# Overview

The routing rules are a set of rules that are applied when a call is received, and help determine how the resulting outbound call is handled.

The routing rules file can be used, together with the PSTN configuration file (see Chapter 5), to define the physical resources used by the Gateway. For example, you might create an element in the PSTN configuration file and then use that element in the routing rules file to write flexible rules that determine how to route calls based on a number of factors.

This chapter contains the following topics:

## What are routing rules?

When an incoming call is received, the routing rules are fetched by the Gateway to a web server. The rules obtained, which are based on the characteristics of the call, are then applied to the incoming call in order to find an outbound location. A routing rule is, by definition, a logical expression that processes an incoming call (regardless of whether it originates from the PSTN or the IP network) and, once the rules' conditions are matched, produces the parameters for the associated outbound call.

The routing rules file is located here:

- *[GATEWAY_HOME]/config/routing-rules.xml*

  where *[GATEWAY_HOME]* is the root folder of the installation (for example, C:\Program Files\Netborder\Express\Gateway*config\routing-rules.xml*).

The location of the routing rules file is specified in the *gw_sangoma.properties* file by the *Netborder.gw.routingRulesUrl* parameter. Since the Gateway embeds a web server, the routing rules are by default obtained from the Gateway web server. However, by modifying the URL location property, any web server can be used.

Out of the box, the Routing Engine is configured to route all incoming PSTN calls to the SIP user agent at the URI specified at installation time. It is also configured to support the calling properties as described in the preceding chapter.

If you are running your target application (either a SIP application or a SIP proxy) at the location specified at installation, and you are using the basic Gateway functionality, **you do not need to modify the routing configuration**.

However, the routing rules file provides powerful customization capabilities, which you may wish to explore.

## Modifying routing rules

The Routing Engine provides application developers with the ability to customize the way calls are routed.

The routing rules file can be modified manually in one of two ways:

- By using a text or XML editor of your choice and modifying the *[GATEWAY_HOME]\config\routing-rules.xml* file (where *[GATEWAY_HOME]* is the root folder of the installation). The Gateway must be restarted for changes to take effect.
- By using the Gateway Web Interface and clicking on the **Routing Rules** tab at the top of the page. Once the file modifications have been saved to disk, the Gateway fetches the new rules on the next incoming call and uses them to make routing decisions.

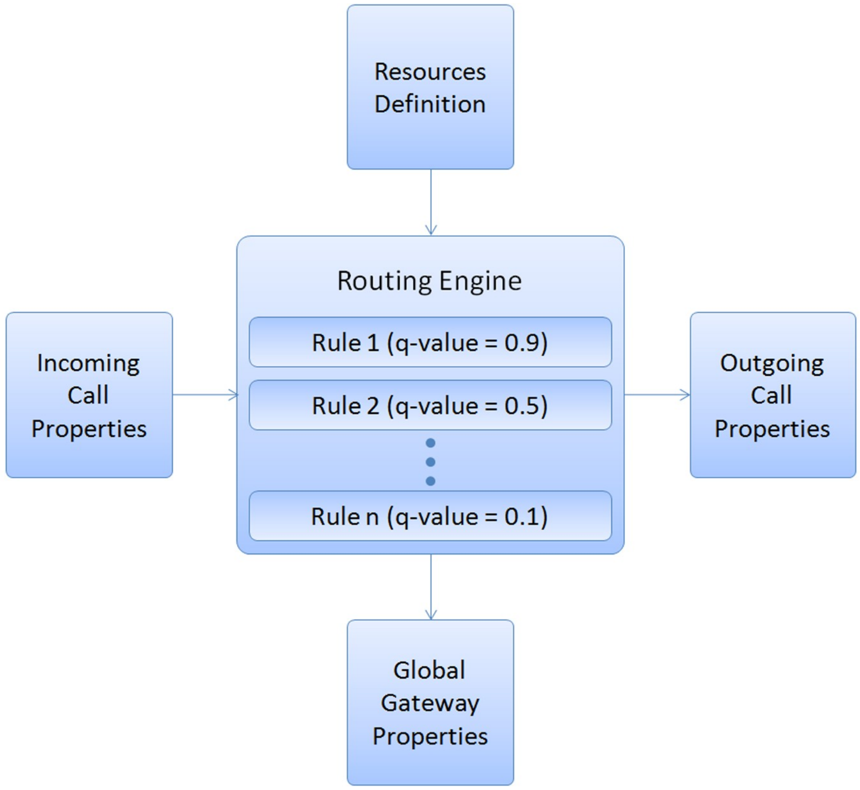
### Routing Engine

When the Routing Engine receives a routing request with some incoming call properties, it does the following:

- fetches the routing rules
- applies the routing rules to the request based on the resource definitions and general Gateway properties (such as the Gateway's host name)
- produces the outgoing call properties.

The call is then relayed to a destination, either PSTN or VoIP, by matching the properties produced by the Routing Engine.

If no routing rule matches the incoming request, the incoming call is rejected with an appropriate SIP response code . If the outbound destination is VoIP, multiple call legs can be established using the same incoming PSTN call.




The following diagram depicts at a high level the operation of the Routing Engine.

## Q-value

Although multiple rules may be active at the same time, not all rules have the same priority. A rule's priority level is defined by its *q-value*. The q-value of a rule is a floating-point number between 0 and 1, with 1 representing the highest priority.

The term "q-value" comes directly from the SIP standard. In SIP, an application can register to a SIP proxy and specify a q-value that represents its availability (floating-point number between 0 and 1).

When you create a new rule, you must assign it a q-value. The Routing Engine processes routing requests by applying the entire set of routing rules, sorted in descending order of q-value. The first rule to trigger on the incoming request's properties will produce the outbound call properties.

If an outbound call cannot be established using the highest priority rule that is triggered, then the Routing Engine will try to establish a call using a lower priority rule that meets all of the conditions (if any), until the maximum value is reached (specified by parameter *Netborder.gw.maxRoutingRulesMatches*).

> **CAUTION:** Assign q-values carefully. If two rules with an equal q-value trigger a request, the Routing Engine cannot guarantee or predict which one will be used at runtime.

# Using definitions and properties in routing rules

When creating a routing rule, the following types of variables can be used to process the incoming request:

- Global gateway properties
- Resource definitions
- Inbound call properties.

## Global Gateway properties

There are a few global Gateway properties that can and should be used in the routing rules for this release. They include the following:

- **GW_SIP_PORT:** The port value of the first IPAddress in the Netborder.*sip.UserAgent.IPAddress* property list.
- **GW_HOST_IP:** Local host IP address.
- **GW_CALL_ID:** The unique call identifier generated by the Gateway for each PSTN-to-SIP call.

## Resource definitions

Resource definitions are used to define the outbound call properties. For example, a rule monitoring SIP-initiated calls could direct the call to a specific device group. A rule references a specific channel group by the ID property of the *<channelGroup>* element of the PSTN configuration. For example, a set of rules could route in priority SIP calls to a priority outbound pool or to a backup outbound pool depending on the type of SIP application requesting the call from the PSTN. A rule can target a specific device group by setting the outbound call property pstn.out.resourcesGroup to the appropriate group name or ID.

## Inbound call properties

Finally, the routing rules can use numerous properties of the incoming call to determine the outbound properties. A call coming into the Gateway is either a PSTN-originated call or a SIP-initiated call. A PSTN-originated call will have non-nil values in the inbound properties *pstn.in.** and nil values in *sip.in.**. For a SIP-initiated call, the opposite is true: it will have nil values in the inbound properties *pstn.in.** and non-nil values in *sip.in.**.

The following table lists the properties that can be accessed from the routing rules.

| ***Inbound Routing Rule Parameters*** | |
|---|---|
| ***Parameter*** | ***Description*** |
| pstn.in.channelName | The Sangoma channel name on which the PSTN call came in. This would be a string in the following formats.<br>For digital channels (PRI/BRI):<br>• b<board-index>di<interface-index>-c<logical-channel-index><br><br>For analog channels (FXS/FXO):<br>• b<board-index>ai<interface-index>-c0 |
| pstn.in.dnis | Dialled number. Available only in digital configurations. |
| pstn.in.ani | Calling number. In analog configurations, this is the CallerID. If not available, this property is set to "*unknown*". |
| pstn.in.ani.presentationIndicator | Calling number presentation indicator; possible values are "allowed," "restricted", "unavailable" , "reserved" or "". |
| pstn.in.ani.screeningIndicator | Calling number screening indicator; possible values are "user-provided not screened", "user-provided verified and passed", "user-provided verified and failed", "network provided" or "" |
| pstn.in.callerName | In ISDN the caller name is extracted from the SETUP or FACILITY message according to the parameter in the PSTN configuration. If not available, this property is set to "unknown-callerName".<br><br>In analog configuration , this information is extracted from the CallerID if this information is available. If not available, this property is set to "unknown-callerName". |
| pstn.in.isdn.setup.iiDigits | Automatic Number Identification (ANI) II digits in the ISDN setup message. These digits identify the type of originating station. Only available with ISDN signalling. |
| pstn.in.isdn.setup.ie.0xZZ.0xYY | The content of ISDN Information Element with codeset **0xZZ** and identifier **0xYY**, in hexadecimal format. All ISDN Information |

| Inbound Routing Rule Parameters | |
|---|---|
| **Parameter** | **Description** |
| | Elements (IE) of the SETUP message can be accessed using this property. The IE content is stored in an hexadecimal format where each octet is separated by a space, **without** the '0x' prefix. |
| sip.in.requestUri<br><br>sip.in.requestUri.canonical | The URI to which the call request was made (and its canonical version). |
| sip.in.from.uri<br><br>sip.in.from.uri.canonical | The caller's URI (and its canonical version). |
| sip.in.from.displayName | The caller's display name if available. |
| sip.in.to.uri<br><br>sip.in.to.uri.canonical | The callee's URI (and its canonical version). |
| sip.in.to.displayName | The callee's display name if available. |
| sip.in.referTo<br><br>sip.in.referTo.canonical | The address in the "Refer To" header of a REFER message (and its canonical version). |
| sip.in.referredBy.uri<br><br>sip.in.referredBy.uri.canonica | The URI of the application requesting the transfer (and its canonical version). |
| sip.in.referredBy.displayName | The display name of the SIP application requesting the transfer. |
| sip.in.redirect.Contact | The content of the contact header on reception of a SIP redirect (3XX) message. |
| sip.in.header.Via | The content of the Via header. |
| sip.in.header.CSeq | The content of the CSeq header. |
| sip.in.header.Call-ID | The content of the Call-ID header. |
| sip.in.header.Content-Length | The content of the Content-Length header. |
| sip.in.header.Contact | The content of the Contact header. |
| sip.in.header.HeaderName | For acquiring the content of an arbitrary header in the incoming SIP message (incoming call or transfer request). To obtain the value of *MyPrivateHeader*, you would use *sip.header.MyPrivateHeader*. Refer to the *Release Notes* for limitations. |

**NOTE:** Canonical versions of the URI parameters are fully expanded of the form *sip:something@hostname:portnumber*. Note that 'sip' must be in lowercase, and that the hostname is expanded to dotted IP notation. For example: *sip:1026@192.168.11.207:5066*.

## Outbound call properties

The purpose of a routing rule is to set the properties of an outgoing call (either PSTN or SIP) based on the information gathered about the incoming call, as well as global properties and resource data.

The following table lists the outbound properties that can be set in a routing rule using the *<out_param>* element:

| Outbound Routing Rule Parameters | |
|---|---|
| **Parameter** | **Description** |
| pstn.out.phoneNumber | For setting the phone number as it should be dialed on the telephony channel. |
| pstn.out.phoneNumber.type | Please refer to Table 4.9 of spefication ITU-T Q931 05/98 |
| pstn.out.ani | For setting the ANI (CAS, default value is 5678), or the calling number information (ISDN). |
| pstn.out.ani.type | Please refer to Table 4.11 of spefication ITU-T Q931 05/98 |
| pstn.out.ani.numberingPlan | Numbering plan applies for type of number = 0, 1, 2 and 4.<br>Please refer to Table 4.11 of spefication ITU-T Q931 05/98 |
| pstn.out.ani.presentationIndicator | Presentation indicator (octet 3a) of the calling party number information element.<br><br>NOTE: meaning and the use of this field is defined in clause 3/ITU-T Q.951 and clause 4/ITU-T Q.951<br><br>Please refer to Table 4.11 of spefication ITU-T Q931 05/98 |
| pstn.out.ani.screeningIndicator | Screening indicator (octet 3a) of the calling party number information element.<br><br>NOTE: The meaning and the use of this field is defined in clause 3/ITU-T Q.951 and clause 4/ITU-T Q.951.<br><br>Please refer to Table 4.11 of spefication ITU-T Q931 05/98 |
| pstn.out.callerName | For ISDN interfaces, the caller name could be carried either in SETUP or in a FACILITY message immediately following a SETUP message. When the caller name is carried in the SETUP message, the caller name is stored in the DISPLAY IE. When the caller name is carried in FACILITY message, the SETUP message contains a facility IE to inform the user if a FACILITY message with caller name is expected.<br><br>NOTE: For analog interfaces, this parameter is ignored for the current version. |
| pstn.out.channelGroup | For setting the resources group to select in making the outbound call. |

| Outbound Routing Rule Parameters | |
|---|---|
| ***Parameter*** | ***Description*** |
| pstn.out.earlyMedia.start.condition | Used only for the analog fxo gateway. Indicates when the gateway will start earlyMedia for analog fxo outboubnd calls. Possible values are 'immediate' or 'after-dialing'.<br><br>If set to 'immediate', the gateway will start the early media by sending a 183 Session In progress as soon as it receives the sip INVITE. If set to 'after-dialing', the gateway will start the eraly media only after having dialed the outbound dtmf digits. If pstn.out.earlyMediaMode is set to 'never', this parameter will have no effect on the gateway, since earlyMedia is disabled. |
| pstn.out.earlyMediaMode | For setting the conditions, if any, under which the Gateway will start the media streams and notify the calling SIP *UA (User Agent)* of an ‘183 Session Progress’ (message that there is some audio prior to call connection that the caller should hear).<br>Valid values are: ‘never’, ‘always’ or ‘as-needed’ (default). |
| pstn.out.progressToneDetectionMode | For setting the conditions, if any, under which the Gateway analyses the PSTN audio stream to detect progress tones (BUSY, REORDER and SIT tones).<br><br>Valid values are: 'never', 'always' or 'as-needed' (default). |
| pstn.out.isdn.setup.iiDigits | For setting the automatic Number Identification (ANI) II digits sent with the SETUP message. These digits identify the type of originating station. Only available with ISDN signalling. |
| pstn.out.isdn.setup.ie.0xZZ.0xYY | For setting the content of ISDN Information Element with codeset **0xZZ** and identifier **0xYY**. Any ISDN Information Elements (IE) of the SETUP message can be set using this property. The IE content is specified in a hexadecimal format where each octet is separated by a space, **without** '0x' prefix.<br>You can also specify string. To specify one or more string, enclose each of them between 's(' your-string ')'.<br>Example:<br><param name="pstn.out.isdn.setup.ie.0x00.0x70" expr="04 s(my string) s(%0)"/><br><br>To send a NOT-END-TO-END-ISDN IE in the setup, one should add to the pstn_out routing rule the following line : <param name="pstn.out.isdn.setup.ie.0x00.0x1E" expr="80 81"/><br>Any IE defined here will be added to the SETUP message **without validation** of its content, overriding any automatically |

| Outbound Routing Rule Parameters | |
|---|---|
| **Parameter** | **Description** |
| | generated IE. Use this feature with care as it may send corrupted message without warning being generated by the application. The IE will be correctly placed in the message according to Q931 specification and many IE may be defined in the same message. |
| pstn.out.networkSpecific | 4ESS and 5ESS switches sometime require you to provide the feature and/or the service you have subscribed. This information is provided by your Telco operator on your subscribtion contract. The following parameters allow you to specify them. Some features and service are specified in the ATT standard TR41459 June 1999 in section 3.6.5.19 Network-Specific Facilities. |
| pstn.out.networkSpecific.feature | This parameter is interpreted by ISDN channels and ignored by analog channels.<br>Can take any of the following values as decribed ATT standard TR41459 June 1999 on Table III - 43. |
| pstn.out.networkSpecific.service | This parameter is interpreted by ISDN channels and ignored by analog channels.<br>Can take any of the following values as decribed ATT standard TR41459 June 1999 on Table III - 43. |
| pstn.out.networkSpecific.identification | In some networks you have to provide Network identification parameters in order to use network specific services. This parameter "pstn.out.networkSpecific.identification" along with the parameters "pstn.out.networkSpecific.identification.type" and "pstn.out.networkSpecific.identification.plan" allows you to specify those.<br><br>This parameter allows you to set the optional octet 3 and 3.1+ as defined in ATT TR 41459, June 1999. The octect 3 and 3.1+ are no inserted in the Network Specific Facility when this parameter is missing.<br><br>The user can specify the type and the plan via the follow parameters:<br>• pstn.out.networkSpecific.identification.type<br>• pstn.out.networkSpecific.identification.plan<br><br>The default type is 2 (National network identification). The default plan is 1 (Carrier identification code). |

| Outbound Routing Rule Parameters | |
|---|---|
| *Parameter* | *Description* |
| pstn.out.networkSpecific.identification.type | Refer to description of parameter pstn.out.networkSpecific.identification. |
| pstn.out.networkSpecific.identification.plan | Refer to description of parameter pstn.out.networkSpecific.identification. |
| pstn.out.ignoreISDNCauseDuringEarlyMedia | If set to true, the gateway will not immediately disconnect the call if it receives a progress announcing some inband info prior to a disconnect with a valid cause while trying to place an inbound call. This could be helpful for users who want the gateway to let some outbound call rejection inband message be played instead of immediately terminate the call.<br><br>Possible values are: true or **false** (default value) |
| sip.out.ignore180 | Can be true or false (default is false). When set to true, it will ignore the 180 messages from the target SIP application and not go in the 'ringing' state.<br>Can be useful if a SIP user agent sends a 180 without SDP followed by a 183 with SDP, as this situation cannot be reproduced in ISDN, and the Gateway will fail to establish early media. |
| sip.out.accept183 | Can be true or false (default is false). When set to true, it will accept a 183 message as a 180 and go in the 'ringing' state. Can be useful in the same situation as the *sip.out.ignore180* property. |
| | |
| sip.out.requestUri | For setting the URI to which the call request is going to be made. |
| sip.out.from.uri | For setting the caller's URI. Typically used to indicate the  the ANI or CallerID of an incoming PSTN call, if available. |
| sip.out.from.displayName | For setting the caller's display name if available. |
| sip.out.to.uri | For setting the callee's URI. |
| sip.out.to.displayName | For setting the callee's display name. |
| sip.out.redirect.Contact | For setting the contact header following a redirect primitive. |
| sip.out.header.HeaderName | For setting arbitrary headers in the outgoing INVITE message. For example, to set *MyPrivateHeader* to value 'foo', you would use *sip.out.header.MyPrivateHeader=foo*. Refer to the *Release Notes* for limitations. |

Now that we have identified the sources of input and output, we will see in the next section how to pull all of this information together into a comprehensive routing rule.

## Routing rule constructs

Routing rules are described using an XML format. The following table lists the elements that are used to define a rule.

| Routing Rule Elements | | |
|---|---|---|
| ***Element*** | ***Description*** | ***Attributes*** |
| <rule> | The <rule> element is used for each of the system's routing rules | • **name** (string): The name of the rule.<br>• **outbound_interface** (string): This is where the call should be directed (outbound direction). Can be set to either 'pstn' or sip'.<br>• **qvalue** (float): Priority of the rule between 0 and 1, 1 being the highest priority. |
| <condition> | This element is used to test one condition on a number of properties. Multiple <condition> elements or "tags" are permitted within one rule; however, all conditions must be met for that rule to trigger. | • **param** (string): The name of the parameter on which the condition is tested. Names of suitable parameters are provided in the Inbound Routing Rule Parameters Table and the Outbound Routing Rule Parameters Table.<br>• **expr** (string): Regular expression. Test submitted to the input parameter. The input parameter must successfully pass the test for the condition to be met.<br>• **except** (string): Regular expression. Negative test to be performed on the input parameter. If the 'except' property is used, then the 'expr' must return true and the 'except' must return false for the condition to be met. |

| Routing Rule Elements | | |
|---|---|---|
| **Element** | **Description** | **Attributes** |
| <out_leg> | An <out_leg> element is used for each outbound call leg resulting from a triggered rule on incoming calls. | • **name** (string): The name of the out leg.<br>• **media_type** (string): The type of media connection. Can only be set to 'sendrecv' or 'recording' in this release.<br>• **outbound_interface** (string): Can be set to 'sip' or 'pstn' to overwrite the outbound interface of the rule, for that <out_leg> only. |
| <param> | This element is used to set an outbound call leg property. One outbound parameter is set per <param> element or "tag". Multiple tags are allowed within a single out leg. | • **name** (string): The name of the outbound call property to be set. Names of suitable parameters are provided in the Outbound Routing Rule Parameters Table.<br>• **expr** (string): Value to set in the outbound call property. May contain the results of regular expressions and/or inbound and general properties. |

The routing rule syntax leverage the full power of a regular expression processor, which is integrated into the Routing Engine. A regular expression is a string that is used to describe or match a set of strings, according to certain syntax rules. If you are not familiar with regular expressions, we encourage you to consult widely available literature on this subject, including the tutorial at http://perldoc.perl.org/perlretut.html.

The specific regular expression package used by the Gateway is documented at the following location: http://www.pcre.org/pcre.txt.

The regular expressions in routing rules are most often used to perform pattern matching on the input parameters and/or to format the outbound properties. An atom within a regular expression is delimited within the rule using parentheses (), and can be called later in the rule by using the percent sign %. The first atom in a rule can be accessed via variable %0, the second one via variable %1, and so on. A regular expression only lives within the scope of the *<rule>* element within which it is defined. See the following routing rules examples for clarification.

## Routing rule examples

Below are a four examples of how you might use routing rules to accomplish specific ends.

### Example 1: Incoming SIP call dialing a 3- or 4-digit extension on PBX

Here is an example of a single rule that triggers when an incoming SIP call has the following format: `sip:<3-4-digit-ext>@<gateway-host>:<gateway-ua-port>`, where:

- `<3-4-digit-ext>` is a three or four-digit extension to dial on the PBX for the outbound call leg
- `<gateway-host>` is the name or IP address of the Gateway host
- `<gateway-ua-port>` is the port number of the Gateway's SIP User Agent (5066 by default).

```
<!-- In SIP, out PSTN, dialing a 3 or 4 digit extension on the PBX -->
< rule name =" outbound_PSTN_extension " outbound_interface =" pstn " qvalue =" 0.1 ">
<!--
 Look if the incoming URI is made of 3 or 4 digits followed by the gateway host (global GW_HOST_IP) and
user agent port (GW_SIP_PORT)
  -->
  < condition param =" sip.in.requestUri.canonical " expr = " ([0-9]{3,4})@GW_HOST_IP:GW_SIP_PORT
" />
<!--
 The text found by regular expression ([0-9]{3,4}) can be accessed via variable %0. Set the outbound phone
number to dial
  -->
  < out_leg name =" default " media_type =" sendrecv " >
       < param name =" pstn.out.phoneNumber " expr =" %0 "/>
    </ out_leg >
/>

</ rule >
```

### Example 2: Incoming SIP call dialing an external number

Here is an the example of another rule that performs exactly the same function as the one above, except that it dials a '9' followed by a pause (represented by a comma in telephony), followed by an external phone number.

```
<!-- In SIP, out PSTN, dialing an external number out of the PBX -->
< rule name =" outbound_PSTN_external " outbound_interface =" pstn " qvalue =" 0.1 ">
<!--
 Look if the incoming URI is made of 7 digits followed by the gateway host (global GW_HOST_IP) and user
agent port (GW_SIP_PORT)
                 -->
  < condition param =" sip.in.requestUri.canonical "
           expr =" ([0-9]{7,})@GW_HOST_IP:GW_SIP_PORT " />
<!-- Set the outbound phone number to dial "9," + phone number -->
  < out_leg name =" default " media_type =" sendrecv " >
< param name =" pstn.out.phoneNumber " expr =" 9,%0 " />
        </ out_leg >
</ rule >
```

The same type of rule could be used to detect a long distance number and handle it appropriately on the PBX line.

### Example 3: DNIS-based routing

Routing rules allow you to select a target SIP destination based on the properties of the incoming call. With analog lines, the number of properties for the incoming call is minimal (with a digital configuration, on the other hand, much more information is known about an incoming PSTN call). Still, you might find it useful, for example, to route all of the incoming calls on a particular port range to a specific application, and calls coming into other ports to another SIP destination.

Here is an example of a rule that could be used to direct the received calls on the first span with different numbers (DNIS) to different SIP URIs.

```
<!– DNIS-based Routing -->

< rule name =“ DNIS_Routing " outbound_interface =“ sip " qvalue =" 0.1 ">

  <!-- Look if the incoming PSTN call is on the first Strong_20_Emphasis -->

< condition param =“ pstn.in.channelName " expr =" B1T.* " />

  <!–- Retrieve and store the ANI in variable %0  -->
  < condition param =“ pstn.in.ani " expr =“ (.*) " />

 <!–- Retrieve and store the DNIS in variable %1 -->
  < condition param =“ pstn.in.dnis " expr =" ([0-9]*) " />

   < out_leg name =“ default " media_type =“ sendrecv “ >
     <!-- Set the CallerID in the ‘From’ URI -->
      < param name =“ sip.out.from.uri "
                  expr =“ sip:%0@GW_HOST_IP:GW_SIP_PORT "/>
      < param name =“ sip.out.from.displayName " expr =“ Gateway "/>

      <!-- Set the outbound URIs to point to URI of the form sip:5551212@acme.com -->
      < param name =“ sip.out.requestUri " expr =“ sip:%1@acme.com “ />
      < param name =“ sip.out.to.uri " expr =“ sip:%1@acme.com "/>
      < param name =“ sip.out.to.displayName " expr =“ App "/>
   </ out_leg >

</ rule >
```

### Example 4: Application failover using q-value

As discussed earlier (Q-value on page 4), when multiple rules are triggered, the Gateway will apply the rule with the highest priority (q-value). Multiple SIP destinations with different priorities could thus be targeted for a single PSTN call. This way, if the Gateway is not able to connect with the highest priority destination, it will try the second highest priority, and so on, according to the maximum number of rules that can be used to establish a call (as per the *Netborder.gw.maxRoutingRulesMatches* global configuration property).

Below are sample routing rules that could be used to provide application failover. In this example, the address *sip:primary@acme.com* is targeted in priority (qvalue is set to 0.2). If this application does not respond in the amount of time specified in *paraxip.gw.ringTimeoutMs*, then the second rule (qvalue set to 0.1) will be executed and attempt to contact ‘*sip:backup@acme.com*’.

```
 < rule name =“ Primary_Routing " outbound_interface =“ sip " qvalue =" 0.2 ">

  <!–- Retrieve and store the ANI in variable %0  -->
  < condition param =“ pstn.in.ani " expr =“ (.*) "/>

 <!–- Retrieve and store the DNIS in variable %1 -->
  < condition param =“ pstn.in.dnis " expr =" ([0-9]*) "/>

   < out_leg name =“ default " media_type =“ sendrecv “ >
     <!-- Set the ANI in the ‘From’ URI -->
      < param name =“ sip.out.from.uri "
                 expr =“ sip:%0@GW_HOST_IP:GW_SIP_PORT "/>
      < param name =“ sip.out.from.displayName " expr =“ Gateway "/>

      <!-- Set the outbound URIs to point to URI of the primary target -->
      < param name =“ sip.out.requestUri " expr =“ sip:primary@acme.com “ />
      < param name =“ sip.out.to.uri " expr =“ %1 "/>
      < param name =“ sip.out.to.displayName " expr =“ App "/>
      <!-- Using a 5 sec ring timeout to detect that the application is down -->
      <param name="paraxip.gw.ringTimeoutMs" expr="5000" />
   </ out_leg >

</ rule >

< rule name =“ Backup_Routing " outbound_interface =“ sip " qvalue =" 0.1 ">

  <!–- Retrieve and store the ANI in variable %0  -->
```

```
  < condition param =“ pstn.in.ani " expr =“ (.*) " />
  <!–- Retrieve and store the DNIS in variable %1 -->
  < condition param =“ pstn.in.dnis " expr =" ([0-9]*) " />

   < out_leg name =“ default " media_type =“ sendrecv “ >
     <!-- Set the ANI in the ‘From’ URI -->
      < param name =“ sip.out.from.uri "
                  expr =“ sip:%0@GW_HOST_IP:GW_SIP_PORT "/>
      < param name =“ sip.out.from.displayName " expr =“ Gateway "/>

      <!-- Set the outbound URIs to point to URI of the backup target -->
      < param name =“ sip.out.requestUri " expr =“ sip:backup@acme.com “ />
      < param name =“ sip.out.to.uri " expr =“ %1 "/
      < param name =“ sip.out.to.displayName " expr =“ Backup App "/>
   </ out_leg >

</ rule >
```

## Default routing rules

The default routing configuration file ships with a number of pre-configured routing rules. To customize the Gateway's routing behaviour, you might choose to add to the list of rules (presumably with higher q-values than the default rules) and/or simply re-write them.

Below is a short description of the routing rules that are provided by default with the Gateway. For complete details, please refer to the rules' syntax in the *routing-rules.xml* file located at *[GATEWAY_HOME]\config\routing-rules.xml* (where *[GATEWAY_HOME]* is the root folder of the installation).

- **Rule "default_sip_out":** All incoming PSTN calls are sent to the SIP URI provided at installation time. The Caller ID, if present, is added in the 'From' URI using the format `sip:CallerID@GW_HOST_IP:`
`GW_SIP_PORT`.

  This rule can be changed to modify the routing of PSTN calls to SIP applications, and/or to modify how the Caller ID information is presented.

- **Rule "default_pstn_out":** All incoming SIP calls that have a Request URI that matches the format `sip:<tel-number>@<gateway-host>:<gateway-ua-port>` are directed to the first available telephony port, where
  - `<tel-number>` is a telephony number to be dialed on the outbound telephony line
  - `<gateway-host>` is the name or IP address of the Gateway host
  - `<gateway-ua-port>` is the port number used by the Gateway's SIP user agent).

  This rule can be changed to modify the format or the field by which a SIP-based application directs outbound PSTN calls.

- **Rule "Redirect_to_sip":** A rule that is used to generate the outbound SIP call caused by a redirect instruction (message 3XX from a SIP server) to a SIP endpoint.
- **Rule "Redirect_to_pstn":** A rule used to generate a PSTN call (bridged on Gateway) when receiving a redirect instruction (message 3XX from a SIP server) to a PSTN phone number.

The role of the default routing rules is to provide basic functionality out of the box, as well as a sample of what can be achieved by customizing the data file.

> **TIP:** Out of the box, the Gateway will route all incoming PSTN calls to the target SIP URI specified at installation time . If there is a SIP proxy in the the network, this is where the calls should be routed. To modify the default SIP destination, simply edit the routing rules file, find the rule named "*default_sip_out*", and modify the value of the outbound property *sip.out.requestUri*. To implement DNIS-based routing, please refer to Example 3: DNIS-based routing on page 18.

## Caching of routing rules

Under high traffic conditions, the routing rules are not fetched on every incoming call. Instead, a caching mechanism is used to keep a temporary copy of the rules inside the Gateway. This cached version of the routing rules is kept for 10 seconds. After 10 seconds, a new *HTTP (HyperText Transfer Protocol)* query is performed. This mechanism helps limit the HTTP traffic.

The caching mechanism is automatically activated under high traffic conditions. Under low traffic conditions (that is, when handling a low number of calls per second), caching is automatically disabled.

The cache timeout value can be changed via the parameter *Netborder.gw.routing.cacheFlushTimeoutSec* in the global configuration file. When set to '0', the cache is disabled and the rules are fetched on every incoming call.

Note that if the fetched routing rules file is invalid or incorrectly formatted, the Gateway will generate a warning and use the previous valid routing rules for new incoming calls.