

Links to Other API pages: [Cisco Unity Connection APIs](#)

CUPI Guide Contents
API Overview
Index of All CUPI Documentation

Contents

- [1 About Routing Rules and Routing Rule Conditions](#)
- [2 Routing Rules](#)
 - ◆ [2.1 Listing and Viewing](#)
 - ◆ [2.2 Searching](#)
 - ◆ [2.3 Creating](#)
 - ◆ [2.4 Updating](#)
 - ◆ [2.5 Deleting](#)
 - ◆ [2.6 Changing Routing Rule Indices](#)
- [3 Routing Rule Conditions](#)
 - ◆ [3.1 Listing and Viewing](#)
 - ◆ [3.2 Creating](#)
 - ◆ [3.3 Updating](#)
 - ◆ [3.4 Deleting](#)
- [4 Routing Rule Examples](#)
 - ◆ [4.1 Transfer to a User or Call Handler](#)
 - ◆ [4.2 Go to a User or Call Handler's Greeting](#)
 - ◆ [4.3 Go to an Interview Handler](#)
 - ◆ [4.4 Go to a Directory Handler](#)
 - ◆ [4.5 Go to a Specific Conversation](#)
- [5 Routing Rule Condition Examples](#)
 - ◆ [5.1 Condition Based on a Phone Number](#)
 - ◆ [5.2 Condition Based on a Port](#)
 - ◆ [5.3 Condition Based on a Phone System](#)
 - ◆ [5.4 Condition Based on a Schedule Set](#)
- [6 Enumeration Types](#)
 - ◆ [6.1 Routing Rule RouteAction](#)
 - ◆ [6.2 Routing Rule RouteTargetHandlerObjectType](#)
 - ◆ [6.3 Routing Rule State](#)
 - ◆ [6.4 Routing Rule Type](#)
 - ◆ [6.5 Routing Rule Condition Operator](#)
 - ◆ [6.6 Routing Rule Condition Parameter](#)
 - ◆ [6.7 Routing Rule RouteTargetConversation](#)

About Routing Rules and Routing Rule Conditions

This page contains information on how to use the API to create, list, update, and delete Routing Rules and their support objects, Routing Rule Conditions. Please note that the syntax for Routing Rules and Routing Rule Conditions is relatively complicated. Retrieving existing rules and conditions is not too difficult, but creating and modifying them via CUPI should only be done with great care. If the Routing Rules or Routing Rule Conditions are misconfigured, they might not display correctly in Cisco Unity Connection Administration (CUCA), and calls into Cisco Unity Connection might not get answered or might route to the wrong Conversation. Routing Rules and Routing Rule Conditions both include some fields that are enumeration types (meaning numeric constants), which makes understanding these fields more complicated. Some examples can be found near the end of this document.

Each Routing Rule has a Type (Direct, Forwarded, or Both) and a RuleIndex (starting at index 0 and increasing by 1 for each additional rule). When Cisco Unity Connection answers a call of the specified type, it attempts to match the Routing Rules in order starting at RuleIndex 0 and continuing with each successive index until a rule matches. The factory default rules (which are built in and cannot be deleted) should match any call that doesn't match a custom rule. An individual Routing Rule also specifies the destination for the call in the event the rule matches, such as a user, call handler, interview handler, directory handler, or conversation.

In addition, each Routing Rule can have 0 or more Routing Rule Conditions, and a call must match all of these conditions before it is considered a match. If a Routing Rule does not have any Routing Rule Conditions, then all calls match it. The default Routing Rule, which transfers to the Opening Greeting, does not have any Routing Rule Conditions and is usually at the highest RuleIndex, so if a call does not match any other Routing Rules it ends up at the Opening Greeting. An individual Routing Rule Condition can specify any of the following conditions: call information such as Calling Number, Called Number, or Redirecting Number (Forwarded Routing Rules only); port or phone system; or schedule.

Routing Rules

Listing and Viewing

The following is an example of a GET that lists all Routing Rules:

```
GET https://<connection-server>/vmrest/routingrules
```

The following is the response from the above GET request:

```
200
OK
```

```
<?xml version="1.0" encoding="UTF-8"?>
<RoutingRules total="3">
  <RoutingRule>
    <URI>/vmrest/routingrules/a2f64719-c82f-491c-996f-ee543e6d7e38</URI>
    <ObjectId>a2f64719-c82f-491c-996f-ee543e6d7e38</ObjectId>
    <DisplayName>Opening Greeting</DisplayName>
    <State>0</State>
    <RuleIndex>2</RuleIndex>
    <Type>3</Type>
    <Flags>3</Flags>
    <RouteTargetConversation>PHTransfer</RouteTargetConversation>
    <RouteTargetHandlerObjectId>198ff222-1129-486c-9bef-c05057dae950</RouteTargetHandlerObjectId>
```

Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_--_Routing_Rules

```
<RouteTargetHandlerDisplayName>Opening Greeting</RouteTargetHandlerDisplayName>
<RouteTargetHandlerObjectType>3</RouteTargetHandlerObjectType>
<RouteAction>2</RouteAction>
<LanguageCode>0</LanguageCode>
<UseDefaultLanguage>true</UseDefaultLanguage>
<UseCallLanguage>true</UseCallLanguage>
<CallType>2</CallType>
<SearchSpaceObjectId>02cb3fba-1525-4768-89f6-64f4712ea177</SearchSpaceObjectId>
<SearchSpaceURI>/vmrest/searchspaces/02cb3fba-1525-4768-89f6-64f4712ea177</SearchSpaceURI>
<Undeletable>true</Undeletable>
<RoutingRuleConditionsURI>/vmrest/routingrules/a2f64719-c82f-491c-996f-ee543e6d7e38/routingrul
</RoutingRule>
<RoutingRule>
  <URI>/vmrest/routingrules/857c5cd6-0153-4673-a917-27c59bf5ce37</URI>
  <ObjectId>857c5cd6-0153-4673-a917-27c59bf5ce37</ObjectId>
  <DisplayName>Attempt Sign In</DisplayName>
  <State>0</State>
  <RuleIndex>1</RuleIndex>
  <Type>1</Type>
  <Flags>3</Flags>
  <RouteTargetConversation>AttemptSignIn</RouteTargetConversation>
  <RouteTargetHandlerObjectType>0</RouteTargetHandlerObjectType>
  <RouteAction>2</RouteAction>
  <LanguageCode>0</LanguageCode>
  <UseDefaultLanguage>true</UseDefaultLanguage>
  <UseCallLanguage>true</UseCallLanguage>
  <CallType>2</CallType>
  <SearchSpaceObjectId>02cb3fba-1525-4768-89f6-64f4712ea177</SearchSpaceObjectId>
  <SearchSpaceURI>/vmrest/searchspaces/02cb3fba-1525-4768-89f6-64f4712ea177</SearchSpaceURI>
  <Undeletable>true</Undeletable>
  <RoutingRuleConditionsURI>/vmrest/routingrules/857c5cd6-0153-4673-a917-27c59bf5ce37/routingrul
</RoutingRule>
<RoutingRule>
  <URI>/vmrest/routingrules/8b7ea5be-48c5-4e7b-bc9d-c06f052074a4</URI>
  <ObjectId>8b7ea5be-48c5-4e7b-bc9d-c06f052074a4</ObjectId>
  <DisplayName>Attempt Forward</DisplayName>
  <State>0</State>
  <RuleIndex>0</RuleIndex>
  <Type>2</Type>
  <Flags>3</Flags>
  <RouteTargetConversation>AttemptForward</RouteTargetConversation>
  <RouteTargetHandlerObjectType>0</RouteTargetHandlerObjectType>
  <RouteAction>2</RouteAction>
  <LanguageCode>0</LanguageCode>
  <UseDefaultLanguage>true</UseDefaultLanguage>
  <UseCallLanguage>true</UseCallLanguage>
  <CallType>2</CallType>
  <SearchSpaceObjectId>02cb3fba-1525-4768-89f6-64f4712ea177</SearchSpaceObjectId>
  <SearchSpaceURI>/vmrest/searchspaces/02cb3fba-1525-4768-89f6-64f4712ea177</SearchSpaceURI>
  <Undeletable>true</Undeletable>
  <RoutingRuleConditionsURI>/vmrest/routingrules/8b7ea5be-48c5-4e7b-bc9d-c06f052074a4/routingrul
</RoutingRule>
</RoutingRules>
```

To retrieve a sorted list of all Routing Rules, add the following query parameter to the GET: sort=(column [asc | desc])

For example, to retrieve a list of all Routing Rules sorted by RuleIndex in ascending order:

```
GET https://<connection-server>/vmrest/routingrules?sort=(RuleIndex%20asc)
```

To retrieve a specific Routing Rule by its ObjectId:

Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_--_Routing_Rules

GET https://<connection-server>/vmrest/routingrules/<objectid>

JSON Example

To list all the routing rules, use the following command:

Request URI:

```
GET https://<connection-server>/vmrest/routingrules
Accept: application/json
Connection: keep-alive
```

The following is the response from the above *GET* request and the actual response will depend upon the information given by you:

```
{
  "@total": "1",
  "RoutingRule": [
    {
      "URI": "/vmrest/routingrules/6ba11848-3eb2-47b9-ab2b-b1464165a1bf",
      "ObjectId": "6ba11848-3eb2-47b9-ab2b-b1464165a1bf",
      "DisplayName": "Opening Greeting",
      "State": "1",
      "RuleIndex": "20",
      "Type": "3",
      "Flags": "3",
      "RouteTargetConversation": "PHTransfer",
      "RouteTargetHandlerObjectId": "e6fd6ae4-7bf5-4a46-b994-e9fd13d83c30",
      "RouteTargetHandlerDisplayName": "Opening Greeting",
      "RouteTargetHandlerObjectType": "3",
      "RouteAction": "2",
      "LanguageCode": "0",
      "UseDefaultLanguage": "true",
      "UseCallLanguage": "true",
      "CallType": "2",
      "SearchSpaceObjectId": "f8a97bde-8f75-4c84-b2e6-aa6e54cc26c2",
      "SearchSpaceURI": "/vmrest/searchspaces/f8a97bde-8f75-4c84-b2e6-aa6e54cc26c2",
      "Undeletable": "true",
      "RoutingRuleConditionsURI": "/vmrest/routingrules/6ba11848-3eb2-47b9-ab2bb1464165a1bf/rc",
      "CallHandlerURI": "/vmrest/handlers/callhandlers",
      "DirectoryHandlerURI": "/vmrest/handlers/directoryhandlers",
      "InterviewHandlerURI": "/vmrest/handlers/interviewhandlers",
      "ConversationURI": "/vmrest/conversations?query=(CnvRoutingRuleForwardedCall%20is%201)",
      "SearchSpaceFetchURI": "/vmrest/searchspaces"
    }
  ]
}
```

Response Code: 200

Searching

To retrieve a list of Routing Rules that meet a specified search criteria, add the following query parameter to a GET: query=(column [is | startswith] value)

For example, to find the Routing Rule at RuleIndex 1:

```
GET https://<connection-server>/vmrest/routingrules?query=(RuleIndex%20is%201)
```

Creating

The only required field for creating a Routing Rule is DisplayName. All other Routing Rule fields are optional. A Routing Rule that is created with default fields has Type=1 (Direct) and routes to the System Directory Handler. Some examples of other types of Routing Rules can be found at the end of this document.

Note that RuleIndex cannot be specified when creating a new Routing Rule; all new Routing Rules are created at RuleIndex 0, and the RuleIndex of all other Routing Rules is incremented by 1 when the new Routing Rule is created.

The following is an example of a POST that creates a Routing Rule with the name "My Routing Rule" of Type 1 (Direct):

```
POST https://<connection-server>/vmrest/routingrules

<RoutingRule>
  <DisplayName>My Routing Rule</DisplayName>
  <Type>1</Type>
</RoutingRule>
```

The following is the response from the above POST request:

```
201
Created
/vmrest/routingrules/bd73528d-e810-4533-8aab-89de32013885
```

JSON Example

To create new routing rule, do the following:

Request URI:

```
POST https://<connection-server>/vmrest/routingrules
Accept: application/json
Content-Type: application/json
Connection: keep-alive
```

Request Body:

```
{
  "DisplayName": "Texoma Routing Rule",
  "Type": "1"
}
```

The following is the response from the above *POST* request and the actual response will depend upon the information given by you:

```
Response Code: 201
```

Updating

The following is an example of a PUT request that modifies the DisplayName of an existing Routing Rule:

```
PUT https://<connection-server>/vmrest/routingrules/<objectid>

<RoutingRule>
```

Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_--_Routing_Rules

```
<DisplayName>Changed Display Name</DisplayName>
</RoutingRule>
```

The following is the response from the above PUT request:

```
204
No Content
```

Note that the RuleIndex of a Routing Rule cannot be modified with this method. Rather, the ordering of the entire Routing Rules collection can be modified, as described later in this document.

JSON Example

To update display name of routing rule, do the following:

```
PUT https://<connection-server>/vmrest/routingrules/<routingrulesobjectId>
Accept: application/json
Content-Type: application/json
Connection: keep-alive
```

Request Body:

```
{
  "DisplayName": "Texoma_1 Routing Rule"
}
```

The following is the response from the above *PUT* request and the actual response will depend upon the information given by you:

```
Response Code: 204
```

Deleting

The following is an example of a DELETE request that deletes a Routing Rule:

```
DELETE https://<connection-server>/vmrest/routingrules/<objectId>
```

The following is the response from the above DELETE request:

```
204
No Content
```

JSON Example

```
DELETE https://<connection-server>/vmrest/routingrules/<routingrulesobjectId>
Accept: application/json
Connection: keep-alive
```

```
Response Code: 204
```

Changing Routing Rule Indices

As noted in previous sections, a Routing Rule's RuleIndex field cannot be modified when creating or updating a Routing Rule. Instead, you can change a Routing Rule's RuleIndex only in the context of ordering

Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_--_Routing_Rules

all of the Routing Rules in a collection.

The following is an example of a PUT request that modifies the RuleIndices of a collection of Routing Rules:

```
PUT https://<connection-server>/vmrest/routingrules

<RoutingRules>
  <RoutingRule>
    <ObjectId>aabbccdd-1111-2222-3333-1234567890ab</ObjectId>
  </RoutingRule>
  <RoutingRule>
    <ObjectId>12345678-abcd-abcd-abcd-123412341234</ObjectId>
  </RoutingRule>
  <RoutingRule>
    <ObjectId>99990000-1234-1234-1234-abcdef012345</ObjectId>
  </RoutingRule>
</RoutingRules>
```

The following is a response from the above PUT request:

```
204
No Content
```

After this PUT request is processed, the Routing Rule with ObjectId aabbccdd-1111-2222-3333-1234567890ab would have RuleIndex 0, the Routing Rule with ObjectId 12345678-abcd-abcd-abcd-123412341234 would have RuleIndex 1, and the Routing Rule with ObjectId 99990000-1234-1234-1234-abcdef012345 would have RuleIndex 2.

JSON Example

To update the order of routing rule, do the following:

Request URI:

```
PUT https://<connection-server>/vmrest/routingrules
Accept: application/json
Content-Type: application/json
Connection: keep-alive
```

Request Body:

```
{
  "RoutingRule": [
    {
      "ObjectId": "d083fa0b-76ec-4bec-add3-c54aae6033df"
    },
    {
      "ObjectId": "983efa07-ccb2-4b61-a1e5-083e16ca7513"
    },
    {
      "ObjectId": "bd8b8092-1c08-4d25-8d35-116d7a677bf4"
    },
    {
      "ObjectId": "7ee01b59-a619-42f9-9311-5df0f27763ac"
    }
  ]
}
```

Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_-_Routing_Rules

The following is the response from the above *PUT* request and the actual response will depend upon the information given by you:

Response Code: 204

Routing Rule Conditions

Each Routing Rule can have a collection of Routing Rule Conditions. The Routing Rule Conditions that belong to a specified Routing Rule can be accessed at `/vmrest/routingrules/<routingruleobjectid>/routingruleconditions`. When first created, a Routing Rule does not have any Routing Rule Conditions, which means that every call matches it and is routed to its destination.

Listing and Viewing

The following is an example of a GET that lists all Routing Rule Conditions that belong to a specified Routing Rule:

```
GET https://<connection-server>/vmrest/routingrules/<routingruleobjectid>/routingruleconditions
```

The following is the response from the above GET request:

```
200
OK
```

```
<?xml version="1.0" encoding="UTF-8"?>
<RoutingRuleConditions total="2">
  <RoutingRuleCondition>
    <URI>/vmrest/routingrules/d2f86bc0-4cab-4c29-9321-d756fe3add6a/routingruleconditions/2a5263b7-
    <ObjectId>2a5263b7-c896-47ff-ad8c-749e0459a28c</ObjectId>
    <RoutingRuleObjectId>d2f86bc0-4cab-4c29-9321-d756fe3add6a</RoutingRuleObjectId>
    <RoutingRuleURI>/vmrest/routingrules/d2f86bc0-4cab-4c29-9321-d756fe3add6a</RoutingRuleURI>
    <Parameter>1</Parameter>
    <Operator>2</Operator>
    <OperandValue>12345</OperandValue>
  </RoutingRuleCondition>
  <RoutingRuleCondition>
    <URI>/vmrest/routingrules/d2f86bc0-4cab-4c29-9321-d756fe3add6a/routingruleconditions/d9635722-
    <ObjectId>d9635722-f1ee-4459-a2d4-533f85cd2c24</ObjectId>
    <RoutingRuleObjectId>d2f86bc0-4cab-4c29-9321-d756fe3add6a</RoutingRuleObjectId>
    <RoutingRuleURI>/vmrest/routingrules/d2f86bc0-4cab-4c29-9321-d756fe3add6a</RoutingRuleURI>
    <Parameter>7</Parameter>
    <Operator>2</Operator>
    <OperandValue>4cc6c1be-e1ba-4fcf-be81-95bce20acbec</OperandValue>
  </RoutingRuleCondition>
</RoutingRuleConditions>
```

To retrieve a specific Routing Rule Condition by its ObjectId:

```
GET https://<connection-server>/vmrest/routingrules/<routingruleobjectid>/routingruleconditions/<objectid>
```


Creating

The required fields for creating a Routing Rule Condition are Parameter, Operator, and OperandValue. All other Routing Rule Condition fields are optional.

The following is an example of a POST that creates a Routing Rule Condition with Parameter=2 (Dialed Number), Operator=2 (Equals), and OperandValue=8675309:

```
POST https://<connection-server>/vmrest/routingrules/d2f86bc0-4cab-4c29-9321-d756fe3add6a/routingrules/d2f86bc0-4cab-4c29-9321-d756fe3add6a/routingruleconditions

<RoutingRuleCondition>
  <Parameter>2</Parameter>
  <Operator>2</Operator>
  <OperandValue>8675309</OperandValue>
</RoutingRuleCondition>
```

The following is the response from the above POST request:

```
201
Created
/vmrest/routingrules/d2f86bc0-4cab-4c29-9321-d756fe3add6a/routingruleconditions/74b50eeb-8c06-4e86-9c00-000000000000
```

Updating

The following is an example of a PUT request that modifies the OperandValue of an existing Routing Rule Condition to 5551212:

```
PUT https://<connection-server>/vmrest/routingrules/d2f86bc0-4cab-4c29-9321-d756fe3add6a/routingrules/d2f86bc0-4cab-4c29-9321-d756fe3add6a/routingruleconditions/74b50eeb-8c06-4e86-9c00-000000000000

<RoutingRuleCondition>
  <OperandValue>5551212</OperandValue>
</RoutingRuleCondition>
```

The following is the response from the above PUT request:

```
204
No Content
```

Deleting

The following is an example of a DELETE request that deletes a Routing Rule Condition:

```
DELETE https://<connection-server>/vmrest/routingrules/d2f86bc0-4cab-4c29-9321-d756fe3add6a/routingrules/d2f86bc0-4cab-4c29-9321-d756fe3add6a/routingruleconditions/74b50eeb-8c06-4e86-9c00-000000000000
```

The following is the response from the above DELETE request:

```
204
No Content
```

Routing Rule Examples

It can be relatively complicated to specify the destination for a Routing Rule. The fields `RouteAction`, `RouteTargetConversation`, and `RouteTargetHandlerObjectId` are used together to specify the destination. The fields `RouteTargetHandlerDisplayName` and `RouteTargetHandlerObjectType` should not be specified when creating or updating a Routing Rule; rather they are set automatically based on the other specified values.

The following sections give some examples of how to specify different types of route destinations by using these fields. Note that all other fields in the Routing Rule objects have been omitted for brevity.

This list of examples covers all of the Routing Rules that can be configured by using Cisco Unity Connection Administration (CUCA). Creating a different type of Routing Rule by using CUPI is likely to yield a Routing Rule that will not display properly in CUCA, and it also may not operate as expected when Connection tries to route a call.

Transfer to a User or Call Handler

The following Routing Rule will go to (`Action=2`) the call handler `aab5eab2-38f7-4231-a3be-bf2a8fde820c` (which happens to be a user's primary call handler) by using the `PHTransfer` Conversation (meaning it will attempt to transfer to the call handler). If you want to transfer to a different call handler, set `RouteTargetCallHandlerObjectId` to it.

```
<RoutingRule>
  <RouteAction>2</RouteAction>
  <RouteTargetConversation>PHTransfer</RouteTargetConversation>
  <RouteTargetHandlerObjectId>aab5eab2-38f7-4231-a3be-bf2a8fde820c</RouteTargetHandlerObjectId>
</RoutingRule>
```

Go to a User or Call Handler's Greeting

The following Routing Rule will go to (`Action=2`) the call handler `aab5eab2-38f7-4231-a3be-bf2a8fde820c` (which happens to be a user's primary call handler) by using the `PHGreeting` Conversation (meaning it will go directly to the call handler's greeting, bypassing transfer). If you want to go to a different call handler's greeting, set `RouteTargetCallHandlerObjectId` to it.

```
<RoutingRule>
  <RouteAction>2</RouteAction>
  <RouteTargetConversation>PHGreeting</RouteTargetConversation>
  <RouteTargetHandlerObjectId>aab5eab2-38f7-4231-a3be-bf2a8fde820c</RouteTargetHandlerObjectId>
</RoutingRule>
```

Go to an Interview Handler

The following Routing Rule will go to (`Action=2`) the interview handler `60367a3d-213e-4696-bd2a-f8d46091f2bf` by using the `PHInterview` Conversation.

```
<RoutingRule>
  <RouteAction>2</RouteAction>
```

Cisco_Unity_Connection_Provisioning_Interface_(CUPI)_API_--_Routing_Rules

```
<RouteTargetConversation>PHInterview</RouteTargetConversation>
<RouteTargetHandlerObjectId>60367a3d-213e-4696-bd2a-f8d46091f2bf</RouteTargetHandlerObjectId>
</RoutingRule>
```

Go to a Directory Handler

The following Routing Rule will go to (Action=2) the directory handler e6304212-d846-4957-b364-ebe323251007 by using the AD Conversation.

```
<RoutingRule>
  <RouteAction>2</RouteAction>
  <RouteTargetConversation>AD</RouteTargetConversation>
  <RouteTargetHandlerObjectId>e6304212-d846-4957-b364-ebe323251007</RouteTargetHandlerObjectId>
</RoutingRule>
```

Go to a Specific Conversation

The following Routing Rule will go to (Action=2) the BroadcastMessageAdministrator Conversation. Note that RouteTargetHandlerObjectId is ignored if the RouteTargetConversation is anything other than PHTransfer, PHGreeting, PHInterview, or AD.

```
<RoutingRule>
  <RouteAction>2</RouteAction>
  <RouteTargetConversation>BroadcastMessageAdministrator</RouteTargetConversation>
</RoutingRule>
```

Routing Rule Condition Examples

Routing Rule Conditions are not as complicated as Routing Rules, but care should still be taken when creating or modifying them. The fields Parameter, Operator, and OperandValue are used to specify the conditions.

This list of examples covers all of the Routing Rule Conditions that can be configured by using Cisco Unity Connection Administration (CUCA). Creating a different type of Routing Rule Condition by using CUPI is likely to yield a Routing Rule Condition that will not display properly in CUCA, and it also may not operate as expected when Connection tries to match a call.

Condition Based on a Phone Number

The following Routing Rule Condition specifies a match of Calling Number (Parameter=1) Equals (Operator=2) 1234 (OperandValue). A Routing Rule Condition to match on Called Number would use Parameter=2, and a Routing Rule Condition to match on Forwarded Number (valid only on Forwarded Routing Rules) would use Parameter=3. Other Operators are allowed when matching Phone Numbers, such as In and Less Than. With the Equals Operator, the OperandValue can include the wildcards * and ?, and with the In Operator, the OperandValue can include comma-separated ranges of numbers like 2000-2199, 3001-3199, 4001.

```
<RoutingRuleCondition>
  <Parameter>1</Parameter>
  <Operator>2</Operator>
  <OperandValue>1234</OperandValue>
</RoutingRuleCondition>
```

Condition Based on a Port

The following Routing Rule Condition specifies a match of a Port (Parameter=5) Equals (Operator=2) ObjectId b79765f1-e14f-47b6-b8e1-00479909f710 (OperandValue). Creating a Routing Rule Condition to match based on a Port has the side effect of setting MediaPortObjectId to the same value as OperandValue. When matching Ports, only the Equals Operator is allowed, and the OperandValue must be a single ObjectId.

```
<RoutingRuleCondition>
  <Parameter>5</Parameter>
  <Operator>2</Operator>
  <OperandValue>b79765f1-e14f-47b6-b8e1-00479909f710</OperandValue>
</RoutingRuleCondition>
```

Condition Based on a Phone System

The following Routing Rule Condition specifies a match of a Phone System (Parameter=9) Equals (Operator=2) ObjectId 2eb79e66-1e53-415c-9222-36665e0e76ae (OperandValue). Creating a Routing Rule Condition to match based on a Port has the side effect of setting MediaSwitchObjectId to the same value as OperandValue. When matching Phone Systems, only the Equals Operator is allowed, and the OperandValue must be a single ObjectId.

```
<RoutingRuleCondition>
  <Parameter>9</Parameter>
  <Operator>2</Operator>
  <OperandValue>2eb79e66-1e53-415c-9222-36665e0e76ae</OperandValue>
</RoutingRuleCondition>
```

Condition Based on a Schedule Set

The following Routing Rule Condition specifies a match of a Schedule Set (Parameter=7) Equals (Operator=2) ObjectId 4cc6c1be-e1ba-4fcf-be81-95bce20acbec (OperandValue). Note that you must specify the ObjectId of a Schedule Set, not a Schedule. When matching Schedule Sets, only the Equals Operator is allowed, and the OperandValue must be a single ObjectId.

```
<RoutingRuleCondition>
  <Parameter>7</Parameter>
  <Operator>2</Operator>
  <OperandValue>4cc6c1be-e1ba-4fcf-be81-95bce20acbec</OperandValue>
</RoutingRuleCondition>
```

Enumeration Types

Routing Rules and Routing Rule Conditions have several fields that are enumeration types, meaning the fields are numeric constants whose meaning is not immediately obvious. The following tables list these constants and their meanings.

Routing Rule RouteAction

The RouteAction field in a Routing Rule is read/write, and can take the following values. It defaults to 2, and should be set to that value in most cases (it is set to 2 in all of the examples above).

Value	Description
0	Ignore (take no action)
1	Hang up call
2	Go to specified object
3	Error (Play error message)
4	Take a message
5	Skip Greeting
6	Restart Greeting
7	Transfer to Alternate Contact Number
8	Route from Next Call Routing Rule

Routing Rule RouteTargetHandlerObjectType

The RouteTargetHandlerObjectType field in a Routing Rule is read-only and can take various values. The relevant values are listed here. As noted previously, it will automatically be set to the type of object specified by the RouteTargetHandlerObjectId field.

Value	Description
3	Call Handler
5	Interview Handler
6	Directory Handler

Routing Rule State

The State field in a Routing Rule is read/write, and can take the following values. It defaults to 0.

Value	Description
-------	-------------

0	Active
1	Inactive
2	Invalid

Routing Rule Type

The Type field in a Routing Rule is read/write, and can take the following values. It defaults to 1.

Value	Description
1	Direct
2	Forwarded
3	Both (normally only used by Opening Greeting)

Routing Rule Condition Operator

The Operator field in a Routing Rule Condition is read/write, and can take the following values. It does not have a default value.

Value	Description
1	In
2	Equals
3	Greater than
4	Less than
5	Less than or equal to
6	Greater than or equal to

Routing Rule Condition Parameter

The Parameter field in a Routing Rule Condition is read/write, and can take the following values. It does not have a default value. Several values are legacy and should not be used.

Value	Description
1	Calling Number
2	Dialed Number
3	Forwarding Number
5	PortID

7	Schedule Set
9	Phone System

Routing Rule RouteTargetConversation

The RouteTargetConversation field in a Routing Rule is read/write, and can take the following values. It defaults to AD. Although it is not an enumeration type, only certain string values are valid Conversation names. For some Conversations, it is required to specify a Route_TargetHandlerObjectId as well. Examples of how to use several of these Conversations can be found earlier in this document.

Value	Description	TargetHandler
AD	Directory Conversation	Directory Handler
PHTransfer	Transfer to User or Call Handler	User or Call Handler
PHGreeting	Play Greeting of User or Call Handler	User or Call Handler
PHInterview	Interview Conversation	Interview Handler
AttemptForward	Forwards the call to the User's Greeting if the Forwarding Number matches a User	n/a
AttemptSignIn	Sends the call to a User's Sign-in if the Calling Number matches a User	n/a
BroadcastMessageAdministrator	Sends the call to a Conversation for Sending Broadcast Messages	n/a
SystemTransfer	Sends the call to a Conversation allowing the caller to transfer to a number they specify (assuming the restriction table allows it)	n/a
CheckedOutGuest	Sends the call to a Conversation for checked-out hotel guests	n/a
GreetingsAdministrator	Sends the call to a Conversation allowing changing greetings by phone	n/a
ReverseTrapConv	Connects to Visual Voicemail	n/a
SubSignIn	Sends the call to the Sign-In Conversation, which prompts the user to enter their ID	n/a
ConvUtilsLiveRecord	Sends the call to the Live-Record pilot number configured on CUCM	n/a
SubSysTransfer	Similar to SystemTransfer, but requires users sign-in first (so unknown callers cannot use it)	n/a