256-ctr |
| aes128-gcm@openssh.com |
| aes256-gcm@openssh.com |
| chacha20-poly1305@openssh.com |
| aes128-cbc |
| 3des-cbc |
| blowfish-cbc |
| cast128-cbc |
| aes192-cbc |
| aes256-cbc |

## Configuring Cipher and Cipher Suite Lists

You can configure each default cipher and cipher suite list to avoid using specified ciphers or cipher suites, or you can reorder each list to change the priority associated with a given cipher or cipher suite.

To configure cipher suite lists for inbound connections, refer to [Configuring Cipher Suites for Inbound Connections](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-Cipher-Suites-for-Inbound-Connections.htm).

To configure cipher suite lists for outbound connections, refer to the following sections:

* For Message VPN and bridge connections, refer to [Configuring VPN Bridges](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-VPN-Bridges.htm).
* For replication bridge and replication Config-Sync connections, refer to [Configuring System-Level Replication Settings](https://docs.solace.com/Configuring-and-Managing-Routers/Replication-Sys-Level-Settings.htm).
* For LDAP connections, refer to [Configuring Cipher Suites for Management Connections](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-Cipher-Suites-for-Inbound-Connections.htm#Configur). The router uses the same cipher suite list for LDAP and management connections.
* For multi-node routing links, refer to [Managing Multi-Node Routing](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-Multi-Node-Routing-Links.htm).

To configure ciphers for SSH connections, refer to [Configuring Ciphers for SSH Connections](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-Cipher-Suites-for-Inbound-Connections.htm#Configur3).

## Supported Signature Algorithms

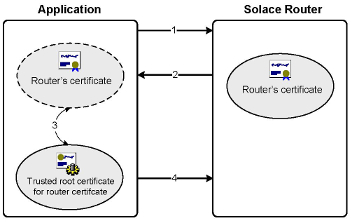
A Solace router can validate a certificate chain if all certificates in the chain are signed with one of the following signature algorithms:

| Supported Signature Algorithms |
| --- |
| **Signature Algorithm** |
| md2withRSAEncryption |
| md4withRSAEncryption |
| md5withRSAEncryption |
| shawithRSAEncryption |
| sha1withRSAEncryption |
| sha1withDSAEncryption |
| sha224withRSAEncryption |
| sha224withDSAEncryption |
| sha256withRSAEncryption |
| sha256withDSAEncryption |
| sha384withRSAEncryption |
| sha512withRSAEncryption |

## Server Certificates Authentication Process

The following process is used when a client or management application connects to a Solace router and requires certificate validation to establish a secure connection.

Certificate Validation Process Using TLS/SSL



1. The client or management application connects to the router’s TLS/SSL listening port.
2. The router responds to the client or management application with its server certificate.

If an Active/Active redundancy configuration is used, each router in the redundant pair should have its own server certificate with its own common name. If an Active/Standby redundancy configuration is used, the two mates can share a server certificate that uses the name of the primary router as the common name.

1. The client or management application verifies that the server certificate is valid, is named as expected, and that it was issued by a trusted root certificate.
2. If the validation succeeds, the client or management application will respond and continue to complete the handshake.

## Certificate Authorities

A Solace router allows clients to authenticate over TLS by presenting a valid client certificate issued by a Certificate Authority (CA). Each participating router must be configured with a list of CAs to be trusted so that the client's certificate obtained during the security protocol exchanges can be verified. The process of configuring each CA in the client certificate chain as a trusted root certificate is required for authentication and is a mandatory step in trusting a CA. A Solace router can also be configured to use a CA with certificate revocation checking.

**Note:**Certificate authority and certificate revocation checking is currently not supported for Solace VMRs. Trusted roots are still used for VMRs to authenticate client certificates.

### Certificate Revocation Checking

As of SolOS version 8.2.0, checking the revocation status of client certificates is supported for Solace appliances. Before this release, while validating the client certificates, the router did not check the revocation status of the certificates.

A Solace router can check the revocation status of the certificate that clients use when attempting to authenticate. The certificate revocation is configurable on a per-router basis with override options per-Message VPN.

The revocation status of a certificate chain can have one of the following states:

* Valid—All certificates in the chain have been checked for revocation and for whether they are valid.
* Unknown—One or more certificate in the chain cannot be verified. It is, however, possible for OCSP to return unknown if the certificate is not revoked and not known to the CA.
* Revoked—One or more certificate in the chain have been revoked.

Certificate revocation checking is a router-wide feature; if the certificate revocation checking is enabled, the revocation status of all the certificates will be checked.

The table below shows the list of clients that support certificate revocation checking.

Clients That Support Certificate Revocation Checking

| **Authentication Source** | **Certificate Revocation Checking Supported** |
| --- | --- |
| SMF client | Yes |
| MQTT client | Yes |
| REST publisher | Yes |
| REST subscriber | No |
| MQTT client | Yes |
| Web Messaging client | Yes |

For information on certificate revocation configurations, see [Managing Certificate Authorities](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-Certificate-Authorities.htm).

### Certificate Revocation Checking Methods

You can configure Solace routers to check the revocation status of client certificates using Certificate Revocation List (CRL), Online Certificate Status Protocol (OCSP), or OCSP-CRL (a combination of both the methods).

**CRL**

CRL contains a list of certificates that have been revoked by the CA. The URL to the CAs certificate revocation list is embedded in each certificate. The router downloads the CRL from the distribution point and keeps a local copy that it refreshes periodically. To authenticate client certificates, the verifying application uses the URL to check the revocation status of all the certificates in the chain and its issuing CA—starting from the last issued certificate—to the root CA itself. If the serial numbers of the certificates do not appear in the CRL, then the certificates pass the revocation check.

**Note:**During CRL revocation check, the root CA gets two counts because it is checking itself once and checking the CA below in the chain once. All the other CAs get one count because they are only checking the one below them or, in the case of the last CA in the chain, it's checking the client that tried to connect.

To enable CRL certificate revocation checking, see [Configuring CRL Certificate Revocation Checking](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-CRL-Certificate-Revocation.htm).

**OCSP**

OCSP allows a router to check the revocation status of a single certificate directly with the CA or through a trusted responder (OCSP responder) inside the firewall. The client certificate contains a URL of the issuer of the certificate, which is used to check the revocation status of the certificate. With the OCSP method, each certificate in the chain is verified by sending a request with the certificate’s serial number to an OCSP responder. The responder must reply to indicate if the certificate has been revoked or whether it is valid or unknown.

To enable OCSP certificate revocation checking, see [Configuring OCSP Certificate Revocation Checking](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-OCSP-Certificate-Revocation.htm).

**OCSP-CRL**

When both the OCSP and CRL certificate revocation checking are used, the revocation status of all the certificates in the chain will be first checked by OCSP. If the result is valid or unknown, then the check is complete. If the OCSP responder returns are unknown (response timeout) or unavailable, then the certificate will be checked by the CRL. This process is performed on each certificate in the chain.

To enable OCSP-CRL certificate revocation checking, see [Configuring OCSP-CRL Certificate Revocation Checking](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-OCSP-CRL-Certificate-Revocation.htm).

# Message Compression

An application using the Solace enterprise or OpenMAMA messaging APIs can enable the compression of message data sent between it and the Solace Messaging Platform so that message data is compressed before transmission and decompressed on reception.

**Note:**Message compression is not supported for clients using Solace Web messaging APIs, REST, or MQTT.

Message compression reduces the size of message data frames to be transmitted over a network link. Reducing the size of a frame reduces the time required to transmit the frame across the network. Data compression provides a coding scheme at each end of a transmission link that allows characters to be removed from the frames of data at the sending side of the link and then replaced correctly at the receiving side. Because the condensed frames take up less bandwidth, greater volumes of data can be transmitted at any one time.

Using data compression may reduce bandwidth consumption and provide lower latency over low-bandwidth connections between clients and routers.

## Functional Description

On Solace routers, compression and decompression of message data is performed by the Network Acceleration Blade (NAB). For clients using the messaging APIs, compression and decompression of message data is performed by the third-party library.

When channel compression is used, a Solace router listens to a special TCP port (by default, 55003). Therefore, the value for the session property on the client side must be set to match the port number on the router.

For compression on the router side, the message data is sent from the publishing client into the NAB compression engine of the Solace router for compressing. Once the compressed data is available, it is sent out to the subscribing clients over the client connection.

For decompression on the router side, the Solace router puts the compressed message data received from the connection into the NAB decompression engine for decompressing. Once the decompressed message data is available, it is handed over to the upper layer application for further processing.

**Note:**If an unrecoverable error is detected during decompression operation, the associated connection is disconnected. This applies to both the router and client API ends of the transmission link.

# Management User Authentication/Authorization

There are two types of management users that can connect to a Solace router:

* CLI user—A user that connects to a router for the purpose of configuring, managing, and monitoring that router. A CLI user can be an administrator that uses the Solace Command Line Interface (CLI) or SolAdmin (a GUI-based equivalent of the Solace CLI). It can also be a management application that uses Solace Element Management Protocol (SEMP) request over HTTP service. (For information on SEMP service, see [Using SEMP](https://docs.solace.com/SEMP/SEMP-Home.htm).)
* File Transfer user—A user that can remotely transfer files to and from specific directories on the selected Solace router using Secure File Transfer Protocol (SFTP) or Secure Copy (SCP).

**Note:**When a user successfully logs in or logs out, or fails to authenticate for a CLI, SEMP, shell, scp, or sftp session to the router, an authentication event is written to the event log. If SEMP is used to manage the router, a persistent SEMP connection should be used if a high volume of authentication event logs is not desired.

## CLI User Authentication

When a CLI user initially establishes a connection to the router, no requests by that user can be processed until it is authenticated according to the authentication type that has been configured for that CLI user account.

The following authentication types can be configured for a provisioned CLI user account:

* Internal—The CLI user’s name and password are authenticated against an internal Solace router database.
* RADIUS—The CLI user’s name and password are sent to an external RADIUS server for authentication.
* LDAP—The CLI user’s name and password are sent to an external LDAP server for authentication. LDAP over TLS is also supported.

The Solace router always first attempts to authenticate a CLI user using internal authentication. If user does not exist in the internal database, the router checks if either RADIUS or LDAP authentication is configured (one of the two can be configured for the router). If RADIUS or LDAP authentication is configured, the router then attempts to authenticate the user using one of those authentication types. For information on how to configure RADIUS or LDAP authentication, see [Setting CLI User Authentication Types](https://docs.solace.com/Configuring-and-Managing-Routers/Setting-CLI-User-Authentication-Types.htm).

If the CLI user is successfully authenticated, the Solace router permits the connection with the access privileges configured for the CLI user account.

**Note:**If you are using RADIUS or LDAP user authentication, it is recommended that you configure a few internal user accounts that can be used to administer the router if the external RADIUS or LDAP servers are unreachable. However, do not duplicate internal user accounts on a RADIUS or LDAP server or vice versa, because this can cause confusion over where user account passwords reside. For example, if a user account password is changed on the RADIUS or LDAP server, and a duplicate account also exists locally on the router, internal authentication will be used and because the old password is expected, the user authentication will fail.

## File Transfer User Authentication

A file transfer user is authenticated through the Internal authentication type. The File Transfer user’s name and password are verified against the internal Solace router database. If the verification is successful, the router permits the connection.

**Note:**File Transfer user accounts do not have a permission level associated with them. All File Transfer users can read, write, and delete files from the file system available to File Transfer users.

## User Access To CLI Commands

Each command offered through the Solace CLI has a particular scope and access level requirement. Therefore, a CLI user can only use a command offered through the CLI if that user’s configured access levels are sufficient for the scope of the command.

### CLI User Access Levels

Access levels are assigned to each CLI user account to control what types of commands the user is allowed to execute. The following access levels are available:

* none—The CLI user cannot execute CLI commands except for a few that pertain to the log in and display preferences for the user’s own account.
* read-only—The CLI user can execute CLI commands to display operational information about the router but cannot change its configuration. A monitoring application would typically use a read-only access level.
* read-write—The CLI user can execute CLI commands to both display operational information about the router and perform most router configurations—including creating other CLI user accounts with access levels of none. A management user responsible for configuring the router or Message VPN services would commonly be granted a read-write access level.
* admin—The CLI user can execute all CLI commands on the router. This includes controlling router-wide authentication and authorization parameters and creating other admin users. (This access level can only be applied to global-scoped commands.)

### Command Scopes

CLI users can be configured to manage the entire router and/or a subset of Message VPNs. To permit this level of control, each CLI user account is assigned access levels for the following different scopes:

* global—a global access level dictates what the user is allowed to do across the entire router
* Message VPN—one or more Message VPN access levels dictate what the CLI user is allowed to do within Message VPNs

An assigned Global access level is all-encompassing in that it gives a CLI user the same level of access to all CLI commands, even those that have a Message VPN scope.

In addition to a global access level, Message VPN access levels are assigned so that users with none or read-only global access levels can be granted increased access Message VPN-scoped CLI commands, as required. An assigned Message VPN access level can only increase–not decrease–the Message VPN access level that a CLI user effectively receives from its assigned global access level.

Assigning both global-scoped and Message VPN-scoped access levels allows the appropriate access to be granted to users based on their administrative roles. Consider the two following examples:

* A system‑wide administrator could be assigned a global access level of admin, which allows that user to run any command on the router, whether it is a global-scoped or Message VPN-scoped command.
* A user who needs to monitor operational statistics within a Message VPN could be assigned a global access level of none and a Message VPN access level of read-only for that Message VPN.

**Note:**For security reasons, only a few administrators should be given access to CLI user accounts with global access levels of read-write or admin.  
In general, Message VPN-scoped CLI commands contain the “message-vpn” keyword, or they are contained in a CLI mode with the “message-vpn” keyword. For example, commands to configure client usernames, durable endpoints, and Distributed Caches are created on a per-Message VPN basis:  
solace(configure)# client-username <username> message-vpn <vpn-name>  
solace(configure/client-username)# acl-profile <name>  
...  
solace(configure)# message-spool message-vpn <vpn-name>  
solace(configure/message-spool)# create queue <name>  
...  
solace(configure)# message-vpn <vpn-name>  
solace(configure/message-vpn)# distributed-cache-management

#### Assigning Access Levels

Each created CLI User must be assigned:

* a single global access level
* one or more specific Message VPN access levels

Message VPN access levels can be assigned through:

* the default Message VPN access level—This provides a consistent Message VPN access level across all provisioned Message VPNs. A default Message VPN access level is always assigned to a CLI user, but it only affects a CLI user’s access to Message VPN-scoped CLI commands if it is greater than the user’s assigned global access level.
* Message VPN exceptions—These are per-Message VPN exceptions to the default Message VPN access level. These exceptions can either increase or decrease the assigned default Message VPN access level for the specified Message VPN.

### Creating and Managing CLI User Accounts

Your ability to create and delete user accounts, and to change the access levels and passwords of existing user accounts, depends on the global access level of the user account you use. The general limitations for each global access levels are as follows:

* A CLI user with a global access level of admin can create, delete, or make changes to other CLI user accounts without any restrictions.
* A CLI user account with a global access level of read-write can:
  + only create or delete other CLI user accounts with a global access level of none
  + set a Message VPN access level for any CLI user account
  + change the password for its own user account and user accounts with an access level of none
* A CLI user account with a global access level of read-only or none can only change the password of its own user account.

When access level and password changes are made to a CLI user account, they do not affect any active sessions that are using that user account. The changes only take effect the next time a user logs in to that CLI user account.

## Access Level Capabilities

There are multiple possible global access levels and Message VPN access levels that can be assigned to a CLI user with various capabilities and limitations.

**Note:**CLI users only see commands that they are authorized to execute. This means, for example, that Help commands will not show unauthorized commands, and CLI tab‑completion will not complete a command for which the user is not authorized. In addition, wildcard characters used in CLI commands will not allow a user to see or access network resources, such as Message VPNs or client profiles, that they are not authorized for.

### Global Access Levels

The table below provides information on the possible global access levels that can be assigned to a CLI user and many of the capabilities and limitations that those access levels present.

| Global Access Levels | |
| --- | --- |
| Level | Capabilities and Limitations... |
| none | The CLI user has access to minimal CLI commands, and no jail file system access. However, a CLI user with this global access level can:   * use ?, help, and tree commands (unauthorized commands are not shown) * use the **show authentication current-user** User EXEC command to view its own access level information * change its own password, if the user is authenticated through the internal database * navigate through command modes, as required, to execute authorized commands * logout of the current session * enable/disable alarm display * set the number of lines to use for paging output * enable/disable strict column wrapping * disconnect CLI sessions that belong to the current user (but not the current session)   **Note:**CLI users with a global access level of none can be given additional access to Message VPN commands by assigning them a default Message VPN access level of read-only or read-write or a Message VPN access level exception of read-only or read-write. |
| read-only  (The default value for the global access level.) | In addition to the capabilities offered by a global access level of none, a CLI user with a global access level of read-only can:   * use **show** User EXEC commands to view status and configuration information for the router and for Message VPNs * clear events (but not statistics) * access the jail file system   **Note:**CLI users with a global access level of read-only can be given additional capabilities at the Message VPN level by assigning them a default Message VPN access level or a Message VPN access level exception of read-write. |
| read-write | In addition to the capabilities offered by a global access level of read‑only, a CLI user with an access level of read-write can perform most configuration changes, including:   * configuring default Message VPN access levels * creating Message VPN access level exceptions * creating, deleting, or renaming internally-authenticated CLI user accounts with an access level of none * restarting the router from its current configuration file through the **reload** Privileged EXEC command |
| admin | The CLI user has full access to all global and Message VPN-scoped CLI commands (it provides an effective Message VPN access level of read-write).  Some configuration changes or actions that can only be performed with a global access level of admin include:   * Creating, deleting, or renaming internally-authenticated CLI user accounts with an access level greater than none. * Changing the global access level of any CLI user account. * Changing the authentication configuration of CLI users with a global access level greater than none. * Changing the authentication configuration of CLI users in LDAP groups. * Restarting the router through either the **boot**, **reload default-config**, or **reload config <config-file>** Privileged EXEC commands. * Making changes at the authentication CONFIG level of the CLI. (The only exception is the ability to make Message VPN-level changes when the CLI user is granted a sufficient default Message VPN access level.) * Changing the configuration database file that the router is currently running—changing to a different configuration database could result in a different authentication configuration. |

### VPN Access Levels

The following table provides information on the possible Message VPN access levels that can be assigned to a CLI user and many of the capabilities and limitations that those access levels present.

| VPN Access Levels | |
| --- | --- |
| Level | Capabilities and Limitations... |
| none  (The default value for the default Message VPN access level.) | The CLI user has no access to Message VPN-scoped CLI commands. |
| read-only | The CLI user can:   * use show commands to view status and configuration information for Message VPNs * clear Message VPN events (but not statistics) |
| read-write | In addition to the capabilities offered by a Message VPN access level of read-only, a CLI user with a Message VPN access level of read-write can perform most Message VPN scoped configuration changes.  Some of configuration changes or actions that are forbidden (and would require a higher global access level of read-write or admin) because they affect system resources include:   * configuring client profiles   Global system administrators are expected to create client profiles that define how clients within a Message VPN are expected to behave. However, a CLI user limited to Message VPN scope does have access to show client profiles and assign client profiles to client username objects.   * creating or deleting Message VPNs * configuring Message VPN parameters that affect system resources, such as:   + export-policy   + management-message-vpn   + max-connections   + max-subscriptions   + semp-over-msgbus   + max-egress-flows   + max-endpoints   + max-ingress-flows   + max-spool-usage   + max-transacted-sessions * configuring the following Message VPN parameters:   + Message VPN authentication   + Replication bridge configuration   + Replication state and Replication [no] shutdown |

# Open APIs & Protocols

Solace facilitates integration with applications, other middleware, and data movement technology with support for open APIs like JMS and OpenMAMA, and open wireline protocols such as AMQP, MQTT and REST. This allows you to get real-time data flowing between diverse applications, big data systems, cloud services and IoT devices.

**AMQP**

Solace VMRs support AMQP 1.0 as a component in Solaceʼs multi-protocol, any-to-any messaging solution based on open source and open standards.

**JMS**

Solace supports persistent and non-persistent JMS, version 1.1. Client applications connect to Solace like any other JMS broker so companies whose applications are struggling with performance or reliability issues can easily upgrade to Solace’s hardware or Virtual Message Router.

**MQTT**

Solace supports the OASIS MQTT 3.1.1 standard to meet the needs of connected devices and mobile applications that need an efficient way to send and receive information that requires very little client-side processing power, memory and/or bandwidth. The Solace message router works with any third-party MQTT 3.1.1 compliant client API, including open source APIs available via an Eclipse initiative called Paho.

**OpenMAMA**

The Open Middleware Agnostic Messaging API (OpenMAMA) is an open source, lightweight, vendor-neutral integration layer for sending and receiving market data information supported by the Linux Foundation. With unified support for many different kinds of data distribution in a compact, cost-effective platform, Solace’s platform is the ideal foundation for OpenMAMA-based market data distribution systems.

**REST**

The Solace REST Messaging API allows HTTP clients to send and receive messages with a Solace message router using HTTP POST requests. This enables REST clients to send messages to and receive messages from any Solace message router clients without needing to use any Solace-provided API.

# Web Messaging Concepts

This section is intended for system architects and developers to understand the architecture, features, and deployment of the Solace Messaging Platform when used for Web Messaging.

It assumes that the reader is familiar with messaging system concepts such as application messaging, topics, subscriptions, and message persistence (for example, as found in Java Message Service [JMS]).

# Web Messaging Overview

Solace Web messaging provides the ability for applications in browsers and mobile devices to receive real-time updates of information pushed to them asynchronously from server applications in a data center over a web infrastructure composed of HTTP proxies, load balancers, and firewalls. It also allows browser and mobile applications to perform request/reply and other message exchange patterns with datacenter applications.

Web Messaging is frequently used to provide a real-time, dynamically updated user display of information for applications such as:

* displaying foreign exchange currency spreads in a Single Dealer Platform
* displaying equity stock prices for retail equity brokerages
* updating bids for an online auction
* updating the real-time location of a moving object on a map
* online chat applications
* sending status alerts about a condition of interest

In these use cases, the real-time information is dynamically sent by a server application in the data center and received by all browser and mobile clients interested in, or subscribing to, this information no matter what the internet connectivity is between the server applications and browsers or mobile devices.

## What is Solace Web Messaging?

Solace Web Messaging is a data movement infrastructure composed of software Application Programming Interfaces (APIs) in various languages and the Solace messaging router. Together, they form an application-to-application messaging fabric that allows server-based applications in a datacenter to communicate with each other as well as with browser and mobile applications over a web infrastructure of proxies, firewalls and HTTP load balancers.

This infrastructure can be used for two main types of applications:

* To asynchronously stream real-time updates to internet client applications for applications such as financial trading, gaming, betting, auctions, taxi and bus location alerts.
* As an alternative to traditional web infrastructures to enable server, browser, or mobile applications to interact with each other using a single uniform messaging paradigm. This is as opposed to using messaging between server applications and web forms, JSPs and servlets between web components which in turn use messaging toward the server applications. This is quickly becoming an attractive architecture as browser and mobile applications become more demanding, their devices become more powerful, and there is an increasing need for device independence.

Support for rich messaging features, built-in high availability, robustness controls and the ability to connect 200,000 concurrent client sessions on a single router make this an ideal infrastructure for large-scale mission-critical web applications.

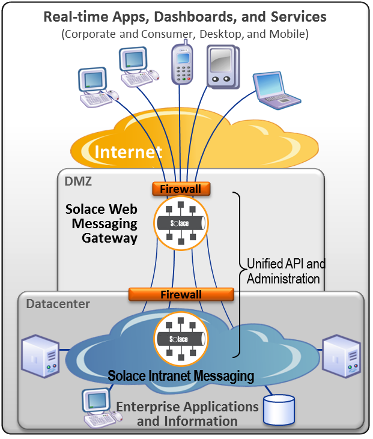
Browser and mobile applications use APIs in JavaScript/HTML5 running in any modern browser (such as Chrome, Safari, Internet Explorer, Firefox) to communicate through messaging directly with server-based applications in the datacenter. This is as opposed to using XHR to send HTTP commands to a web server, which then relays them on to server applications.

A key advantage of Solace Web Messaging over standard HTTP is that it allows server based applications in a datacenter to asynchronously send real-time updates to browser and mobile applications over a web infrastructure. Web transport techniques (known as Comet, Long Polling, HTTP Streaming, HTTP Push, and WebSockets) are used by the Solace router and Web messaging APIs to provide asynchronous updates to HTTP client applications.

However, Solace Web Messaging provides far more than this. It provides a complete topic-based, publish/subscribe messaging fabric with ubiquitous reach between browser/mobile applications and server applications without needing any other infrastructure or bridging software. No web streamers, no web servers, no gateways to integrate with other messaging systems.

Web Messaging can be used to provide connectivity to browser and mobile apps both within an enterprise, as well as over the public internet.

As shown in the following figure, Solace Web Messaging both in the DMZ toward the internet combined with Messaging Oriented Middleware (MOM) services provided by Solace routers inside the datacenter support this ubiquitous application messaging fabric.



With Solace Web Messaging, both the server and web applications benefit from a functionally rich messaging API and message system capabilities that can be used by applications whether inside the data center or over the internet. The Solace architecture allows application designers to seamlessly extend the reach of their server-based applications not only to desktop users within their intranet, but also to customers, employees and partners over the internet through desktop and mobile applications. Extending an integrated messaging paradigm all the way to browser and mobile applications with a single infrastructure allows architects and developers to focus on their business problem rather than the plumbing of application communication.

## Programming Model with Solace Web Messaging

Solace Web Messaging can be used by mobile and browser applications instead of typical XHR GET/POST operations to perform all interactions with back end server applications – and for message exchange patterns not supported by HTTP.

Browser, mobile, and server based applications all use Solace messaging APIs in the language of their choice to first establish a session with a Solace router. As part of establishing a session, the client application authenticates to the message bus using username/password which the Solace appliance validates against either its internal or an external identity management system.

Once a session is established, publishing applications can send messages addressed to “topics”, and consuming applications can subscribe to topics to receive the messages of interest without connecting directly to the publishing application or knowing anything about the publishing application. As such, once the web application in JavaScript is downloaded to a browser, all interactions with back end systems can be done using the Solace messaging API. Whether to receive real-time updates or to perform request/reply interactions with back end services, thereby replacing the typical HTTP GET/POST methods to web servers, Solace Web Messaging can be used to provide uniform access to any service from any channel/device.

For example, topic-based messaging is what is used by the Java Message Service (JMS) specification, and is widely used by Message Oriented Middleware (MOM) products, including those offered by Solace. This topic-based messaging paradigm is more powerful than the semantics of HTTP as it can be used for:

* Publish/subscribe (that is, one-to-many message exchange patterns) where the topic of a message indicates the type of data being carried, such as a bid or ask for a particular equity symbol or a foreign exchange currency pair. This information is asynchronously published in real time by a back end application within the data center. It is typical for such a message to be delivered to multiple consuming applications either inside the enterprise or over the internet. These updates allow Graphical User Interfaces (GUIs) to provide a responsive, real-time, dynamically updating display of relevant information.
* Point-to-point request/reply interactions where one topic indicates a well-known “service” that is to handle the request, and another topic uniquely identifies the requestor to whom the reply is to be sent. These applications can then communicate “point-to-point” without needing to establish a connection to each other. This message exchange pattern can be used, for example, to query the status of a trade, or retrieve a stock price chart, or verify the status of a payment or an auction bid.

In typical web applications, point-to-point request/reply interactions are accomplished using normal HTTP GET/POST commands sent from the browser into the web server tier, which could then either:

* do a database lookup directly, or
* (more typically) translate this request into some internal format that is then sent toward the back-end application tier through a messaging system. The messaging system reply is correlated with the original HTTP GET/POST request by the web server and sent back to the browser.

In the Web Messaging model, the heavy processing and integration in the web tier is removed in favor of intelligent front-end applications that know how to send messages on well-known topics, which are then routed in a loosely coupled manner by the Solace messaging system to applications in the back end. With the advent of web applications that are more intelligent and feature rich than static HTML, using a messaging-based application architecture is much more efficient than using web servers to perform data conversion for all the same reasons that messaging has long been used by thick client front ends.

However, dynamically updating information is much more difficult for traditional web applications to support due to the client-driven request/reply nature of the HTTP protocol. This type of asynchronous, streaming of real-time messages is a key problem solved by Solace Web Messaging.

In practice, browser-based Web Messaging applications typically use a combination of the following techniques:

* HTTP GET operations to a traditional web server to download the browser application (for example, JavaScript), as well as any static data, such as forms and images.
* Web Messaging to subscribe to and receive asynchronous updates from server applications.
* Web Messaging to submit requests for data or processing to back end server applications. For example, to buy or sell securities, bid at an auction, or download a graph that will be dynamically updated.

Web Messaging is also well suited for mobile applications wishing to receive real-time updates where the more traditional HTTP/HTML/XML environment may not be as convenient for request/reply interactions.

In addition to pure communication and connectivity, Solace Web Messaging provides many features typically required by real-time streaming applications such as:

* data access security controls
* per-client rate limiting
* prioritized message delivery
* runtime-independent and self-describing data types for cross-platform interoperability

### Publish/Subscribe Message Exchange Pattern

One capability that Web Messaging provides that cannot be accomplished with standard XmlHttpRequest() is real-time streaming to a browser – also known as HTTP Push. This type of 1-to-many message exchange pattern is typical of publish/subscribe messaging systems and is often used in stock or FX front ends as well as in chats, online betting and auctions.

With Web Messaging, browser and mobile applications use a Solace API to dynamically subscribe to (and unsubscribe from) the hierarchical topics of their choice, even using wildcard subscriptions, to allow messages published on only those topics of interest to be asynchronously delivered to the application. These applications do not need to worry about the transport mode, about message delineation and marshalling and subscription management – this is all done by the Web Messaging infrastructure.

### Request/Reply Message Exchange Pattern

In addition to asynchronous publish/subscribe messaging, Solace Web Messaging can be used to perform request/reply message exchanges to get information from or send information to server applications similar to the way XHR is used to send HTTP GET and POST commands. For example, an application could send a message to retrieve the current state of the user’s blotter or portfolio – similar to sending an HTTP GET – with the blotter or portfolio information returned in the reply. Similarly, an application could send an order message to an Order Management System (OMS) application to buy or sell a stock and receive the status in a response – similar to sending an HTTP POST. In addition, if the order does not fill immediately, then Web Messaging can be used by the OMS to send real-time updates to the issuing client application as the order changes state with partial fills, for example.

With a Web Messaging architecture, there are no web servers or web server applications that need to be implemented, deployed and managed other than to serve static content. All application communication uses only a Solace messaging fabric as infrastructure.

Applications address request messages to a well-known topic which identifies the service of interest rather than to a well-known URL which must then be translated by a web server to an internal application name or bridged to an internal messaging system. Messages are sent using the Solace sendRequest() API method rather than using XHR. The reply, along with the associated context for the original request, is returned in a callback or delegate function provided by the application.

Applications can have several requests outstanding at any given time and the responses come in without any ordering dependencies, unlike some cases with HTTP.

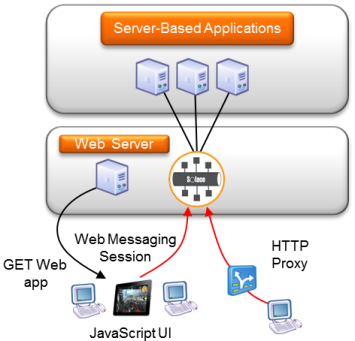
Each request message is sent with a ReplyTo field which identifies who the reply is to be sent back to by the server application handling the request. This ReplyTo value, in the form of a “#P2P/…” topic, is generated automatically by the Solace infrastructure and uniquely identifies this particular API session, making it simple for applications to send requests and receive the associated reply. This default ReplyTo value can also be overridden by the application.

# Web Messaging Architectures

This section describes the architecture of typical Solace Web Messaging systems.

## Internal Web Application Deployment Architecture

In the following figure, server applications within the data center communicate with each other through an internal messaging system—the Solace messaging infrastructure—in the normal distributed systems manner. These applications can be written in Java, .NET, C++, or other convenient languages, and the messaging system provides services such as pub/sub, request/reply, and various messaging qualities of service such as persistent and non-persistent messaging.



Desktop users can use traditional thick GUI in .NET or Java to interact with the back end server applications to request information, perform transactions, and receive real-time streaming updates.

With Web Messaging, mobile and browser-based applications can also connect directly to this same messaging platform without requiring an additional web fan-out tier between the browser/mobile and the internal MOM. This allows organizations to take advantage of the many benefits of web UI technologies, such as zero-footprint application install and ubiquitous deployment, within their organization using a clean and simple architecture.

The user accesses the web application by clicking a link or entering a URL on their browser or starting a mobile application. For browser-based applications, the browser performs HTTP GET operations to download the web application (for example, JavaScript). The web application and any static content are served from a standard web server and not by the Solace router to allow maximum flexibility in managing and deploying applications.

The web application in the browser or mobile device uses the JavaScript API to connect and authenticate to the Solace router, and all real-time communication with the server applications goes through the Solace router not the web server.

The JavaScript API automatically uses the most efficient communication transport allowed by both internet intermediaries and the browser/mobile without needing any intervention from the application. See [Web Transport Protocols](https://docs.solace.com/Features/Web-Messaging-Concepts/Web-Messaging-Architectures.htm#Web-Transport-Protocols) more information on web transports.

Since Solace Web Messaging supports Cross Origin Resource Sharing (CORS) and Adobe’s cross-domain policy, the web server that downloads the application need not share the same DNS name as the Solace router, and no reverse proxy is needed between the browser/mobile and the router. See [Cross Origin Resource Sharing – CORS](https://docs.solace.com/Features/Web-Messaging-Concepts/Web-Messaging-Architectures.htm#CORS) for more information on CORS and Same Origin Policy.

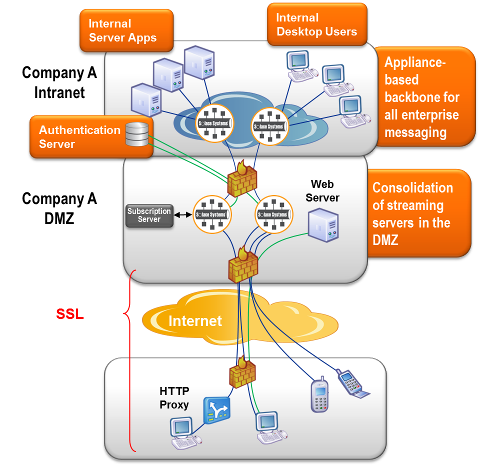
Once authenticated, the web application uses its JavaScript API to communicate directly with the Solace router to register topic subscriptions and receive messages published on those topics from any publisher.

The same JavaScript is used to send a request to back end services and receive the reply to this request.

## Internet-Based Web Application Deployment Architecture

This section describes how internet-based browser and mobile users can access internal applications.

As shown in the following figure, external browser users download web applications over the internet by clicking a link or typing in a URL. The web application is dynamically served by a web server typically within the Demilitarized Zone (DMZ) of Company A. These web applications use a Solace Web messaging API to connect to a Solace router inside Company A, and subscribe to topics of their choice to receive streaming updates or engage in a request/reply interaction with some back end application in the data center of Company A. This real-time streaming connection may need to pass through various internet intermediaries such as HTTP proxies, firewalls, and HTTP load balancers. This is all handled by the Solace JavaScript API without intervention by the application developer.



Although not technically required, Solace routers providing messaging services to external/internet users are typically deployed within the DMZ as a result of enterprise security practices. The DMZ appliances are then configured to be peered with the intranet Solace routers so that communication can be both automatically and securely performed between server-based applications and external web applications.

Multiple Solace routers are typically deployed within the DMZ for fault tolerance and load balancing purposes. [Session Load Balancing to DMZ Appliances](https://docs.solace.com/Features/Web-Messaging-Concepts/Web-Messaging-Architectures.htm#Session-Load-Balancing) describes how to load balance sessions to Solace routers and [Network-Level DMZ Connectivity Architecture](https://docs.solace.com/Features/Web-Messaging-Concepts/Network-Level-DMZ-Connectivity.htm) explains network architectures used in the DMZ.

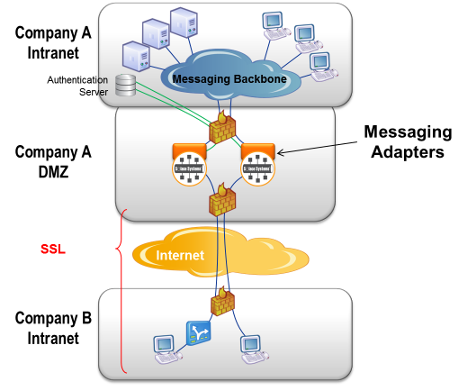
The topic routing protocols that run between the routers allow automatic discovery and dynamic propagation of client topic subscriptions for both publish/subscribe and peer-to-peer message exchanges. Thus, the only messages flowing from the intranet routers to the DMZ routers are those for which there is at least one interested internet user, which reduces total bandwidth requirements and load.

Messages are fanned out to internet users by the DMZ routers based on subscription interest. Similarly, messages from internet users can be sent to internal applications (typically a request/reply message exchange pattern) and these messages are routed (and perhaps load balanced) by the Solace routers to the correct server application based on the topic in the message, with the reply routed back to the requesting internet user.

If transport security is required by the web application, Secure Sockets Layer (SSL) encryption is performed by the Web messaging API using the facilities of the browser or mobile operating system. This SSL connection is typically terminated at the external firewall of Company A. This allows all internet security (for example, certificate management, Denial of Service (DoS) attack prevention, access controls) to be managed from a single point in the infrastructure (because messaging is typically just one of many internet services).

## Using Solace Only for Web Messaging

The following figure shows how Solace Web Messaging can be used with an existing non-Solace intranet messaging system.



In this case, a software messaging adaptor or bridge must be implemented that uses the messaging API of the internal messaging system on one side, and a Solace messaging API on the other side, to bridge communications between the two messaging systems. Web applications communicate with the Solace DMZ routers using Solace JavaScript APIs. The messaging adaptor transforms messages to and from the internal messaging system to allow bi-directional communication.

## Cross Origin Resource Sharing – CORS

Cross Origin Resource Sharing (CORS - <http://en.wikipedia.org/wiki/Cross-origin_resource_sharing>) is a web browser technology specification that defines ways for a web server or a device like the Solace router to allow its resources to be accessed by a web page that was served from a different domain/web site. It provides a way to create a web application that uses the resources of multiple web sites in a Web2.0 manner.

The Solace router supports CORS for JavaScript. CORS support on the Solace router allows a web application retrieved from a web server using one URL to access the Solace router using a completely different and unrelated URL without requiring an intermediate reverse HTTP proxy, as shown in the figure in [Internal Web Application Deployment Architecture](https://docs.solace.com/Features/Web-Messaging-Concepts/Web-Messaging-Architectures.htm#Internal).

Without CORS support, browsers adhere to the Same Origin Policy (<http://en.wikipedia.org/wiki/Same_origin_policy>) which prevents a web application served from one (web) site from accessing resources of a different (web) site due to security concerns.

## Web Transport Protocols

Raw TCP cannot be used from browsers and cannot pass through HTTP proxies and other internet intermediaries. HTTP must be used in these situations. Before the advent of WebSockets, the HTTP protocol and its support in browsers and internet intermediaries did not natively support asynchronous, bidirectional communication between a web application and a web server, as is required by messaging applications. HTTP only supports client initiated request/reply. Various techniques need to be applied to be able to support asynchronous bidirectional communication over an HTTP transport. These techniques vary based on the client runtime and feature support in intermediaries. Solace Web Messaging handles this automatically for applications so application developers can be free from needing to worry about such transport complexities and can instead focus on their business application.

Solace Web Messaging automatically determines the most efficient web transport protocol to use to carry messages to/from a browser or mobile application when the session is connected by the application. The most efficient transport protocol used in any particular situation depends on a combination of the runtime container (browser type and version, mobile framework, mobile OS) and any internet intermediaries (proxies, load balancers, firewalls). Internet Explorer 8 and 9, for example, do not support the WebSocket protocol for JavaScript applications so only a Comet/Long Polling transport can be used. Chrome, Safari, and other modern browsers allow the Solace API to use WebSockets to provide a more efficient transport from the Solace router to the browser/mobile.

The “best” transport to use is automatically determined by the JavaScript API without the application needing to be involved, although the application can choose to influence this using the transportProtocol session property. Applications can determine which transport protocol is currently in use by a session by querying the transportProtocolInUse session property. Similarly, the protocol in use for a given connection can be viewed from the Solace router using the show client <name> web-transport CLI command.

The web transports supported by Solace Web Messaging are listed below in order of most efficient to least efficient. In all cases, the transport channel can be SSL encrypted or clear text as directed by the application when the session is established.

* WebSockets – WebSockets is a wireline protocol defined in IETF [RFC 6455](http://tools.ietf.org/html/rfc6455) as well as a browser API defined by [W3C](http://www.w3.org/TR/2011/WD-websockets-20110419/). Applications using Solace web APIs do not use the WebSocket API, rather the Solace API uses the browser’s WebSocket API (when available) to produce the WebSocket wireline protocol interactions with the Solace router and applications use a Solace Web Messaging API. WebSockets is the most efficient and lowest latency web transport.
* HTTP Streaming – HTTP Streaming is a form of [long polling](http://en.wikipedia.org/wiki/Push_technology#Long_polling) where HTTP Chunked Transfer Coding (see [RFC 2616](http://tools.ietf.org/html/rfc2616)) is used by the Solace router to send a stream of Solace messages from the router to the web client without needing to terminate the HTTP response. It is therefore more efficient than Comet for streaming data to web clients.
* HTTP Comet Binary – HTTP Comet is a simpler form of long polling, which is used when a more efficient transport is not available. In this mode, multiple Solace messages can be packed in a single HTTP interaction however the HTTP reply from the router is not long running as with HTTP Streaming – it is terminated with each group of messages sent to the web client and a new HTTP long poll is applied by the API. In the Binary mode, the payload remains as binary data for increased efficiency.
* HTTP Comet Base64 – This mode of Comet is the same as HTTP Comet Binary except that the HTTP payload is Base64 encoded due to the fact that some older browsers do not support binary HTTP payloads.

Support for the different transport modes across Solace messaging APIs is shown in the following table.

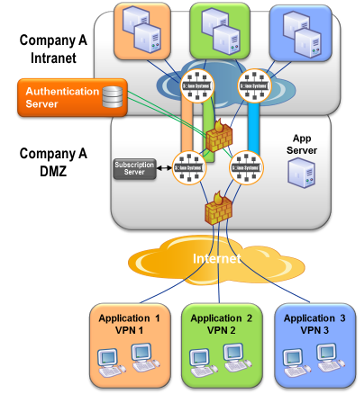
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | **Web Transport** | | | |
| WebSockets | HTTP Streaming | HTTP Comet Binary | HTTP Comet Base64 |
| **Web API** | JavaScript | checkmark-16 | checkmark-16 | checkmark-16 | checkmark-16 |
| iOS | checkmark-16 | checkmark-16 | checkmark-16 |  |
| .NET | checkmark-16 | checkmark-16 | checkmark-16 |  |
| C | checkmark-16 | checkmark-16 | checkmark-16 |  |

## Multiple Applications Sharing the Same Infrastructure

In some cases, especially for private cloud infrastructures, there are several distinct applications that require web messaging. In this case, as shown in the following figure, service can be provided to these multiple applications on the same shared Solace infrastructure using a feature called Message Virtual Private Networks (VPNs).

Message VPNs provide a method of virtualizing a Solace router so that multiple application groups can share the same physical router while still being isolated from each other, like Virtual Local Area Networks (VLANs) on an Ethernet switch.

The following figure shows an example of three applications with server applications and web clients sharing the same DMZ and internal messaging infrastructure. Each application is deployed within its own Solace Message VPN which spans both the DMZ routers and the internal routers as a result of the Solace routing protocols.



Back end server applications connect into their own Message VPN and offer streaming data or request/reply services using topics that are only accessible by applications in this same Message VPN.

When web applications establish a session with a Solace router in the DMZ, they provide their Message VPN name along with their username and password. This locks them into a virtual partition of the router specifically for their application, allowing them to communicate only with the back end processes that are providing service for this application but no others.

## Session Load Balancing to DMZ Appliances

The format of URLs used for Web Messaging is completely determined by the system architect.

When Web Messaging is deployed for access by browser or mobile applications within an enterprise, as in [Internal Web Application Deployment Architecture](https://docs.solace.com/Features/Web-Messaging-Concepts/Web-Messaging-Architectures.htm#Internal), there are no additional considerations.

When Web Messaging is deployed for internet access, as in [Internet-Based Web Application Deployment Architecture](https://docs.solace.com/Features/Web-Messaging-Concepts/Web-Messaging-Architectures.htm#Internet), information about these URLs must be configured on both the Solace router and the HTTP load balancer to allow the system to

* load balance new Web Messaging sessions among the several Solace routers in the DMZ, and
* ensure that HTTP requests from established messaging sessions are routed in a “sticky” manner to the same Solace router that received the session initiation request.

The Solace router is configured via the web-url-suffix command. Example configuration for an F5 Big IP firewall is provided in [Using F5 Big-IP as a Load Balancer for External Internet Connectivity](https://docs.solace.com/Features/Web-Messaging-Concepts/Using-F5-Big-IP-as-a-Load.htm).

The URLs to consider and associated rules are:

1. URL used to get the web application

Format: https://<hostname>/<download-my-app-from-server>

This URL is the one used by the user to “GET” the browser application. Because the Solace router does not serve static content to browsers, this URL must be served by a separate web server as shown in the figure of [Internal Web Application Deployment Architecture](https://docs.solace.com/Features/Web-Messaging-Concepts/Web-Messaging-Architectures.htm#Internal). This URL does not apply for non-browser applications. This URL is visible to the application user.  
For example: https://my-company.com/download-my-sdp-application

1. URL used to initiate the messaging session  
   Format: https://<hostname>/<application>   
   This URL is not seen by the application user, but is provided by your web application to theJavaScript API through the url session property. This URL identifies a set of possible Solace routers to connect to.   
   Configuration of the external load balancer for this URL must allow incoming sessions to be load balanced among the IP interfaces of applicable DMZ Solace routers for that application. For example, referring to the figure in [Network-Level DMZ Connectivity Architecture](https://docs.solace.com/Features/Web-Messaging-Concepts/Network-Level-DMZ-Connectivity.htm), to load balance sessions for <application1> among both Solace routers in the DMZ, this URL needs to be mapped with a load balanced distribution method to all of (IP-2 to IP-5) and (IP-7 to IP-10).   
   For example: https://my-company.com/my-sdp-application
2. URL used after messaging session initiation

Format: https://<hostname>/<application><web-url-suffix>   
This URL is not seen by the application user or your application. It is used by the JavaScript API to send subsequent HTTP requests to the specific Solace router that received its session initiation request. This provides a “sticky” communication between this API session and the particular router.   
The “web-url-suffix” value is configured on each DMZ router using the “web-url-suffix” command within the “web-transport” configuration. Cookies and other such mechanisms are not used to provide session stickiness since these can be disabled in browsers.   
The web-url-suffix configuration rules are:

* 1. The “web-url-suffix” value must uniquely identify a particular Solace router among all other DMZ appliances (for example, the hostname of the appliance) to effect session affinity.
  2. The suffix can be any valid URL suffix syntax, but for convenience and readability is often in the form of a URL parameter (for example, “?appliance=Prod-DMZ-123”).
  3. As part of the session initiation process, this suffix value is returned by the Solace router to the JavaScript API and then used by the API in subsequent HTTP requests for this API session. No application involvement is required.
  4. This URL must be configured on the external firewall to map to only the IP addresses of this particular Solace router, but it can map to any or all of the externally-facing IP interfaces. For example, referring to the figure in [Network-Level DMZ Connectivity Architecture](https://docs.solace.com/Features/Web-Messaging-Concepts/Network-Level-DMZ-Connectivity.htm), the “web-url-suffix” for the left-most appliance could map to all of IP2 through IP5.

For example: https://my-company.com/my-sdp-application?appliance=Prod-DMZ-158

In summary, the “web-url-suffix” is:

* determined by the system architect to uniquely identify a Solace router in the DMZ
* configured on the Solace router through the “web-url-suffix” command
* configured on the external firewall along with the application URL root to map to the IP addresses of the specific Solace router
* completely invisible to the application

# Useful Solace Messaging Platform Features

This section describes the following key messaging features from the Solace Messaging Platform that are useful for Web Messaging use cases. This is not a complete list of features from the Solace Messaging Platform, nor is it a detailed presentation of each feature. The intent is to introduce the reader to key messaging features that may be of interest for Web Messaging applications, and then refer to the appropriate document for further information.

## Solace Client APIs

The Solace provides Java, C, .Net, and JMS messaging Application Programming Interfaces (APIs) for developing server-based applications, as well as source code for various sample applications to get you going quickly and easily. They provide all the information developers need to write and test server applications.

Solace also provides a JavaScript Web messaging API. This API follows the same structure as the server-based APIs to allow application designers to be very proficient regardless of the language and development environment they use.

## Multi-Node Routing

Solace routers support the ability to peer together to dynamically exchange topology and subscription information. This feature can be used to link together routers in the DMZ with those inside the intranet to:

* allow JavaScript API to communicate with back-end server applications to receive streaming information or perform request/reply interactions
* link DMZ routers to help reduce the flow of streaming data over the internal firewall of the DMZ

The internal routers could be in the same data center, or could be in remote data centers if required.

Subscriptions from Web messaging API clients are aggregated by the routers in the DMZ to reduce the subscription and traffic load on the internal messaging system, thereby ensuring a highly scalable solution.

## Client Authentication

Solace routers in the DMZ support the Lightweight Directory Access Protocol (LDAP) for authentication using the username and password model from Web Messaging applications. Thus, a centralized authentication repository can be used to perform all external user authentication.

This mechanism can also be used to support Single Sign On (SSO) authentication models whereby the Web Messaging application retrieves the SSO token from the SSO server and uses this token in the password field to connect to the DMZ Solace router. The Solace router uses LDAP to transfer the username/password==SSO Token back to the SSO authority to allow or disallow the login.

## Data Access Controls

Per-user, per-topic customer access control rules for both publish and subscribe operations can be enforced by the Solace router by:

* using Access Control List (ACL) profiles
* implementing custom access control rules using the “On Behalf Of” subscription capability

ACL profiles are rules provisioned on the Solace routers. They are used to provide coarse-grained access control to topics based on the hierarchical nature of the topic structure. ACL Profiles contain the list of topics that a client is or is not allowed to publish to, and the list of topics that a client is or is not allowed to subscribe to. These rules are enforced by the router in real time as applications dynamically publish and subscribe. A given ACL Profile is typically referenced by multiple clients, so the rules are configured once for a “class” of user.

The “on behalf of” subscription feature allows a Subscription Manager application you write to:

1. Receive all subscription requests from a set of clients.
2. Then apply your own business rules to allow or disallow the subscription.
3. Then add the subscription for the data “on behalf of” the client.

This feature is very useful where:

* tiered information is available, as in FX Single Dealer Platforms, and the client needs to receive the right “tier” of streaming data based on who the client is
* the number of rules and clients is very large
* the access control rules are not static

For example, in a foreign exchange Single Dealer Platform application, a client may want to receive streaming USD/CAD spread quotes, but he needs to receive the quote stream that is determined by his business tier. In this case, the Subscription Manager application, upon validating access to USD/CAD quotes, can add a topic subscription for the topic as appropriate for the business tier level of the client.

For an application example of how to use the “subscription on behalf of” feature, refer to the sample applications in the applicable server-side Solace messaging API.

## Limiting Client Message Stream Rates

In many applications where streaming quotes are to be delivered to end users, it is desirable to reduce the rate of updates to the end user to be less than the rate produced by the source—especially if the end user is human or reachable over a WAN or the internet. This is typical for both equity and foreign exchange market data.

Solace supports a feature called message eliding which allows a maximum per-topic message rate to be configured individually for each end user application. In this case, sources of market data can publish at the message rate they like, with some (perhaps internal) applications receiving the full unlimited message rate, while other applications (internal or external) can have this same message stream limited to the rate they desire—all without additional infrastructure—to ensure that these applications always receive only the most recent updates, rather than queued or outdated information.

## Last Value Caching

SolCache is a software product from Solace that provides message caching and retrieval services. It caches up to a configured number of messages deep per topic for configured topics and makes them available to requesting clients. SolCache provides various plug-ins to allow customer-specific message handling – such as merging delta updates from a streaming publisher into a larger “initial image” stored in SolCache to create a complete book image from delta updates, for example.

SolCache is often used in applications such as price and odds distribution for client applications to acquire the latest value of some stock or currency pair without needing to wait for the next update message.

Querying SolCache can be performed by some applications directly from the JavaScript API by web client applications, thereby reducing the amount of infrastructure that application developers need to implement.

## Message Priority

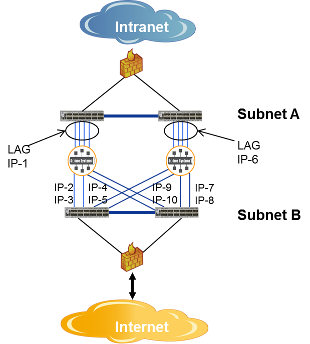
It is sometimes desirable to deliver high priority messages ahead of lower priority messages when there is congestion and queueing occurs. Such queueing can occur, for example, toward an internet application due to the lack of bandwidth or slow processing by the application.

For these reasons, the Solace Messaging System allows publishers to assign each message a priority which then causes higher priority messages to be delivered ahead of lower priority messages to end applications during congestion.

# Network-Level DMZ Connectivity Architecture

This section provides information for network engineering teams responsible for the deployment of Solace appliances in the DMZ to provide high-availability connectivity into the underlying Layer 2 switch infrastructure.

If there is no requirement for physical separation between internally-facing subnets and externally-facing subnets within the DMZ, then the typical Solace network connectivity options can be used, that is, a Link Aggregation Group (LAG) for 1GE connectivity or port bonding for 10GE connectivity.



In cases where physical separation between subnets within the DMZ is required, more care must be taken. This figure shows a typical deployment using an 8x1GE NAB to provide redundancy on both the intranet-facing (that is, internal) and internet-facing (that is, external) sides. In this case, a LAG is used to provide port and link redundancy facing the intranet, while individual IP interfaces (one per port) are used facing the internet. Typically, the external firewall (for example, BigIP appliance from F5) monitors the availability of each internet-facing IP interface on the Solace appliance, and then load balances incoming HTTP requests to available IP interfaces based on URLs (as discussed in [Session Load Balancing to DMZ Appliances](https://docs.solace.com/Features/Web-Messaging-Concepts/Web-Messaging-Architectures.htm#Session-Load-Balancing)).

In deployments using a 2x10GE NAB, one port is connected to the subnet facing the intranet, while the other port is connected to the subnet facing the internet. Thus, there is no port or link redundancy.

Solace appliance active/active redundancy is typically not used with appliances in the DMZ since the URL routing rules of the external firewall, combined with the ability of the firewall to monitor the status of Solace interfaces, allows for an N:1 resiliency model, whereby the load from one failed appliance can either be switched to another spare or standby appliance, or spread among existing appliances, thereby providing more cost-effectiveness with high availability.

**Using F5 Big-IP as a Load Balancer for External Internet Connectivity**

This section describes how to configure an N+1 redundancy configuration using the F5 Big-IP appliance as a load balancer for external internet connectivity through Solace Web messaging, as follows:

1. Create a pool for web servers and add the internet-facing web server interface addresses to it.
2. Create a pool for Solace routers and add the Internet-facing interface addresses of the Solace routers to it.
3. Create a virtual server. This is the server clients connect to.
4. Configure the F5 Big-IP appliance as a load balancer to route requests to the Web servers or Solace routers based on tags in the request URLs. This is done by creating an F5 iRule and associating it with the virtual server.

For example, consider the network shown in the figure below with two web servers, two Solace appliances, and one F5 Big-IP appliance acting as a load balancer. In this network the following is true:

* Each of the Solace appliances is configured with a web-url-suffix matching its name. For example, the CLI configuration performed on Solace1 is:

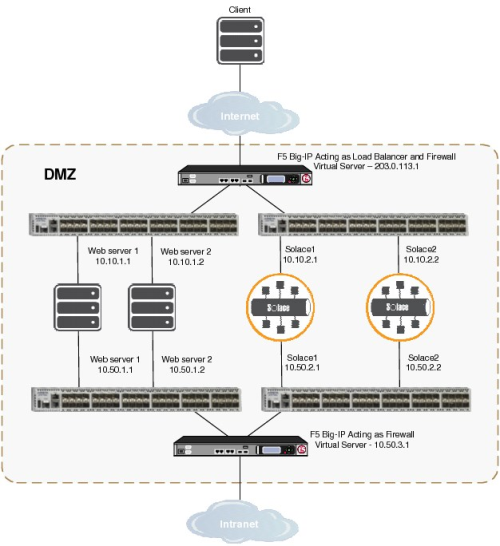
Solace1# configure

Solace1(config)# service web-transport

Solace1(config-service-web)# web-url-suffix ?appliance=Solace1

* On the F5 Big-IP appliance acting as a load balancer, the web servers are added to a pool called web\_server\_pool while the Solace appliances are added to a pool called solace\_appliance\_pool.
* The URL specified in the session properties by the application to the Solace Web messaging API contains the string https://203.0.113.1/FxPortal.

Figure 1 - N+1 Redundancy Configuration for External Internet Connectivity



Given the above, the F5 iRule associated with the virtual server configured on the F5 Big-IP appliance acting as load balancer is:

when HTTP\_REQUEST {

  if { [HTTP::path] starts\_with "/FxPortal" } {

    if { [HTTP::uri] ends\_with "?appliance=Solace1" } {

      node 10.10.2.1 80

    } elseif { [HTTP::uri] ends\_with "?appliance=Solace2" } {

      node 10.10.2.2 80

    } else {

      pool solace\_appliance\_pool

    }

  } else {

    pool web\_server\_pool

  }

}

The client connects to the virtual server/load balancer. If, for example, the user types the following address into their browser address bar: https://203.0.113.1/

The F5 Big-IP appliance routes the above request to one of the web servers, which returns the application to the client’s browser. The application code specifies a session URL as a session property to the Solace Web messaging API to connect to the Solace appliance. For example, it uses a parameter such as:

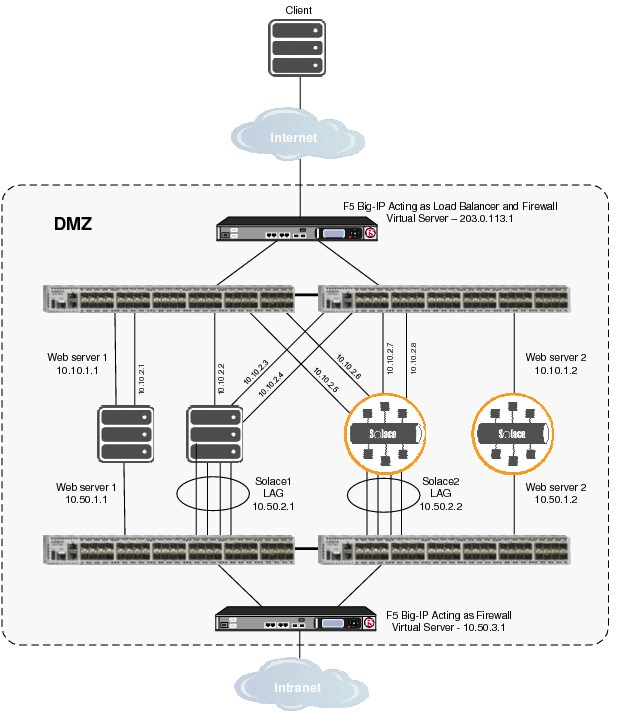
https://203.0.113.1/FxPortal

The F5 Big-IP appliance routes this request to one of the Solace appliances in the pool in a load-balanced manner. The handshake between the Solace appliance and the Solace Web messaging API includes the web-url-suffix configured on the Solace appliance. Subsequent requests from the application include the web-url-suffix in the request URL, which is added by the Solace Web messaging API, thereby allowing the F5 Big-IP appliance to route the requests to the same Solace appliance handling the session. For example, if an application was originally load balanced to the appliance named Solace1, subsequent requests would look like:

https://203.0.113.1/FxPortal?appliance=Solace1

For a more advanced example, consider the network shown in the figure below.

Figure 2 - Advanced N+1 Redundancy Configuration for External Internet Connectivity



The configuration in Figure 2 is created as follows:

1. Configure Solace1 with the web-url-suffix “?appliance=Solace1”.
2. Configure Solace2 with the web-url-suffix “?appliance=Solace2”.
3. On the F5 Big-IP appliance acting as a load balancer, add:
   * the web servers to a pool called “web\_server\_pool”.
   * all eight of the internet-facing Solace appliance interfaces to a pool called “solace\_appliance\_pool”.
   * the four Iinternet-facing Solace1 appliance interfaces to a pool called “solace\_appliance1\_pool”.
   * the four internet-facing Solace2 appliance interfaces to a pool called “solace\_appliance2\_pool”.
4. Create the virtual server.

Given the above, the F5 iRule associated with the virtual server configured on the F5 Big-IP appliance acting as load balancer is:

when HTTP\_REQUEST {

  if { [HTTP::path] starts\_with "/FxPortal" } {

    if { [HTTP::uri] ends\_with "?appliance=Solace1" } {

      pool solace\_appliance1\_pool

    } else if { [HTTP::uri] ends\_with "?appliance=Solace2" } {

      pool solace\_appliance2\_pool

    } else {

      pool solace\_appliance\_pool

    }

  } else {

    pool web\_server\_pool

  }

}

Based on the IP addresses in Figure 1, the pools are:

|  |  |
| --- | --- |
| web-server\_pool: | solace\_appliance\_pool: |
| 10.10.1.1 | 10.10.2.1 |
| 10.10.1.1 | 10.10.2.2 |
|  | 10.10.2.3 |
|  | 10.10.2.4 |
|  | 10.10.2.5 |
|  | 10.10.2.6 |
|  | 10.10.2.7 |
|  | 10.10.2.8 |
|  |  |
| solace\_appliance1\_pool: | solace\_appliance2\_pool: |
| 10.10.2.1 | 10.10.2.5 |
| 10.10.2.2 | 10.10.2.6 |
| 10.10.2.3 | 10.10.2.7 |
| 10.10.2.4 | 10.10.2.8 |

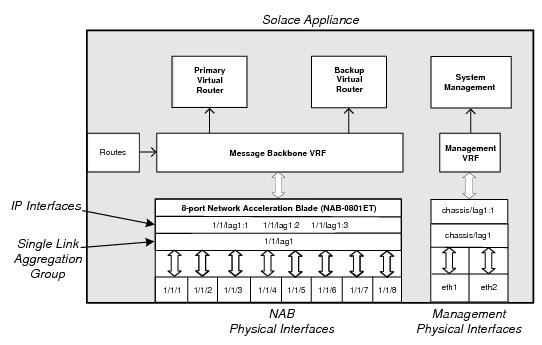
# Working with Appliance Interfaces

A Solace appliance relies on several interfaces and internal routing components to enable the transfer of data between the appliance either management applications that connect to the appliance’s management interfaces or client applications that connect to the appliance's Network Acceleration Blade (NAB) interfaces. These components include:

* management and network physical interfaces, which can be:
  + configured and used directly as independent physical interfaces;
  + grouped into virtual interfaces, using link aggregation groups (LAGs);
  + or used in a combination of independent physical interfaces and LAGs
* IP interfaces—One or more IP interfaces are created for each independent physical interface and for each LAG. Each IP interface must be assigned a unique IP address.
* Message Backbone VRF—This Virtual Routing and Forwarding (VRF) object contains all the NAB interfaces (physical, LAG, and IP), and transports message and topic subscription traffic between client applications and the virtual routers on the physical Solace appliance.
* Management VRF—This VRF object contains all the management interfaces (physical, LAG, and IP), and transports management traffic between management applications and the router.
* IP routes—IP routes are defined independently for the Management and Message Backbone VRFs, associating destination CIDR addresses in the VRF with next-hop gateway addresses. Default routes may also be defined.

The following figure shows the relationship of these components.

Default Solace Interface Configuration After Running the Setup



**Note:**The configuration shown above is that which is created after running the initial appliance set up through the **setup** privileged EXEC CLI command using default configuration parameters. By default, the physical interfaces on the NAB are grouped into a single LAG named lag1 upon completing the initial software configuration procedure. This provides a single IP address associated with the NAB interfaces through lag1.

As further discussed in this section, the configuration of the Solace interfaces can be modified as required to best suit your deployment requirements.

## Physical Interfaces

A Solace appliance uses two types of physical Ethernet interfaces:

* management interfaces—These interfaces handle traffic from management applications such as SolAdmin, Secure Shell (SSH), and Simple Network Management Protocol (SNMP). There are two management interfaces on the appliance motherboard, and their physical Ethernet ports can be found at the rear panel of a Solace appliance. The management interfaces are identified as eth0 and eth1 for a 3230 appliance and eth1 and eth2 for a 3260, 3530, or 3560 appliance.
* NAB interfaces—These interfaces handle message and topic subscription traffic from client applications. The NAB interfaces are physically located on the NAB, which is installed in one of the fabric expansion slots at the rear panel of a Solace appliance. The number of physical Ethernet ports on a NAB depends on the model of NAB installed in the Solace appliance. The network interfaces are identified as <cartridge>/<slot>/<port> (for example, 1/1/1 through 1/1/8).

**Note:**Physical interfaces are configured through the Interface CONFIG CLI commands, and their system configuration can be viewed through the **show interface** User EXEC command.

### LAGs

Solace appliances support LAGs, as defined in IEEE standard 802.3ad, so that multiple physical Ethernet interfaces can be grouped together to form a single, virtual, link layer interface.

A maximum of one LAG (identified as chassis/lag1) is supported for the management interfaces, while multiple LAGs are supported for the network interfaces on the NAB. The number of LAGs possible for the network interfaces is limited by the number of Ethernet ports present (due to the fact that LAGs reuse MAC addresses allocated for Ethernet ports). LAGs on the NAB are identified as <cartridge>/<slot>/lag<N>.

To client and management applications connecting to the appliance, a LAG appears as a single IP interface, but inside the LAG, packets are transmitted and received on the bundled physical ports. Link Aggregation Control Protocol (LACP) actively monitors the status of those grouped physical interfaces.

Some of the advantages of using LAGs are:

* simplified application integration—For Solace appliances and applications connecting to them, the group of aggregated Ethernet interfaces in a LAG appear as a single IP address.
* increased network bandwidth—The capacity of multiple links is combined into one logical link.
* increased availability—Traffic on a failed link can be redirected to one of the other links in the LAG (that is, the logical port transparently continues to function over the remaining physical ports). This allows for larger client loads per appliance with a longer meantime between service outages.
* more efficient bandwidth utilization—All traffic to or from the logical port is transparently load shared among all of the available physical ports.

https://docs.solace.com/Resources/Images/banner_notice2.gif

**NOTICE**: The use of 802.3ad link aggregation is restricted in that only point-to-point links between a single pair of devices may be aggregated. Multipoint aggregations among more than two devices is generally not supported, although some Layer 2 switch vendors may allow the links of a LAG to be split between two switches.

You can create independent LACP LAGs and LAGs that act in an active/backup mode, whereby two interfaces or two LACP LAGs are bonded together so that one member acts as the active member, the other as the backup. For more information on active/backup LAGs, refer to [Active/Backup LAGs](https://docs.solace.com/Features/Working-With-Appliance-Interfaces.htm#Active-Backup-LAGs).

**Note:**The 802.3ad specification defines two modes of LACP–Active mode and Passive mode. LAGs on the appliance always operate in Active mode. Thus, the third-party Layer 2 switches and appliances that support 802.3ad link aggregation can be configured as either Active or Passive mode when connected to the Solace appliances.

### Active/Backup LAGs

A LAG can optionally be configured in an active/backup mode whereby two interfaces are bonded together to provide link redundancy protection: one interface in the pair is active, while the other sits idle waiting to become active should a failure be detected on the active link. Typically, these two links terminate on different Layer 2 devices in the network to add protection should one of those devices fail.

The bonded interfaces can be separate physical interfaces (either management or network interfaces). In addition, if you are using a NAB with more than two Ethernet ports (for example, 4x1GE, 8x1GE, or 6x10GE NAB models), the network interfaces on the NAB can also be grouped into an even number of LACP LAGs, and then those LACP LAG pairs can be given an active/backup bonding. That is, the LACP LAGs can be bonded together in an active/backup LAG.

**Note:**The MAC address for the LAG in active/backup mode is the MAC address of its first member (that is, the first member added to the LAG).

When using the active/backup mode, there are two LAG members: primary and backup. Only one LAG member can be designated active at a time, and the primary LAG member is always designated active unless the backup LAG has more available bandwidth (that is, unless the backup LAG member has more active ports than the primary LAG member), or if it is a NAB port LAG, the **switch-active** Interface Admin EXEC command is run.

**Note:**The switch-active command is not persistent, so when used on an LACP LAG pair, subsequent changes in the operational status of one or more physical interfaces in the LACP LAGs may cause activity to revert to the previously active LAG member. If you want to force the backup LAG to remain active under these circumstances, you must shutdown the physical ports in the previously active LAG member.

All traffic is sent out or received on the active LAG member only. All incoming messages on the non-active port are dropped.

* For a NAB port LAG, once the backup LAG member becomes active, that switch from the primary to backup LAG member will not be reverted until one of the following events occurs:
  + the primary LAG member has more available bandwidth than the backup LAG member (that is, the primary LAG member has more active ports than the backup LAG member)
  + the appliance restarts, and the primary LAG member has equal or more available bandwidth than the backup LAG member (that is, at least as many active ports as the backup LAG member)
  + the **switch-active** Interface Admin EXEC command is run
* For a management interface LAG, the switch from the backup LAG member back to the primary automatically occurs as soon as the primary LAG member becomes available again (thus, the **switch-active** Interface Admin EXEC command is not supported by management interface LAGs).

**Note:**An event log is generated for every switch from primary to backup link or vice versa, regardless of its causes (that is, for example, whether it is a link failure or an execution of the **switch-active** Interface Admin EXEC command).

#### Possible Active/Backup LAG Configurations

LAGs can be used in active/backup mode for both management and network interfaces:

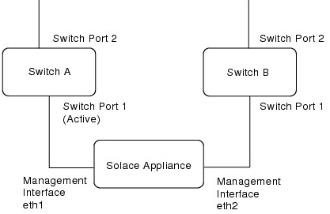
* For the management interfaces, a single LAG can be created with the two physical Management network interface ports, where one interface is active and the other is the backup. See [Active/Backup LAG Pairing for Management Interfaces](https://docs.solace.com/Features/Working-With-Appliance-Interfaces.htm#Active-Backup-Pairing-Mgmt).
* For the network interfaces, a LAG can be created with:
  + Two physical network interfaces on a NAB, where one interface is active and the other is the backup. See [Active/Backup LAGs for NAB Network Interfaces](https://docs.solace.com/Features/Working-With-Appliance-Interfaces.htm#Active-Backup-for-NAB).
  + Two LACP LAGs that are then bonded together as an active/backup pair. See [LACP LAGs in Active/Backup Pairs](https://docs.solace.com/Features/Working-With-Appliance-Interfaces.htm#LACP).

#### Active/Backup LAG Pairing for Management Interfaces

The figure immediately below shows an example of how the two physical Management network interface ports can be given an active/backup bonding and be connected to separate Layer 2 switches in a data center. This active/backup configuration provides Ethernet port, cable, and Layer 2 switch redundancy for the management access to the Solace appliance.

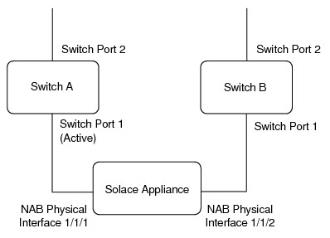
**Note:**When a LAG active/backup mode is not being used, it is possible to configure the Management network interface ports as two separate management IP interfaces to the same appliance. This allows for manual redundancy between the Management Ethernet interfaces, which allows an administrator to connect to the IP address of the backup interface if the primary interface is unreachable for any reason.

LAG in Active/Backup Mode on Appliance Management Ports



#### Active/Backup LAGs for NAB Network Interfaces

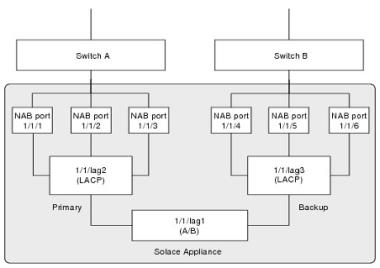
The figure immediately below shows an example of two physical network interfaces on a NAB, where one interface is active and the other is the backup. In this example, Port 2 of Switch A and B are both connected to the outside world, while Port 1 of both switches are connected to the NAB. In the event that Switch A fails, physical interface 1/1/2 becomes active and one or more gratuitous Address Resolution Protocol (ARP) announcements are sent out to Switch B.

LAG in Active/Backup Mode  


#### LACP LAGs in Active/Backup Pairs

The figure immediately below shows how two LACP LAGs can bonded together in a single LAG as an active/backup pair. In this example, 1/1/lag1 contains 1/1/lag2 as the primary member and 1/1/lag3 as the backup member.

LACP LAGs in Active/Backup Pairing



## IP Interfaces

Each Active/Backup LAG, each LACP LAG that is not part of an Active/Backup LAG, and each physical interface that is not part of a LAG, must have one or more of the following IP interfaces configured for it:

* static—a static IP interface that can be used to verify that the interface is physically connected and functional (once configured and enabled, a static IP interface is always reachable provided link layer connectivity is intact). This interface type is available in both the Message Backbone VRF and the Management VRF. The static IP interface in the Message Backbone VRF is also used by Solace routing protocols. Client applications can only connect using this address in the Message Backbone VRF if the system is not configured for redundancy.
* primary—an interface for the primary virtual router. This interface type is only available for interfaces contained in the Message Backbone VRF, it is not supported in the Management VRF.
* backup—an interface for the secondary virtual router. This interface type is only available for interfaces contained in the Message Backbone VRF, it is not supported in the Management VRF.

An IP interface is identified by the physical interface or LAG it is associated with, and is given an interface index number of 1, 2, or 3. For example, for physical interface 1/1/5, 1/1/5:1 could identify the primary IP interface, while 1/1/5:2 could identify the backup IP interface on that same physical interface. A given index can be assigned any of the interface types, that is, primary, backup, or static.

**Note:**Interface indices “2” and “3” are only supported for interfaces contained in the Message Backbone VRF, they are not available for interfaces in the Management VRF.

Primary interfaces are activated when the primary virtual router is locally active. Similarly, secondary interfaces are activated when the backup virtual router is locally active. Therefore, from an IP routing standpoint, the primary and backup IP interface states can go in and out of activity depending on the configuration and state of the virtual routers.

**Note:**

* IP interfaces contained within a VRF are configured through the VRF IP CONFIG CLI commands, and their system configuration can be viewed through the **show interface** User EXEC command.
* Shutting down the physical interface also disables any associated IP interfaces. However, the IP interfaces can be individually configured to be shutdown separate from their associated physical interface.

## VRFs

Virtual Routing and Forwarding (VRF) is a network technology that allows multiple instances of a routing table to coexist within the same router at the same time. Because the routing instances are independent, the same or overlapping IP addresses can be used in the VRFs without conflicting with each other.

Each IP interface on a Solace appliance is contained within either the Management VRF and Message Backbone VRF objects. These default VRF objects keep management and message traffic and routing tables separate.

The Management VRF and Message Backbone VRFs are preconfigured on an appliance through the **setup** Privileged EXEC CLI command. The Management VRF is assigned to interface eth1, and the Message Backbone VRF is assigned to <cartridge>/<slot>/lag1.

### Management VRF

The Management VRF handles the following types of traffic for a Solace router:

* CLI or SSH sessions
* Secure File Transfer Protocol (SFTP)
* Secure Copy (SCP)
* SNMP
* Solace Element Management Protocol (SEMP)

This management traffic is handled through the following physical interfaces:

* for Solace 3260, 3530, or 3560 appliances, eth1 and eth2 (and associated static IP interfaces eth1:1 and eth2:1, or LAG interface chassis/lag1:1)
* for Solace 3230 appliances, eth1 and eth0 (and associated static IP interfaces eth1:1 and eth0:1, or LAG interface chassis/lag1:1)

### Message Backbone VRF

The Message Backbone VRF handles the client messaging and topic subscription traffic that is exchanged between the network interfaces on the NAB and the virtual routers on the appliance. To support high-availability (HA), redundant Solace router pairs, a primary and backup virtual router can be configured on each physical appliance.

**Note:**The primary and backup virtual routers always exist on an appliance—they cannot be created nor deleted. However, if you are using a non‑redundant Solace router, configuring a backup virtual router is optional because the primary virtual router is always active, and the backup virtual router would be idle.

Primary IP interfaces are created in the Message Backbone VRF to support the primary virtual router, and these primary IP interfaces are only activated when the primary virtual router is locally active. Likewise, secondary IP interfaces are used to support the backup virtual router, and these secondary IP interfaces are only activated when the backup virtual router is locally active.

## IP Routes

When a client or management application connects to the router, the router will always send traffic to that application over the same IP interface that initially received the application’s incoming TCP connection (that is, the interface that the application is “connected to” at the TCP layer). If the application’s IP address is part of the same IP subnet as the selected egress interface, the outbound traffic can be sent directly to the application over that interface, and no additional routing is needed.

However, if the application’s IP address is in a different IP subnet than the egress interface’s subnet, than the outbound traffic must be forwarded to a next-hop IP router or gateway. IP routes are used to determine the IP address of that next-hop gateway.

When messages need to be forwarded to a next-hop gateway, the Message Backbone and Management VRFs use the following types of routes to choose the correct next-hop gateway:

* global specific routes

You configure the CIDR address used by the applications, and the associated next-hop IP address. This next-hop address must be reachable over the interface that the client is connected to.

* per-interface default routes

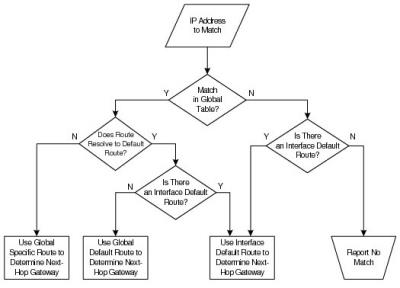
Per-interface default routes may be used for any traffic addressed to an IP address for which a global specific route has not been defined within the VRF. You configure the default next-hop IP address, and the interface within the VRF that the default route is associated with. For the Message Backbone VRF, the interface specified must be a physical or LAG interface, for the Management VRF, the interface specified must be an IP interface. Per-interface default routes are recommended in situations where the individual physical or LAG interfaces in the VRF are connected to different subnets.

* global default routes

Global default routes may be used for any traffic addressed to an IP address for which a global specific route has not been configured within the VRF and a per-interface default route has not been configured on the interface that the application is connected to. You configure the default next-hop IP address. This next-hop address must be reachable over the interface that the client is connected to. Global default routes are recommended in situations where the individual physical or LAG interfaces in the VRF are all connected to the same subnet.

**Note:**The routing information can be configured through the **route** VRF IP CONFIG command.

As shown in below, the Solace router uses these different types of IP routes in the order listed above. That is, during a routing table lookup, the router first attempts to use global routes that use a specific CIDR address. If these routes are not found, then it attempts to use the per-interface default route. for the interface that the destination is connected to, and if those routes are not found, it attempts to use the global default route.

Routing Table Lookup Flowchart  


## Physical Interface Configurations

This section lists the configuration options supported for both the management interfaces and the network interfaces on the NAB.

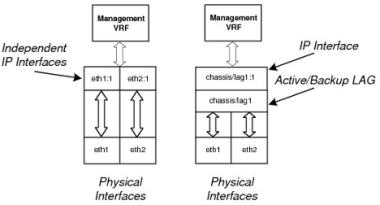
**Note:**The configuration that is possible for the network interface depends on the model of NAB used by the router and the number of Ethernet ports it offers. For each of the following configuration examples, an eight-port NAB is used.

## Management Interface Configurations

The two following configuration options are supported for the two Management network interfaces:

* assign independent IP interfaces to each Management port (that is, no LAG configured)
* group the Management ports into a single LAG in active/backup mode

Grouping the two ports together provides link redundancy protection. Only one link in the pair is active, while the other port sits idle waiting to become active should a failure be detected on the active link.

Possible Management Interface Configurations  


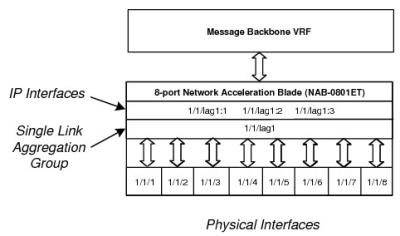
### NAB Interface Configurations

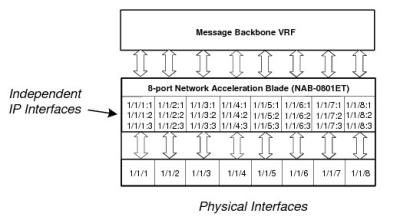
The following configuration options are supported for the network interfaces on the NAB:

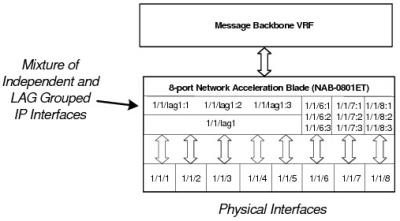
* Group the NAB ports into one or more LAGs. The number of LAGs possible for the network interfaces is limited by the number of Ethernet ports present (due to the fact that LAGs reuse MAC addresses allocated for Ethernet ports).
* Assign independent IP addresses to each NAB port (that is, no LAG configured).
* A mixture of both, that is, have some of the Ethernet ports grouped into a single LAG, and the remaining ports independently addressed.

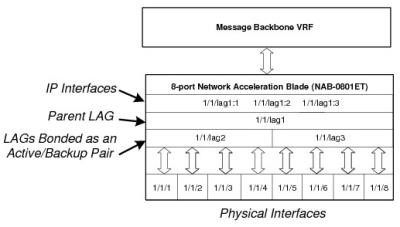
Also, as described in [Possible Active/Backup LAG Configurations](https://docs.solace.com/Features/Working-With-Appliance-Interfaces.htm#Poss-Active-Backup-Config), the network interface ports on the NAB support the concept of a LAG in active/backup mode that allows you to group two ports or two LACP LAGs together to provide link redundancy protection. Only one port/LAG in the pair is active, while the other port/LAG sits idle, waiting to become active should a failure be detected on the active port/LAG.

The following figures show the types of configuration that can be applied to the Solace physical interfaces.

All Network Interfaces Grouped Into a Single LAG  


All Network Interfaces Assigned Independent IP Addresses (No LAGs)  


Network Interfaces with Independent IP Addresses and Grouped Into a LAG  


Network Interfaces Grouped Into Active/Backup LAGs  


#### Egress Traffic Shaping

Some Solace NABs implement egress traffic shaping that can be used to delay the output data at rates for its physical interfaces. Although using a NAB’s maximum egress data rate is suitable for a well-designed IP network, such an egress data rate may not be appropriate for an IP network that many contain a mix of high-speed and low-speed interfaces, switches and routers with limited buffering, lossy links, and/or background traffic levels that are reach the capacity of the network. If these conditions exist, large numbers of packets may be discarded in the network when the NAB outputs data at the full line rate, resulting very poor goodput from the router.

The NAB egress traffic shaping is available on the following NABs:

* NAB-0610EM (all variants)
* NAB-0210EM-04 and later
* NAB-0401ET-04 and later
* NAB-0801ET-04 and later
* NAB-0810EM-01

When egress traffic shaping is offered for a NAB, you can individually configure maximum sustained egress bit rates (in Mbps) for each of its given physical interfaces.

# Multi-Node Routing

The Multi-Node Routing (MNR) feature allows multiple Solace routers (that is, nodes) to be networked together so that Direct messages published from clients connected to one router can be delivered to clients connected to the other routers. This effectively distributes the message routing load across multiple Solace routers in the network, resulting in aggregate message forwarding rates across the network that exceed the capacity of any one router.

**Note:**

* The Multi-Node Routing feature is for use with Direct messages only. Multi‑Node Routing does not route Guaranteed messages between routers.
* Network-wide load balancing with (Deliver-To-One) DTO, when used in conjunction with Multi-Node Routing, is not supported. A single node will always give priority to locally-connected clients when load balancing. Only when there are no available local clients eligible for DTO delivery will the router pass the DTO request to downstream routers. In such scenarios, no attempt will be made to load balance amongst multiple neighboring routers.

Multi-node Routing uses the Solace Content Shortest Path First (CSPF) protocol to link neighbor routers and allow them to discover the complete message-routing network topology to which they belong. Solace routers can then determine which neighbor would be the optimal node for the forwarding of messages addressed to specific destination routers. This topology discovery is continuous and dynamically updated as routers within the network go on or offline.

Multi-Node Routing also uses the Subscription Management Routing Protocol (SMRP) to enable linked neighbor routers to propagate topic subscriptions added by clients of one router throughout the message‑routing network. This allows messages that are published to one router to be routed to other routers that also have connected clients interested in those topics.

By default, the data and routing traffic carried over Multi-Node Routing links is carried in plain text over TCP. Data compression can optionally be applied on these links. However, if you will be transmitting sensitive data in the messages sent between routers, Transport Layer Security (TLS)/ Secure Sockets Layer (SSL) encryption can be applied to the Multi-Node Routing linksʼ data channels.

**Note:**

* Compression is only available on plain-text over SMF client connections to the neighbor router; compression is not available on TLS/SSL encryption is used for the data connections (that is, secure/encrypted client connections) to the neighbor.
* TLS/SSL encryption is only supported on the data links between Solace appliances. It is not currently supported for VMRs.

## Multi-Node Routing Compared to Message VPN Bridging

Messages published on one router can also be delivered to another networked router through the use of Message VPN Bridging. However there are some fundamental operational differences between the Multi-Node Routing and Message VPN Bridging. The table below lists some of the differences between the two.

| Comparing Multi-Node Routing and VPN Bridging | | |
| --- | --- | --- |
|  | Multi-Node Routing | Message VPN Bridging |
| Supports Direct Messaging | Yes. | Yes. |
| Supports Guaranteed Messaging | No. | Yes. |
| Only links between Message VPNs with the same names | Yes. | No. Bridges can link Message VPNs with the same or different names. |
| All subscriptions in a Message VPN are used for topic matching | Yes. | No. Only those subscriptions manually added to a bridge or queue associated with a bridge are used. |
| Topic subscriptions are static | No. Topic subscriptions are dynamically updated as clients add or remove subscriptions. | Yes. Administrators manually add subscriptions to bridges (Direct) or queues associated with bridges (Guaranteed). |
| Supports linking of Message VPNs of different names | No. Only Message VPNs with the same name on separate routers can be linked. | Yes. Message VPNs with different names can be linked. |
| Supports linking of Message VPNs on the same router | No. | Yes. Message VPNs on the same router can be linked if they have different names. |
| Supports data compression on links | Yes. | Yes. |
| Supports TLS/SSL encryption on links | Yes (appliance only). | Yes. |

Note that both features can be used together in a messaging network when carefully deployed. For more information on Message VPN Bridging, refer to [Working with Message VPN Bridges](https://docs.solace.com/Features/Working-With-Message-VPN-Bridges.htm).

After considering the differences between Multi-Node Routing and Message VPN Bridging shown in the preceding table, some general recommendations on when to use each feature are provided below.

### You should use Multi-Node Routing when:

* Guaranteed Messaging is not required
* Message VPNs span multiple routers, with the same Message VPN names on each router
* there is no need to limit the advertised subscriptions in a Message VPN (that is, subscriptions can be advertised to all routers in the network)

Other Multi-Node Routing advantages:

* very simple to configure—the only configuration required is identifying the neighbor links between routers and identifying which Message VPNs should advertise their subscriptions into the network
* uses a common name for the Message VPN across the network
* dynamically discovers the routers that are part of the same Message VPN
* dynamically reroutes around failed or offline routers or failed inter-router links
* guarantees a loop-free forwarding topology at all times
* automatically propagates subscriptions for the Message VPN between routers in the network
* subscriptions are automatically withdrawn from the network if all clients who subscribed to the topic have disconnected from the router

### You should use Message VPN Bridging when:

* Guaranteed Messaging is required
* you need to connect two distinct Message VPNs
* you need to carefully control which subscriptions get advertised between Message VPNs on the routers

**Note:**When you use Message VPN bridging, be careful with complex network messaging configurations to ensure a loop-free forwarding topology and that subscriptions on bridge links do not attract excessive traffic for which there are no interested clients on the local router.

**Related Provisioning and Configuration Information**

For information on how to configure and manage routing links and how to enable subscriptions to be exported over those links, refer to [Managing Multi-Node Routing](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-Multi-Node-Routing-Links.htm).

## Linking Routers

A routing link defines the route for connectivity between Solace routers. When building a messaging network, administrators must explicitly configure the routing links between router neighbors. Using these bi-directional links, those neighbors are then able to exchange network topology using the CSPF protocol and subscription information using SMRP. These configured neighbor links allow Direct messages published by clients to travel between neighbors on their way to the final subscriber clients.

Compression can optionally used on neighbor router data connections (only data connections are affected).

**Note:**Both ends of a message routing link must be configured. That is, to link routers A and B, there must be a neighbor configuration in A for B, and a neighbor configuration in B for A.

Multiple bi-directional routing links can be used to route messages between routers in your network for load balancing or redundancy, and the routing protocols that are used will calculate the shortest path (as defined by link costs). Routes with lower link costs are preferred over those with higher link costs. If there are multiple routing links with equal cost paths, then a route with less hops will be chosen as the preferred route.

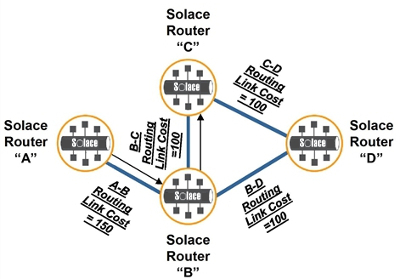
Where alternative routes are possible, an administrator can assign explicit link costs to indicate the best inter-neighbor message route. The rationale for assigning a higher link cost to a particular route could be based on considerations such as the speed, latency, or monetary cost of the underlying communication link.

### Routing Link Cost Examples

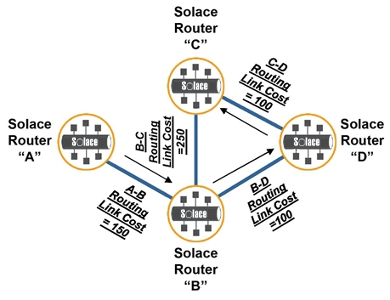
To prefer one route over another (for example, to specify a default link), you can change the routing link costs to set the preferred message route.In the example shown in [Routing Link Cost—Example 1](https://docs.solace.com/Features/Multi-Node-Routing.htm#multi-node-routing-link-cost-ex-1), there are four bi-directional routing links available between routers A, B, C, and D:

* A-B routing link
* B-C routing link
* B-D routing link
* C-D routing link

If the cost associated with the A-B routing link is set to 150, and the cost associated with the three other routing links is set to 100, messages routed from router A to C travel from router A to router B and then on to router C.

Routing Link Cost—Example 1  


However, as shown in [Routing Link Cost Example 2](https://docs.solace.com/Features/Multi-Node-Routing.htm#multi-node-routing-link-cost-ex-2) , if the cost associated with the A-B routing link is set to 150, while the cost associated with the B-C routing link is set to 250, and the cost associated with the C-D and B-D routing links is set to 100, messages routed from router A to C travel from router A to Site B, on to router D, and then to router C. Such a cost configuration may be desirable in cases where, for example, two router sites in one city are connected by a high-speed link, while other overseas router sites are connected by low-speed links.

Routing Link Cost Example 2   


## Subscription Propagation and Management

When multiple-node routing is used, topic subscriptions and matching messages are only distributed between matching Message VPNs. This means that a router with a topic subscription in one message VPN can only receive matching messages from a neighbor router that were published to a Message VPN of the same name. For example if a message was published to Message VPN Blue on Router A, that message may be delivered to a client on Router B with a matching topic subscription in a Message VPN Blue, but it may not be delivered to a client on Router B with a matching topic subscription in a Message VPN Green.

Topic subscriptions can be added by external clients and internal clients (for example, the internal client of the Message VPN, #client). When external client subscriptions are associated with a specific virtual router name; internal client subscriptions are associated with a physical router name.

### Exporting Topic Subscriptions

Each Message VPN has a topic subscription export policy associated with it, which controls whether or not topic subscriptions on the Message VPN get advertised to other Solace routers in the network. The default policy is to not export subscriptions.

To receive messages from other routers, the subscription export policy in a Message VPN must be set to export subscriptions. This causes all subscriptions added locally to the Message VPN to be advertised to the other routers in the network.

# Working with Message VPN Bridges

Message Virtual Private Network (VPN) bridges can be used to link two Message VPNs so that messages published to one Message VPN that match the topic subscriptions set for the bridge are also delivered to the linked Message VPN.

A Message VPN bridge allows for the delivery of Direct messages that match an explicit set of topic subscriptions from a remote Message VPN to a local Message VPN. Guaranteed messages can also be delivered over Message VPN bridges when additional parameters are configured for the remote Message VPN (for more information, refer to [Avoiding Loss of Guaranteed Messages with VPN Bridges](https://docs.solace.com/Features/Working-With-Message-VPN-Bridges.htm#working_with_bridges_avoid_loss)).

The possible scenarios in which Message VPN bridges can be used include:

* Link two Message VPNs with different names to enable Direct or Guaranteed messages published to one Message VPN to also be delivered to the other Message VPN. The linked Message VPNs can be either on the same router or on two separate routers.
* Link two Message VPNs with the same names on two separate routers to enable Guaranteed messages published to one Message VPN to also be delivered to the other Message VPN.
* Link two Message VPNs with the same names on two separate routers to enable Direct messages published to one Message VPN can also be delivered to the other Message VPN.

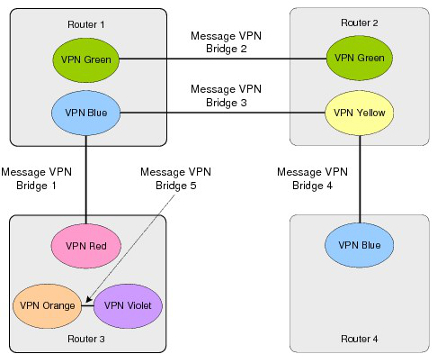
**Note:**That this scenario operates differently than multi-node routing, in that Message VPN bridges only transfer those Direct messages that match the topic subscriptions explicitly set on the Message VPN bridge. In addition, Message VPN bridges do not exchange the complete set of topic subscriptions used by the routers and their topology.

A bridge can be uni-directional (messages pass over the bridge in only one direction) or bi-directional (messages pass over the bridge in both directions). The messages that pass in either direction may be different, depending on the remote topic subscriptions that are assigned to the bridge for each local Message VPN.

## Controlling Access to Messages

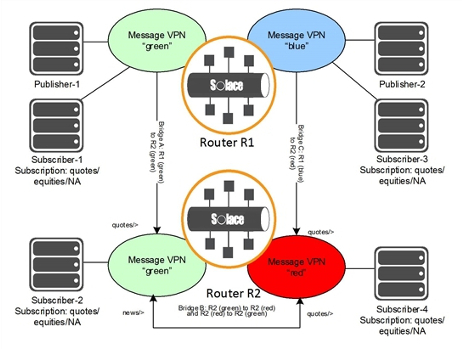
Solace Messaging Platform networks may use Message VPN bridges to control whether messages may pass from one Message VPN to another. As shown in below, when a Message VPN bridge is used to link two Message VPNs, the topic subscriptions assigned to the bridge (or the bridge queue for Guaranteed messages) determines which messages may be delivered from a remote Message VPN to the local Message VPN it is linked to.

Typical Message VPN Bridge Configurations for Controlled Access



The next figure shows a deployment where there are clients across three different Message VPNs and two routers not participating in multiple-node routing that all have subscriptions for the same topic–“quotes/equities/NA”.

Message VPN Bridge Example



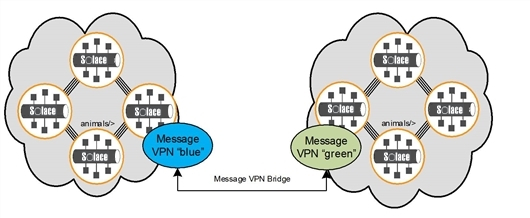
The same message is also sent from Message VPN green on router R2 across Message VPN bridge B to Message VPN red to fulfill the topic subscription quotes/>. Note that Message VPN bridge B is a bi-directional bridge, so that messages published to Message VPN red could also be delivered to Message VPN green on router R2 if they match the news/> subscription set for that end of the bridge.

## Internetworking

As shown in the figure below, a Message VPN bridge can also be used to link two networks together. In this use case, an entire remote network (rather than just a single remote client) can join to another network to form an extended intranet.

This configuration is useful for companies with offices and private networks located in different physical locations that need to carefully control the message flow over expensive WAN links between sites. For this scenario, each location uses multiple-node routing for fully‑distributed Message VPN connectivity for Direct messages, and Message VPN bridging is used to transfer messages for specific topics over a LAN or MAN.

Bridging Message VPNs for Internetworking

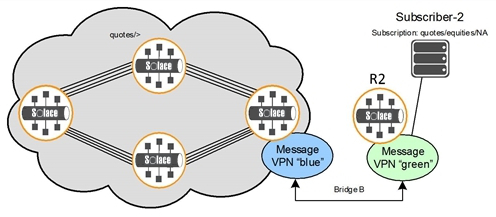


As shown in the figure below, a group of routers can be configured as neighbors for multiple-node routing to efficiently route traffic through a network. Edge routers in that in group can then be linked by Message VPN bridges so that messages in the network that match select topics can be delivered to clients connected to routers outside of those groups of neighbors.

In this type of Message VPN bridge application:

* publishers are typically in the core network with subscribers on the edge of the network
* Message VPN bridges subscribe to a small subset of the topics in Message VPN blue in the core

Message VPN Bridges with Multi-Node Routing in the Core



## Establishing VPN Bridges

To establish a Message VPN bridge, you must create it in a local Message VPN. An internal client is then automatically created for the bridge. This client allows a connection to be established to a remote Message VPN. A Message VPN bridge connection looks like a client connection to the remote router, and its login attempt to the remote router is authenticated in the same manner as other client connections. Through this client, control and data messages are sent between the local and remote Message VPNs.

Once a Message VPN bridge is created, the following parameters must be configured for the bridge before it can be enabled and messages can begin to be transferred:

* remote Message VPN and router—This is the remote Message VPN and router to receive messages from. The router is specified either by its name or “connect-via” address expressed as an IP address or DNS name. Multiple remote Message VPNs can also be specified in order of preference. These redundant remote Message VPNs provides some protection if the first remote Message VPN the bridge connects to becomes unavailable.

**Note:**If you want to establish a bridge link to a remote Message VPN that is also on the local router, use an IP address of 127.0.0.1 and do not specify a physical interface. This configuration will establish a loopback Message VPN bridge.

* remote Message VPN authentication scheme—the authentication scheme that the bridge uses to authenticate with the remote Message VPN. An authentication scheme is required for each remote Message VPN that is configured for the bridge.
  + basic authentication client username—the username used to authenticate with the remote Message VPN, if basic authentication is configured. An optional password can also be specified.
  + client certificate authentication certificate file—the certificate file used to authenticate with the remote Message VPN, if client certificate authentication is configured. If no certificate file is specified, the bridge will present the router’s server certificate for authentication.

**Note:**By default, remote Message VPN bridge connections are not available to clients—permission must first be configured in the client profile that is assigned to the client username. Permission to create bridge connections can be enabled through the clientsʼ assigned client profile (refer to [Allowing Bridge Connections](https://docs.solace.com/Configuring-and-Managing-Routers/Configuring-Client-Profiles.htm#allow-bridge-connections)).

* remote subscription topic—If you want to transfer Direct messages over the Message VPN bridge, configure one or more topic subscriptions for the bridge that will attract Direct messages published to the remote Message VPN. Topics are added to the top level of the bridge and apply across all hosts. The set of topics may be changed while the bridge is enabled.
* remote message spool queue—If you want to transfer Guaranteed messages over the Message VPN bridge, use this parameter to specify an existing durable queue (with an exclusive access type) on the remote Message VPN to which a consumer flow will connect. Topic subscriptions must be added to the queue to attract messages published to specific topics of interest.

The following parameters can also be configured (if they are not modified, the default values will be used):

* remote deliver-to-one priority—The deliver-to-one priority given to the subscriptions that are assigned to the bridge. By default, the highest priority is used.
* remote Message VPN SSL—Sets whether to use TLS/SSL encryption for the remote Message VPN bridge link. This can be configured for each remote Message VPN that is configured for the bridge. If TLS/SSL encryption is used, the port on the router used by the remote Message VPN must also be a TLS/SSL port (port 55443 is the default TLS/SSL port). For information on using TLS/SSL with Message VPN bridges, refer to [Managing TLS/SSL Service](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-TLS-SSL-Service.htm).
* remote Message VPN compressed data—Sets whether to use data compression for the remote Message VPN bridge link. This can be configured for each remote Message VPN that is configured for the bridge. If data compression is used, the port on the router used by the remote Message VPN must also be a compression port (port 55003 is the default compression port).
* remote Message VPN connect order—Configures the order in which the bridge attempts to connect to configured redundant remote Message VPN hosts. This is configured per host (that is, for each remote Message VPN that is configured for the bridge).
* SSL parameters—Sets the cipher suites to be used for TLS/SSL encryption as well as the list of trusted common names used to verify the identity of remote server certificates. For information on using TLS/SSL encryption with Message VPN bridges, refer to [Managing TLS/SSL Service](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-TLS-SSL-Service.htm).
* remote Message VPN message spool window size—Sets the requested transport window size for the consumer Flow that is bound to the remote queue. This determines how many outstanding Guaranteed messages can be delivered from the remote queue over the Message VPN connection before an acknowledgment must be received by the remote queue. Modifying the window size can be helpful for bridges that go across high-latency links.

### VPN Bridging and Fault Tolerance

For details on how to establish Message VPN bridge connections to remote routers when those remote routers have been deployed in high-availability (HA) redundant router pairs for fault tolerance, refer to [Bridging With Appliances Using Active/Standby Redundancy](https://docs.solace.com/Configuring-and-Managing-Routers/VPN-Bridging-to-Remote-Appliances.htm#Active-Standby).

## Guaranteed Messaging Over VPN Bridges

To use a Message VPN bridge to transfer Guaranteed messages from a remote Message VPN to a local Message VPN, there must first be:

* a durable queue provisioned on the remote Message VPN
* a durable topic subscription added to that durable queue so that messages with a topic match can be spooled to the queue

Then, when you create the bridge on the local Message VPN, you must specify the following:

* the remote Message VPN to connect the bridge to
* the provisioned queue on that remote Message VPN

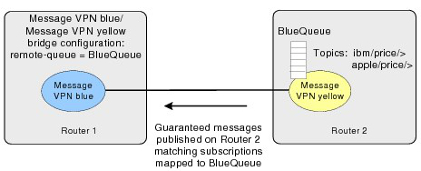
When the bridge is established, a consumer flow is bound to the queue. Then messages published to the remote Message VPN with matching topics can be transported over the bridge to the consuming router using the bound flow.

For clients to receive these messages, topic subscriptions must be added to the local Message VPN:

* For the Guaranteed messages received from the bridge to remain as Guaranteed messages (that is, retain their non-persistent or persistent delivery modes), a durable queue that has been assigned a matching topic subscription is required. In this case, the received messages are persisted because they are saved to the message spool.
* If the Guaranteed messages received from the bridge only match client topic subscriptions on the local Message VPN, the messages are converted to Direct messages and are not persisted.
* If there are no matching subscriptions on the local Message VPN, the router discards the messages.

In the example shown in the figure below, the Message VPN bridge on Router 1 is configured to establish a Guaranteed message flow to the durable queue named BlueQueue. That queue is configured through the CLI or SolAdmin on Router 2, along with a list of topics which are to be mapped to BlueQueue (in this example ibm/price/> and apple/price/>). Any messages published to these topics are written to BlueQueue, and then transported through the Guaranteed message flow over the bridge to Router 1. Upon reaching Router 1, the messages are delivered to all queues and topic endpoints with matching subscriptions.

Message VPN Bridge Configuration for Guaranteed Messaging



If multiple consumer flows are going to be established to a remote Message VPN, then a separate durable message queue should be created in the Message VPN for each of the inbound bridges that want to receive Guaranteed messages.

### Avoiding Loss of Guaranteed Messages with VPN Bridges

When properly configured, Guaranteed messages will not be lost when sent over a Message VPN bridge. The normal operational behavior is as follows:

* Guaranteed messages do not unspool from the remote Message VPN of a Message VPN bridge unless they can be spooled by all matching endpoints on the local Message VPN. When a Guaranteed message cannot be successfully spooled to all endpoints with matching subscriptions on the local Message VPN, the router periodically retransmits the message across the Message VPN bridge. Once the Guaranteed message is successfully spooled to all matching endpoints on the local Message VPN, then it is unspooled from the remote Message VPN.
* If a Guaranteed message is rejected by the local Message VPN when it is retransmitted across the bridge, no other Guaranteed messages can traverse the bridge. This can cause the bridge queue on the remote Message VPN to fill up. If the bridge queue reaches its quota, it will reject Guaranteed messages just published to the remote Message VPN. Direct messages, however, including those that are promoted on the local Message VPN, continue to traverse the bridge while in this state.
* Once the condition that prevents the Guaranteed messages to be spooled by the endpoints on the local Message VPN is cleared, Guaranteed messages begin to traverse the Message VPN bridge again. Once the bridge queue has dropped below quota, it starts accepting newly published Guaranteed messages again.

The correct configuration for this behavior requires the reject‑msg‑to‑sender‑on‑discard option to be enabled on the durable endpoints and for a matching durable subscription to be configured on the local Message VPN. By default, the reject-msg-to-sender-on-discard option is enabled on queues but is disabled on topic endpoints.

Although the reject-msg-to-sender-on-discard option only needs to be enabled on one matching endpoint in the local Message VPN to get the desired behavior, it is recommended that it be enabled on all endpoints in the local Message VPN. For more information on this feature, refer to [Message VPN-Level Guaranteed Messaging Configuration](https://docs.solace.com/Configuring-and-Managing-Routers/VPN-Level-Msg-Spool-Config.htm).

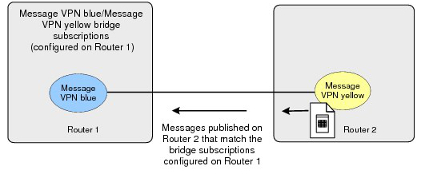
The local Message VPN must also have a matching durable subscription for the Guaranteed message. If there are no endpoints with matching subscriptions, then the message is discarded as “No Eligible Destination” on the local Message VPN and the message is unspooled from the bridge queue on the remote Message VPN. This prevents the bridge consumer flow from being blocked by messages that have no destination on the local Message VPN.

**Note:**If the Guaranteed messages received from the bridge match client topic subscriptions on the local Message VPN, the messages are converted to Direct messages and are not persisted.

## Uni-Directional Versus Bi-Directional VPN Bridges

Subscriptions must be statically configured on a Message VPN bridge to attract Direct messages from a remote Message VPN. The figure below shows a uni‑directional bridge that has been configured on Router 1 so that messages from that remote Message VPN with matching topics can be transferred over the bridge from Router 2 to Router 1 using a single TCP connection.

Uni-Directional Message VPN Bridge

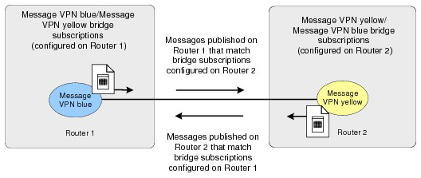


It is possible to extend the configuration shown above to create a bi-directional bridge between the two Message VPNs. A bi-directional bridge essentially creates a bridge link in the opposite direction of an existing uni-directional bridge; where the local Message VPN of the uni-directional bridge is the remote Message VPN that the new bridge links to, and the remote Message VPN that the uni-directional bridge links to is the local Message VPN of the new bridge link.

As shown in the figure below, by adding a bridge configuration on Router 2 from Message VPN yellow to Message VPN blue on Router 1, a bi-directional bridge is established, and a single TCP connection is used to transport all messages between the two Message VPNs.

The No Local Delivery property is automatically enabled for Message VPN bridges to prevent forwarding loops. This means that messages transferred across a Message VPN bridge to a remote Message VPN cannot then be sent back to the local Message VPN they originally came from to fulfill matching topic subscriptions again.

Bi-Directional Message VPN Bridge



### Establishing Bi-Directional Bridge Links

Two uni-directional bridges cannot be established between the same two Message VPNs—a bi-directional bridge must be used. Therefore, before a local router attempts to establish a bridge to a remote router, it inspects the existing incoming bridges to see if there is a compatible bridge already established from the remote router that can be used. To be compatible, an existing bridge from the remote router must pass the following tests:

* the remote router name from the peer router must match the local router name
* the router name of the peer must match the specified remote router name
* the remote Message VPN of the peer must match the local Message VPN
* the Message VPN of the peer must match the specified remote Message VPN
* if TLS/SSL encryption is to be used, then TLS/SSL must be configured on both bridges (that is, an TLS/SSL bridge is not compatible with a non-TLS/SSL bridge)

If there is a compatible incoming bridge, then no new Message VPN bridge connection is required and an existing uni‑directional Message VPN bridge is transformed into a bi-directional Message VPN bridge. The login is performed with the existing connection.

# Appliance Redundancy and Fault Tolerance

Solace appliances can operate in high-availability (HA) redundant pairs for fault tolerance. Redundancy provides 1:1 appliance sparing to increase overall service availability. HA redundancy eliminates the potential for a single point of failure in the Solace Messaging Platform by allowing a network administrator to define two appliances as a redundant pair. If one of the appliances is taken out of service or fails, the other appliance automatically takes over responsibility for the clients typically served by the out-of-service appliance.

The redundancy feature is largely transparent to clients and other Solace appliances in the network. Only the two appliances that are paired as mates require explicit configuration to take advantage of the feature.

Similarly, there is no configuration needed on a client host system to take advantage of the HA redundancy facility. The only visible impact to clients during a redundancy failover is non-delivery of messages for a short period of time, and the clients are forced to reconnect.

The Virtual Router Redundancy Protocol (VRRP), as defined by Request For Comments (RFC) 3768, is used to manage the IP address used by clients to communicate with Solace appliances. The appliance that is active for a given IP address terminates all client messages sent to that IP address. If the active appliance fails, the backup appliance takes over the IP address. Therefore, migration of the IP address and message routing services from one appliance in the HA pair to its mate appliance allows clients to reconnect to the same IP address which is now managed by the mate appliance.

Solace redundancy models are built around the concept of “virtual routers”:

* If an active/standby model is deployed, one virtual router with a unique IP address must be configured for the redundant pair. This virtual router is configured as the primary on one appliance and as the backup on its mate.

Under normal operating conditions, the physical appliance that is “active” for the primary virtual router handles messaging activity. If that appliance fails, the mate appliance takes over activity for the virtual router.

* If an active/active model is deployed, two virtual routers must be configured within the redundant pair, each with its own unique IP address. One appliance is configured as the “primary” for one of the virtual routers (and “backup” for the second virtual router), while the second physical appliance is configured as the “primary” for the other virtual router (and “backup” for the first virtual router).

Under normal operating conditions, each appliance is “active” for its primary virtual router, and “standby” for the mate’s virtual router. If one of the appliances fails, the mate appliance takes over activity for the virtual router associated with the failed appliance.

## Active/Standby Redundancy Model

Active/standby redundancy is supported by the Solace Messaging Platform for both Guaranteed and Direct Messaging.

In the active/standby model, the primary appliance provides service to clients and sends and receives data and messages, while the backup appliance waits in standby mode but only provides service should the primary appliance fail (because load sharing between the two appliances is not supported).

In the active/standby model:

* All clients connect to the same virtual router, which is configured as primary for one appliance and as backup on its mate appliance.
* The backup appliance acts only as a standby. While the primary appliance is active, no clients can connect to the backup, and no messaging traffic can flow through the backup.
* If the primary appliance fails for any reason, the backup appliance takes over the IP address of the virtual router associated with the failed appliance by sending gratuitous Address Resolution Protocol (ARP) requests, and all the clients reconnect to the backup appliance.

## Active/Active Redundancy Model

Active/active redundancy is supported by the Solace Messaging Platform for Direct Messaging. In this active/active model, clients may be distributed between the two appliances, and both appliances can provide service simultaneously during normal operating conditions. This allows for load-sharing while both appliances are functional. However, should one of the appliances fail for any reason, the active appliance can provide the service ordinarily provided by both of the appliances individually.

In the active/active model:

* One appliance is active for one virtual router, while the other appliance is active for the second virtual router.
* Clients actively connect to both virtual routers (thus to both appliances) and both appliances carry messaging traffic.
* If one of the appliances fails, the mate appliance takes over the IP address of the virtual router associated with the failed appliance by sending gratuitous ARPs, and all the clients associated with the failed appliance reconnect to the backup appliance.

Two variants of the active/active redundancy model can be deployed, whereby:

* Individual Message VPNs are Active/Standby.

In this deployment model, all clients for one group of Message VPNs are configured to connect to virtual router #1, and all clients for a separate group of Message VPNs are configured to connect to virtual router #2. That is, all clients for a given Message VPN are configured to be serviced by one appliance in the redundant pair.

* Individual Message VPNs are Active/Active.

In this deployment model, clients of a given Message VPN span the two appliances in the redundant pair, with the clients divided between the virtual routers. For this model to provide full reachability between all clients of the distributed Message VPN at all times, either multiple-node routing links or Message VPN bridge links must be provisioned to provide connectivity between the two appliances.

**Note:**Guaranteed Messaging may also be configured in conjunction with Direct Messaging, however, only one appliance may provide Guaranteed Messaging service at any given time. That is, while the Redundancy Mode can be configured as Active/Active where both appliances support Direct Messaging concurrently, Guaranteed Messaging will operate in Active/Standby mode on one virtual router.

## Recommended Redundancy Model

Solace recommends deploying appliances in the active/standby model, for the following reasons:

* Active/standby supports both Guaranteed and Direct Messaging.
* Active/standby avoids the risk of inadvertent overbooking an appliance, whereby each appliance has sufficient capacity to handle the messaging load for its own virtual router, but under a failure scenario, a single appliance has insufficient capacity to handle the workload of the two virtual routers, resulting in an unacceptable service degradation that is not exposed until a failover occurs.
* Lower engineering/planning overhead. No need to decide which Message VPNs are going to reside within each virtual router, or which clients will connect to which virtual router.
* Active/standby is an easier model to test and validate prior to going into production.

## Fault Tolerance

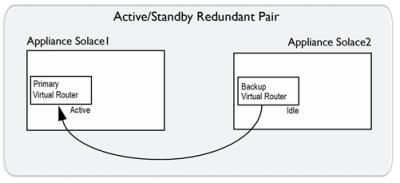
The following sections discuss the interaction between primary and virtual routers and primary and backup IP interfaces in appliance failover situations. It also provides network topology examples.

### Primary and Backup Virtual Routers

To support redundancy, each Solace appliance uses a primary and backup virtual router. To enable the backup virtual router to assume the role of its mate’s primary virtual router when a failure occurs, the configuration of the virtual routers on each appliance must mirror one another. That is, the backup virtual routers must have the same configuration as the primary virtual routers they backup.

For an active/standby redundant pair, the primary virtual router is on the primary appliance, and the backup virtual router is on the standby appliance. If the primary appliance goes out of service, the backup virtual router of the standby appliance changes to an active state, and it provides service for clients and handles the data and messages that typically use the primary virtual router of the primary appliance that has gone out of service. The figure below shows the relationship between the virtual routers of an active/standby redundant pair.

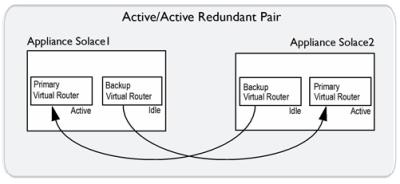
Virtual Router Relationships in an Active/Standby Redundancy



|  |  |  |
| --- | --- | --- |
| Appliance Solace2 Backup Virtual Router | supports... | Appliance Solace1 Primary Virtual Router |

For an active/active redundant pair, the primary virtual routers on both appliances are active, but the backup virtual routers are idle. If one of the appliances in the redundant pair goes out of service, the backup virtual router of the inactive appliance changes to an active state, and it provides service for clients and handles the data and messages that typically use the primary virtual router of the appliance that is out of service. The figure below shows the relationship between the virtual routers of an active/active redundant pair.

Virtual Router Relationships in Active/Active Redundancy



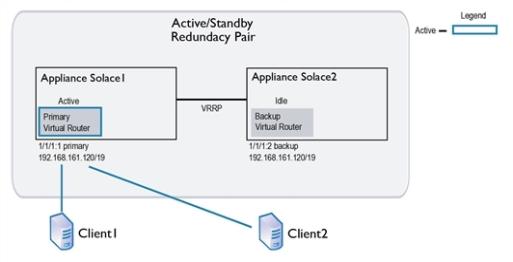
|  |  |  |
| --- | --- | --- |
| Appliance Solace1 Backup Virtual Router | supports... | Appliance Solace2 Primary Virtual Router |
| Appliance Solace2 Backup Virtual Router | supports... | Appliance Solace1 Primary Virtual Router |

The primary and backup virtual routers are served by the Message Backbone Virtual Routing and Forwarding (VRF) object. Through the physical interfaces located on the Network Acceleration Blade (NAB), the Message Backbone VRF handles the client traffic (including subject subscriptions) for the virtual routers.

### Primary and Backup IP Interfaces

For each Solace appliance, physical interfaces on the NAB must be bound to distinct IP interfaces that are identified by an IP address and subnet mask. It is these IP interfaces that clients connect to. These primary and backup IP interfaces are associated with the primary and backup virtual routers on each Solace appliance in the redundant pair.

The example immediately below shows a simplified example of the primary and backup IP interfaces and virtual routers used by an active/standby redundant pair. In a failover situation, clients can reconnect to the backup IP interface.

Simplified Active/Standby Configuration  


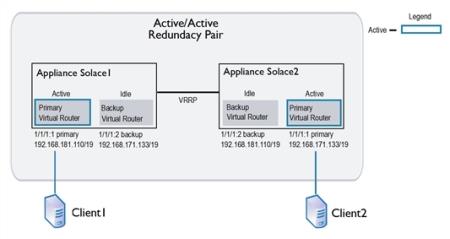
To enable active/active redundancy, primary and backup instances of the IP interfaces are created for each appliance in a redundant pair. The same IP interfaces are used by each appliance, but they are assigned as primary on one and as backup on the other. Therefore, if one appliance goes out of service, a backup IP interface can still be accessed by the client on the inactive appliance.

Below is shown a simplified example of the primary and backup IP interfaces and virtual routers used by an active/active redundant pair, and then the same redundant pair is shown in a failover situation.

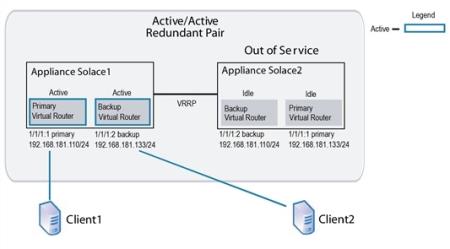
* On appliance Solace2, the physical interface 1/1/1 contains two IP interfaces. 1/1/1:1 is configured as the primary IP interface with IP address 192.168.181.133/24, and 1/1/1:2 is configured as the backup IP interface with IP address 192.168.181.110/24.
* To maintain service if appliance Solace2 goes down, the mate appliance, Solace1, also contains the IP interfaces 1/1/1:1 and 1/1/1:2. However the IP addresses assigned to these IP interfaces are reversed. For Solace1, the primary IP interface 1/1/1:1 is configured with IP address 192.168.181.110/24, and the backup IP interface 1/1/1:2 is configured with IP address 192.168.181.133/24.

**Note:**An IP number from 1 to 3 indicates the type of IP interface (primary, backup, or static). A typical association is 1 for primary, 2 for backup, and 3 for static, but this is not enforced.

Simplified Active/Active Configuration



Simplified Active/Active Configuration in Failover



In addition to the IP interface that clients connect to, a Solace appliance uses a VRRP interface. Through this VRRP interface, both appliances of the redundant pair can communicate the status of their virtual routers. Primary and backup instances are required for each VRRP interface, and to enable redundancy, the IP addresses assigned for both primary VRRP interfaces must match the IP addresses for their backup VRRP interfaces.

### Active/Standby Network Topology Example

The example shown immediately below shows a physical network topology example for active/standby redundancy when used for Guaranteed Messaging. Only one Solace appliance in the redundant pair (Solace1) receives messages through the primary virtual router. For the other appliance (Solace2), the same virtual router is configured is configured as backup. However, should Solace1 fail, Solace2 will detect the failure and start providing service to the clients that were connected to Solace1.

When active/standby appliance is used with Guaranteed Messaging, a customer‑supplied external disk storage array is required, and each Solace appliance must have an ADB and an HBA.

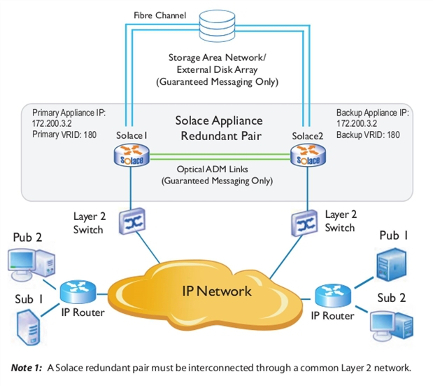
In addition, when Guaranteed Messaging is used, a pair of redundant optical links connect the ADBs between the two appliances. The ADB optical link is used to propagate Guaranteed messages and states from the active appliance to the standby appliance. Whenever the standby appliance becomes reachable over the ADB link, the active appliance copies the entire ADB contents from the active to the standby appliance over the ADB link. Any subsequent messages and state information which the active appliance writes to the ADB is automatically propagated over the optical link to the ADB of the backup appliance. This ensures that the message contents and state of the ADB in the backup appliance are always in sync with the ADB of the active appliance.

#### SAN Behavior for Guaranteed Messaging Through Active/Standby Redundancy

Each pair of Solace appliances in an active/standby redundancy configuration require a separate Logical Unit Number (LUN) on the Storage Area Network (SAN), whereby two partitions are created on the LUN. One of the appliances mounts the first partition as “read/write” and the second partition as ‘read only, while the mate appliance mounts the first partition as ‘read only’ and the second partition as “read/write” (that is, the inverse of its paired appliance).

The active appliance in an active/standby redundant pair only ever writes to its “read/write” or active partition. In the case of a failover, it provides service for its mate by reading the messages spooled by its mate through the standby partition mounted as “read only”.

Physical Network Topology Example for Active/Active Redundancy



#### Basic Operation

If a Solace appliance detects that its mate is down, it asserts an activity switch on behalf of its mate.

The basic operational activity is:

1. The primary appliance periodically advertises that it is active for its VRID.
2. If the primary appliance stops advertising for a predetermined period of time, the backup appliance using the same VRID takes over message forwarding responsibilities from the primary. The backup appliance is now active.

## Failure Detection

As shown in the figure of the redundant HA pair immediately above, there are several paths of connectivity between the redundant appliance pair. The redundancy state machines within each appliance monitors the status of the mate appliance through the following communication paths:

1. VRRP over the Layer 2 Network – VRRP is always employed between a redundant pair of appliances to communicate activity status, and to detect a failure of the mate appliance.
2. Multiple-Node Routing – If the appliances have been configured as CSPF neighbors in multiple-node routing, then the activity status of the appliances is advertised as part of the periodic “hello” messages sent by CSPF.
3. ADB Optical Links – If the appliances are equipped with ADBs for Guaranteed Messaging, then the activity status of the appliances is advertised over the fibre links that interconnect the ADBs on the two appliances.

An active appliance may decide to voluntarily release activity to the standby appliance by sending a “releasing-activity” indication over all the inter-router communication paths. However, to prevent the possibility of paired appliances becoming primary simultaneously and creating a split-brain scenario, the standby appliance does not take activity until it receives a “releasing-activity” indication, or detects a loss of connectivity on all three of the communication paths described above.

Once an appliance has taken activity, it typically does not pro-actively send a “releasing-activity” indication to its mate unless one of the following events occur:

1. A **release-activity** Redundancy CONFIG or **revert-activity** Admin EXEC command is executed through the CLI or SEMP.
2. The appliance detects that it has lost all Layer 2 connectivity on the NAB.
3. The appliance detects that it has lost all connectivity to the SAN.
4. The appliance detects a fault that requires it to restart.

**Note:**Under some fault conditions, the failing appliance may not always be able to send a “releasing-activity” indication to its mate before restarting. However, such a restart causes an interruption of all three communication paths, which after a debounce time of several seconds on the mate appliance, causes the mate appliance to take activity.

## Failover Sequences

This section discusses the failover sequences that occur for:

* [Router Failure Activity Switch](https://docs.solace.com/Features/Redundancy-and-Fault-Tolerance.htm#redundancy_and_fault_tolerance_344917737_258022)
* [Appliance Recovery Activity Switch](https://docs.solace.com/Features/Redundancy-and-Fault-Tolerance.htm#redundancy_and_fault_tolerance_344917737_211395)

**Note:**

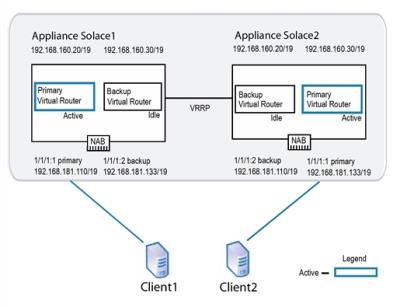
* Many other router activity switch scenarios are possible, but they are beyond the scope of this section. For more information, contact Solace.
* A number of Solace router priority levels have been defined for redundancy. These priority levels are advertised by the virtual routers. When activity switches occur between redundant pairs, the priority levels advertised by the virtual routers of each appliance change to indicate their current state and role. These advertisements are broadcast between redundant pairs through VRRP. For example, the virtual router that advertises the highest priority level for a given VRID is the active virtual router for that VRID. All messages and system requests for that VRID are then forwarded to that virtual router.

### Router Failure Activity Switch

* reset or power-cycle of the router (transient outage, planned or unplanned)
* hardware failure of the router (longer-term outage, unplanned)
* physical link to the router is down (could be transient or longer-term, planned or unplanned)
* software upgrade of the router (transient outage, planned)
* router has been placed in the standby state as a result of the network operator entering the **release-activity** Redundancy CONFIG command through the Solace CLI (longer-term outage, planned)

The following figure shows an example of an active/active network configuration.

Active/Active Redundant Pairing



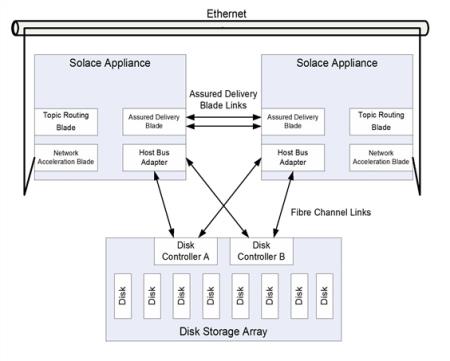
Using this example the following series of operations occur if one of the appliances (Solace2 in this case) goes out of service for any reason:

1. VRRP on appliance Solace1 detects the failure of appliance Solace2.
2. Solace1 becomes active for the IP address 192.168.181.133/19. When this occurs, the backup virtual router for Solace1 goes from an idle state to an active state.
3. Solace1 increases the priority of its VRRP advertisements for the IP address 192.168.181.133/19 to Assert-Activity. It also sends out a gratuitous ARP to let other network devices know that it is taking over the IP address. At this point, network traffic to the IP address 192.168.181.133/19 flows to Solace1.
4. Clients that had already established TCP connections to Solace2 receive error messages back from Solace1 because the TCP connections are not present on Solace1. This causes the clients to tear down the existing TCP sessions and immediately establish new TCP sessions. These new connections are accepted by Solace1.
5. Solace1 accepts subscription updates and published messages from its own clients and those that typically have service from Solace2.

#### Active/Standby Using Guaranteed Messaging

If an active/standby redundancy model for Guaranteed Messaging is used for the redundant pair, the failure of the primary appliance is detected through the VRRP link between the appliances. In addition, the failure is also detected by the ADB link—the standby appliance uses the state from its local ADB to continue to provide service. The figure below shows the links used between active/standby appliances.

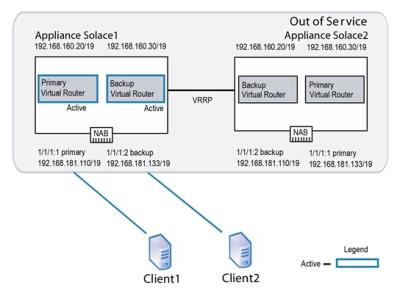
Links Used by Active/Standby Appliances Using Guaranteed Messaging



### Appliance Recovery Activity Switch

The figure below shows a common recovery activity switch scenario for an active/active redundant pair, where appliance Solace2 has failed and its mate appliance, Solace1, is acting on behalf of Solace2.

Activity Switch Behavior Failure



When appliance Solace2 comes back online, the following series of operations occur:

1. As Solace2 starts up, its primary virtual router uses a VRRP advertisement of Primary-Reconcile, and its backup virtual router uses a VRRP advertisement of Backup-Reconcile. These advertisements indicate that the virtual routers are initializing, but are not yet ready to take on client activity.

**Note:**If Guaranteed Messaging is enabled for the redundant pair, the ADB on Solace1 transfers its entire contents to the ADB on Solace2 over the ADB links.

If auto revert is enabled (b), then the following series of operations also occur:

**Note:**By default auto revert is not enabled, which is preferable for most situations. However, you may want to enable auto revert so that clients revert to their primary appliance as soon as it comes back online when all of the following conditions are met:

* + The appliances are deployed in an active/active redundancy model.
  + The redundant pair has been deliberately “overbooked” beyond the capacity of a single appliance, so that service will degrade when one of the appliances is off-line.
  + The disruption caused by the momentary service outage during auto revert is considered less harmful than continuing to run in an overbooked state until service can be manually reverted during a service window.

1. Once has fully started, its primary virtual router increases the priority of its VRRP advertisements to Assert-Activity. This priority level causes the backup virtual router of its mate system to relinquish activity. Solace2 then sends a gratuitous ARP to let the other network services know it is taking over the IP address 192.168.181.133/19 for client connections.
2. Solace1 disconnects any TCP sessions that it had established for clients receiving service from IP address 192.168.181.133/19.

**Note:**If Guaranteed Messaging is enabled for the redundant pair, on redundancy switchover the following steps also occur:

* + The old active appliance unmounts the disk partitions.
  + The newly active appliance mounts both disk partitions as “read only” then probes the AdKey. If no conflict is detected, the disk partitions are mounted as described in [SAN Behavior for Guaranteed Messaging Through Active/Standby Redundancy](https://docs.solace.com/Features/Redundancy-and-Fault-Tolerance.htm#redundancy_and_fault_tolerance_344917737_295043) (that is, the active LUN partition is mounted as “read/write”, and the standby LUN partition is mounted as “read only”).
  + Garbage collection may be done in the background where messages spooled on the newly active partition have already been delivered or administratively deleted (since it was previously read-only so file cleanup was not possible).

1. The primary virtual router for Solace1 advertises a local priority level of Active, and the backup virtual router advertises a local priority level of Backup.

**Note:**Through the **auto-revert** Redundancy CONFIG command, the network administrator can configure whether the standby appliance in a redundant pair should automatically revert back at this point to the primary when the primary comes back online after a service outage.

1. Solace2 accepts new TCP connections to IP address 192.168.181.133/19 and delivers messages to, and accepts messages from, clients using this IP address.

## Subscription Export Considerations for Message VPNs

As discussed in [Working with Message VPNs](https://docs.solace.com/Features/Working-with-Message-VPNs.htm), each Message VPN has a topic subscription export policy associated with it and the default mode is set to not exporting subscriptions.

If a Message VPN is not exporting topic subscriptions, then clients connected to one appliance in an active/active appliance pair do not receive messages from clients connected to another appliance in the pair. However, if a failure occurs such that the clients from both virtual routers end up connected to the same appliance, then messages can be passed between clients on different virtual routers.

This is because the topic subscriptions are then maintained within the scope of the appliance, since the Message VPN export policy does not apply between virtual routers hosted on the same appliance.

To avoid this scenario, either:

* Deploy the redundant pair in an active/standby configuration.
* For Message VPNs not exporting topic subscriptions all clients of the Message VPN should connect to the same virtual router so that the Message VPN is effectively deployed in an active/standby configuration.

## VPN Bridging and Fault Tolerance

For details on how to establish Message VPN bridge connections to remote appliances when those remote appliances have been deployed in a redundant configuration to provide fault tolerance, refer to [Bridging to Remote Appliances That Use Redundancy](https://docs.solace.com/Configuring-and-Managing-Routers/VPN-Bridging-to-Remote-Appliances.htm).

# VMR Redundancy and Fault Tolerance

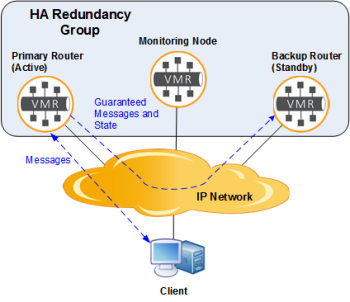
You can deploy Solace Virtual Message Routers (VMRs) in high-availability (HA) redundancy groups for fault tolerance. HA redundancy provides 1:1 router sparing to increase overall service availability. If one of the routers fails or is taken out of service, the other router automatically takes over and provides service to the clients that were previously served by the now-out-of-service router.

## VMR HA Redundancy Model

Solace VMRs support an active/standby redundancy model. With this model, a primary router provides messaging services to clients, while a backup router waits in standby mode—it only provides service should the primary router fail. A third router acts as a monitoring node, to act as a tie-breaker and prevent split-brain scenarios that would otherwise cause both the primary and backup router to become active simultaneously.

The VMR HA redundancy model supports both Direct and Guaranteed Messaging clients.

Active/Standby Redundancy Model



In the active/standby model:

* All clients connect to the active router in the redundancy group (typically the primary router).
* The other router (typically the backup router) acts only as a standby. While one primary router is active, clients cannot connect to the standby router, and no messaging traffic can flow through the standby.
* The active router uses the IP network to automatically propagate all Guaranteed messages and Guaranteed Messaging state to the standby router.
* If the primary router fails for any reason, the backup router will become active and provide messaging services to the clients.
* When the primary router comes back online, the backup router continues to provide service to the clients, while it automatically resynchronizes its Guaranteed messages and the Guaranteed Messaging state with the primary router. Once resynchronized, the backup router can either continue to provide service to the clients (the default behavior for the VMR), or it can transfer activity back to the primary router (if you have configured the VMRs to auto-revert to the primary).

### Synchronizing Router Configurations

The primary and backup VMRs in a VMR HA redundancy group must have the same system and Message VPN level configurations, and this configuration must remain in sync while the routers are running. The Config-Sync facility is used to automatically synchronize their configurations.

The mate link service is also used for the synchronization of Guaranteed Messages and message state between the primary and backup router. The mate link service uses the Management VRF for the messaging traffic to/from the clients and the Guaranteed Messaging data synchronization between the active and standby routers is all done over the same physical Ethernet interface.

## Failover Mechanism

The VMR supports host list failover mechanism through which client connections are transferred from one message routing node to another upon the node failure. This mechanism uses lists of IP addresses, or corresponding DNS names, of both the primary and backup router. The primary and backup routers have different IP addresses at all time, but only one of them is active and accepts connections. Connecting clients know these IP addresses, and the clients (not VMRs of the HA group) handle reconnecting from one IP address to the other.

The client API is responsible for connecting to whichever router is active in the HA redundancy group. This kind of configuration would be common in cloud environments.

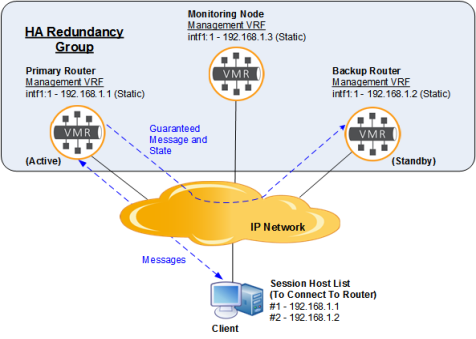
**Note:**Your client APIs and VPN bridge connections must be configured with the host lists for the primary and backup routers in the HA redundancy group. Once configured, if the primary router becomes unavailable for any reason, the backup router will take over activity, and the client APIs and Message VPN bridges will reconnect to the newly active router without impacting the client applications.

### Client Host Lists

This failover mechanism relies on client applications using configured host lists to connect and reconnect to valid hosts for the HA redundancy group. For Solace APIsʼ use of host lists, refer to [Host](https://docs.solace.com/Solace-Messaging-APIs/Session-Props-Required.htm#Creating_ent_sessions_sess_props_host)

When using host lists, the active VMR will accept client connections on the Management VRF static IP address, and the standby VMR will reject such connection requests. Primary and backup VMR IP interfaces are ignored, and client connections to these interfaces are rejected too.

Client Using the Host List to Connect



Notice that the client application’s host list is configured with two IP addresses:

* the primary router’s static IP address for the Management VRF (or corresponding DNS name)
* the backup router’s static IP address for the Management VRF (or corresponding DNS name)

## VMR IP Addressing

VMRs rely on a Message Backbone service for all messaging traffic to and from clients, and on a Management service for management traffic. Both Message Backbone and Management services share the Management Virtual Routing and Forwarding (VRF) object that is used to connect to the IP network.

By default, VMRs have a single network interface that is mapped to the Management VRF. This is different from Solace appliances that have two separate network interfaces.

By default the VMR network interface is configured as a DHCP client. However, to use VMR Redundancy, each VMR instance in the HA redundancy group, including the monitoring node, must have a unique static IP address associated with the Management VRF, and this IP address must be in the same subnet and statically configured (that is, DHCP is not supported). Using static IP addresses in the HA redundancy group is a prerequisite for the VMR redundancy functionality.

## Failure Detection

All three nodes in the HA redundancy group—primary, backup, and monitoring—continuously communicate with each other using a protocol that runs over the static IP interfaces in the Management VRF, and, by default, uses ports 8300, 8301, and 8302.

If the active router in the group becomes unreachable for any reason, and neither the monitoring node nor the backup router can see the active router, but they can still see each other, then the backup router will take activity, and provide messaging services to the clients.

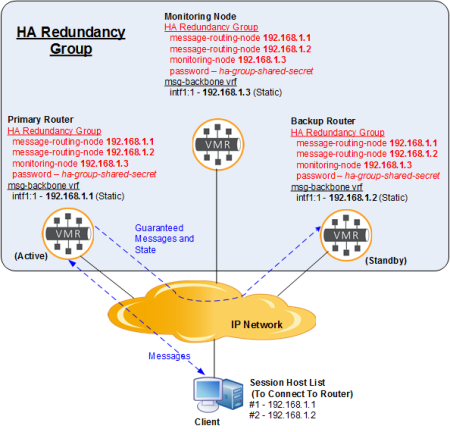
Similarly, if the active router loses connectivity with both the standby router and monitoring node, the active router will give up activity, to eliminate the possibility it might be operating in a split-brain fashion. This implies that for a router to take (or keep) activity and provide service, it must be able to communicate with at least one other node in the group—either the mate router and/or the monitoring node.

For redundancy to function properly, all three nodes in the group need to be configured with the Management VRF static IP addresses of the other nodes in the group and the assigned role of each of the nodes (message-routing-node, or monitoring-node).

All three nodes in the HA redundancy group also need to be configured with the same HA redundancy group password as a security mechanism to ensure that only the nodes in the group can communicate with each other, and that other hosts on the network cannot impersonate the routers or attempt to join the HA redundancy group.

The following figure shows a correct failure detection configuration.

Configuring for Failure Detection



Notice above how all three nodes in the HA redundancy group are configured with static IP addresses.

## Fail-Over Sequence

If the active router goes offline, a failure is detected within a HA redundancy group. Subsequently, a failover occurs in the following sequence:

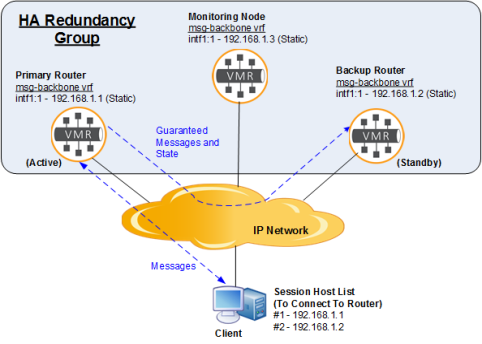
1. The backup router takes over messaging activity.
2. Once the failed primary router comes back on-line, it resynchronizes to match the currently active backup router.
3. The primary router takes on the “Standby” role, or, if auto-revert is enabled, messaging activity automatically switches back to the primary router.

The diagrams below show a failover sequence in detail.

### Normal Operation

The diagram below shows a typical HA redundancy group under normal operation, when both the primary and backup routers are online and capable of providing service to the clients. This group is configured with the host lists for client connections (192.168.1.1 and 192.168.1.2).

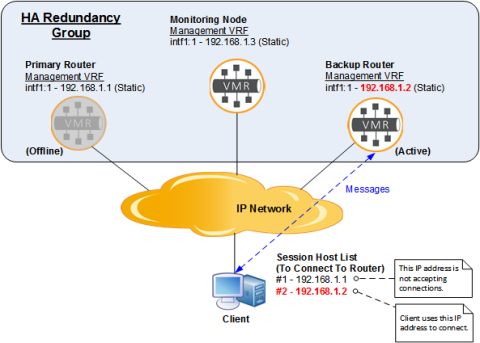
Typical Operation with host list



### Taking Over Activity

If the active router fails or is taken offline, the backup router and monitoring node will detect the failure, and the backup router will take over activity. When the backup router takes activity, it will start accepting connections on the Management VRF static IP address. The connecting clients will traverse their host lists and connect to the backup router using the backup router’s static IP address (on the Management VRF).

The diagram below shows the failover mechanism. Notice how the client uses the backup router IP address 192.168.1.2 after the backup router took over messaging activity.

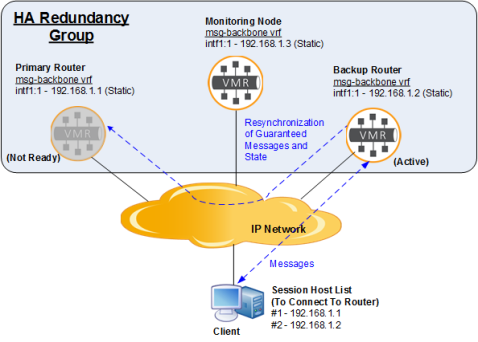


### Resynchronization

Once the failed router comes back on-line, it will use the mate link VRF to resynchronize its message-spool contents to match the active router. This process may take a few seconds if the differences in the message-spool contents are minimal between the two routers, but it may also take several hours if the failed router was offline for a long time, and large quantities of data have been spooled on the active router.

Resynchronization is not a service-affecting operation, and the backup router continues to service connected clients while the resynchronization is taking place. However, the primary router is not able to provide service to clients during the resynchronization process. (Note that when disk resynchronization occurs, the redundancy status will be displayed as down.)

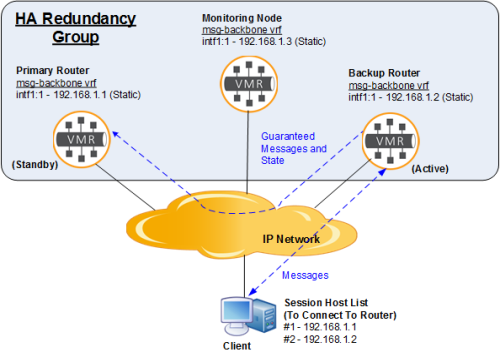
The following diagram shows the resynchronization process.



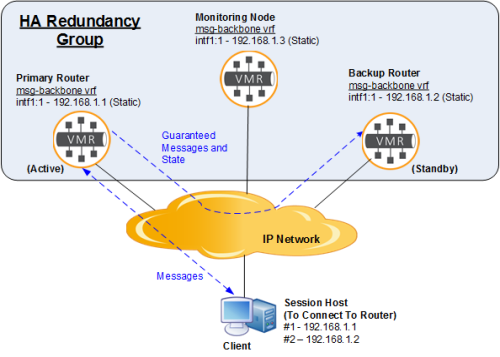
### Taking on the Standby Role

Once resynchronization has completed, the primary router takes on the Standby role, and is available to provide service to clients should the backup router go offline for any reason.

The following diagram shows this state of the HA redundancy group.



If auto-revert is enabled on the routers of this group, then activity will automatically switch back to the primary router. The diagram below shows this scenario. You’ll notice that this diagram is the same as that showing the normal operating state.



# Data Center Replication

The Solace Replication facility provides a data center redundancy and disaster recovery solution for the Solace Message platform. It allows mission‑critical applications to continue to function during a major service outage to a data center.

**Note:**To implement Replication, Config-Sync must be enabled for each router in a Replicated site. The Config‑Sync facility provides automatic synchronization of Message VPN configuration parameters that are required to match between Replicated routers. For more information, see [Config-Sync](https://docs.solace.com/Features/Config-Sync.htm). For redundant appliances that are handling Guaranteed Messaging, durable endpoint information such as queue and topic endpoints, topic-to-queue mappings and queue options are automatically propagated whether Config-Sync is enabled or not.

When the Replication facility is implemented, Guaranteed messages that are published to Message VPNs with Replication active state at one data center are automatically propagated across Replication bridge links to matching Message VPNs with Replication standby state at another data center located in a separate geographic location. In addition, if the messages are part of a local or XA transaction, the transaction is propagated to the standby site and transaction semantics are respected. For example, rolling back a transaction would roll it back on both sites. Preparing an XA transaction would prepare the transaction on both sites. In a scenario where a major service outage occurs for one Replicated data center (that is, one Replication site), a service fail-over to the operational mate Replication site can be performed.

A typical customer deployment model for replicated data center infrastructure is to have a pair of Replication sites located some distance apart (perhaps 50 or 100 miles). These sites are considered Replication mates. The main or primary site will use a high-availability (HA) pair of routers to protect against a local failure of a router or equipment in that site. The secondary or standby site may have a single router or an HA pair of routers. The primary site provides service unless there is a failure of the primary site. If the primary site fails, service is failed over to the backup site. Once the primary site is restored, service can be failed back to the primary site.

The fail-over of a Replication site is often not an action that can just be performed at the messaging layer—typically there are servers, critical applications, and other infrastructure that must be switched as part of the fail-over. Therefore the fail-over is a co-ordinated operation that must be performed by network administrators. It does not happen automatically.

**Note:**Replication is not a replacement for HA router redundancy within a data center. Router redundancy provides automatic protection against a single router failure. Replication protects against more catastrophic events in the data center and requires manual intervention to effect a fail over.

## How Replication Works

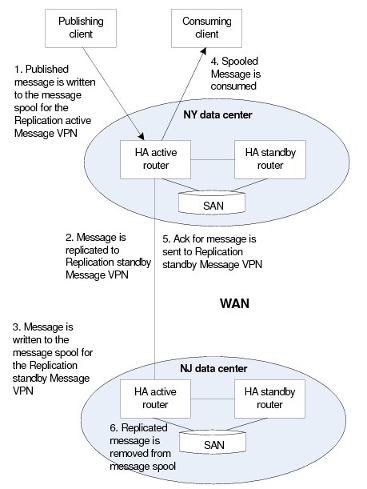
This section provides a simplified description of how Replication works. All diagrams and descriptions are for the synchronous mode of Replication, where messages are stored on both sites before responding to the client. For more details of synchronous Replication, see [Synchronous and Asynchronous Message Replication](https://docs.solace.com/Features/Data-Center-Replication.htm#Synchron).

The details of how Replication works depends on whether the Guaranteed messages are part of a transaction or not. There are also differences in how local and XA transactions are handled.

### Non-Transacted Messages

The figure below provides an example of how Replication uses the Guaranteed Messaging capabilities of Solace routers to propagate non-transacted Guaranteed messages published to a topic that you want to replicate.

Non-Transacted Message Replication Process



Here is how it works when publishing and consuming non-transacted messages:

1. The client publishes a message that matches a Replicated topic subscription that is configured for the Message VPN with an active Replication state. The message is persisted on the Solace router.
2. A copy of the message is sent over a Replication bridge (named #MSGVPN\_REPLICATION\_BRIDGE) that is automatically created when Replication is enabled for a Message VPN. This bridge links the Message VPN with the active Replication state and its mate Message VPN of the same name that has a standby Replication state.

**Note:**For added security, Transport Layer Security (TLS) / Secure Sockets Layer (SSL) encryption can be used on Message VPN Replication bridges and Replication Config-Sync bridges. For information on using SSL encryption on Replication Config-Sync bridges and Message VPN replication bridges, refer to [Managing TLS/SSL Service](https://docs.solace.com/Configuring-and-Managing-Routers/Managing-TLS-SSL-Service.htm).

1. The message is also persisted on the Replication mate.
2. When the message is consumed from an endpoint on the Replication active Message VPN, it is removed from the persistent store.
3. An acknowledgment (ack) is propagated to the Replication standby Message VPN that the message has been delivered to the consumer.
4. The message is then also removed from the persistent store of the router the Replication standby Message VPN resides on.

### Local Transactions

With a local transaction, the messages and acknowledgments are propagated to the Replication mate as a part of an internal XA transaction.

This is how it works for when publishing messages in a local transaction:

1. The client opens a local transaction and publishes multiple messages that matches a Replicated topic subscription. The messages are persisted on the active site.
2. The client commits the transaction. An internal XA transaction containing the messages is created.
3. Copies of the messages are sent over the Replication bridge to standby site as part of the internal XA transaction and persisted on the standby site.
4. The internal XA transaction is committed on the standby site. The messages are inserted into the destination endpoint(s) standby site.
5. The messages are inserted into the destination endpoints on the active site. The internal XA transaction completes. The commit response is sent back to the client.

This is how it works when consuming messages in a local transaction:

1. The client opens a local transaction and consumes multiple messages that have been replicated (that is, when they were published, they matched a replicated topic) from an endpoint.
2. The client commits the transaction. An internal XA transaction containing an acknowledgment of the consumed messages is created.
3. The acknowledgment is sent over the Replication bridge to standby site as part of the internal XA transaction.
4. The internal XA transaction is committed on the standby site. The consumed messages are removed from the endpoint on the standby site.
5. The internal XA transaction completes. The consumed messages are removed from the endpoint on the active site. The commit response is sent back to the client.

**Note:**It is possible for a single local transaction to include both the publishing and consuming of messages. For simplicity, the above transaction example only showed the publishing and consuming of messages.

### XA Transactions

With an XA transaction, the XA transaction is extended to include the standby site in the distributed transaction.

This is how it works when publishing messages in an XA transaction:

1. The client starts an XA transaction and publishes multiple messages that match a Replicated topic pattern. The messages are persisted on the active site.
2. Copies of the messages are sent immediately over the Replication bridge to the standby Replication site. The messages are persisted on the standby site.
3. The client ends the XA transaction. The transaction state is updated on the active site.
4. The client prepares the XA transaction. The active site checks if the transaction can succeed. If it can, the prepare request is sent to the standby site, where it is checked. If it can also succeed on the standby site, the XA transaction state is updated on both active and standby sites. The prepare response is sent back to the client.
5. The client commits the XA transaction. The messages are inserted into the destination endpoints on the active site. The commit request is sent to the standby site and the messages are inserted into the destination endpoints on the standby site.
6. The commit response is sent back to the client.
7. The XA transaction is completed.

This is how it works when consuming messages in an XA transaction:

1. The client starts an XA transaction and consumes multiple messages that has been replicated (that is, when they were published, they matched a replicated topic).
2. The client ends the XA transaction. The transaction state is updated with the acknowledgment on the active site.
3. The client prepares the XA transaction. The active site checks if the transaction can succeed. If it can, the prepare request is sent to the standby site, where it is checked. If it can also succeed on the standby site, the XA transaction state is updated on both active and standby sites. The prepare response is sent back to the client.
4. The client commits the XA transaction. The messages are removed from endpoint on the active site. The commit request is sent to the standby site and the messages are removed from the endpoint on the standby site.
5. The commit response is sent back to the client.
6. The XA transaction is completed.

**Note:**It is possible for an XA transaction to include both the publishing and consuming of messages. For simplicity, the above transaction example only showed the publishing and consuming of messages.

## Replicated Topics

To indicate which messages should be replicated between the active and standby site, you must configure a replicated topic subscription for a Message VPN. This topic pattern can be a topic subscription or a queue name subscription (a subscription for a queue is #P2P/QUE/<queueName>). If a published message matches both a replicated topic and an endpoint on the active site, then the message is replicated to the standby site.

You can also specify a topic subscription exception by adding a leading "!" character to the topic subscription. In this case, messages whose topics match the configured topic subscription exception are not copied to the corresponding Message VPN on the replication site. For more information, see [Configuring Replicated Topics](https://docs.solace.com/Configuring-and-Managing-Routers/Replication-VPN-Level-Settings.htm#Config-Rep-Topics).

### Transactions

Which messages published within a transaction (local and XA) are replicated is determined by the same replicated topic subscriptions as non-transacted messages. Only those messages that match a replicated topic subscription are part of the extended transaction to the mate Replication site. Because of this, messages published within a transaction should all match a replicated topic subscription, otherwise parts of the transaction will not be sent to the mate replication site. In the event of a fail-over, the transaction contents will not match.

For example, if an XA transaction contains 10 messages, six of which match a replicated topic and four of which do not, only the six matching messages become part of the transaction interactions with the standby site. The active site will have 10 messages and the standby site will have only six when the transaction is committed.

## Synchronous and Asynchronous Message Replication

Replication can be performed in one of two modes:

* Synchronous Replication—A message or transaction is not considered persisted until it has been confirmed to be stored on both the active and standby sites. While providing a greater guarantee that the published message or transaction will not be lost in an uncontrolled fail-over, synchronous Replication incurs a performance penalty for the publisher, especially blocking publishers. This is because the publisher has to wait for communication between the two Replication sites to complete before publishing the next message or transaction. In those use cases, the maximum message rate of a single publisher is limited by the round-trip time and available bandwidth between the active and the standby sites.
* Asynchronous Replication—A message or transaction is considered persisted once it has been stored on the active site and put into the Replication queue (#MSGVPN\_REPLICATION\_DATA\_QUEUE) on the active site to be delivered to the standby site. This type of Replication gives improved performance, since it does not have to wait for communication with the standby site to complete, but during a uncontrolled failure of the active site there is a chance that a message or transaction which the client has been told has completed has not been delivered to the standby site. In this uncontrolled case, messages or transactions may be lost or duplicated.

When using non-transacted messages, replicated topics are configured to use either synchronous or asynchronous Replication mode. Messages published to a replicated topic that is configured as synchronous are not acknowledged until the message is stored on both the active and standby site. Messages published to a asynchronous topics are acknowledged once the message is stored on the active site and put into the Replication queue for delivery to the standby site. If a message matches both a synchronous and an asynchronous replicated topic, synchronous Replication is used.

When using transactions, the Replication mode is set at the message VPN level. All local and XA transactions in the message VPN use the same Replication mode. Synchronous transactions must be stored on the standby site before responding to the client. Asynchronous transactions only need to be stored in the Replication queue. The Replication mode of the replicated topics is ignored when using transactions.

### Downgrading to Asynchronous When Sync Ineligible

When using synchronous Replication, if the Replication bridge connection is very slow or goes down, processing of replicated messages and transactions will stop. To allow message and transactions to continue to be processed, by default the Message VPN will switch to asynchronous Replication when the standby site is unreachable or slow. The message VPN is considered synchronous ineligible and the Replication service will be considered degraded. All messages published to synchronous replicated topics and synchronous transactions will be treated using asynchronous handling.

This behavior helps to avoid a business interruption when the standby site is temporarily unreachable.

The Replication service is considered degraded when any of the following occurs:

* The Replication bridge is disconnected from the Replication queue.
* A message can be put in the Replication queue but cannot be sent immediately (streamed) to the Replication mate because the link is slow or the replication queue is backed up. This state must persist for 30 seconds before replication is considered degraded.
* A message put in the Replication queue can be sent immediately but is not acknowledged by the replication mate within 30 seconds (and is therefore retransmitted). This state must persist for 30 seconds before Replication is considered degraded.

The router generates notification events whenever it transitions between synchronous eligible and ineligible (degraded or not degraded).

### Preventing Downgrade to Asynchronous Replication

If the risks associated with asynchronous replication are not acceptable, it is possible to ensure synchronous replication is always strictly enforced. To maintain a synchronous replication mode, you can enable the reject-msg-when-sync-ineligible option for a Message VPN. With this enabled, synchronous replicated messages or transactions are rejected if they cannot be successfully stored on both the active and standby site.

## Replication Queue Full

If the standby site is down or unreachable for an extended amount of time, the Replication queue (#MSGVPN\_REPLICATION\_DATA\_QUEUE) can eventually become full. It is assumed that replication is a high-priority service, so the replication queue size should be set to a large value. A system administrator can adjust the quota of a Replication queue. Like all queues, the Replication queue has threshold event logs as it fills up, which warns you that the queue may become full and action should be taken.

By default, if the Replication queue becomes full, publishing to replicated topics or using replicated transactions is rejected. No new replicated messages or transactions can be processed. To restore non-replicated service, the administrator has the option of disabling this behavior (by disabling the reject-msg-to-sender-on-discard feature on the replication queue). With this configuration, messages and transactions are not replicated, but local service can be provided. If connectivity is restored and the replication queue is drained, replication will start again. At this point, the administrator should re-enable reject-msg-to-sender-on-discard on the Replication queue.

### Pruning Replication Queues

When the standby site is down and the active site is storing messages and transactions in the replication queue, there is an optimization to help prevent the replication queue from becoming full. Replicated messages that are waiting to be sent to the standby site do not need to be sent if the message has subsequently been consumed on the active site. Since the message is no longer present on the active site, there is no need to send it to the standby site for replication purposes. In this case, those messages are removed, or pruned, from the replication queue, helping to prevent it from becoming full.

## Switching Service from Site to Site

There are two aspects to switching replication service from site to site. The router replication service must be manually switched and the clients must be configured to automatically connect to the newly active site.

### Router Service Switch

In the event of a failure at the active replication site, a network operator can change the replication state of the Message VPNs on routers at the standby replication site from standby to active. The messaging clients can then reconnect to the Message VPNs of the same name on the newly active replication site, and the processing of guaranteed messages and transactions can continue.

There is a small possibility that under high traffic rates or unfortunate timing of a fail-over to the standby site, some messages could be duplicated following a fail-over. It is recommended that applications that cannot tolerate duplicate message delivery under any scenario should implement application-layer mechanisms (for example, globally-unique message IDs) to detect duplicate message delivery.

When using transactions (local and XA), not all transaction states on the active site are mirrored on the replication standby site. Only those states that are necessary to preserve the transactional behaviors on a fail-over are preserved. For example, XA transactions in the ACTIVE state are not mirrored. In the event of a replication fail-over, an application server transaction manager is expected to detect this and will handle it properly.

For procedural information on how to perform replication fail-overs, refer to [Switching Replication Service from Site to Site](https://docs.solace.com/System-and-Software-Maintenance/Switching-Replication-Service.htm).

### Client Configuration

The host list feature of the Solace messaging APIs provides messaging clients with ability to automatically switch to a newly active replication site. The host list provides the IP addresses or hostnames of the routers in both the replication sites. Typically, the host list contains the primary site, followed by the backup site. When its connection to the primary site drops or is changed to standby state, the client automatically attempts to connect to the backup site, which is the next host in the host list.

**Note:**When a Message VPN that has a replication active state is switched to replication standby, all active clients are disconnected.

## 

## Replication Fail-Over Types

There are four types of replication fail-overs:

* Controlled Fail-over
* Uncontrolled Fail-over, Short-Term Outage
* Uncontrolled Fail-over, Long-Term Outage
* Uncontrolled Fail-over, Complete Failure

In a uncontrolled fail-over, there are some special considerations when failing back to the originally active site.

### Controlled Fail-over

This is a fail-over triggered by an administrator following the documented procedure. In this case, both sites are in service and can communicate with each other.

### Uncontrolled Fail-Over, Short-Term Outage

In this fail-over type, the active site is out-of-service or isolated from the clients and the standby site for a short duration (minutes, hours). In this case, the standby site can be made active and there is enough capacity in the replication queue (#MSGVPN\_REPLICATION\_DATA\_QUEUE) to store all the replicated messages and transactions (using asynchronous replication) until service or connectivity to the failed site is restored. The recovered site becomes the standby, drains the replication queue (#MSGVPN\_REPLICATION\_DATA\_QUEUE) from the active site, and full replication behavior is restored.

### Uncontrolled Fail-over, Long-Term Outage

In this fail-over type, the active site is out-of-service or isolated for a long duration (days, weeks). In this case, the standby site can be made active, but the replication queue (#MSGVPN\_REPLICATION\_DATA\_QUEUE) may become full. When the replication queue fills up, reject-msg-to-sender-on-discard can be disabled on the replication queue to provide non-replicated service on the newly active site. Once the failed site is restored, the recovered site becomes standby and drains the replication queue from the active site. Messages and transactions that did not fit in the replication queue were not replicated.

### Uncontrolled Fail-over, Complete Failure

In this fail-over type, the active site is out of service and cannot be recovered (for example the entire router has been replaced with a new one). In this case, the standby site can be made active while waiting for a replacement and replicated messages and transactions are stored in the replication queue. Depending on how long it takes to source the replacement, the short-term or long-term outage behaviors will apply.

### Failing-Back to the Originally Active Site

After an uncontrolled fail-over and once the failed site has been restored, a fail back to the originally active site may be desirable. This is especially true if the backup site has lower capacity or fault protection than the primary site. When failing back after a controlled fail-over, there are no special considerations. However, after an uncontrolled fail-over, there are some considerations, depending on the fail-over type:

* **Uncontrolled Fail-over, Short-Term Outage**—When failing back replicated messages or transactions that were in progress when the fail-over occurred and have not been consumed are at risk of loss or duplication.
* **Uncontrolled Fail-over, Long-Term Outage**—When failing back replicated messages or transactions that were in progress when the fail-over occurred and have not been consumed are at risk of loss or duplication. In addition, only those messages and transactions that made it into the replication queue will be available.
* **Uncontrolled Fail-over, Complete Failure**—Depending on how long it takes to replace the failed replication site, either the short- or long-term failure considerations will apply. In addition, the replacement router will have no historical data, so replicated messages from before the fail-over that have not been consumed would be lost.

In all cases, the risk of loss or duplication when failing back to the originally active site can be eliminated if all replicated messages published before the fail-over and were in-progress during the fail-over have been consumed on the newly active site. In other words, if you leave the backup replication site active long enough to be certain that all messages published prior to the fail-over have been consumed.

## Allowing Clients to Connect to Standby Sites

By default, clients consume replicated messages (as part of a transaction or not) from Message VPNs with replication active states. If a client attempts to connect to a standby replication site, the connect attempt is rejected. However, it is possible to allow clients to connect and consume messages from endpoints on a replication active Message VPN and its mate replication standby Message VPN.

An example of this type of deployment would be one using a replay server. A replay server must consume all messages so they can be later replayed. For the purposes of disaster recovery, an instance of the replay server must run on both the active and the standby sites. In the event of a fail-over, the replay server on the standby site will have consumed the same set of messages as the server on the active site and therefore would be able to provide the same replay service.

To enable a replication deployment like this, some extra configuration is required:

* the allow-clients-when-standby option must be enabled for all client profiles that are used by clients allowed to connect on to the standby site.
* the propagation of consumer acks must be disabled for each endpoint that these standby clients will consume messages from. Ack propagation is the mechanism by which the consuming of messages on the active site is signaled to the standby site. Since messages are being consumed on both sites, this signaling needs to be disabled. The configuration is the same whether or not the messages are being consumed in a transaction or not.

Although clients are allowed to connect to the standby site, their capabilities are limited. They are not allowed to publish guaranteed messages, create endpoints, or modify existing endpoints in any way. They are, however, allowed to publish direct messages and add direct subscriptions. This allows for direct message communication between clients connected to the standby site for co-ordination purposes.

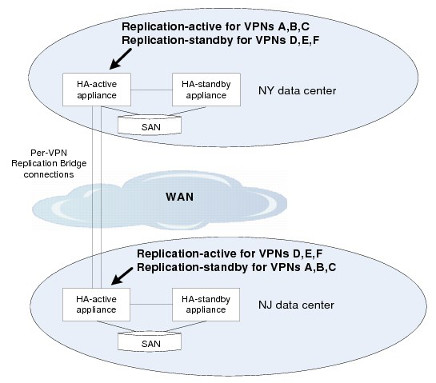
**Note:**Host lists should not be used for a standby client. In this scenario, there should be a client for each site and the clients should only connect to their specific site, so a list of hosts is not needed.

## Providing Service at Both Replication Sites

A possible customer deployment model for replicated data center infrastructure is to have a pair of replication sites located some distance apart (perhaps 50 or 100 miles). Each replication site will use high-availability (HA) pairs of redundant routers: one replication site can actively service one set of Message VPNs, while the other replication site actively services another set of Message VPNs. That is, the HA pairs at one site have Message VPNs with replication active states and other Message VPNs with replication standby states, and the HA pairs at the other replication site have the same Message VPNs but with the opposite replication states.

This allows both replication sites to provide messaging services to separate sets of clients. However, they should maintain enough capacity to provide service to all clients of the two replication sites if there is a planned or unplanned service outage at one of the sites.

The figure below provides a simplified example, where at the New York data center the Message VPNs A, B, C have an replication active state, and the Message VPNs D, E, F have a replication standby state; whereas at the New Jersey data center, the Message VPNs D, E, F are active and the Message VPNs A, B, C are standby.



**Note:**Replication can also be deployed with single routers at one or both sites, rather than redundant pairs. However this configuration reduces the fault tolerance of the solution.

# Config-Sync

A Solace router (VMR or a physical appliance) maintains two separate types of configuration information:

* system-level configuration information
* configuration information for each Message VPN

If you are deploying routers in high-availability (HA) redundant pairs or in replicated data centers, the use of Config-Sync will automatically propagate changes in both types of configurations between two routers, so that they remain in sync.

In both kinds of deployments, Config‑Sync relies on Guaranteed Messaging that must be configured and enabled on each router involved.

**Note:**

* For redundant appliances that are handling Guaranteed Messaging, durable endpoint information (for example, queue and topic endpoints, topic-to-queue mappings, queue options) is automatically propagated whether Config-Sync is enabled or not.
* Some configuration parameters that are considered unique to a router or related to licensing or security are not automatically propagated between mates when Config-Sync is enabled. Therefore, they must be manually configured for each router. For more information, see [Properties That Are Not Synchronized](https://docs.solace.com/Configuring-and-Managing-Routers/Using-Config-Sync.htm#Properti).

## Synchronizing High Availability Deployments

When HA redundancy is used, the configurations of both routers in an HA pair must be the same and remain in sync while the routers are running. The Config-Sync functionality automatically synchronizes configuration of such pairs.

In HA deployments Config-Sync must be instructed which of the paired routers is the master for both the system-level configuration and for each Message VPN-level configuration. This is done by asserting the master configuration over its mate. For each HA pair, the system-level configuration must be asserted, and each Message VPN-level configuration must also be asserted.

## Synchronizing Replication Deployments

Config-Sync must be used in replication deployments to synchronize Message VPN-level configuration parameters between replication mates.

When replication is enabled on a router, a replication Config-Sync bridge is automatically created to link a replicated router with its mate, and Message VPNs are automatically synchronized.

For replication deployments, the master Message VPN is always the one with the “Replication Active” state, and any configuration changes made to it are automatically pushed to its corresponding Message VPN in the “Replication Standby” state. Configuration changes are not allowed on the Message VPN in the “Replication Standby” state.

Notice that the Message VPN is also in an HA redundant configuration, its changes are also pushed to the redundant mate routers.

## System Resources Used by Config-Sync

The Config-Sync facility consumes additional system resources on a Solace router, specifically it creates and uses:

* one Message VPN
* four clients itself and two clients for each synchronized Message VPN
* nine topic subscriptions for itself and six topic subscriptions for each synchronized Message VPN
* three queues for itself and two queues for each synchronized Message VPN

It is important to count in the resources listed above as they are limited by system-level limits. See [Guaranteed Message Queuing Limits](https://docs.solace.com/Features/G-Msg-Queueing-Limits.htm).

All resources created and used by Config-Sync are all named as #config-sync, except for its clients, which are named #config-sync/<router-name>. Any of the resources created and used by Config-Sync cannot be deleted or directly edited.