

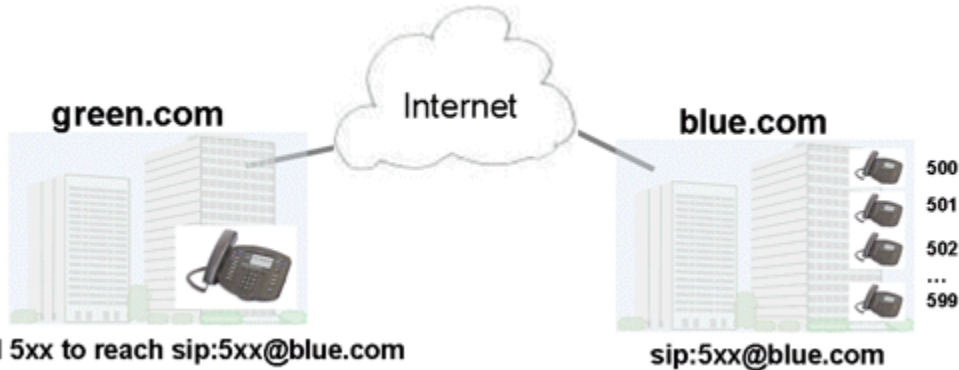
Defining SIP URI Mapping Rules

<?xml version="1.0" encoding="utf-8"?>
<html>

Defining SIP URI Mapping Rules

The sipX dial plan allows the definition of rules for the handling and transformation of dialed **numbers**. However, what if you wanted to manipulate SIP addresses (SIP URIs) instead? Let's assume you are at company "green" and you would like to define a dial plan so that all the 5xx numbers are routed to company "blue" directly over IP (see figure below). Users with hard phones typically are only willing to dial numbers and not SIP URIs. You as the administrator would like to route calls to company "blue" over an IP connection, especially because you know that your employees call company blue a lot.

<center>



</center>

Different combinations of SIP URI mapping rules are desirable. Let's assume you have a 4 digit dial plan and you want to define a prefix, e.g. "8", that would cause all dialed numbers that start with "8" to be routed to company "blue" as follows: *8,xxx --> sip:xxx@blue.com*. Or you would like to define a specific number to call company blue's operator: *300 --> sip:operator@blue.com*.

Starting with sipX 3.2 such mapping rules can be defined using the sipX configuration server.

SIP URI Mapping Rules Defined in ConfigServer

In order to define such a SIP URI mapping rule a virtual gateway has to be created first (see below). the domain name of the destination is given as an address and no serial number (MAC address) is specified.

<center>

Gateway Details

Name	<input type="text" value="CompanyblueGW"/>
Address	<input type="text" value="blue.com"/> <small>IP address (example: 10.1.1.1) or a fully qualified hostname (example: gateway.example.com)</small>
Serial Number	<input type="text"/> <small>Serial number contains 12 hexadecimal digits (0-9 and a-f), for example: 0040214131fa. Usually the serial number is set to the device's MAC address. For an unmanaged gateway, the serial number is only displayed, not used.</small>
Description	<input type="text" value="This is a virtual GW that allows certain dialed numbers to be mapped to the blue.com domain."/>

</center>

Now a custom dial rule can be defined that performs the desired mapping. Using the "Dialed Number Prefix" and the "Resulting Call Prefix" all kinds of desired combinations can be specified.

Note: The "Resulting Call Prefix" should also be able to accommodate an alphanumeric string. However, ConfigServer currently still limits this capability to numeric values (see XCF-1086).

<center>

Custom Dialing Rule

Enabled

Name

Description

Dialed Number

Prefix

Required Permissions

- NoAccess
- Voicemail
- 900Dialing
- TollFree
- LongDistanceDialing
- InternationalDialing
- LocalDialing
- AutoAttendant
- ForwardCallsExternal
- RecordSystemPrompts
- ValidUser

Resulting Call

Prefix :

Gateways

[Add New Gateway](#) [Add Existing Gateway](#)

<input type="checkbox"/>	Name	Address	Description
<input type="checkbox"/>	CompanyBlueGW	blue.com	This is a virtual GW that allows certain dialed numbers to be mapped to the blue.com domain.

</center>

Note: The order in which dialing rules are specified matters in sipX. More specific rules have to appear **before** more generic rules. Since SIP URI mapping rules are generally more specific than some of the default rules, make sure you move them up before activation.

Manually Defining SIP Routing

{{Box Important| The following only applies to expert users; use these manual processes only as a last resort. In most cases it's better to use dialing rules, hunt groups or call forwarding to achieve the same functionality.}}

For background, see [how the actual routing decisions are made in the Registry/Redirect server](#)

Manual Aliases

Sometimes you need to specify many such SIP URI routing rules. An example could be that you would like to define numeric aliases for many destinations registered at many different domains. Such a service could also be looked at as **global speed dialing of SIP URIs**. If you are an expert user, there is a possibility to define *external aliases* in an XML configuration file that can be added to the sipXecs system. The location of this file can be specified in *sipxco.nfig.properties* (in 3.X and below, use *sipxconfig.properties.in*)

Look for the *external aliases* section and add the name of your file(s) to *externalAliases.aliasAddins*. You can add multiple comma separated file names. `{{Box File| /etc/sipxpbx/sipxconfig.properties (/etc/sipxpbx/sipxconfig.properties.in in 3.X and below) |`

```
# external aliases
externalAliases.addinsDirectory=/usr/share/java/sipXecs
externalAliases.aliasAddins=ext_aliases.xml, my_ext_aliases.xml
```

```
}}
```

An example of such an XML file, *my_ext_aliases.xml*, looks like this:
`{{Box File| Example file /usr/share/java/sipXecs/my_ext_aliases.xml|`

```
<items type="alias" xmlns="http://www.sipfoundry.org/sipX/schema/xml/alias-00-00">
  <!-- map 77000 to another domain -->
  <item>
    <identity>77000@example.com</identity>
    <contact>sip:77000@host.example.com</contact>
  </item>
  <!-- map 1234567 to another number at another domain -->
  <item>
    <identity>1234567@example.com</identity>
    <contact>sip:7654321@sipfoundry.org</contact>
  </item>
</items>
```

```
}}
```

Note: The URL stated in the file above is just a namespace identifier and does not exist on the Internet. It needs to match the targetNamespace attribute in the schema, which is in the *sipXregistry/meta* directory.

This example file shows how to add a set of fixed mappings from one SIP address to another. Each `<item>` element below creates an alias that directs incoming requests for the `<identity>` address to the `<contact>` address. This does not specify patterns, only exact matches. (For patterns, use the dialing rules interface as described above.)

To create a set of aliases:

- Copy this file to `/usr/share/java/sipXecs` directory (or to the directory designated by *externalAliases.addinsDirectory* property)
- Edit the example to create the set of aliases you want
- Check that your copied file syntax is valid by running the following command. If no errors are reported the file is ok and you can now restart sipXconfig.
`{{Box Code| Optional: Validate XML file syntax|`

```
sipx-validate-xml /usr/share/java/sipXecs/copiedfile.xml
```

```
}}
```

After this change every time sipXconfig generates its set of aliases it will include aliases from your file as well. You can trigger generation of aliases by adding and removing an alias to any user at any time using the sipXconfig UI. You can do this as an alternative to manually restarting sipXconfig after you modified the external aliases file.

Manual Mapping and Fallback Rules

When you can express the routing change that you want to make as a pattern, and especially if you don't know all the possible addresses you need to route to, then you can use an external Mapping rule or Fallback rule. Again **try to use the facilities in the web interface to do this first**.

You can specify an external mapping rule or fallback rule that will be inserted by sipXconfig into the rules it generates. Usually, mapping rules are only for destinations *within* your domain, and fallback rules are for destinations *outside* your domain.

The location of the rules file can be specified in *sipxconfig.properties.in*

Look for the *fallbackRules.externalRulesFileName* property and add the name of your file.
`{{Box File| /etc/sipxpbx/sipxconfig.properties.in|`

```
# external additions to external call fallback rules
fallbackRules.externalRulesFileName=/etc/sipxpbx/external-fallback.xml
```

```
}}
```

An example of such an XML file, *external-fallback.xml*, looks like this:
{{Box File| Example file /usr/share/sipxpbx/my_ext_aliases.xml|

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
< mappings xmlns='http://www.sipfoundry.org/sipX/schema/xml/urlmap-00-00' >

  < hostMatch >
    < hostPattern > gateway.provider.net </ hostPattern >
    < userMatch >
      < userPattern > . </ userPattern >
      < permissionMatch >
        < transform >
          < host > proxy.provider.net </ host >
        </ transform >
      </ permissionMatch >
    </ userMatch >
  </ hostMatch >

</ mappings >
```

```
}}
```

Note: The URL stated in the file above is just a namespace identifier and does not exist on the Internet. It needs to match the targetNamespace attribute in the schema, which is in the *sipXregistry/meta* directory.

This example file shows how to change one domain name (gateway.provider.net) to another (proxy.provider.net). The name is changed *only* in the request URI of the forwarded request, *not* in the To or From headers.

- Before you install the file, check that your file syntax is valid by running the following command. If no errors are reported the file is ok and you can now restart sipXconfig.
{{Box Code| Optional: Validate XML file syntax|

```
sipx-validate-xml /etc/sipxpbx/external-fallback.xml
```

```
}}
```

After this change every time sipXconfig generates its rules, it will include the rules from your file as well. You can trigger generation of rules by using the *Activate Dial Plans* button.

</html>