

# Using IOS Translation Rules – Creating Scalable Dial Plans for VoIP Networks

Document ID: 20595

## Contents

### Introduction

#### Prerequisites

- Requirements
- Components Used
- Conventions
- Background Information

#### Configure

- Network Diagram
- Configurations

#### Verify

#### Troubleshoot

#### Related Information

## Introduction

This document provides a sample configuration for creating scalable dial plans for a VoIP network using IOS translation rules. As you install integrated voice and data networks, one issue frequently encountered is how to manage the numbering plans of the indial ranges at different locations. Depending on the type of exchange, signaling protocol standards and even location, the service provider could pass similar number ranges to the subscriber equipment at each remote site. If these calls are being routed back to a central site, there could be an overlap in the called numbers that originate from each of the remote sites. Since the PBX makes the routing decision based on unique called numbers, this could cause problems with automatic call distribution (ACD) queues on private branch exchange (PBX) systems . For example, calls from each site may need to be directed to particular operators who speak the local language from where the call originated. If the called numbers from each site overlap, there is not any way of identifying the origin of a call, therefore the PBX is not able to route the call to the correct ACD queue.

Some remote sites may be provided with a 2–digit indial number range while other sites may have 3– or 4–digit indial ranges, so the called numbers could be from [00 – 99] to [0000 – 9999]. With these number ranges, the main site router would need configurations to handle 2–, 3– and 4–digit numbering plans. This could add to the overall complexity of the router configuration.

The solution to these issues is to use IOS digit translation rules at each remote site to prepend digits to the number range that comes in from the telephone network. This then creates a standard numbering plan across the customer's network and allows new sites to be gradually added without major changes to the rest of the network.

**Note:** Cisco IOS digit translation rules allow users to manipulate the automatic number identification (ANI) digits of a call which is the *calling* number or Dialed Number Identification Service (DNIS) digits of a call which is the *called* number. Optionally, the numbering types of these fields can also be modified using the translation rules.

In the case study presented in this document, the goal is to standardize the numbering plan. Suggestions include:

- Use a trunk access code.
- Use a site (area) code. This is an easy way to create a unique site code to use the local public switched telephone network (PSTN) area code of the remote site.
- Expand (pad out) the original called number with leading zeros to produce a 4–digit extension number.

## Prerequisites

## Requirements

Readers should be knowledgeable of:

- Configuring voice and pots dial–peer

## Components Used

The information in this document is based on these software and hardware versions:

- Cisco 3660 and 7206VXR routers running IOS version 12.3(4)
- IOS translation rules were introduced in IOS version 12.07XK1 and are available on all Cisco platforms that support voice ports.

The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using it.

## Conventions

For more information on document conventions, refer to the Cisco Technical Tips Conventions.

## Background Information

This section uses the case study: "Calcutta (remote site)" example to explain the implementation for translation rules. Refer to the Network Diagram for more information.

In our example, the remote site (Calcutta) has an E1 trunk that passes a 3–digit dial string in the range of 000 to 999. The Main Site router accepts calls from multiple remote sites that may also cover the range of 000 to 999.

Translation rules are used in the Calcutta router to prepend a site code that uniquely identifies calls from Calcutta (based on the called number) to the Main Site PBX. For example, the area code dial string "033" is used as a site code. In addition, a trunk access code digit is added to allow selection of specific trunks at the destination Main Site router. These statements illustrate the translation procedure in the Calcutta router:

- Original Called number (any 3 digits in the range 000 to 999) " ... "
- Translated Called number "**1033**... " digit "1": trunk access code, digits "033": site code

When a call is placed from the Calcutta site, the Calcutta router applies the translation rule applied in its POTS voice port . You can verify this in the voice–port 1/0:1 configuration of the Calcutta router.

The translation rule is applied to the voice port where the call comes in to the router. When a call comes in from the telephone network towards the router, the called number is translated before it is

matched on any dial peers.

The translation rule is applied as follows for the Calcutta site for the dialed number – 987 to the main site.

- ◆ In the called 3–digit number 987, the first digit 9 from the LHS (Left Hand side) is taken and compared in the translation rule for replacing the digits.
- ◆ The replacing number for 9 in the translation rule is – 103309, this number is replaced for the first digit 9.
- ◆ After replacing the first digit 9 with 103309, the remaining 2 digits – 87 are prepended to the replaced digit 103309 which becomes 10330987 and the call is routed to the voip dial peer as the destination pattern – 10330... matches.

Two more remote sites need to be integrated into the network. The New Delhi site uses a 2–digit indial range while the Chennai site uses a 4–digit indial range. By prepending a trunk access code and a site code to the called number, you now have a dial plan as follows:

This trunk access/site code...	...represents this dial plan
" 1011.. "	2–digit indial with an area code of 011
" 1033... "	3–digit indial with an area code of 033
" 1044.... "	4–digit indial with an area code of 044

This dial plan consists of 6–,7– and 8–digit numbers, so they can be standardized by padding the original called numbers with leading zeros to give a 1–digit access code, 3–digit site code and 4–digit extension:

This trunk access/ site code...	...is translated to
" 1011.. "	" 101100.. "
" 1033... "	" 10330... "
" 1044.... "	" 1044.... "

The number plan at the Main Site gateway/router is now a consistent 8 digits: (1–digit trunk access code, 3–digit site code, and 4–digit called number padded with leading zeros). Once the call reaches the Main Site gateway/router, the access code is stripped off and the subsequent seven digits are sent to the PBX.

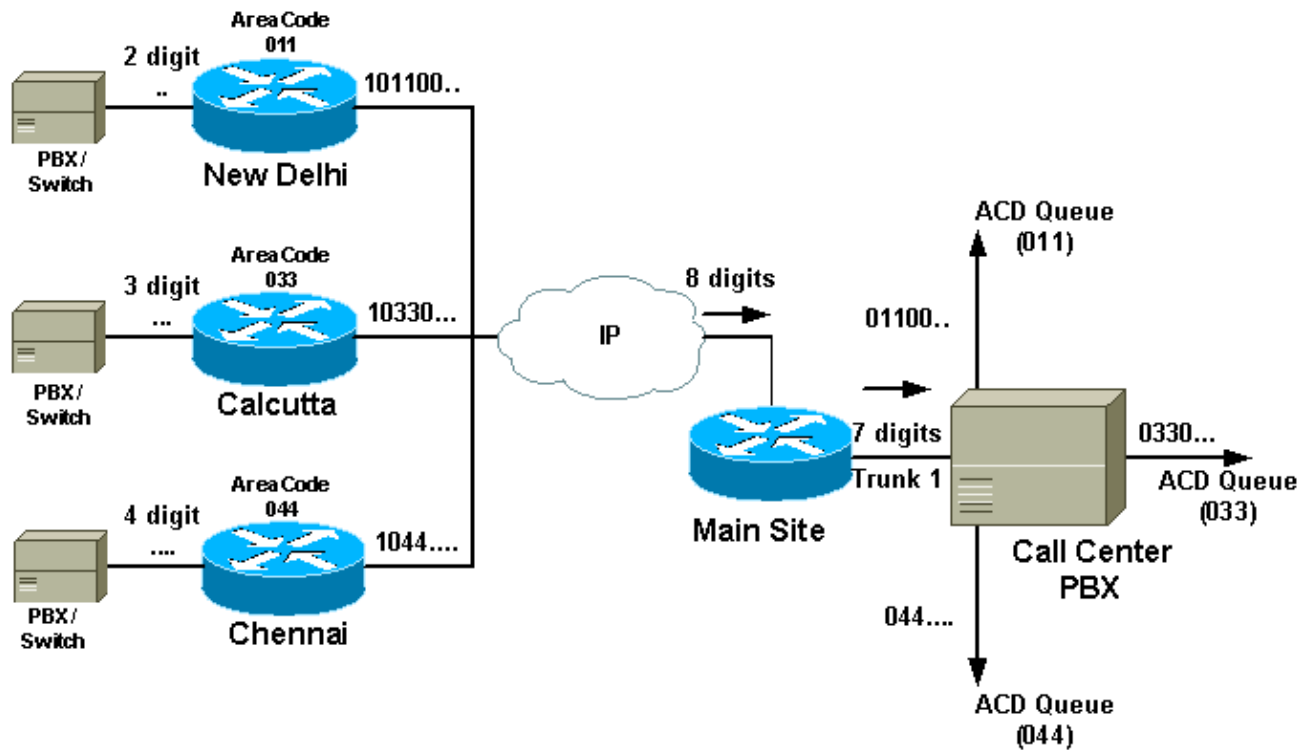
## Configure

In this case study there are three remote sites (New Delhi, Calcutta and Chennai) that are directing VoIP calls to the Main Site Call Center. The customer's requirement is for a system that only directs calls from remote sites to the main site. There is no provision for return calls from the main site back towards the remote sites. The Main Site gateway/router has 4 trunks that connect to the Call Center PBX.

**Note:** To find additional information on the commands used in this document, use the Command Lookup Tool (registered customers only) .

# Network Diagram

This document uses the network setup shown in the diagram here:



Remote Router	Trunk Code	Site (area) Code	Indial Range	Zero Padding
New Delhi	1	011	2 digit ..	00
Calcutta	1	033	3 digit ...	0
Chennai	1	044	4 digit ....	N/A

# Configurations

This document uses the configurations shown here:

```

New Delhi (2-digit indial range)

!---- Only relevant "IOS translation rule" output is presented

!
translation-rule 1

!-- The "1" above is the tag for the set.

rule 0 ^0. 1011000
rule 1 ^1. 1011001
rule 2 ^2. 1011002
rule 3 ^3. 1011003
rule 4 ^4. 1011004
rule 5 ^5. 1011005
    
```

```

rule 6 ^6. 1011006
rule 7 ^7. 1011007
rule 8 ^8. 1011008
rule 9 ^9. 1011009
!

!-- These rules replace the first digit of a 2-digit number with the corresponding
!-- translation. The router looks for a 2-digit number starting with a leading [0-9].
!-- The caret, "^" ensures the match only happens at the start of the digit string
!-- rather than any occurrence in a digit string. This ensures the router makes the
!-- translation only for the leading digits. By default, if an explicit match is made
!-- on a digit (in this case the first digit) the router replaces it with the new
!-- digits. Therefore, to keep the original numbering, the matched digit needs to be
!-- replaced with the same digit at the end of the modified string. Once the call
!-- comes in, the called number prepended with 101100 followed by the
!-- original 2 digits.
!
voice-port 1/0:1
translate called 1
cptone IN
compand-type a-law
!
!-- The translation rule is applied to the voice port where the
!-- call comes in to the router. When a call comes in from the
!-- telephone network towards the router, the called number
!-- is translated before it is matched on any dial peers.

!
dial-peer voice 100 voip
destination-pattern 101100..
session target ipv4:main site IP address
ip precedence 5
dtmf-relay h245-alphanumeric
!

!-- The VoIP dial peer needs to be configured to match on the new numbering plan

```

### Calcutta (3-digit indial range)

```

!--- Only relevant output is presented

!
translation-rule 1
rule 0 ^0.. 103300
rule 1 ^1.. 103301
rule 2 ^2.. 103302
rule 3 ^3.. 103303
rule 4 ^4.. 103304
rule 5 ^5.. 103305
rule 6 ^6.. 103306
rule 7 ^7.. 103307
rule 8 ^8.. 103308
rule 9 ^9.. 103309
!

!-- The router looks for a 3-digit called number in the range 000 - 999. It
!-- then prepends 10330 to this number before making a match on any other dial peers.
!-- Note: rules are executed before incoming POTS dial peer matching takes place.
!
voice-port 1/0:1
translate called 1

```

```
!-- The translation rule is applied to the voice port:
```

```
cptone IN
compand-type a-law
!
dial-peer voice 100 voip
destination-pattern 10330...
session target ipv4:main site IP address
ip precedence 5
dtmf-relay h245-alphanumeric
!
```

```
!-- The VoIP dial peer needs to be configured to match on the new numbering plan.
```

### Chennai (4-digit indial range)

```
!
translation-rule 11
rule 0 ^0... 10440
rule 1 ^1... 10441
rule 2 ^2... 10442
rule 3 ^3... 10443
rule 4 ^4... 10444
rule 5 ^5... 10445
rule 6 ^6... 10446
rule 7 ^7... 10447
rule 8 ^8... 10448
rule 9 ^9... 10449
!
```

```
!-- The router looks for a 4-digit called
!-- number in the range 0000 - 9999. !-- It then prepends 1044 to this number
!-- before making a match on any other dial peers.
```

```
!
voice-port 1/0:1
translate called 1
cptone IN
compand-type a-law
!
```

```
!-- The translation rule is applied to the voice port:
```

```
dial-peer voice 100 voip
destination-pattern 1044&.
session target ipv4:main site IP address
ip precedence 5
dtmf-relay h245-alphanumeric
!
```

```
!-- The VoIP dial peer needs to be configured to match on the new numbering plan
```

### Main Site

```
!-- By default, in POTS dial peers, matched digits get stripped off after a dial
!-- peer is matched. Then, the remaining digits are outputted to the PSTN/PBX.
```

```
dial-peer voice 20 pots
description - Call Center voice port 2/0
destination-pattern 1.....
```

```

port 2/0:1
!
dial-peer voice 21 pots
description - Call Center voice port 2/1
destination-pattern 1.....
port 2/1:1
!
dial-peer voice 30 pots
description - Call Center voice port 3/0
destination-pattern 1.....
port 3/0:1
!
dial-peer voice 31 pots
description - Call Center voice port 3/1
destination-pattern 1.....
port 3/1:1
!

!-- The trunk access code (leading 1) can be used to route the call out the Main Site
!-- router towards the Call-Center PBX. This digit is stripped off and the following
!-- 7 digits are sent to the external PBX. A single dial peer for each active voice
!-- port on the main site gateway router directs calls from the remote sites to
!-- the PBX.

```

**Note:** Since the Main Site PBX does not require the trunk access code, the Main Site router would strip it off. Therefore, the call is identified to the main site PBX by a 3–digit site code and the 4–digit extension. The PBX would need some configuration changes to match this numbering plan, but the advantage is that it can recognize the site code from the called number and route the call to particular ACD queue set up for each location. When a call comes in from one particular national region, it is directed to agents who speak the local language.

**Note:** For more information on commands, refer to Command Lookup Tool (registered customers only) .

## Verify

This section provides information you can use to confirm your configuration is working properly.

Certain **show** commands are supported by the Output Interpreter Tool (registered customers only) , which allows you to view an analysis of **show** command output.

- **test translation–rule** *name–tag input–number* Used to test the execution of the translation rules on a specific name tag. The syntax for this command is described in this table:

Syntax	Description
<i>name–tag</i>	The tag number by which the rule set is referenced.
<i>input–number</i>	

- **show translation–rule** *[name–tag]* Used to display the contents of the rules that have been configured for a specific translation name. The syntax for this command is described in this table:

Syntax	Description
--------	-------------

<i>name-tag</i>	(Optional) The tag number by which the rule set is referenced
-----------------	---

This output was captured from the NewDelhi router which shows the translation rules applied while dialing from the NewDelhi site.

```

NewDelhi - Output

!-- It is possible to confirm the translation rules are working:
!
!
NewDelhi#
test translation-rule 1 99

!-- Original called number is "99"

The replaced number: 10110099

!-- Translated to 8 digits

NewDelhi#
show translation-rule 1

Translation rule address: 0x64ADC2A8
Tag name: 1
Translation rule in_used 1
**** Xrule rule table ****
    Rule : 0
        in_used state: 1
        Match pattern: ^0.
        Sub pattern: 1011000
**** Xrule rule table ****
    Rule : 1
        in_used state: 1
        Match pattern: ^1.
        Sub pattern: 1011001
**** Xrule rule table ****
    Rule : 2
        in_used state: 1
        Match pattern: ^2.
        Sub pattern: 1011002
**** Xrule rule table ****
    Rule : 3
        in_used state: 1
        Match pattern: ^3.
        Sub pattern: 1011003
**** Xrule rule table ****
    Rule : 4
        in_used state: 1
        Match pattern: ^4.
        Sub pattern: 1011004
**** Xrule rule table ****
    Rule : 5
        in_used state: 1
        Match pattern: ^5.
        Sub pattern: 1011005
**** Xrule rule table ****
    Rule : 6
        in_used state: 1
        Match pattern: ^6.
        Sub pattern: 1011006

```



```

**** Xrule rule table *****
    Rule : 7
    in_used state: 1
    Match pattern: ^7.
    Sub pattern: 1011007
**** Xrule rule table *****
    Rule : 8
    in_used state: 1
    Match pattern: ^8.
    Sub pattern: 1011008
**** Xrule rule table *****
    Rule : 9
    in_used state: 1
    Match pattern: ^9.
    Sub pattern: 1011009
NewDelhi#

```

## Troubleshoot

This section provides information you can use to troubleshoot your configuration.

**Note:** Before issuing debug commands, refer to Important Information on Debug Commands.

- **debug translation {detail | min}** Used to debug number translation traces with the options described in this table:

Option	Description
detail	Enables a detailed debugging trace.
min	Enables a minimum debugging trace.

### NewDelhi – Output

```

NewDelhi#debug translation detail
xrule detail tracing is enabled
*Apr  4 04:52:57.124: xrule_checking
*Apr  4 04:52:57.124: xrule_checking calling , called
*Apr  4 04:52:57.124: xrule_checking peer_tag 0, direction 1, protocol 6
*Apr  4 04:52:57.124: xrule_translation
*Apr  4 04:52:57.124: xrule_translation callednumber , strlen 0
*Apr  4 04:52:57.124: xrule_translation callednumber null xruleCalledTag=1
*Apr  4 04:52:57.124: xrule_translation called Callparms Numpertype 0x81,
    match_type 0x0
*Apr  4 04:52:57.124: xrule_translation Xrule index 0, Numpertype 0x9
*Apr  4 04:52:57.124: dpMatchString, target_number null, match_number ^0.
*Apr  4 04:52:57.124: dpMatchString match_tmp 0, match_len 1
*Apr  4 04:52:57.124: dpMatchString beginning_replace 1, match_tmp 0,target null
*Apr  4 04:52:57.124: dpMatchString: len 4
*Apr  4 04:52:57.124: xrule_translation there was no match index 0
*Apr  4 04:52:57.124: xrule_translation called Callparms Numpertype 0x81,
    match_type 0x0
*Apr  4 04:52:57.124: xrule_translation Xrule index 1, Numpertype 0x9
*Apr  4 04:52:57.124: dpMatchString, target_number null, match_number ^1.
*Apr  4 04:52:57.124: dpMatchString match_tmp 1, match_len 1
*Apr  4 04:52:57.124: dpMatchString beginning_replace 1, match_tmp 1,target null
*Apr  4 04:52:57.124: dpMatchString: len 4
*Apr  4 04:52:57.124: xrule_translation there was no match index 1
*Apr  4 04:52:57.124: xrule_translation called Callparms Numpertype 0x81,

```



```
*Apr 4 04:52:59.396: xrule_checking calling , called 9
*Apr 4 04:52:59.396: xrule_checking peer_tag 0, direction 1, protocol 6
*Apr 4 04:52:59.396: xrule_translation
*Apr 4 04:52:59.396: xrule_translation callednumber 9, strlen 1
*Apr 4 04:52:59.396: xrule_translation callednumber 9 xruleCalledTag=1
*Apr 4 04:52:59.396: xrule_translation called Callparms Numpertype 0x0,
    match_type 0x0
*Apr 4 04:52:59.396: xrule_translation Xrule index 0, Numpertype 0x9
*Apr 4 04:52:59.396: dpMatchString, target_number 9, match_number ^0.
*Apr 4 04:52:59.396: dpMatchString match_tmp 0, match_len 1
*Apr 4 04:52:59.396: dpMatchString beginning_replace 1, match_tmp 0,target 9
*Apr 4 04:52:59.396: dpMatchString: len 1
*Apr 4 04:52:59.396: xrule_translation there was no match index 0
*Apr 4 04:52:59.396: xrule_translation called Callparms Numpertype 0x0,
    match_type 0x0
*Apr 4 04:52:59.396: xrule_translation Xrule index 1, Numpertype 0x9
*Apr 4 04:52:59.396: dpMatchString, target_number 9, match_number ^1.
*Apr 4 04:52:59.396: dpMatchString match_tmp 1, match_len 1
*Apr 4 04:52:59.396: dpMatchString beginning_replace 1, match_tmp 1,target 9
*Apr 4 04:52:59.396: dpMatchString: len 1
*Apr 4 04:52:59.396: xrule_translation there was no match index 1
*Apr 4 04:52:59.396: xrule_translation called Callparms Numpertype 0x0,
    match_type 0x0
*Apr 4 04:52:59.396: xrule_translation Xrule index 2, Numpertype 0x9
*Apr 4 04:52:59.396: dpMatchString, target_number 9, match_number ^2.
*Apr 4 04:52:59.396: dpMatchString match_tmp 2, match_len 1
*Apr 4 04:52:59.396: dpMatchString beginning_replace 1, match_tmp 2,target 9
*Apr 4 04:52:59.396: dpMatchString: len 1
*Apr 4 04:52:59.396: xrule_translation there was no match index 2
*Apr 4 04:52:59.396: xrule_translation called Callparms Numpertype 0x0,
    match_type 0x0
*Apr 4 04:52:59.396: xrule_translation Xrule index 3, Numpertype 0x9
*Apr 4 04:52:59.396: dpMatchString, target_number 9, match_number ^3.
*Apr 4 04:52:59.396: dpMatchString match_tmp 3, match_len 1
*Apr 4 04:52:59.396: dpMatchString beginning_replace 1, match_tmp 3,target 9
*Apr 4 04:52:59.396: dpMatchString: len 1
*Apr 4 04:52:59.396: xrule_translation there was no match index 3
*Apr 4 04:52:59.396: xrule_translation called Callparms Numpertype 0x0,
    match_type 0x0
*Apr 4 04:52:59.396: xrule_translation Xrule index 4, Numpertype 0x9
*Apr 4 04:52:59.396: dpMatchString, target_number 9, match_number ^4.
*Apr 4 04:52:59.396: dpMatchString match_tmp 4, match_len 1
*Apr 4 04:52:59.396: dpMatchString beginning_replace 1, match_tmp 4,target 9
*Apr 4 04:52:59.396: dpMatchString: len 1
*Apr 4 04:52:59.396: xrule_translation there was no match index 4
*Apr 4 04:52:59.396: xrule_translation called Callparms Numpertype 0x0,
    match_type 0x0
*Apr 4 04:52:59.396: xrule_translation Xrule index 5, Numpertype 0x9
*Apr 4 04:52:59.396: dpMatchString, target_number 9, match_number ^5.
*Apr 4 04:52:59.396: dpMatchString match_tmp 5, match_len 1
*Apr 4 04:52:59.396: dpMatchString beginning_replace 1, match_tmp 5,target 9
*Apr 4 04:52:59.396: dpMatchString: len 1
*Apr 4 04:52:59.396: xrule_translation there was no match index 5
*Apr 4 04:52:59.400: xrule_translation called Callparms Numpertype 0x0,
    match_type 0x0
*Apr 4 04:52:59.400: xrule_translation Xrule index 6, Numpertype 0x9
*Apr 4 04:52:59.400: dpMatchString, target_number 9, match_number ^6.
*Apr 4 04:52:59.400: dpMatchString match_tmp 6, match_len 1
*Apr 4 04:52:59.400: dpMatchString beginning_replace 1, match_tmp 6,target 9
*Apr 4 04:52:59.400: dpMatchString: len 1
*Apr 4 04:52:59.400: xrule_translation there was no match index 6
*Apr 4 04:52:59.400: xrule_translation called Callparms Numpertype 0x0,
    match_type 0x0
*Apr 4 04:52:59.400: xrule_translation Xrule index 7, Numpertype 0x9
*Apr 4 04:52:59.400: dpMatchString, target_number 9, match_number ^7.
*Apr 4 04:52:59.400: dpMatchString match_tmp 7, match_len 1
```

```
*Apr 4 04:52:59.400: dpMatchString beginning_replace 1, match_tmp 7,target 9
*Apr 4 04:52:59.400: dpMatchString: len 1
*Apr 4 04:52:59.400: xrule_translation there was no match index 7
*Apr 4 04:52:59.400: xrule_translation called Callparms Numpertype 0x0,
      match_type 0x0
*Apr 4 04:52:59.400: xrule_translation Xrule index 8, Numpertype 0x9
*Apr 4 04:52:59.400: dpMatchString, target_number 9, match_number ^8.
*Apr 4 04:52:59.400: dpMatchString match_tmp 8, match_len 1
*Apr 4 04:52:59.400: dpMatchString beginning_replace 1, match_tmp 8,target 9
*Apr 4 04:52:59.400: dpMatchString: len 1
*Apr 4 04:52:59.400: xrule_translation there was no match index 8
*Apr 4 04:52:59.400: xrule_translation called Callparms Numpertype 0x0,
      match_type 0x0
*Apr 4 04:52:59.400: xrule_translation Xrule index 9, Numpertype 0x9
*Apr 4 04:52:59.400: dpMatchString, target_number 9, match_number ^9.
*Apr 4 04:52:59.400: dpMatchString match_tmp 9, match_len 1
*Apr 4 04:52:59.400: dpMatchString beginning_replace 1, match_tmp 9,target 9
*Apr 4 04:52:59.400: dpMatchString 0. target 9,match_tmp 9,match_len 1
*Apr 4 04:52:59.400: replace_string
*Apr 4 04:52:59.400: replace_string match ^9., replace 1011009
*Apr 4 04:52:59.400: translation_format replace_rule ^9., strip_proceeding 0
*Apr 4 04:52:59.400: replace_string match_tmp ^9., strip_proceeding 0
*Apr 4 04:52:59.400: replace_string match_tmp 9
*Apr 4 04:52:59.400: replace_string direction 1, callparty 2
*Apr 4 04:52:59.400: replace_string direction 1, callparty 2, target 9
*Apr 4 04:52:59.400: replace_string match_tmp 9,replace 1011009
*Apr 4 04:52:59.400: replace_string0.replace19,target,current,match_tmp
*Apr 4 04:52:59.400: replace_string0.1 compare_len 1,match_len 1
*Apr 4 04:52:59.400: replace_string 3. replacel 9, compare_len 1
*Apr 4 04:52:59.400: replace_string 4. replacel 1,compare_len 0,replace 011009
*Apr 4 04:52:59.400: replace_string 4. replacel 10,compare_len -1,replace 11009
*Apr 4 04:52:59.400: replace_string 4. replacel 101,compare_len -2,replace 1009
*Apr 4 04:52:59.400: replace_string 4. replacel 1011,compare_len -3,replace 009
*Apr 4 04:52:59.400: replace_string 4. replacel 10110,compare_len -4,replace 09
*Apr 4 04:52:59.400: replace_string 4. replacel 101100,compare_len -5,replace 9
*Apr 4 04:52:59.400: replace_string 4. replacel 1011009,compare_len -6,replace
*Apr 4 04:52:59.400: replace_string 5. replacel 1011009, compare_len -6,match_l
en 1
*Apr 4 04:52:59.400: replace_string 6. replacel 1011009,compare_len -6,current
*Apr 4 04:52:59.400: replace_string buffer 1011009
*Apr 4 04:52:59.400: xrule_translation index 9,xrule_number 1011009, callparty
2
*Apr 4 04:52:59.400: xrule_translation Return rc = 0 any_match 0
*Apr 4 04:52:59.400: xrule_checking Return rc = 0
*Apr 4 04:53:00.156: xrule_checking
*Apr 4 04:53:00.156: xrule_checking calling 4444, called 10110099
*Apr 4 04:53:00.156: xrule_checking peer_tag 100, direction 2, protocol 0
*Apr 4 04:53:00.156: xrule_checking Return rc = -5
```

NewDelhi#debug translation min

```
*Apr 4 02:37:17.045: xrule_checking
*Apr 4 02:37:17.045: xrule_translation
*Apr 4 02:37:17.045: xrule_translation callednumber , strlen 0
*Apr 4 02:37:17.045: xrule_translation callednumber null xruleCalledTag=1
*Apr 4 02:37:19.457: xrule_checking
*Apr 4 02:37:19.457: xrule_translation
*Apr 4 02:37:19.457: xrule_translation callednumber 9, strlen 1
*Apr 4 02:37:19.457: xrule_translation callednumber 9 xruleCalledTag=1
*Apr 4 02:37:19.457: dpMatchString 0. target 9,match_tmp 9,match_len 1
*Apr 4 02:37:19.457: replace_string
*Apr 4 02:37:19.461: replace_string0.replace19,target,current,match_tmp
*Apr 4 02:37:19.461: replace_string0.1 compare_len 1,match_len 1
*Apr 4 02:37:19.461: replace_string 3. replacel 9, compare_len 1
*Apr 4 02:37:19.461: replace_string 4. replacel 1,compare_len 0,replace 011009
*Apr 4 02:37:19.461: replace_string 4. replacel 10,compare_len -1,replace 11009
```

```
*Apr 4 02:37:19.461: replace_string 4. replacel 101,compare_len -2,replace 1009
*Apr 4 02:37:19.461: replace_string 4. replacel 1011,compare_len -3,replace 009
*Apr 4 02:37:19.461: replace_string 4. replacel 10110,compare_len -4,replace 09
*Apr 4 02:37:19.461: replace_string 4. replacel 101100,compare_len -5,replace 9
*Apr 4 02:37:19.461: replace_string 4. replacel 1011009,compare_len -6,replace
*Apr 4 02:37:19.461: replace_string 5. replacel 1011009, compare_len -6,match_l
en 1
*Apr 4 02:37:19.461: replace_string 6. replacel 1011009,compare_len -6,current
*Apr 4 02:37:19.461: replace_string buffer 1011009
*Apr 4 02:37:19.461: xrule_translation index 9,xrule_number 1011009, callparty
2
*Apr 4 02:37:19.841: xrule_checking
```

## Related Information

- **Dial Peer Enhancements**
- **Understanding Dial Peers and Call Legs on Cisco IOS Platforms**
- **Understanding Inbound and Outbound Dial Peers on Cisco IOS Platforms**
- **Understanding How Inbound and Outbound Dial Peers are Matched on Cisco IOS Platforms**
- **Understanding the Operational Status of Dial Peers on Cisco IOS Platforms**
- **Understanding Direct-Inward-Dial (DID) on Cisco IOS Digital (T1/E1) Interfaces**
- **Voice Technology Support**
- **Voice and Unified Communications Product Support**
- **Troubleshooting Cisco IP Telephony** [🔗](#)
- **Technical Support – Cisco Systems**

---

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2014 – 2015 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

---

Updated: Feb 02, 2006

Document ID: 20595

---