Configuring Variables

Expanded Call Context Variables

Expanded Call Context (ECC) variables are variables that you define and enable in the Configuration Manager to store values for a call. You can specify the variable name and data type. The name must begin with the string "user." ECC variables are in addition to the variables the system software defines for each call (PeripheralVariable1 through PeripheralVariable10, CallerEnteredDigits, CallingLineID, and so on).

An ECC variable name can be up to 33 bytes long (1–32 usable characters). Use the following naming convention when creating an ECC variable:

user.<CompanyName>.<VariableDescription>

In this syntax:

- **<CompanyName>** is the name of your company.
- **<VariableDescription>** is a descriptive tag for the variable.

For example:

```
user.Cisco.AcctNum
```

Using this naming convention prevents naming conflicts with any third-party applications that interface with the system software.

Note For a large corporation, you can break **<VariableDescription>** down to include the Business Unit, Division, or other organizational entities.

ECC variables follow these size rules:

- An ECC variable can be either a scalar variable or an array element, each with a maximum length of 210 bytes.

Note Array types are not supported for an agent request.

- The maximum number of elements in an array is 255.
- The maximum buffer size for each scalar variable = 5 + the maximum variable length. The 5 bytes includes 4 bytes to tag the variable and 1 byte for the null terminator.
- The maximum buffer size for each array = 5 + (1 + the maximum length of an array element) * (the maximum elements in the array). There is a null terminator for each element, and a null terminator for the array as a whole.
- You pass ECC variables in an ECC payload which has a 2000-byte limit. The total sum of all the maximum buffer sizes for each variable and each array in one ECC payload cannot exceed 2000 bytes.

  For example, if you intended to use the following:

  - A scalar ECC variable with a maximum length of 100 bytes
  - A scalar ECC variable with a maximum length of 80 bytes
  - An ECC array with a maximum of 9 elements with each element having a maximum length of 200 bytes

  Totaled the buffer size is: (5+100) + (5+80) + (5 + (1+200)*9) = 2004. Because this size is too large, you must change the length of one of the scalar ECC variables or the length of the array ECC variables.

ECC Payloads

You can define as many ECC variables as necessary. But, you can only pass 2000 bytes of ECC variables on a specific interface at any one time. To aid you in organizing ECC variables for specific purposes, the solution has *ECC payloads*.

An ECC payload is a defined set of ECC variables with a maximum size of 2000 bytes. You can create ECC payloads to suit the necessary information for a given operation. You can include a specific ECC variable in multiple ECC payloads.The particular ECC variables in a given ECC payload are called its *members*.

Note For ECC payloads to a CTI client, the size limit is 2000 bytes plus an extra 500 bytes for the ECC variable names. Unlike other interfaces, the CTI message includes ECC variable names.

In certain cases, mainly when using APIs, you might create an ECC payload that exceeds the CTI Server message size limit. If you use such an ECC payload in a client request, the CTI Server rejects the request. For an OPC message with such an ECC payload, the CTI Server sends the message without the ECC data. In this case, the following event is logged, “CTI Server was unable to forward ECC variables due to an overflow condition.”

You can use several ECC payloads in the same call flow, but only one ECC payload has scope at a given moment. TCDs and RCDs record the ID of the ECC payload that had scope during that leg of the call. The *Call.ECCPayloadID* variable contains the ID of the ECC payload which currently has scope.

In solutions that only use the default ECC payload, the system does not create an ECC variable that exceeds the 2000-byte limit for an ECC payload or the 2500-byte CTI Message Size limit. The system does this because it automatically adds all ECC variables to the default ECC payload if that is the only ECC payload.

If you create another ECC payload, the system no longer checks the 2000-byte limit when creating ECC variables. The system creates the ECC variables without assigning them to an ECC payload. Assign the new ECC variable to an appropriate ECC payload yourself through the ECC Payload Tool.

You can create and modify ECC payloads in the **Configuration Manager** > **List Tools** > **Expanded Call Variable Payload List** tool.

Default ECC Payload

The solution includes an ECC payload named "Default" for backward compatibility. If your solution does not require more ECC variable space, you only need the Default payload. The solution uses the Default payload unless you override it.

If your solution only has the Default payload, the solution automatically adds any new ECC variables to the Default payload until it reaches the 2000-byte limit.

Note You cannot delete the Default payload. But, you can change its members.

Important During upgrades, when the system first migrates your existing ECC variables to the Default payload, it does not check the CTI message size limit. The member names might exceed the extra 500 bytes that is allocated for ECC payloads to a CTI client. Manually check the **CTI Message Size** counter in the **Expanded Call Variable Payload List** tool to ensure that the Default payload does not exceed the limit. If the Default payload exceeds the limit, modify it to meet the limit.

In a fresh install, the Default payload includes the predefined system ECC variables. In an upgrade, the Default payload's contents depend on whether the starting release supports ECC payloads:

- **ECC payloads not supported**—During the upgrade, a script adds your existing ECC variables to the Default payload.
- **ECC payloads are supported**—The upgrade brings forward the existing definition of your Default payload.

Note If your solution includes PGs from a previous release that does not support ECC payloads, the Router always sends the Default payload to those PGs. Those PGs can properly handle the Default payload.

ECC Payload Node

The **ECC Payload** node is available from the **General** tab on the **Object Palette**:

Figure 1: Payload icon

Use this node to change the ECC payload that has scope for the following part of your script. Once you select an ECC payload, it has scope for all non-VRU operations until changed. You can select the ECC payload either statically or dynamically by the payload's EnterpriseName or ID.

ECC Payload Use by Interface

This table summarizes the use of ECC payloads in various operations:

Condition	ECC Payload That Is Used
Routing to VRU	Default payload If an ECC payload is specified in the configuration of that VRU, it overrides the Default payload.
Routing to Application Gateway	ECC payload that currently has scope in the script
Routing to Agent PG (including the Unified CM PG and Avaya PG)	ECC payload that currently has scope in the script
Routing to Media Routing PG	Default payload If an ECC payload is specified in the configuration of the VRU for that MR PG, it overrides the Default payload.
Routing to pre-12.0 PG	Always Default payload
Routing to System PG (Agent or VRU)	Always Default payload
Routing to Avaya Aura Symposium PG	Always Default payload
Routing to Aspect PG	Always Default payload
Contact Director to target Unified CCE	ECC payload that currently has scope in the script
Routing to INCRP NIC	ECC payload that currently has scope in the script
Pre-route to Gateway PG on Parent in Parent/Child	Always Default payload

Note If you do not create another ECC payload, the solution uses the Default payload for everything.

ECC Variables for Blended Collaboration or Voice MRDs with Collaboration

ECC variables must be configured in Configuration Manager's Expanded Call Variable List tool (for each integrated application) to route requests using the voice Media Routing Domain.

For Cisco Blended Collaboration or Voice MRDs with Collaboration, the ECC variables are:

- user.cisco.cmb
- user.cisco.cmb.callclass
- user.ece.activity.id
- user.ece.customer.name

Important While their default size is 40 characters, use the Expanded Call Variable List tool in the Configuration Manager to limit the user.cisco.cmb variable to 8 bytes and the user.cisco.cmb.callclass variable to 10 bytes to prevent ECC space limitation issues.

Expanded Call Context Variable Configuration

Expanded call context variable configuration consists of two steps:

- Enable ECC variables
- Define ECC variables

For Web Callback and Delayed Callback to work properly, an ECC variable (also known as a named variable) must be defined. The Cisco CTI driver supports the use of ECC variables in addition to the standard call variables associated with a call. Before an ECC variable can be used, it must be defined in the Unified ICM ECC variable database table.

Note For more information, refer to the *Database Schema Handbook for Cisco Unified Contact Center Enterprise*.

Enable ECC Variables

Step 1 Within the Configuration Manager, double-click **Tools** > **Miscellaneous Tools** > **System Information**.

The System Information window appears.

Step 2 Select the **Expanded call context enabled** check box.

For additional information, refer to the online Help.

Step 3 Click **Save** to apply your changes.

Define ECC Variables

Step 1 Within the Configuration Manager, double-click **Tools** > **List Tools** > **Expanded Call Variable List**.

The **Expanded Call Variable List** window appears.

Step 2 Click **Retrieve** to enable adding ECC variables.

Step 3 Click **Add**.

The **Attributes** property tab appears.

Step 4 Complete the **Attributes** property tab. See the *List Tools Online Help* for details on the **Attributes** property tab.

Step 5 Click **Save** to apply your changes.

What to do next

If you change the configuration of any ECC variable with the **Expanded Call Variable List** tool, restart the Unified CVP Call Server or VRU PIM to force a renegotiation of the ECC variables.

Before you can use the new ECC variable, you must add it to an ECC payload.

Note If your solution only has a Default payload, the solution automatically adds any new ECC variables to the Default payload until it reaches the 2000-byte limit.

Define ECC Payloads

You can create and modify ECC payloads in the **Expanded Call Variable Payload List** tool.

Note The tool checks that the ECC payload does not exceed the 2000-byte limit only when you save your changes. The counters on the **Members** tab only show what the current size is with all the selected members. They are only informational and do not enforce the limit. The limit is enforced when you attempt to save the changes.

To define an ECC payload, you create the ECC payload and then add its members.

Step 1 In the Configuration Manager, open **Tools** > **List Tools** > **Expanded Call Variable Payload List**.

The **ECC Payload List** window appears.

Step 2 Click **Retrieve** to enable adding ECC payloads.

Step 3 Click **Add**.

The **Attributes** property tab appears.

Step 4 Complete the **Attributes** property tab. See the *List Tools Online Help* for details on the **Attributes** property tab.

Step 5 On the **Members** tab, click **Add**.

A dialog box listing all the existing ECC variables appears.

Step 6 Select the members for your ECC payload and click **OK**.

Watch that the **ECC Variable Size** counter does not exceed 2000 bytes. For ECC payloads that go to CTI clients, watch that the **CTI Message Size** counter does not exceed 2500 bytes.

Step 7 Click **Save** to apply your changes.

Validate ECC Variable Size for CTI Server

Before configuring ECC variables, validate the total size of the ECC variables against the following rules and limits:

- Because the total size of the buffer used to store the variables in CTI Server internally is 2500 bytes, the total sum of all the maximum buffer sizes for each scalar variable and arrays must be no greater than 2500.
- The maximum buffer size for each scalar variable = 4 + length of the ECC name + the maximum length of the variable where the 4 bytes includes a 1 byte tag, 1 byte to define the length, and 2 terminating NULL characters.
- The maximum buffer size for each array = (5 + length of the ECC name + the maximum length of array element) * (the maximum number of elements in the array) where the 5 bytes includes a 1 byte tag, 1 byte to define the length, 1 byte for the array index, and 2 terminating NULL characters.
- For example, if you intend to use one scalar ECC variable with a maximum length of 100 bytes named *user.var*, one scalar ECC variable with a maximum length of 80 bytes named *user.vartwo*, and an ECC array named *user.varthree* with a maximum of 9 elements with each element having a maximum length of 200 bytes, the buffer size would be:

  (4+8+100) + (4+11+80) + ((5 + 13 + 200)*9)) = 2169

  where 8 is the length of *user.var*, 11 is the length of *user.vartwo* and 13 is the length of *user.varthree*.

User Variables

You can also create global user variables; for example, you can create a user variable called usertemp to serve as a temporary storage area for a string value used by an If node.

After you have defined a user variable, you can then use the Script Editor Formula Editor to access the variable and reference it in expressions, just as you would with a “built-in” variable.

Each user variable must:

- Have a name that begins with **user**.

Note This name cannot contain the dot/period (.) character.

- Be associated with an object type, for example, Service. (This enables the system software to maintain an instance of that variable for each object of that type in the system.)
- Be checked as persistent. A persistent variable maintians its value between script invocations. This allows you to set the variable in one script and reference later in another script.

Note Because these variables may be persisted, do not use User Variables to store sensitive information belonging to the customer or company. Using these variables to store confidential information could lead to violation of security standards, such as PCI, the Common Criteria, HIPAA, or FIPS 140-2.

A user variable can store a value up to 40 characters long.

Define User Variables

Step 1 Within the Configuration Manager, select **Tools** > **List Tools** > **User Variable List**.

The User Variable List window appears.

Step 2 In the User Variable List window, click **Retrieve** to enable Add.

Step 3 Click **Add**.

The Attributes property tab appears.

Step 4 Complete the Attributes property tab.

Note The **Variable name**, **Object type**, and **Data type** fields are required. All other fields are optional. For additional information refer to the online Help.

Step 5 Click **Save** to apply your changes.