Oracle Utilities Cloud Services

IP Whitelisting Guide

For 18.x Releases

E96481-01

May 2018

Oracle Utilities Customer Cloud Services 18.x IP Whitelisting Guide

Contents

Chapter 1

Chapter 2

Chapter 1

Oracle Utilities Cloud Services IP Whitelist Configiration

This guide provides information about IP whitelist configuraiton, including:

- Introduction
- Whitelist Update Permission

Introduction

This guide provides instructions to be used by a super-user to change the IP Whitelist configuration for the following Oracle Utilities cloud services:

- Oracle Utilities Meter Solution Cloud Service (OUMSCS)
- Oracle Utilities Analytics Cloud Service (OUACS)
- Oracle Utilities Customer Cloud Service (OUCCS)
- Oracle Utilities Work and Asset Cloud Service (OUWACS)
- Oracle Utilities Operational Device Cloud Service (OUODCS)
- Oracle Mobile Workforce Cloud Service (OMWCS)

IP Whitelist configuration allows for access to these cloud services without the use of a VPN connection. Each service instance has the choice to expose particular resources via VPN or IP whitelist or both VPN and IP whitelist. Each cloud service instance may make different choices about how services are exposed. For example, if subscribed to multiple cloud services they do not all need to have the same choice. One can use VPN and the next IP whitelist.

Each subscriber has one Common Services instance regardless of which of the above Cloud Services are subscribed to. Common Services includes SFTP Server and the Identity Management stack.

VPN access is optional, but note that these cloud services will be unreachable until either VPN access or IP Whitelist has been setup.

Whitelist Update Permission

A "whitelist permission representative" must be delegated. This representative is responsible for providing Oracle Customer Success with a list of users that have permission to update the whitelist.

Chapter 2

IP Whitelist Update Process

This chapter describes the process by which customers can update the IP Whitelist configuration for their Oracle Utilities cloud services.

Updating the IP Whitelist requires the following steps:

- Open a Service Request
- Get the Template
- Get the Whitelist Validator and Schema
- Create the Whitelist
- Attach Whitelist to the Service Request
- Verify Whitelist

Each of these steps is described in detail below.

Open a Service Request

Create a service request for the appropriate cloud service. If the updates apply to more than one, create a service request for one of them. Use the following values for this service request:

- **Problem Type**: Accessibility and Security Issues
- **Problem Summary**: Update Whitelist

Get the Template

Oracle Customer Success will attach two files to the service request:

- **template_whitelist.xml**: A template for making the whitelist
- **example_whitelist.xml**: An example of a populated whitelist

Download both of these files. Note that these files may be different based on current cloud subscriptions.

Get the Whitelist Validator and Schema

The whitelistvalidator.jar tool is provided with your cloud service documentation (OR is available from Oracle Customer Success). Save the validator to the same folder as the template and run the following command from this folder to get the XML schema.

```
java -jar whitelistvalidator.jar -schema whitelist_config.xsd
```

This command will work in both the Windows and Linux command prompts assuming Java 8 is in your path.

The XML schema is not mandatory, but it is helpful when editing the whitelist in an XML editor.

Create the Whitelist

Make a copy of the file template_whitelist.xml. The copy can have any name as long as it retains .xml file extension, but for the purposes of this document it will be called "whitelist.xml".

Open the whitelist.xml file in the XML editor of your choice.

Editing

The whitelist is an XML document with four sections as described below. Any XML editor should work and most XML editors will have the ability to validate the document against the XML schema if the schema is in the same folder as the XML document and is named "whitelist_config.xsd".

IpSets

The <ipSets> section is a series of lists of the IP addresses and IP subnets used for inbound whitelists. Each list is called a set (or IP set) and is represented by a <set> element in the document. Each set has a unique name which is represented by the "name" attribute.

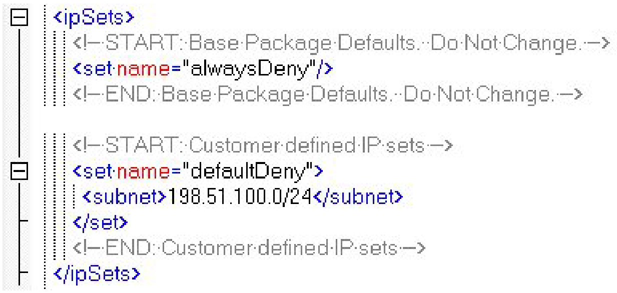
The contents of a set is a series of <subnet> and/or <include> elements.

- <subnet> elements represent an subnet of IPv4 IP addresses. This is expressed in Classless Inter-Domain Routing (CIDR) notation (e.g., "198.51.100.0/24") or as an IPv4 address (e.g., "198.51.100.37").
- <include> elements allow the contents of a previously defined set to be included in a set. The content of the <include> element is the name of the set to be included.

All of the changes made to the <ipSets> section must be between the "START: Customer-defined IP sets" and "END: Customer-defined IP sets" comments. A set named "defaultDeny" exists in the template. The simplest case for allowing external access is to have one IP set that covers all of the allowed inbound connections.

Note: The IP subnets and addresses that are specified must belong to your organization.

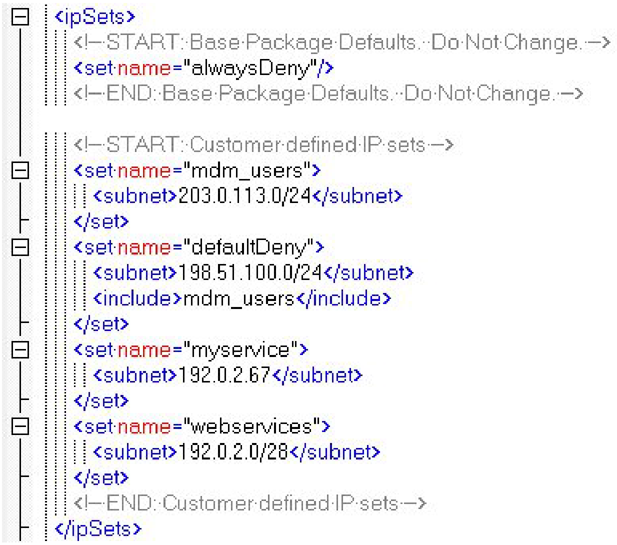
Example 1: One IP Set for Everything



```
<ipSets>
    <!-- START: Base Package Defaults.  Do Not Change. -->
    <set name="alwaysDeny"/>
    <!-- END: Base Package Defaults.  Do Not Change. -->

    <!-- START: Customer defined IP sets -->
    <set name="defaultDeny">
        <subnet>198.51.100.0/24</subnet>
    </set>
    <!-- END: Customer defined IP sets -->
</ipSets>
```

Example 2: Separate Whitelist for MDM and Web Services



```
<ipSets>
    <!-- START: Base Package Defaults.  Do Not Change. -->
    <set name="alwaysDeny"/>
    <!-- END: Base Package Defaults.  Do Not Change. -->

    <!-- START: Customer defined IP sets -->
    <set name="mdm_users">
        <subnet>203.0.113.0/24</subnet>
    </set>
    <set name="defaultDeny">
        <subnet>198.51.100.0/24</subnet>
        <include>mdm_users</include>
    </set>
    <set name="myservice">
        <subnet>192.0.2.67</subnet>
    </set>
    <set name="webservices">
        <subnet>192.0.2.0/28</subnet>
    </set>
    <!-- END: Customer defined IP sets -->
</ipSets>
```

Inbound

The <inbound> section lists the externally-accessible hosts. Each host has a list of their valid paths for inbound connections and the IP set.

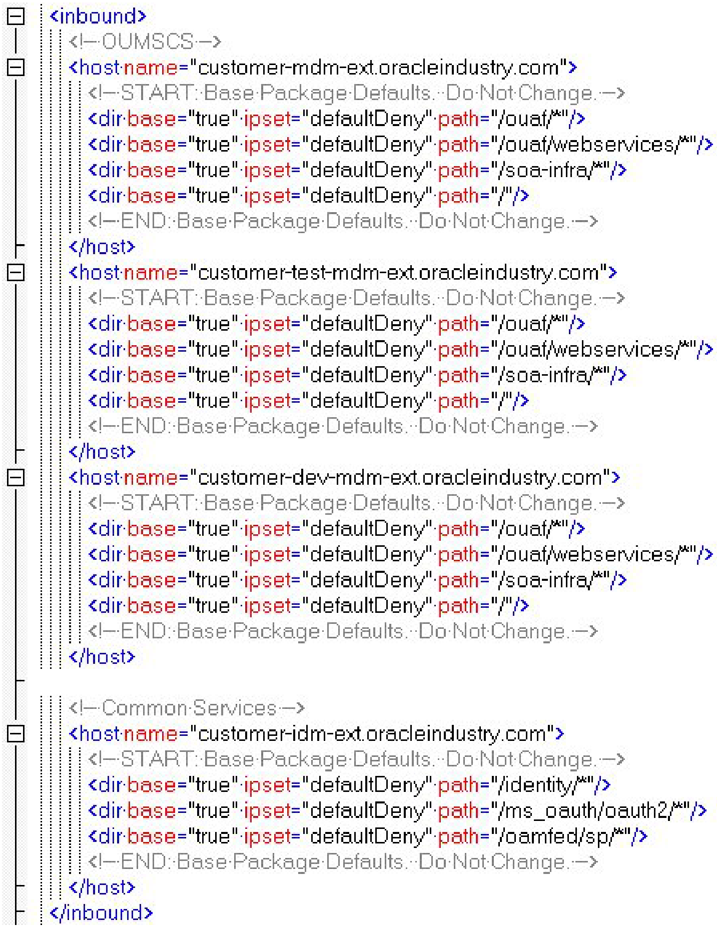
- Paths are listed as <dir> elements with the attribute base="true".
- Existing <dir> elements should not be modified.
- Path mappings are changed by adding new <dir> elements.
- Added <dir> elements must have the "ipset" and "path" attributes set and must not have the "base" attribute.
- An added <dir> with the same path as an existing base <dir> will override the ipset of the existing <dir>.
- An added <dir> with a new path will add a new path.
- The order of the <dir> elements does not matter.
- If the path of an incoming HTTP request matches the path of more than one <dir> element then the <dir> element with the longest path will be used.
- The path attribute can use the following wildcard characters:
    - * - matches 0 or more characters
    - ? - matches exactly 1 character

Additional Notes:

- <host> elements should not be added or removed.
- The name attribute of <host> elements should not be modified.
- <dir> elements with the attribute base="true" should not be added, removed, or modified.

Example 1: One IP Set for Everything

The <inbound> section that matches the first IpSets example would use the default values.

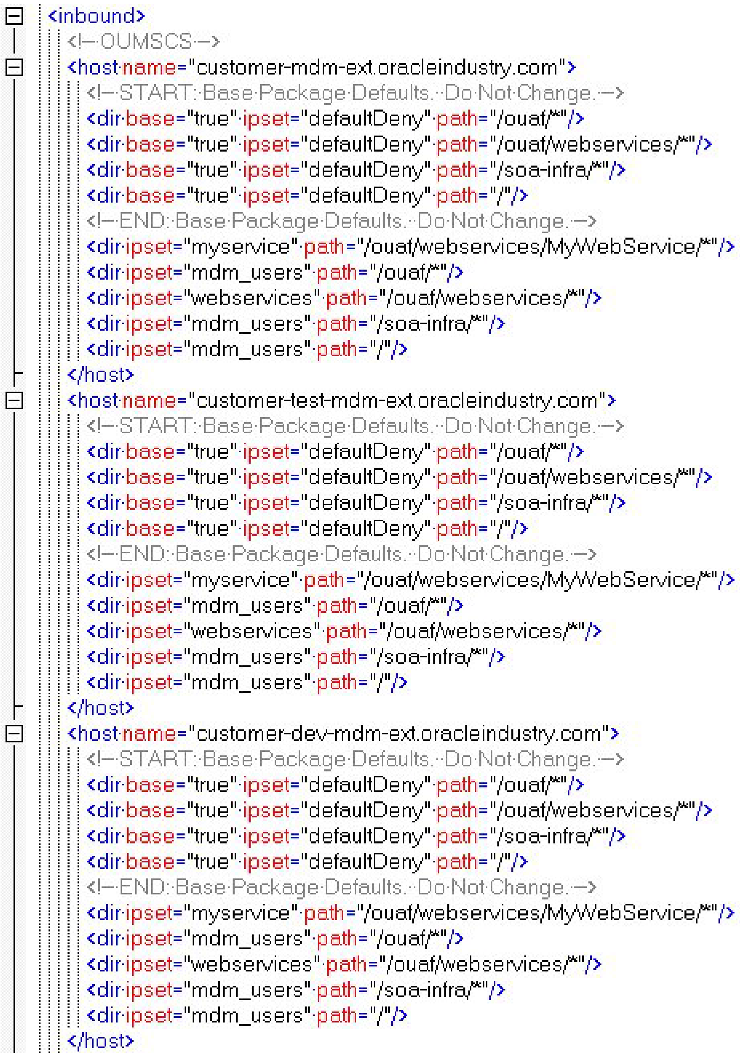


```
<inbound>
  <!-- OUMSCS -->
  <host name="customer-mdm-ext.oracleindustry.com">
    <!-- START: Base Package Defaults.  Do Not Change. -->
    <dir base="true" ipset="defaultDeny" path="/ouaf/*"/>
    <dir base="true" ipset="defaultDeny" path="/ouaf/webservices/*"/>
    <dir base="true" ipset="defaultDeny" path="/soa-infra/*"/>
    <dir base="true" ipset="defaultDeny" path="/"/>
    <!-- END: Base Package Defaults.  Do Not Change. -->
  </host>
  <host name="customer-test-mdm-ext.oracleindustry.com">
    <!-- START: Base Package Defaults.  Do Not Change. -->
    <dir base="true" ipset="defaultDeny" path="/ouaf/*"/>
    <dir base="true" ipset="defaultDeny" path="/ouaf/webservices/*"/>
    <dir base="true" ipset="defaultDeny" path="/soa-infra/*"/>
    <dir base="true" ipset="defaultDeny" path="/"/>
    <!-- END: Base Package Defaults.  Do Not Change. -->
  </host>
  <host name="customer-dev-mdm-ext.oracleindustry.com">
    <!-- START: Base Package Defaults.  Do Not Change. -->
    <dir base="true" ipset="defaultDeny" path="/ouaf/*"/>
    <dir base="true" ipset="defaultDeny" path="/ouaf/webservices/*"/>
    <dir base="true" ipset="defaultDeny" path="/soa-infra/*"/>
    <dir base="true" ipset="defaultDeny" path="/"/>
    <!-- END: Base Package Defaults.  Do Not Change. -->
  </host>

  <!-- Common Services -->
  <host name="customer-idm-ext.oracleindustry.com">
    <!-- START: Base Package Defaults.  Do Not Change. -->
    <dir base="true" ipset="defaultDeny" path="/identity/*"/>
    <dir base="true" ipset="defaultDeny" path="/ms_oauth/oauth2/*"/>
    <dir base="true" ipset="defaultDeny" path="/oamfed/sp/*"/>
    <!-- END: Base Package Defaults.  Do Not Change. -->
  </host>
</inbound>
```

Example 2: Separate Whitelist for MDM and Web Services

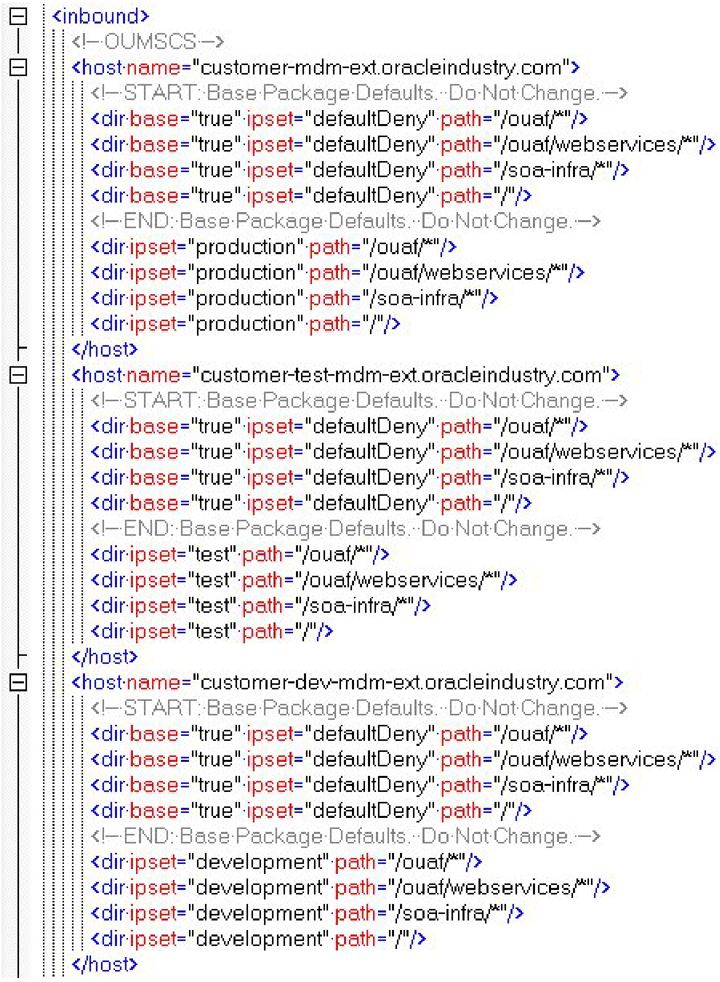
This <inbound> section matches the <ipsets> example 2.



```
<inbound>
  <!-- OUMSCS -->
  <host name="customer-mdm-ext.oracleindustry.com">
    <!-- START: Base Package Defaults.  Do Not Change. -->
    <dir base="true" ipset="defaultDeny" path="/ouaf/*"/>
    <dir base="true" ipset="defaultDeny" path="/ouaf/webservices/*"/>
    <dir base="true" ipset="defaultDeny" path="/soa-infra/*"/>
    <dir base="true" ipset="defaultDeny" path="/"/>
    <!-- END: Base Package Defaults.  Do Not Change. -->
    <dir ipset="myservice" path="/ouaf/webservices/MyWebService/*"/>
    <dir ipset="mdm_users" path="/ouaf/*"/>
    <dir ipset="webservices" path="/ouaf/webservices/*"/>
    <dir ipset="mdm_users" path="/soa-infra/*"/>
    <dir ipset="mdm_users" path="/"/>
  </host>
  <host name="customer-test-mdm-ext.oracleindustry.com">
    <!-- START: Base Package Defaults.  Do Not Change. -->
    <dir base="true" ipset="defaultDeny" path="/ouaf/*"/>
    <dir base="true" ipset="defaultDeny" path="/ouaf/webservices/*"/>
    <dir base="true" ipset="defaultDeny" path="/soa-infra/*"/>
    <dir base="true" ipset="defaultDeny" path="/"/>
    <!-- END: Base Package Defaults.  Do Not Change. -->
    <dir ipset="myservice" path="/ouaf/webservices/MyWebService/*"/>
    <dir ipset="mdm_users" path="/ouaf/*"/>
    <dir ipset="webservices" path="/ouaf/webservices/*"/>
    <dir ipset="mdm_users" path="/soa-infra/*"/>
    <dir ipset="mdm_users" path="/"/>
  </host>
  <host name="customer-dev-mdm-ext.oracleindustry.com">
    <!-- START: Base Package Defaults.  Do Not Change. -->
    <dir base="true" ipset="defaultDeny" path="/ouaf/*"/>
    <dir base="true" ipset="defaultDeny" path="/ouaf/webservices/*"/>
    <dir base="true" ipset="defaultDeny" path="/soa-infra/*"/>
    <dir base="true" ipset="defaultDeny" path="/"/>
    <!-- END: Base Package Defaults.  Do Not Change. -->
    <dir ipset="myservice" path="/ouaf/webservices/MyWebService/*"/>
    <dir ipset="mdm_users" path="/ouaf/*"/>
    <dir ipset="webservices" path="/ouaf/webservices/*"/>
    <dir ipset="mdm_users" path="/soa-infra/*"/>
    <dir ipset="mdm_users" path="/"/>
  </host>
```

Example 3: Different Whitelists for Different Environments

If each environment type (production, test, and development) needed a different IP set, the <inbound> section would look like this. The <ipsets> section would define sets named "production", "test", and "development".



```
<inbound>
  <!-- OUMSCS -->
  <host name="customer-mdm-ext.oracleindustry.com">
    <!-- START: Base Package Defaults.  Do Not Change. -->
    <dir base="true" ipset="defaultDeny" path="/ouaf/*"/>
    <dir base="true" ipset="defaultDeny" path="/ouaf/webservices/*"/>
    <dir base="true" ipset="defaultDeny" path="/soa-infra/*"/>
    <dir base="true" ipset="defaultDeny" path="/"/>
    <!-- END: Base Package Defaults.  Do Not Change. -->
    <dir ipset="production" path="/ouaf/*"/>
    <dir ipset="production" path="/ouaf/webservices/*"/>
    <dir ipset="production" path="/soa-infra/*"/>
    <dir ipset="production" path="/"/>
  </host>
  <host name="customer-test-mdm-ext.oracleindustry.com">
    <!-- START: Base Package Defaults.  Do Not Change. -->
    <dir base="true" ipset="defaultDeny" path="/ouaf/*"/>
    <dir base="true" ipset="defaultDeny" path="/ouaf/webservices/*"/>
    <dir base="true" ipset="defaultDeny" path="/soa-infra/*"/>
    <dir base="true" ipset="defaultDeny" path="/"/>
    <!-- END: Base Package Defaults.  Do Not Change. -->
    <dir ipset="test" path="/ouaf/*"/>
    <dir ipset="test" path="/ouaf/webservices/*"/>
    <dir ipset="test" path="/soa-infra/*"/>
    <dir ipset="test" path="/"/>
  </host>
  <host name="customer-dev-mdm-ext.oracleindustry.com">
    <!-- START: Base Package Defaults.  Do Not Change. -->
    <dir base="true" ipset="defaultDeny" path="/ouaf/*"/>
    <dir base="true" ipset="defaultDeny" path="/ouaf/webservices/*"/>
    <dir base="true" ipset="defaultDeny" path="/soa-infra/*"/>
    <dir base="true" ipset="defaultDeny" path="/"/>
    <!-- END: Base Package Defaults.  Do Not Change. -->
    <dir ipset="development" path="/ouaf/*"/>
    <dir ipset="development" path="/ouaf/webservices/*"/>
    <dir ipset="development" path="/soa-infra/*"/>
    <dir ipset="development" path="/"/>
  </host>
```

Outbound

The <outbound> section contains the whitelist of hosts that are allowed for outbound connections. If outbound connections from the application to your hosts are desired then <value> elements should be added to the existing <routingAlias> element. Any number of <value> elements can be added. The values are one of the following:

- An IPv4 subnet in CIDR notation. Examples: "198.0.2.0/24", "198.51.100.0/26"
- An IPv4 address. Examples: "203.0.113.27", "198.51.100.88"
- A fully qualified domain name. Examples: "host1.example.com", "host2.subdomain.example.com"
- A fully qualified domain name with wildcards. The asterisk (*) and question (?) characters are used as wildcards with the same meaning as the wildcards in path attributes. Examples: "*.example.com", "*.subdomain.example.com", "abc*.example.com"

Make sure to include values in this section for every host to which the application would make an outbound connection. This includes web services and cloud storage.

Listing each individual server is preferable to subnets and wildcards where feasible. If wildcards are used, they should be in the hostname part of the fully qualified domain name rather than the domain name or top-level domain. These are examples of bad/ unsafe use of wildcards:

- companyname*.com
- company*name.com
- *companyname.com
- companyname.*com
- companyname*.co.uk
- *companyname.co.uk

Examples of better use of wildcards:

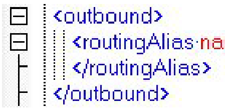
- *.companyname.com
- abc*.companyname.com
- *def.companyname.com
- *.companyname.co.uk

When in doubt of a fully qualified domain (with or without wildcards), "whois" can be used to verify the owner. Remove everything up to and including the first dot (.) after the last wildcard character (* or ?) and do a whois lookup of the name to verify the owner of the domain. The following sites allow whois lookup by domain name:

- https://www.whois.com/whois/
- https://www.networksolutions.com/whois

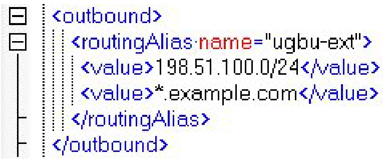
Also note that all outbound connections will use HTTPS protocol and the hosts must have a valid TLS certificate signed by a trusted public certificate authority (Digicert, Symantec/VeriSign, Comodo, etc.).

Example 1 - No Outbound Allowed (default)



```
<outbound>
  <routingAlias name="ugbu-ext">
  </routingAlias>
</outbound>
```

Example 2 - Outbound Subnet and Wildcard Host



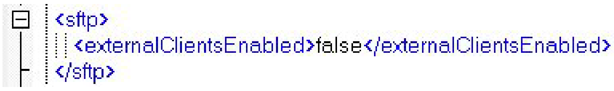
```
<outbound>
  <routingAlias name="ugbu-ext">
    <value>198.51.100.0/24</value>
    <value>*.example.com</value>
  </routingAlias>
</outbound>
```

Sftp

The <sftp> section has a single XML element: <externalClientsEnabled>. The value of this element is either "true" or "false" (default).

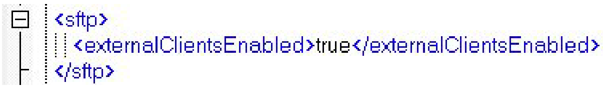
- **false**: do not allow external (non-VPN) clients to access sftp server. Only access via VPN will be allowed
- **true**: allow external clients in addition to clients connecting via VPN to access sftp server.

Example 1: False



```
<sftp>
  <externalClientsEnabled>false</externalClientsEnabled>
</sftp>
```

Example 2: True



```
<sftp>
  <externalClientsEnabled>true</externalClientsEnabled>
</sftp>
```

Validation

To validate your whitelist, execute the following command (assuming the file is named whitelist.xml):

```
java -jar whitelistvalidator.jar whitelist.xml
```

Note that The "template_whitelist.xml" template file should exist in the same folder as the whitelist being validated.

If the whitelist is valid, this command will output a message similar to the following

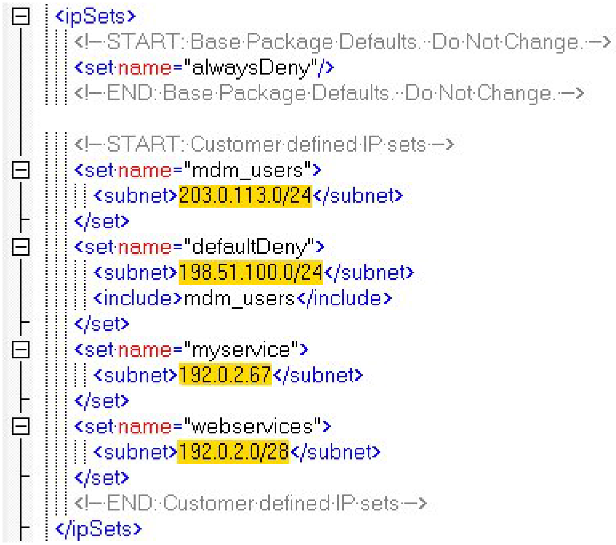
```
file:/C:/Documents/whitelist.xml is valid
```

Otherwise this command will output one or more error messages. See **Validation Errors** on page 2-9 section for details of the various messages.

Host and Subnet Validation

To validate the hosts and subnets defined in your whitelist, open the "whitelist.xml" file in the editor of your choice (Text or XML).

Copy the contents of each of the subnet elements in the <ipSets> section. Note the highlighted text in the screenshot



```
<ipSets>
  <!-- START: Base Package Defaults. Do Not Change. -->
  <set name="alwaysDeny"/>
  <!-- END: Base Package Defaults. Do Not Change. -->

  <!-- START: Customer defined IP sets -->
  <set name="mdm_users">
    <subnet>203.0.113.0/24</subnet>
  </set>
  <set name="defaultDeny">
    <subnet>198.51.100.0/24</subnet>
    <include>mdm_users</include>
  </set>
  <set name="myservice">
    <subnet>192.0.2.67</subnet>
  </set>
  <set name="webservices">
    <subnet>192.0.2.0/28</subnet>
  </set>
  <!-- END: Customer defined IP sets -->
</ipSets>
```

For each of the subnets in your whitelist (excluding duplicates since each subnet only needs to be checked once):

1. Verify that the subnet does not overlap with reserved IP addresses.
2. Find the first IP address in the subnet and do a "whois" lookup to ensure that the subnet is owned by your organization and that the subnet in the whitelist does not extend past the subnet returned by the whois lookup. The following sites allow whois lookup by IP address:
   - https://whois.arin.net/ui/
   - https://www.ultratools.com/tools/ipWhoisLookup
   - https://www.whois.com/whois/
   - https://www.networksolutions.com/whois

Validation Errors

These are the common errors that can appear when running whitelistvalidator.jar.

Unsupported Class Version

This error message indicates that the Wrong java version was used. The whitelistvalidator.jar tool requires Java 8 or later.

Example:

```
java.lang.UnsupportedClassVersionError: com/oracle/ugbu/whitelist/
WhitelistValidator : Unsupported major.minor version 52.0
```

Set must be empty

This error indicates that a <set> element with the given name has child elements (i.e., <subnet> or <include>) when it should not. The set named "alwaysDeny" is reserved to be an empty set.

Example:

```
ERROR: The set named "alwaysDeny" must be empty
```

Duplicate path under host

This error indicates that the multiple <dir> elements exist with the same path under the given <host> element.

Example:

```
ERROR: Duplicate path "/identity/*" under host "customer-idm-
ext.oracleindustry.com"
```

Include is prior to set definition

This error indicates that an <include> element exists for an IP set prior to the definition of the IP set.

Example

```
ERROR: include of "foo" inside set "bar" is prior to set definition
```

Include is self-referential

This error indicates that an <include> element exists for an IP set inside the definition of the same IP set.

Example

```
ERROR: include of "foo" inside set "foo" is self-referential
```

Not valid

This error indicates that the whitelist does not validate against the XML schema. The message should include line number, column number, and a brief message

Example:

```
whitelist.xml is NOT valid.  Reason:
org.xml.sax.SAXParseException; systemId: file:/C:/whitelist.xml;
lineNumber: 1; columnNumber: 147; cvc-elt.1: Cannot find the
declaration of element 'whitelistConfig'.
```

Additional routing alias not from template

This error indicates that a <routingAlias> element in the whitelist does not match one of the <routingAlias> elements from the template.

Example:

```
ERROR: Additional routing alias not from template: "abc", "def"
```

Routing alias from template not found

This error indicates that a <routingAlias> element from the template was not included in the whitelist.

Example:

```
ERROR: Routing alias from template not found in whitelist: "ugbu-
ext"
```

Host element names in whitelist and template do not match

This error indicates that the names of the <host> elements in the whitelist do not match the ones in the template. Either a <host> was added or removed or the name attribute of a <host> was modified.

Example:

```
ERROR: <host> element names in whitelist and template do not match
```

Base dir elements do not match template

This error indicates that the <dir> elements with the attribute base="true" do not match the ones in the template. Either these <dir> elements were modified/removed or new <dir> elements with the attribute base="true" were added.

Example:

```
ERROR: base <dir> elements under <host> "cust-idm-
ext.oracleindustry.com" do not match template
```

Could not find template

This error indicates that the template file could not be found. Without the template, only minimal validation can be done.

Example:

```
WARNING: Could not find template_whitelist.xml
```

Could not open template

This error indicates that the template file was found, but could not be opened. Without the template, only minimal validation can be done.

Example:

```
WARNING: Could not open C:\template_whitelist.xml
```

Attach Whitelist to the Service Request

Once you have completed the edits to your whitelist and validated the file, attach the "whitelist.xml" file to the service request and ask for the whitelist to be applied. If this is an update to an existing whitelist and the Common Services host and IP sets are not changed from the previous whitelist, indicate that common services are unchanged in the service request.

- If common services are unchanged, then whitelist changes will be rolled out on development and test domains first. Once these are validated, then the changes will be rolled out in production and disaster recovery.
- Otherwise, changes will be rolled out on development, test, production, and disaster recovery.

Note that updating the whiltelist requires some downtime.

Verify Whitelist

Once Oracle Customer Success has updated the service request to indicate that the whitelist has been applied, the whitelist must be verified. Log in and use the cloud domains from various hosts to verify that clients are allowed or denied as appropriate.