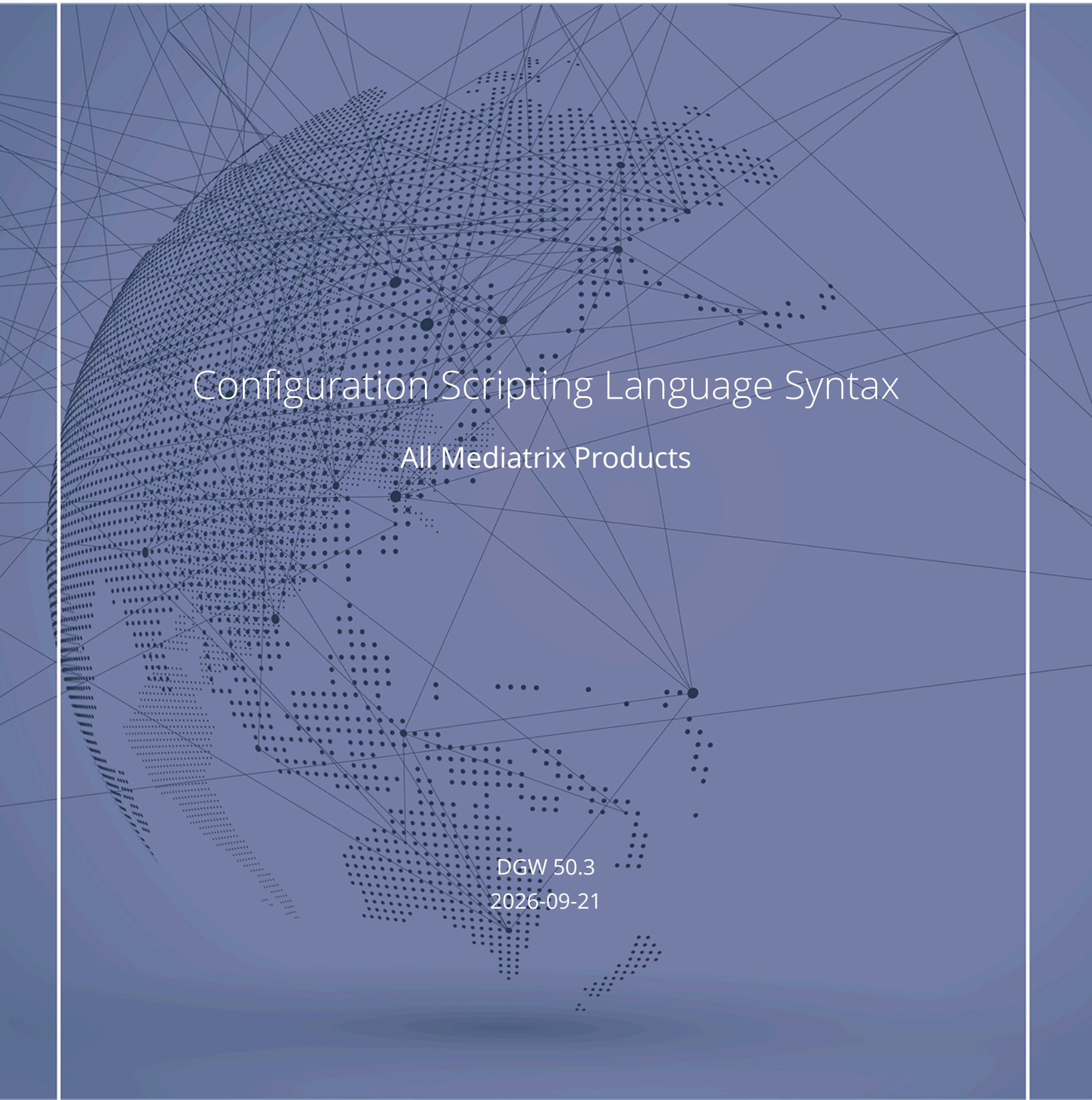


# Configuration Scripting Language Syntax

## All Mediatrix Products

DGW 50.3

2026-09-21

# Table of Contents


## Media5 Proprietary Scripting Language 4

Language Characteristics 4

## Line Syntax 5

Token Types 5
Scripting with Special Characters 6
Call Router Specific Naming Rules 7
Text vs Number Resolution 7

## Expression Syntax 9

Table Syntax 9
Row Syntax 9
Column Syntax 9
Cell Syntax 10
Scalar Syntax 10
Enum Parameter and Command Arguments 11

## Command Syntax 12

Row Command Syntax 12
Table Command Syntax 13

## Command Lines to Display CLI Information 14

## Special Scripting Commands 20

## Basic Call Scripting Examples 21

Generic Configuration Template 21
Script to Configure the ISDN Interface 22
Script to Configure an ISDN/SIP Call Route 23
Script to Configure the SIP Endpoint 24
Script to Configure the Call Property Transformations 26

## Management Scripting Examples 27

Script to Configure a Backup/Restore 27
Script to Configure a User Password 27
Script to Perform a Firmware Upgrade 27

## Debugging 28

Script to Enable the Syslog 28
Script to Configure PCM Traces 28

## Advanced Script Examples 29

Script to Configure SIP Servers and Registrations (C7/S7/G7) 29
Script to Configure an ISDN PRI Interface 29
Script to Configure an ISDN BRI Interface 30
Script with Paused Execution 31

## Copyright Notice 32

# Media5 Proprietary Scripting Language

The Media5 proprietary scripting language can be used to assign values to configuration parameters and to execute configuration commands. The scripting language may be used when creating a configuration script or when working with the Command Line Interface (CLI).

The scripting language uses the following general syntax:

```
(keyword) (Context_Name (separator expression (operator constant)))
 (#comment)
```

All specific syntaxes are derived from this general syntax, except for the parenthesis symbols, "(" and ")", which are used to mark optional arguments; they are not part of the syntax.

## Language Characteristics

- Scripts are line-based, i.e. the end of the line marks the end of the sentence.
- The "enter" command indicates that the next line is a new command.
- Scripts are currently stateless (no text can change how the script parser will interpret the next line).
- Indentation is ignored.
- With the exception of the values of some parameters, scripts are NOT case-sensitive.
- Only constant expressions (numeric or textual) can be used as right-hand values of assignment operations.
- All configuration objects referred to in a script line MUST exist, except when using the `insertrow` keyword, in which case the script refers to a configuration object (row) that is about to be created.

# Line Syntax

A line includes a set of tokens written in a defined syntax.

The syntax of a line is defined by tokens with the following general syntax:

```
[keyword] [context [separator expression [operator constant]]] [#comment]
```

**Note:** Brackets ([ and ]) are used to mark optional arguments. They are not part of the syntax.

- The standard delimiter in a line syntax is a space, but all blanks (any combination of spaces and tabs) are allowed.
- Separators and operators are delimiters by themselves but can be surrounded on one or two sides by blanks.
- Unquoted blanks are prohibited in all tokens but constants.
- If a token is to include blanks, the token MUST be quoted.
- Except when enclosed in double quotes, no token is allowed to contain a sequence matching an operator or a separator.
- The total length of a line, including white space and comments, must be lower than 65535.

## Token Types

Refer to Text vs Number Resolution (p.7).

| Token type | Description |
|---|---|
| Comment | Anything following the comment marker (#) up to the end of the line is ignored. |
| Constant | A textual string or a number to assign to the expression. The constant value MUST be quoted. For more details on text and number resolution in the scripting language, refer to Text vs Number Resolution (p.7). |
| Context_Name | Defines the service the following expression belongs to. |
| Expression | Defines the task to be performed, which can either be a command to execute or a parameter to read or write. The expression can resolve to either:<br>• a scalar parameter<br>• a cell<br>• a column<br>• a row<br>• a table |
| Keyword | Defines the type of operation to execute on the expression or the type of data to retrieve from the expression. For a list of allowed keywords refer to Command Lines to Display CLI Information (p.14). |
| Operator | Defines the relation between an expression and a constant. Only the = (equal) assignment operator is defined. It is mandatory when the keyword is `set`. |
| Separator | Delimiter defined as `.` (coma). |

## Scripting with Special Characters

- Pound (#) sign is the comment marker. It can be placed in-line, not only at the beginning of a line. Anything following the comment marker (#) up to the end of the line is ignored.
- Double quotes (") are recommended to protect content that would otherwise be interpreted by the parser. Content enclosed in a double quoted block is always literally interpreted as text. Double quotes are always removed from the token.
- Double quotes (") can be escaped using the backslash character (\). They are then considered as usual characters.

- Backslash (\) can be escaped using the backslash character. It is then considered as a usual character.
- * (asterix) can be used to abstract the index list in a `set` command. It can only be used for the `set` and `get` commands.

## Call Router Specific Naming Rules

When working with call router parameters, you must prefix the name of:

- a route with "route-", for instance: route-isdn_sip.
- a SIP interface with "sip-", for instance: sip-default.
- an ISDN interface with "isdn-", for instance: isdn-default.
- a hunt with "hunt-", for instance: hunt-trunkLines.

## Text vs Number Resolution

- A token containing only numbers and the + or - sign is interpreted as a number.
- A token containing at least one non-numeric character (except + and -) is interpreted as a string.
- Numbers and text MUST be quoted.
- Double quotes (") and backslash (\) MUST be escaped using the backslash character (\).
- A text containing a special character MUST be quoted.
- A text containing the separator '.' or the operator '=' MUST be quoted.

| Script | Resolves as | |
|---|---|---|
| | Number | Text |
| 432 | 432 | |
| "432" | 432 | |
| Allo | | Allo |
| 43p2 | | 43p2 |
| "42 big | | Syntax error: double quotes (") are a special character. |
| 43"2 | | Syntax error: double quote (") is a special character. |
| "twenty" | | twenty |
| \"twenty\" | | Syntax error: A text with special characters must be quoted. |
| "\"twenty\"" | | "twenty" |
| Twen\"ty | | Syntax error: A text with special characters must be quoted. |
| Twen\ty | | Syntax error: backslash is a special escape character and the escapable characters the double quote (") and the backslash (\). |
| Twen\\ty | | Syntax error: A text with special characters must be quoted. |
| "Twen\\ty" | | Twen\ty |
| "Twen=ty" | | Twen=ty |
| "10.5.56.6" | | 10.5.56.6 |

# Expression Syntax

The purpose of an expression is to describe a configuration object. However an expression by itself is not enough, it must be prefixed by a context and a separator to specify the service the configuration object belongs to.

**Note:** When a service, a scalar, a command, or a table parameter has the same name as a keyword, use the `get` or `set` command to access the parameter.

## Table Syntax

The syntax for a table is the name of the service and the name of the table separated by a separator. Table commands cannot be provided with arguments, therefore, the `set` command cannot be used.

```
get Service_Name.Table_Name
```

## Row Syntax

The syntax of a row is the name of the service and the name of the table followed by the index identifying the row. The index is in the `name=value` form, enclosed within brackets.

```
set Service_Name.Table_Name.Row_Name[index1=value1]
get Service_Name.Table_Name.Row_Name
```

**Note:** [ and ] (brackets) are part of the syntax.

## Column Syntax

The syntax for a column is the service name, the table name, and the column name all separated by a separator.

```
get Service_Name.Table_Name.Column_Name
set Service_Name.Table_Name.Column_Name="Value"
```

## Cell Syntax

The syntax for a cell is the service name, the table name (followed by the index), and the column name. They are all separated by a separator.

```
get Service_Name.Table_Name[Index=key].Column_Name
set Service_Name.Table_Name[Index=key].Column_Name="value"
get Service_Name.Table_Name[*].Column_Name
get Service_Name.Table_Name.Column_Name
```

- [ ] (brackets) are part of the syntax.
- The index is in the `name=value` form, enclosed within brackets.
- The syntax is column-based.
- * (asterix) can be used to replace the index list and apply the command on each row in the table.
- * can only be used for the `set` and `get` commands.

| Command | Description |
|---|---|
| Column_Name | Name of the column that contains the cell. |
| index=key | Index identifying the row on which the cell is located. The syntax of the index is `name=value`, enclosed within brackets.<br>The index is always the first column of a table. |
| Service_Name | Defines which service should process the expression. |
| Table_Name | Name of the table that contains the cell. |
| Value | A textual string or a number to assign to the argument. The value SHOULD be quoted. |

## Scalar Syntax

The syntax for a scalar is the name of the service with the name of the scalar, separated by a separator.

```
set Service_Name.Scalar_Name="value"
get Service_Name.Scalar_Name
```

| Command | Description |
|---|---|
| Service_Name | Defines which service should process the expression. |
| Scalar_Name | Name of the specific parameter to which to assign a value. |
| Value | A textual string or a number to assign to the argument. The value MUST be quoted. |

## Enum Parameter and Command Arguments

Enum type scalars, columnars, table keys, and command arguments can be set with their textual or numerical values. Example (Uppercase values are enum values in textual format):

```
Service_Name.Scalar_Name = ENABLE
Service_Name.Scalar_Name = 100
Service_Name.Table_Name[Index=CONF].Column_Name=FTP
Service_Name.Table_Name[Index=CONF].Column_Name=400
Command ParameterName = HIGH
Command ParameterName = 200
```

# Command Syntax

The syntax of the normal command is the name of the service, the name of the command, separated by a separator, and the arguments with their values.

```
Service_Name.Command_Name arg1="value1"
```

| Command | Description |
|---|---|
| Service_Name | Defines which service should process the expression. |
| Command_Name | The command to execute. |
| arg(n) | scalar arguments (with mandatory '=' and following value). |
| value(n) | A textual string or a number to assign to the argument. The value MUST be quoted. |

## Row Command Syntax

The syntax for a row command (a "command in a cell") is the table name and the name of the command separated by a separator, followed by index identifying the row the command is part of. Row commands appear as table cells and allow you to perform an action on a specific row of the relevant table.

```
Context_Name.Table_Name[index=value].Row_Command=execute_value
```

**Note:** [ and ] (brackets) are part of the syntax.

The specified syntax is column-based and the index is in the `name=value` form, enclosed within brackets.

| Command | Description |
|---|---|
| Service_Name | Defines which service should process the expression. |
| Table_Name | Table where the row command is located. |
| index=name | Index identifying the row on which to execute the command. The syntax of the index is `name=value` form, enclosed within brackets.<br>The index is always the first column of a table. Refer to Cell Syntax (p.10). |
| Row_Command | The row command to execute. |
| execute_value | Numerical value of the enum. |

### Row Commands

| Command | Description |
|---|---|
| DeleteAllRows | A table command used to delete all rows of a specific table to start anew. It can be used as follows: `Service_Name.Table_Name.DeleteAllRows` |

## Table Command Syntax

The syntax for a table command is the table name followed by a separator and the name of the row command.

```
Service_Name.Table_Name.Table_Row_Command_Name
```

**Note:** Table commands cannot use arguments.

# Command Lines to Display CLI Information

The following command lines can be used in the CLI to display available keywords, commands, arguments, etc.

| Cli Keywords | Command Line | Descriptions |
|---|---|---|
| [tab key] | List all available contexts . (Services, tables, parameters, indexes, commands, columns, and key) | |
| | Global>tab | List all service names. |
| | Global>XX+tab | List all service names starting with XX. |
| | Global>ServiceName.+tab | List all available commands for the specified service. |
| | Global>ServiceName.CommandName+tab | List available parameters for the specified command of the service. |
| access | Global>access ServiceName.Parameter | Retrieve the access type of the parameter (read only, read/write). |
| alias | Create another global name for accessing a context (Scalars, cells, columns, rows,tables, and services). | |
| | Global>alias ServiceName=ChosenAlias | Create another global name for accessing a service. |
| | Global>alias ServiceName.Parameter=ChosenAlias | Create another global name for accessing a parameter. |
| | Global>alias ServiceName.TableName=ChosenAlias | Create another global name for accessing a **TableName**. |
| cd | Global>cd ServiceName | Change to the specified service. |
| columnars | Global>columnars ServiceName.TableName | List all columns associated with the table. |
| commands | Global>commands ServiceName | List all commands associated with the service. |
| defval | Global>defval ServiceName.Scalar | Retrieve the default value of the scalar indicated in the current profile. |
| | Global>defval ServiceName.Cell | Retrieve the default value of the cell indicated in the current profile. |
| | Global>defval ServiceName.Column | Retrieve the default value of the column indicated in the current profile. |
| help | Global>help | List all available keywords. |

| Cli Keywords | Command Line | Descriptions |
|---|---|---|
| | Global>help ServiceName | Display the description of the service. |
| | Global>help ServiceName.ObjectName | Display the description of the object (scalars, tables, columns, and commands). |
| indexes | Global>indexes ServiceName.TableName | List the indexes associated with the table. |
| logs on | Global>logs on | Display syslog messages in the console. |
| logs off | Global>logs off | Do not display syslog messages in the console. |
| ls | Global>ls ServiceName.TableName | List all objects associated with the specified table (scalar, column,cells, and indexes ). |
| objects | Global>objects ServiceName | List all objects associated with the specified service (scalars, tables, columns, cells, or commands). |
| pcapture | pcapture -i any | Start network capture. |
| | pcapture -D | Display the list of available network interfaces. |
| | pcapture -i eth5 | Start the network capture on the eth5 interface. |
| | pcapture -i eth5 -c 100 | Stop the capture automatically when 100 packets are captured. |
| | pcapture -i eth5 -e | Display the link layer header on each dump line. |
| | pcapture -i eth5 -p | Do not put the interface into promiscuous mode. (Only what the unit really sees not what is blocked by the network interface). |
| | pcapture -i eth5 -T rtp | Force the captured packet to be interpreted as 'rtp' packets. |
| | pcapture -i eth5 -T rtcp | Force the captured packet to be interpreted as 'rtcp' packets. |

| Cli Keywords | Command Line | Descriptions |
|---|---|---|
| | pcapture -i eth5 -T snmp | Force the captured packet to be interpreted as 'snmp' packets. |
| | pcapture -i eth5 -T tftp | Force the captured packet to be interpreted as 'tftp' packets. |
| | pcapture -i eth5 -raw | Use to transform the CLI session as a remote binary capture device to be put in a capture file or directly in a network capture software. Unreadable output. |
| ping | Global>ping 10.0.0.1 | Test whether or not a particular host is reachable across an IPv4 network, by sending 1 ping. |
| | Global>ping -c 4 10.0.0.2 | Test whether or not a particular host is reachable across an IPv4 network by sending only 4 pings (1 and more). |
| | Global>ping -s 1024 10.0.0.3 | Test whether or not a particular host is reachable across an IPv4 network by sending a ping of 1024 bytes (default is 56). |
| | Global>ping -q -c 10 10.0.0.5 | Test whether or not a particular host is reachable across an IPv4 network by sending 10 pings but only display start and stop messages. |
| ping6 | Global>ping6 2001:db8::1 | Test whether or not a particular host is reachable across an IPv6 network by sending only 1 ping. |
| | Global>ping6 -c 4 2001:db8::2 | Test whether or not a particular host is reachable across an IPv6 network by sending only 4 pings (1 and more). |

| Cli Keywords | Command Line | Descriptions |
|---|---|---|
| | Global>ping6 -s 1024 2001:db8::3 | Test whether or not a particular host is reachable across an IPv6 network by sending a ping of 1024 bytes (default is 56). |
| | Global>ping6 -q -c 10 2001:db8::4 | Test whether or not a particular host is reachable across an IPv6 network by sending 10 pings but only display start and stop messages. |
| scalars | Global>scalars ServiceName | List the scalars associated with the service. |
| services | Global>services | Retrieve all the services. |
| sysinfo | Global>sysinfo | Display current system information (Mac address, firmware versions, serial number, model, system uptime, current time, and information on installed cards) |
| tables | Global>tables ServiceName | List the tables associated with the service. |
| traceroute | Global>traceroute 8.8.8.8 | Trace the route to a particular host across an IPv4 network. |
| | Global>traceroute -n 8.8.8.8 | Trace the route with numeric addresses (don't resolve DNS). |
| | Global>traceroute -m 3 8.8.8.8 | Trace the route, stopping after 3 hops (default is 30). |
| | Global>traceroute -s 192.168.0.10 192.168.0.3 | Trace the route to the host across an IPv4 network using alternative source address. Note that you must select the address of one of the interfaces. By default, the address of the outgoing interface is used. |
| traceroute6 | Global>traceroute6 2620:119:35::35 | Trace the route to a particular host across an IPv6 network. |

| Cli Keywords | Command Line | Descriptions |
|---|---|---|
| | Global>traceroute6 -q 6 2620:119:35::35 | Trace the route to the host across an IPv6 network using 6 probes per TTL (default 3). |
| type | Global>type ServiceName.Expression | Retrieve the type of expression (int32, uint64, and string) (the enum values are numerical values). |
| unalias | Global>unalias S | Remove alias of expression. |

# Special Scripting Commands

| Command | Description |
|---|---|
| Pause ss | Pause the script for ss seconds. The value of ss must be between 1 and 300 seconds.<br>This command cannot be aborted. The timer of the Pause command must be elapsed before another action can be performed.<br>This command is only available in configuration scripts. It is not available with Cli or with Snmp.<br>If a longer pause is necessary, multiple consecutive Pause commands can be used.<br>Long pauses must be used with caution. Pauses extend the duration of a script execution, and there are several restrictions during a script execution. For instance, starting or stopping services is delayed until the end of the script. |

# Basic Call Scripting Examples

## Generic Configuration Template

```
##################################################################
# Description        : Generic configuration Template File  #
##################################################################
# DNS - Uncomment the following two lines if are using DNS server
 addresses specified by Broadsoft
#Hoc.StaticDnsServers[Priority=1].IpAddress = "IP address of Primary DNS
 server"
#Hoc.StaticDnsServers[Priority=2].IpAddress = "IP address of secondary DNS
 server"

# Time Server
Hoc.SntpConfigSource = "Static"
# Static, or Automatic (dhcp)
Hoc.StaticSntpServerHost = "NTP server IP address or FQDN"

# Config File Download
Conf.ScriptsLocation = "config file folder"
Conf.ScriptsTransferSrvHostname = provisioning server fqdn or ip address"
Conf.ScriptGenericFileName = "generic_config.txt"
Conf.ScriptSpecificFileName = "unit_specific-%mac%.txt"
Conf.ScriptsTransferOnRestartEnable = "Enable"
Conf.ScriptsTransferProtocol = "http"
Conf.ScriptsTransferUsername ="username"
Conf.ScriptsTransferPassword = "password"

# DTMF
# DTMF Transport Method: Inband, OutOfBandUsingRtp (RFC 2833),
#  OutOfBandUsingSignalingProtocol (SIP Info)
Mipt.DefaultDtmfTransportMethod = "OutOfBandUsingRtp"

# RFC 2833 Event payload number
Mipt.DefaultDtmfTransportPayloadType = "101"

# Country Tone set, e.g. NorthAmerica1, UK1, China1, UAE2, UAE2,
# Brazil1, Mexico1, Germany1, Spain1, France1 and more
TelIf.CountrySelection ="NorthAmerica1"

# Digit Map
EpServ.CallDtmfMapAllowed[Index=2].DtmfMap = "*xx"

# CRout Service
CRout.AutoRoutingEnable ="Enable"
# Attach additional Interop settings here
```

```
# SIP Servers
SipEp.DefaultStaticMessagingHost = "voicemail server IP or FQDN"

# Voicemail server - SIP Subscription host
SipEp.DefaultStaticRegistrarServerHost = "SIP Registration server IP
 address of FQDN"
SipEp.DefaultStaticProxyHomeDomainHost = "SIP Proxy server IP address or
 FQDN"
SipEp.DefaultStaticProxyOutboundHost = "IP address or FQDN of the SBC "
SipEp.SupportedDnsQueries = "Srv"

# DNS Query: Srv,Naptr, Address (A-record)
# SIP Transport -Only one (UDP, TCP, TLS) can be enabled at a time
# Default is SIP over UDP
SipEp.TransportConfig[GatewayName=default].UdpEnable ="Enable"
SipEp.TransportConfig[GatewayName=default].TcpEnable ="Disable"
SipEp.TransportConfig[GatewayName=default].TlsEnable ="Disable"

# SIP Registration Time
# RefreshTime = Time to register before registration expires
# By default, a SIP Register is sent 60sec before it expires
SipEp.DefaultRegistrationRefreshTime = "60"
SipEp.DefaultRegistrationExpirationValue = "3600"

# SIP Penalty BoxSipEp.PenaltyBoxEnable = "Disable"
# Enable or Disable
SipEp.PenaltyBoxTime = "300"

# SIP Timer
SipEp.InteropTransmissionTimeout = "32"
#Enable SIP Remote Reboot and Check-Sync
SipEp.GwEventHandling[GatewayName=default].Reboot = "Restart"
SipEp.GwEventHandling[GatewayName=default].CheckSync = "TransferScript"

# Restart services
Scm.RestartRequiredServices
```

## Script to Configure the ISDN Interface

To send and receive calls, you must minimally configure the ISDN interface.

### Mediatrix C7 Series (BRI interface only)

```
Isdn.BasicRateInterface[Name=Bri1].EndpointType = "Nt"
Isdn.BasicRateInterface[Name=Bri1].ConnectionType = "PointToPoint"
```

### Mediatrix G7 and Sentinel 100 Series (BRI and PRI interfaces)

```
# For BRI interface
```

```
Isdn.BasicRateInterface[Name=BRI1].EndpointType = "Nt"
Isdn.BasicRateInterface[Name=BRI1].ConnectionType = "PointToPoint"
```

```
# For PRI interface
Isdn.PrimaryRateInterface[Name=PRI1].EndpointType = "Nt"
Isdn.PrimaryRateInterface[Name=PRI1].PortPinout = "Nt"
```

### Sentinel 400 Series (BRI and PRI interfaces)

```
# For BRI interface
Isdn.BasicRateInterface[Name=Slot1/BRI1].EndpointType = "Nt"
Isdn.BasicRateInterface[Name=Slot1/BRI1].ConnectionType = "PointToPoint"
```

```
# For PRI interface
Isdn.PrimaryRateInterface[Name=Slot1/E1T1].EndpointType = "Nt"
Isdn.PrimaryRateInterface[Name=Slot1/E1T1].PortPinout = "Nt"
```

## Script to Configure an ISDN/SIP Call Route

You must create routes that will route calls from ISDN to SIP and from SIP to ISDN.

### Mediatrix C7 Series (BRI interface only)

```
Crout.InsertRoute    # From ISDN to SIP
Crout.Route[Index=1].SourceCriteria="Bri1"
Crout.Route[Index=1].PropertiesCriteria=""
Crout.Route[Index=1].ExpressionCriteria=""
Crout.Route[Index=1].Destination="sip-default"
Crout.InsertRoute    # From SIP to ISDN
Crout.Route[Index=2].SourceCriteria="sip-default"
Crout.Route[Index=2].PropertiesCriteria=""
Crout.Route[Index=2].ExpressionCriteria=""
Crout.Route[Index=2].Destination="Bri1"
Crout.ApplyConfig
```

### Mediatrix G7 and Sentinel 100 Series (BRI and PRI interfaces)

```
# For BRI interface
Crout.InsertRoute    # From ISDN to SIP
Crout.Route[Index=1].SourceCriteria="BRI1"
Crout.Route[Index=1].PropertiesCriteria=""
Crout.Route[Index=1].ExpressionCriteria=""
Crout.Route[Index=1].Destination="sip-default"
Crout.InsertRoute    # From SIP to ISDN
Crout.Route[Index=2].SourceCriteria="sip-default"
Crout.Route[Index=2].PropertiesCriteria=""
Crout.Route[Index=2].ExpressionCriteria=""
Crout.Route[Index=2].Destination="BRI1"
```

```
Crout.ApplyConfig
```

```
# For PRI interface
Crout.InsertRoute    # From ISDN to SIP
Crout.Route[Index=1].SourceCriteria="PRI1"
Crout.Route[Index=1].PropertiesCriteria=""
Crout.Route[Index=1].ExpressionCriteria=""
Crout.Route[Index=1].Destination="sip-default"
Crout.InsertRoute    # From SIP to ISDN
Crout.Route[Index=2].SourceCriteria="sip-default"
Crout.Route[Index=2].PropertiesCriteria=""
Crout.Route[Index=2].ExpressionCriteria=""
Crout.Route[Index=2].Destination="PRI1"
Crout.ApplyConfig
```

Sentinel 400 Series (BRI and PRI interfaces)

```
# For BRI interface
Crout.InsertRoute    # From ISDN to SIP
Crout.Route[Index=1].SourceCriteria="Slot1/BRI1"
Crout.Route[Index=1].PropertiesCriteria=""
Crout.Route[Index=1].ExpressionCriteria=""
Crout.Route[Index=1].Destination="sip-default"
Crout.InsertRoute    # From SIP to ISDN
Crout.Route[Index=2].SourceCriteria="sip-default"
Crout.Route[Index=2].PropertiesCriteria=""
Crout.Route[Index=2].ExpressionCriteria=""
Crout.Route[Index=2].Destination="Slot1/BRI1"
Crout.ApplyConfig
```

```
# For PRI interface
Crout.InsertRoute    # From ISDN to SIP
Crout.Route[Index=1].SourceCriteria="Slot1/E1T1"
Crout.Route[Index=1].PropertiesCriteria=""
Crout.Route[Index=1].ExpressionCriteria=""
Crout.Route[Index=1].Destination="sip-default"
Crout.InsertRoute    # From SIP to ISDN
Crout.Route[Index=2].SourceCriteria="sip-default"
Crout.Route[Index=2].PropertiesCriteria=""
Crout.Route[Index=2].ExpressionCriteria=""
Crout.Route[Index=2].Destination="Slot1/E1T1"
Crout.ApplyConfig
```

## Script to Configure the SIP Endpoint

Configuring the SIP endpoint allows you to register your ISDN telephone to a SIP server.

### Mediatrix C7 Series (BRI interface only)

```
SipEp.DefaultStaticRegistrarServerHost="Ip Address of the Default Static
 Registrar Server Host"
SipEp.DefaultStaticProxyHomeDomainHost="Ip address of the Default Static
 Proxy Home Domain Host"
sipEp.userAgent[EpId=Bri1].SipUsername="Unit model number"
sipEp.userAgent[EpId=Bri1].FriendlyName="Unit model number"
sipEp.userAgent[EpId=Bri1].GatewayName="default"
sipEp.userAgent[EpId=Bri1].Register="Enable"
SipEp.registrationrefresh
```

### Mediatrix G7 and Sentinel 100 Series (BRI and PRI interfaces)

```
# For BRI interface
SipEp.DefaultStaticRegistrarServerHost="Ip Address of the Default Static
 Registrar Server Host"
SipEp.DefaultStaticProxyHomeDomainHost="Ip address of the Default Static
 Proxy Home Domain Host"
sipEp.userAgent[EpId=BRI1].SipUsername="Unit model number"
sipEp.userAgent[EpId=BRI1].FriendlyName="Unit model number"
sipEp.userAgent[EpId=BRI1].GatewayName="default"
sipEp.userAgent[EpId=BRI1].Register="Enable"
SipEp.registrationrefresh
```

```
# For PRI interface
SipEp.DefaultStaticRegistrarServerHost="Ip Address of the Default Static
 Registrar Server Host"
SipEp.DefaultStaticProxyHomeDomainHost="Ip address of the Default Static
 Proxy Home Domain Host"
sipEp.userAgent[EpId=PRI1].SipUsername="Unit model number"
sipEp.userAgent[EpId=PRI1].FriendlyName="Unit model number"
sipEp.userAgent[EpId=PRI1].GatewayName="default"
sipEp.userAgent[EpId=PRI1].Register="Enable"
SipEp.registrationrefresh
```

### Sentinel 400 Series (BRI and PRI interfaces)

```
# For BRI interface
SipEp.DefaultStaticRegistrarServerHost="Ip Address of the Default Static
 Registrar Server Host"
SipEp.DefaultStaticProxyHomeDomainHost="Ip address of the Default Static
 Proxy Home Domain Host"
sipEp.userAgent[EpId=Slot1/BRI1].SipUsername="Unit model number"
sipEp.userAgent[EpId=Slot1/BRI1].FriendlyName="Unit model number"
sipEp.userAgent[EpId=Slot1/BRI1].GatewayName="default"
sipEp.userAgent[EpId=Slot1/BRI1].Register="Enable"
SipEp.registrationrefresh
```

```
# For PRI interface
SipEp.DefaultStaticRegistrarServerHost="Ip Address of the Default Static
 Registrar Server Host"
```

```
SipEp.DefaultStaticProxyHomeDomainHost="Ip address of the Default Static
 Proxy Home Domain Host"
sipEp.userAgent[EpId=Slot1/E1T1].SipUsername="Unit model number"
sipEp.userAgent[EpId=Slot1/E1T1].FriendlyName="Unit model number"
sipEp.userAgent[EpId=Slot1/E1T1].GatewayName="default"
sipEp.userAgent[EpId=Slot1/E1T1].Register="Enable"
SipEp.registrationrefresh
```

## Script to Configure the Call Property Transformations

You must create Call Property Transformations that will allow you to properly communicate between an ISDN telephone and an analog telephone behind a SIP gateway.

In this example, it is assumed that the ISDN telephone number is 800 and the analog telephone number is 101.

```
Crout.InsertMappingType
Crout.mappingtype[Index=1].Name="ISDN_To_SIP"
Crout.mappingtype[Index=1].Criteria="CalledE164"
Crout.mappingtype[Index=1].Transformation="CalledE164"
Crout.InsertMappingExpression
Crout.mappingexpression[Index=1].Name="ISDN_To_SIP"
Crout.mappingexpression[Index=1].Criteria="Unit model number"
Crout.mappingexpression[Index=1].Transformation="101"
Crout.mappingexpression[Index=1].SubMappings=""
```

```
Crout.InsertMappingType
Crout.mappingtype[Index=2].Name="SIP_to_ISDN"
Crout.mappingtype[Index=2].Criteria="CalledE164"
Crout.mappingtype[Index=2].Transformation="CalledE164"
Crout.InsertMappingExpression
Crout.mappingexpression[Index=2].Name="SIP_to_ISDN"
Crout.mappingexpression[Index=2].Criteria="Unit model number"
Crout.mappingexpression[Index=2].Transformation="800"
Crout.mappingexpression[Index=2].SubMappings=""
```

```
Crout.Route[Index=1].Mappings="ISDN_To_SIP"
Crout.Route[Index=2].Mappings="SIP_to_ISDN"
Crout.ApplyConfig
```

# Management Scripting Examples

## Script to Configure a Backup/Restore

Each of the two following commands must be created on one line.

### Creating a backup of the unit configuration

```
# This command must be written on one continous line
Conf.BackupImage FileName="image.text" Location="resultfolder"
  TransferProtocol="Tftp" TransferUsername="" TransferPassword=""
  TransferSrvHostname="Ip address of Host server"
```

### Restoring a backup of the unit configuration

```
# This command must be written on one continous line
Conf.RestoreImage FileName="image.text" Location="resultfolder"
  TransferProtocol="Tftp" TransferUsername="" TransferPassword=""
  TransferSrvHostname="Ip address of Host server"
```

## Script to Configure a User Password

If you are using the CLI, the new password will be used the next time you connect to the Mediatrix.

```
Aaa.Users[UserName=admin].Password=TestPwd
```

## Script to Perform a Firmware Upgrade

```
Fpu.MfpTransferUsername=""
Fpu.MfpTransferPassword=""
Fpu.AutomaticRestartEnable="1"
Fpu.MfpUrl="url of firmware (http(s)://, ftp://, tftp://, etc)"
Fpu.Install
```

# Debugging

## Script to Enable the Syslog

```
Nlm.SyslogRemoteHost="Ip address of Remote host"
Cli.MinSeverity="Warning"
Bni.MinSeverity="Error"
Hoc.MinSeverity="Critical"
```

## Script to Configure PCM Traces

```
Mipt.PcmCaptureEnable=1
Mipt.PcmCaptureEndpoint="EndpointNameToCapture"
Mipt.PcmCaptureIpAddr="IP address to send the traces to"
Mipt.restart
```

# Advanced Script Examples

## Script to Configure SIP Servers and Registrations (C7/S7/G7)

```
SipEp.UserAgent[EpId=FXS1].Username = "18885551211"
SipEp.UserAgent[EpId=FXS2].Username = "18885551212"
SipEp.UserAgent[EpId=FXS1].Register = "Enable"
SipEp.UserAgent[EpId=FXS2].Register = "Enable"
SipEp.DefaultStaticProxyHomeDomainHost = "ims.example.com"
SipEp.DefaultStaticProxyOutboundHost = "proxy.example.net"
SipEp.Authentication.DeleteAllRows
SipEp.Authentication[Index=1].CriteriaSelection = "Username"
SipEp.Authentication[Index=2].CriteriaSelection = "Username"
SipEp.Authentication[Index=1].UsernameCriteria = "18885551211"
SipEp.Authentication[Index=2].UsernameCriteria = "18885551212"
SipEp.Authentication[Index=1].ValidateRealm = "Disable"
SipEp.Authentication[Index=2].ValidateRealm = "Disable"
SipEp.Authentication[Index=1].Username = "18885551211@ims.example.com"
SipEp.Authentication[Index=2].Username = "18885551212@ims.example.com"
SipEp.Authentication[Index=1].Password = "password_1"
SipEp.Authentication[Index=2].Password = "password_2"
SipEp.DefaultStaticRegistrarServerHost = "ims.example.com"
SipEp.DefaultStaticMessagingHost = "ims.example.com"
# Restart all the services that require it.
Scm.RestartRequiredServices
```

## Script to Configure an ISDN PRI Interface

```
Hardware.Cards[*].LineType = "T1"
Hardware.Cards[*].Signaling = "Isdn"
Isdn.PrimaryRateInterface[*].EndpointType = "Te"
Isdn.PrimaryRateInterface[*].PortPinout = "Auto"
Isdn.PrimaryRateInterface[*].LineCoding = "B8zs"
Isdn.PrimaryRateInterface[*].LineFraming = "Esf"
Isdn.PrimaryRateInterface[*].NetworkLocation = "User"
Isdn.PrimaryRateInterface[*].PreferredEncodingScheme = "G711ulaw"
Isdn.PrimaryRateInterface[*].FallbackEncodingScheme = "G711alaw"
Isdn.PrimaryRateInterface[*].ChannelRange = "1-23"
Isdn.PrimaryRateInterface[*].ChannelAllocationStrategy = "Ascending"
Isdn.PrimaryRateInterface[*].MaxActiveCalls = "0"
Isdn.PrimaryRateInterface[*].SignalInformationElementEnable = "Disable"
Isdn.PrimaryRateInterface[*].InbandToneGenerationEnable = "Enable"
Isdn.PrimaryRateInterface[*].InbandDtmfDialingEnable = "Enable"
```

```
Isdn.PrimaryRateInterface[*].OverlapDialingEnable = "Disable"
Isdn.PrimaryRateInterface[*].CallingNameMaxLength = "34"
Isdn.PrimaryRateInterface[*].ExclusiveBChannelSelectionEnable = "Disable"
Isdn.PrimaryRateInterface[*].SendingCompleteEnable = "Enable"
Isdn.PrimaryRateInterface[*].ClipEnable = "Enable"
Isdn.PrimaryRateInterface[*].ClirEnable = "Disable"
Isdn.PrimaryRateInterface[*].ClirOverrideEnable = "Disable"
Isdn.PrimaryRateInterface[*].SendRestartOnStartupEnable = "Enable"
Isdn.SignalingChannel[*].Protocol = "Ni2"
Isdn.SignalingChannel[*].FacilityServicesEnable = "Enable"
Isdn.SignalingChannel[*].ColpEnable = "Disable"
Isdn.SignalingChannel[*].ColrEnable = "Disable"
Isdn.SignalingChannel[*].ColrOverrideEnable = "Disable"
Isdn.SignalingChannel[*].OutgoingNotifyEnable = "Disable"
Isdn.SignalingChannel[*].AcceptedProgressCauses = "1-127"
Isdn.SignalingChannel[*].AutoCancelTimeout = "0"
Isdn.SignalingChannel[*].MaintenanceServiceCallTermination = "Graceful"
Isdn.SignalingChannel[*].LinkEstablishment = "OnDemand"
Isdn.SignalingChannel[*].AcceptedStatusCauses = ""
Isdn.SignalingChannel[*].SendIsdnProgress = "SendAll"
Isdn.SignalingChannel[*].SendProgressIndicatorIE = "SendInbandOnly"
Isdn.SignalingChannel[*].AocESupport = "No"
Isdn.SignalingChannel[*].CallReroutingBehavior = "Unsupported"
Isdn.SignalingChannelInterop[*].CallProceedingDelay = "0"
Isdn.SignalingChannelInterop[*].CallingNameDelivery = "SignalingProtocol"
Isdn.SignalingChannelInterop[*].InteropPlayLocalRingbackWhenNoMediaStream
 = "Disable"
Isdn.PhysicalLink[*].L1TimerT3 = "3000"
Isdn.PhysicalLink[*].ClockMode = "Auto"
Isdn.PhysicalLink[*].MonitorLinkStateEnable = "Enable"
Isdn.PrimaryRateInterfaceInterop[*].MaximumFacilityWaitingDelay = "100"
# Restart all the services that require it.
Scm.RestartRequiredServices
# Initiate a unit reboot only if the new hardware settings require it.
Conf.RebootIfNeeded
```

## Script to Configure an ISDN BRI Interface

```
Isdn.BasicRateInterface[*].EndpointType = "Te"
Isdn.BasicRateInterface[*].ConnectionType = "PointToPoint"
Isdn.BasicRateInterface[*].OverlapDialingEnable = "Disable"
Isdn.BasicRateInterface[*].SendingCompleteEnable = "Disable"
Isdn.BasicRateInterface[*].SendRestartOnStartupEnable = "Disable"
Isdn.BasicRateInterface[*].PreferredEncodingScheme = "G711alaw"
Isdn.BasicRateInterface[*].FallbackEncodingScheme = "G711ulaw"
Isdn.SignalingChannel[*].Protocol = "Dss1"
Isdn.PhysicalLink[*].L1TimerT3 = "3000"
Isdn.PhysicalLink[*].ClockMode = "Auto"
Isdn.PhysicalLink[*].MonitorLinkStateEnable = "Enable"
# Restart all the services that require it.
```

```
Scm.RestartRequiredServices
```

## Script with Paused Execution

```
# Syslog Server IP address here
Nlm.SyslogRemoteHost = "192.168.2.97"

# Turn on TAC Diagnostics Syslog
Nlm.DiagnosticTracesEnable = "Enable"
Nlm.DiagnosticTracesFilter = ""

# Syslog Duration timer
Pause 180

# Turn off TAC Diagnostics Syslog
Nlm.SyslogRemoteHost = ""
Nlm.DiagnosticTracesEnable = "Disable"
```

# Copyright Notice

Copyright © 2026 Media5 Corporation.

This document contains information that is confidential and proprietary to Media5 Corporation.

Media5 Corporation reserves all rights to this document as well as to the Intellectual Property of the document and the technology and know-how that it includes and represents.

This publication cannot be reproduced, neither in whole nor in part, in any form whatsoever, without written prior approval by Media5 Corporation.

Media5 Corporation reserves the right to revise this publication and make changes at any time and without the obligation to notify any person and/or entity of such revisions and/or changes.

media5 corporation

media5corp.com