

# Using TCL Scripts for CNR in a Telco-Return Environment

---

## Contents

[Introduction](#)[Before You Begin](#)[Conventions](#)[Prerequisites](#)[Components Used](#)[Background Theory - Understanding Telco-Return](#)[The Need for Using TCL Scripts](#)[Configurations](#)[Verify](#)[Troubleshoot](#)[Related Information](#)

---

## Introduction

This document describes how to use Toolkit Command Language (TCL) scripts in a Telco-Return environment using Cisco Network Registrar (CNR) as your Dynamic Host Configuration Protocol (DHCP) server. This document solves Telco-Return issues where cable modems and hosts get IP addresses from the same subnet. The intent is to have PCs behind cable modems receive IP addresses from a different subnet.

## Before You Begin

### Conventions

For more information on document conventions, see the [Cisco Technical Tips Conventions](#).

### Prerequisites

The reader should have a basic understanding of Data-over-Cable Service Interface Specifications (DOCSIS), Point-to-Point Protocol (PPP), and Unix or WinNT server.

### Components Used

The information in this document is based on the software and hardware versions below.

- Unix workstation
- Cisco uBR7246 running Cisco IOS® version ubr7200-ist-mz.121-1a.bin
- Cisco Network Registrar 3.5.3

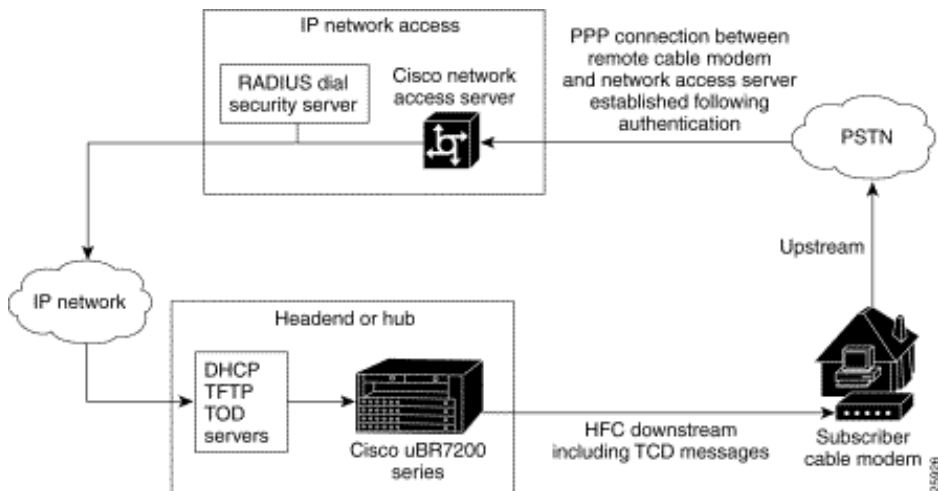
The information presented in this document was created from devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If you are working in a live network, ensure that you understand the potential impact of any command before using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

## Background Theory - Understanding Telco-Return

In Telco-Return configurations, the Cisco uBR7200 series universal broadband router provides downstream dataflow from cable modem cards connected to the cable system, and accepts upstream traffic via a combination of the local Public Switched Telephone Network (PSTN) and IP network path that terminates at the Cisco uBR7200 series I/O controller or applicable port adapter. Upstream data is through a telephone modem (external or internal to a cable modem, as well as a cable modem card in a PC, based on the third-party cable modem vendor) connected to an analog telephone line typically.

Telco-Return gives cable companies that have not upgraded their cable plants or specific service areas to support two-way RF transmission the ability to offer fast downstream data services via the cable plant and upstream transmission via the PSTN.

Refer to the diagram below.



## The Need for Using TCL Scripts

In a Telco-Return environment, the DHCP server does not have a way to distinguish between DHCP DISCOVER packets coming from Cable modems or hosts. Therefore, both devices end up getting assigned IP addresses from the same scope, which is not scalable and wastes valuable IP address since only hosts require a routable subnet. When trying to attempt to solve this problem, you can use two TCL scripts: `postclientlookup.tcl` and `prepacketencode.tcl` in Cisco's DHCP server (CNR).

The key point in understanding what pool you belong to is based on the source of the request. When the DISCOVER is from a cable modem, it is sourced from the AS5300 network, usually the 192.168 network or the PPP network address. If that is the case, the CNR server learns that the device is a Telco-Return cable modem, and points them to the appropriate 10-net pool. With DISCOVER packets from hosts behind cable modems, they are sourced by the cable modem's Ethernet address. These DHCP DISCOVER packets are forwarded by the cable modems within a UDP packet to the CMTS. The CMTS then forwards these DHCP requests to the DHCP server as specified in cable helper-address. The GIADDR of those relayed requests, however, are set to the primary network because the GIADDR is already populated by the cable modem. DHCP requests from hosts are sourced from modems and therefore will have the GIADDR set to the primary address on the cable interface. This is the downstream channel IP address in the TSI message specified in section 3.4.4.2 of the TRI spec, which states that the cable modem must populate the GIADDR of DHCP Discovers from hosts behind them. When the DHCP server sees that the DISCOVER packets are sourced from anything else other than the 192.168 network, it thinks it must be a host and **not** a cable modem.

## Configurations

You need to use two scripts, the `postclientlookup.tcl` and `prepacketencode.tcl`. You only need to edit the `postclientlookup.tcl` to have the 192.168 address (or network address your using for PPP) instead of all zeros however.

The main objective is to set `PPPnetaddress 192.168`. Initially, the address is all zeros.

The main objective is to set `PPPnetaddress 192.168`. Initially, the address is all zeros.

The following actions are required when setting up CNR using these scripts.

1. In CNR, use `tagTelcoModem` for the Telco-Return modem.

2. In CNR, use tagComputer for the CPE device behind the modem.
3. Edit the postclientlookup script, enable checking for Telco-Return modems.
4. Also, in the script, change the PPPNetAddress to the network address of the PPP dial pool.
5. Extensions must be created for the scripts.
6. These extensions must be attached to the DHCP server.
7. Enable Client Class Processing on CNR using the proper tags.
8. Place the .tcl files in the following directory when running NT:

C:\Program Files\Network Registrar\EXTENSIONS\DHCP\TCL

If running CNR on Solaris, place the appropriate scripts in the following directory:

/opt/nwreg2/extensions/dhcp/scripts/tcl

9. Run the following commands from the CNR command line interface:

```
nrcmd>extension PostClientLookup create tcl postclientlookup.tcl PostClientLookup
nrcmd>dhcp attachExtension post-client-lookup PostClientLookup
nrcmd>extension PrePacketEncode create tcl prepaketencode.tcl PrePacketEncode
nrcmd>dhcp attachExtension pre-packet-encode PrePacketEncode
```

10. In CNR create scopes, client classes, and policies using the tags that are given in the scripts. Enable Client Class Processing.

Below is the script **postclientlookup.tcl**. All you need to do is edit the PPPnetaddress. This section is in **bold**.

#### Postclientlookup.tcl

```
# Copyright (C) 1998 by Cisco Systems.
# This program contains proprietary and confidential information.
# All rights reserved. This software shall not be used by any party
# except by prior written consent of Cisco Systems.
#
# These scripts are an experimental version and in current flux, to obtain a
# stable version for production environments please contact Cisco/AIC/engineering
support.
#
# 12/14/98 jtb Created by jtb
# 1/25/99 jtb added support for BOOTP routers
# 2/11/99 jtb Added support for selection-criteria merging..
# 3/29/99 jtb Cleanup of SetRouter and reorganized and
# added Router ARP "Switched Network Flag" and
# renamed to postclientlookup.tcl
# 4/5/99 jtb fixed remaining problems with exp version 7
# 4/10/99 jtb added PostClientLastDynamicNameinWins
# 4/11/99 jtb Removed PostClientLastDynamicNameinWins and Modified
# DelayALLScopes
#
# VERSION 12
#
#
# NEEDS:
# Add doc on policies for VOIP, Telco, Cable, system_default_policy
#
# To configure this extension, perform the following steps:
#
# *. Move this file (postclientlookup.tcl) to the directory
# <your-install-path>/extensions/dhcp/tcl
#
# *. Configure the extension:
```

```

#          nrcmd -N <user> -P <Password> extension PostClientLookup create tcl
postclientlookup.tcl
          PostClientLookup
#          nrcmd -N <user> -P <Password> dhcp attachExtension post-client-lookup
PostClientLookup
#
# This proc is attached to the post-client-lookup extension point and can
# be used to:
# *. If you wish to use SelectClientType then;
# set SelectClientType "1" ELSE set SelectClientType "0"
set SelectClientType "1" # ALSO;
# ** Ensure that Client-Class is enabled and these specific selection tags
# (no spaces or quotes and case sensitive) exist.
# # nrcmd -N <user> -P <Password> dhcp enable client-class
# # nrcmd -N <user> -P <Password> scope-selection-tag tagCablemodem create
# # nrcmd -N <user> -P <Password> scope-selection-tag tagTelcoModem create
# # nrcmd -N <user> -P <Password> scope-selection-tag tagComputer create
# # nrcmd -N <user> -P <Password> scope-selection-tag tagSelsiusIPPhone create
#
# *. The following instructions assume that the primary interface on each of the
# line cards on the uBR have been assigned a subnet for cable modems and or
# VOIP phones eg. 10.X.X.X
# and that the secondary interfaces correspond to subnets for computers.
# e.g. 24.X.X.X
# The IOS command for creating a secondary IP address on an interface is:
# ip address <ip-address> <mask> secondary
#
# *. They also can use the relay-agent-option if it has been enabled on the uBR (IOS
11.x).
# The [no] cable relay-agent-option command is enabled on a
per-interface
# basis.
#
# *. OR (IOS 12.x).
# The ip dhcp relay info option command is enabled.
#
# ***. AND To work around bug CSCdm19529 in IOS 12.0 > 12.5
# The no ip dhcp relay info option check command is disabled.
#
# *. They also assume that the uBR has been setup correctly as a BOOTP forwarder
device so
# that DHCP broadcast messages are being forwarded on to Network Registrar
# The IOS command for this is: ip helper-address <address of CNR>
#
# *. Ensure that the scopes created in CNR that correspond to the *Primary*
interfaces
# assigned on the uBR 72xx line cards:
# - Are not marked as secondary to any other scope
# - Have at least one scope-selection-tag assigned to them.
# - They should have the tags: tagCablemodem
# - They *MUST NOT* have the tags: tagComputer,tagTelcoModem,tagSelsiusIPPhone
# (no spaces or quotes and case sensitive)
#
#          nrcmd -N <user> -P <Password> scope <CM-scope-name> create <addr> <mask>
#          nrcmd -N <user> -P <Password> scope <CM-scope-name> set
selection-tags=tagCablemodem
#
# *. If you are using Telco Return Modems, then modify PPPnetaddress below to match

```

```

the
    masked network.
# OR set PPPnetaddress "0.0.0.0" to Disable checking for Telco Return.
set PPPnetaddress "192.168."

# *. Ensure that the scopes created in CNR that correspond to an interfaces on the
NAS (Cisco 5xxx):
# - *Are* each marked as secondary to the correct primary scope.
# - Have at least one scope-selection-tag assigned to them.
# - They should have at least the tag: tagTelcoModem
# - They *MUST NOT* have the tag: tagCablemodem, tagComputer
# (no spaces or quotes and case sensitive)
#
# nrcmd -N <user> -P <Password> scope <TelcoCM-name> create <addr> <mask>
# nrcmd -N <user> -P <Password> scope <TelcoCM-name> set
selection-tags=tagTelcoModem
# *. If you are using Cisco/Selsius IP Phones, then set SelsiusIPPhone "1"
# else set SelsiusIPPhone "0"
set SelsiusIPPhone "0"

# *. Ensure that the scopes created in CNR that correspond to the *Secondary*
interfaces
# assigned on the uBR 72xx line cards:
# - *Are* each marked as secondary to the correct primary scope.
# - Have at least one scope-selection-tag assigned to them.
# - They should have at least the tag: tagComputer, tagSelsiusIPPhone
# - They *MUST NOT* have the tag: tagCablemodem,tagTelcoModem
# (no spaces or quotes and case sensitive)
#
# nrcmd -N <user> -P <Password> scope <name> create <addr> <mask>
# nrcmd -N <user> -P <Password> scope <name> set selection-tags=tagComputer,
tagSelsiusIPPhone
#
# *. Be sure to reload the server when the configuration is completed
# # nrcmd -N <user> -P <Password> server dhcp reload
#
# -If PPPnetaddress != "0.0.0.0" , and Source Addr. matches PPPnetaddress (i.e.
192.168.)
# then will be treated as a Telco Return cablemodem w/ selection-criteria set to
tagTelcoModem
# - Else If SelsiusIPPhone == "1", and Vendor MAC address = "^00:10:EB" for VOIP,
then will be treated as
# Cisco/Selsius IP Phones w/ selection-criteria set to tagSelsiusIPPhone
# -Else If the option82 RemoteID = chaddr, the device will be treated as a 2 way
Cablemodem
# w/ selection-criteria set to tagCablemodem
# -Else the device will be treated as a PC w/ selection-criteria set to tagComputer

# *. If you want to use this as your backup server delay script then;
# set PostClientDelayScope "1" ELSE set PostClientDelayScopes "0"
set PostClientDelayScopes "0"

# ALSO
# *. If you only want to ONLY delay SOME scopes then
# *. Ensure that Client-Class is enabled and these specific selection tags
# (no spaces or quotes and case sensitive) exist.
# # nrcmd -N <user> -P <Password> dhcp enable client-class

```

```

# # nrcmd -N <user> -P <Password> scope-selection-tag tagBackupScope create
#
# *. Ensure that the scopes created in CNR that are Backup Scopes
# - Have at least one scope-selection-tag assigned to them.
# - They should have at least the tag: tagBackupScope
# (no spaces or quotes and case sensitive)
# # nrcmd -N <user> -P <Password> scope <name> set selection-tags=tagBackupScope
# ELSE
# *. If you want to delay ALL scopes then
# set PostClientDelayALLScopes "1" ELSE set PostClientDelayALLScopes "0"
set PostClientPostClientDelayALLScopes "0"

# This proc is attached to the post-client-lookup extension point
#
proc ModSelectCriteria { request environ scriteria } {
global SelsiusIPPhone
global PPPnetaddress
    set RelayAgentOption [ $request get relay-agent-info ]
set chaddr [ $request get chaddr ]
    if { [ string length $scriteria ] > 0 } {
        # $environ log LOG_INFO "<$chaddr> selection-criteria was <$scriteria>"
        set scriteria "$scriteria,"
    }

if { [ regexp "^$PPPnetaddress" "0.0.0.0" ] == 0 &&
    [ regexp "^$PPPnetaddress" [ $request get client-ipaddress ] ] } { # is
TelcoModem
    # $environ log LOG_INFO "<$chaddr> is a TelcoModem"
    append scriteria tagTelcoModem

} elseif { $SelsiusIPPhone == "1" &&
    [ regexp -nocase "^00:10:EB" $chaddr ] } { # is an SelsiusIPPhone
    # $environ log LOG_INFO "<$chaddr> is a SelsiusIPPhone"
    append scriteria tagSelsiusIPPhone

} elseif { [ string length $RelayAgentOption ] > 0 } {
    # $environ log LOG_INFO "Relay agent option from <$chaddr> is
<$RelayAgentOption>"
    set temp [ split $RelayAgentOption : ]

    set opt1len [ lindex $temp 1 ]
    set opt2len [ lindex $temp [ expr $opt1len + 3 ] ]
    set opt2off [ expr $opt1len + 4 ]
    set remoteid [ join [ lrange $temp $opt2off [ expr $opt2off + $opt2len ] ] : ]
    # $environ log LOG_INFO "Relay agent RemoteID option from <$chaddr> is
<$remoteid>"
    if { [ regexp "^$chaddr" $remoteid ] } { # is a 2 way Cablemodem
        # $environ log LOG_INFO "<$chaddr> is a 2 way cablemodem"
        append scriteria tagCablemodem

    } else { # is a Computer
        # $environ log LOG_INFO "<$chaddr> is a computer"
        append scriteria tagComputer

    }

} else { # is a Computer
    # $environ log LOG_INFO "<$chaddr> is a computer"
    append scriteria tagComputer
}
}

```

```

}
    # $enviro n log LOG_INFO "<$chaddr> selection-criteria is now <$scriteria>"
    $request put selection-criteria $scriteria
}

proc PostClientLookup { request response environ } {
global SelectClientType
global PostClientDelayScopes
global PostClientDelayALLScopes
set msgtype [ $request get dhcp-message-type ]

    if { $msgtype != 4 && $msgtype != 7 } {
        # Reply && !DHCPDECLINE && !DHCPRELEASE

        set scriteria ""
        set scriteria [ $request get selection-criteria ]
if { $SelectClientType == "1" } {
    ModSelectCriteria $request $enviro n $scriteria
}

        if { $PostClientDelayScopes == "1" } {
            if { $PostClientDelayALLScopes == 1 || [ regexp tagBackupScope
$scriteria ] } {
                # $enviro n log LOG_INFO "Proc PostClientdelayhalfsec: delaying
500ms."
                after 500
            }
        }

        # $enviro n log LOG_INFO "Proc PreEncodeLastDynamicNameinWins:
        Environ host-name is '[$enviro n get host-name]'"
        # $enviro n log LOG_INFO " Response; Client-requested-host '
        [$request get client-requested-host-name]'
        host-name-changed =
        '[$request get host-name-changed]' to '[$request get client-host-name]'.
    }
}

```

Below is the script **prepacketencode1.tcl**, which does **not** need to be edited.

### Prepacketencode1.tcl

```

# Copyright (C) 1998 by Cisco Systems.
# This program contains proprietary and confidential information.
# All rights reserved. This software shall not be used by any party
# except by prior written consent of Cisco Systems.
#
# 12/14/98 jtb Created by jtb
# 1/7/99 jtb added Scope Netmask Override
# 1/25/99 jtb added support for BOOTP routers
# 2/11/99 jtb Added support for policy overrides script
# 2/11/99 jtb Added support for tagClientNetmask
# 3/29/99 jtb Cleanup of SetRouter and reorganized and
# added Router ARP "Switched Network Flag" and
# renamed to prepacketencode.tcl
# 4/5/99 jtb fixed remaining problems with exp version 7
# 4/10/99 jtb added PreEncodeLastDynamicNameinWins
# 4/12/99 jtb Modified DelayALLScopes and Removed PreEncodeLastDynamicNameinWins
#
# VERSION 12

```

```
#
#
# NEEDS:
# Add doc on system_default_policy
#
# To configure this extension, perform the following steps:
#
# *. Move this file (prepacketencode.tcl) to the directory
# <your-install-path>/extensions/dhcp/tcl
#
# *. Configure the extension:
# nrcmd -N <User> -P <Password> extension PrePacketEncode create tcl
prepacketencode.tcl PrePacketEncode
# nrcmd -N <User> -P <Password> dhcp attachExtension pre-packet-encode
PrePacketEncode

# This proc is attached to the pre-packet-encode extension point and can
# be used to:
# *. If you wish to set the router via this script then;
# set UseSetRouter "1" ELSE set UseSetRouter "0"
set UseSetRouter "1"

# ALSO;
# ** Configure System_Default_Policy or scope associated policy
routers=255.255.255.255 for Router
  ARP "Switched Network Flag"
# nrcmd -N <User> -P <Password> policy system_default_policy set
routers=255.255.255.255
# OR;

# ** Configure system_default_policy routers=0.0.0.0 for Router Rule AND
# MODIFY routeroffset accordingly Zero == none
# nrcmd -N <User> -P <Password> policy system_default_policy set routers=0.0.0.0
set routeroffset1 "1"

# *. If you wish to use "Scope Netmask Override" then;
# edit OverrideScopeNetmask appropriately, else set OverrideScopeNetmask "0.0.0.0"
;
set OverrideScopeNetmask "0.0.0.0"

# ALSO;
# *. If want to use scope netmask on some scopes:
# nrcmd -N <user> -P <Password> dhcp enable client-class
# nrcmd -N <User> -P <Password> scope-selection-tag tagUseScopeNetmask create
#
# *. AND Ensure that the scopes created in CNR that MUST use the Scope Netmask (No
Override)
# have at least the tag: tagUseScopeNetmask
# nrcmd -N <User> -P <Password> scope <name> set
selection-tags=tagUseScopeNetmask

# *. If you want to use this as your backup server delay script then;
# set PreEncodeDelayScope "1" ELSE set PreEncodeDelayScopes "0"
set PreEncodeDelayScopes "0"

# ALSO
# *. If you only want to ONLY delay SOME scopes then
# *. Ensure that Client-Class is enabled and these specific selection tags
```

```

# (no spaces or quotes and case sensitive) exist.
# # nrcmd -N <user> -P <Password> dhcp enable client-class
# # nrcmd -N <user> -P <Password> scope-selection-tag tagBackupScope create
#
# *. Ensure that the scopes created in CNR that are Backup Scopes
# - Have at least one scope-selection-tag assigned to them.
# - They should have at least the tag: tagBackupScope
# (no spaces or quotes and case sensitive)
# # nrcmd -N <user> -P <Password> scope <name> set selection-tags=tagBackupScope
# ELSE
# *. If you want to delay ALL scopes then
# set PreEncodeDelayALLScopes "1" ELSE set PreEncodeDelayALLScopes "0"
set PreEncodeDelayALLScopes "0"

# This proc is attached to the pre-packet-encode extension point
#
proc UseRouterRule { response environ scopeNetwkNum scopeSubnetMask Offset } {

    # extract the scope network number and mask
    regexp {([0-9]*)\.([0-9]*)\.([0-9]*)\.([0-9]*)} $scopeNetwkNum dummy byte1
byte2 byte3 byte4
    regexp {([0-9]*)\.([0-9]*)\.([0-9]*)\.([0-9]*)} $scopeSubnetMask dummy
byte11 byte22 byte33 byte44

    # correct for incorrect network numbers e.g. 192.168.10.129 255.255.255.192
    set byte1 [ expr $byte1 & $byte11 ]
    set byte2 [ expr $byte2 & $byte22 ]
    set byte3 [ expr $byte3 & $byte33 ]
    set byte4 [ expr $byte4 & $byte44 ]

    # do a quick and dirty sanity check.
    set netbrdcast [ expr 255 - $byte44 + $byte4 ]
    set router [ expr $byte4 + $Offset ]
    # $environ log LOG_INFO "proc PrePacketEncode from Subnet =
$byte1.$byte2.$byte3.$byte4, Router = .$router"
    if { $router > [ expr $netbrdcast - 1 ] } {
        $environ log LOG_INFO "proc PrePacketEncode from Subnet =
$byte1.$byte2.$byte3.$byte4,
Broadcast = .$netbrdcast, Router = .$router"
        $environ log LOG_INFO "proc PrePacketEncode: Router Greater Than Broadcast !!! :
Setting to Broadcast - 1."
        set router [ expr $netbrdcast - 1 ]
    }

    set routers [ append tmp $byte1 . $byte2 . $byte3 . $router ]
    # load the result into the routers field.
    $response put routers $routers
    # $environ log LOG_INFO "proc PrePacketEncode from [ $response get chaddr ],
Routers = $routers"
}

proc SetRouter { response environ offset1 } {

    # system_default_policy Router Rule
    set scopeNetwkNum [ $response get scope-network-number ]
    set scopeSubnetMask [ $response get scope-subnet-mask ]

    UseRouterRule $response $environ $scopeNetwkNum $scopeSubnetMask $offset1
}

```

```

# DEBUG
# $enviro log LOG_INFO "proc PrePacketEncode from [ $response get chaddr ], Routers
= [ $response get routers ]"
}

proc PrePacketEncode { request response environ } {
global UseSetRouter
global routeroffset1
global PreEncodeDelayScopes
global PreEncodeDelayALLScopes
global OverrideScopeNetmask
set msgtype [ $response get dhcp-message-type ]
set routers ""
    if { $msgtype != 4 && $msgtype != 6 } {
        # Reply && !DHCPDECLINE && !DHCPNACK

if { $UseSetRouter != "0"} {
    set routers [ $response get routers ]

        if [ regexp "^255.255.255.255" $routers ] {
            # system_default_policy ARP Router "Switched Network Flag"

            $response put routers [ $response get lease-ipaddress ]

        } elseif { [ regexp "^0.0.0.0" $routers ] && $routeroffset1 > "0" } {
            SetRouter $response $enviro $routeroffset1

        }
    }

if { [ regexp "^0.0.0.0" $OverrideScopeNetmask ] == 0 &&
    [ regexp tagUseScopeNetmask [ $response get scope-selection-tags ] ] == 0 }
{
    # $enviro log LOG_INFO "Proc OverrideMask"
    $response put subnet-mask $OverrideScopeNetmask
}

    if { $PreEncodeDelayScopes == "1" } {
        if { $PreEncodeDelayALLScopes == "1" ||
            [ regexp tagBackupScope [ $response get scope-selection-tags ] ] }
        {
            $enviro log LOG_INFO "Proc PreEncodedelayhalfsec: delaying 500ms."
            after 500
        }
    }

    # $enviro log LOG_INFO "Proc PreEncodeLastDynamicNameinWins:
    Environ host-name is '[$enviro get host-name]'"
    # $enviro log LOG_INFO " Txt-string is '[$enviro get txt-string]',
    ignore is '[$enviro get ignore-prerequisites]'"
    # $enviro log LOG_INFO " Response; Client-requested-host '
    [$response get client-requested-host-name]' host-name-changed =
    '[$response get host-name-changed]' to '[$response get client-host-name]'"
}

```

## Verify

There is currently no verification procedure available for this configuration.

## Troubleshoot

There is currently no specific troubleshooting information available for this configuration.

---

## Related Information

- [How to Switch Over Cable Telco-Return to Two-Way RF Return Path](#)
  - [Technical Support - Cisco Systems](#)
- 

<a href="#">Home</a>	<a href="#">What's New</a>	<a href="#">How to Buy</a>	<a href="#">Login</a>	<a href="#">Profile</a>	<a href="#">Feedback</a>	<a href="#">Search</a>	<a href="#">Map/Help</a>
----------------------	----------------------------	----------------------------	-----------------------	-------------------------	--------------------------	------------------------	--------------------------

All contents are Copyright © 1992-2003 Cisco Systems, Inc. All rights reserved. [Important Notices](#) and [Privacy Statement](#).