# Oracle® Communications Convergent Charging Controller

## Data Access Pack Help

Release 12.0.6
F41574-01
August 2022

Oracle Communications Convergent Charging Controller Data Access Pack Help, Release 12.0.6

F41574-01

# Contents

## 1 Resources

## 2 Import WSDL

# 1
# Resources

This chapter explains how to use the Resources screen to configure Application Service Providers (ASP) and make operations available to DAP.

This chapter contains the following topics.

## Resources Screen

The DAP Resources screen is used to configure the Oracle Communications Convergent Charging Controller Data Access Pack (DAP) application. It contains these tabs:

- ASP
- Operations
- Operation Sets

## ASP

You use the **ASP** tab to configure the ASPs that are available to the DAP application.

### About Specifying URLs

When you configure an ASP you must specify the URL to which requests are sent. You can also specify an URL as the file location to use for importing WSDLs. The URL can contain either the host name of the ASP or its Internet Protocol (IP) address, and an optional port number. You can specify an IP version 6 (IPv6) or an IP version 4 (IPv4) address.

If you specify an IPv6 address and port number in the URL, then you must enclose the IPv6 address in square brackets []; for example: **[2001:db8:**`n:n:n:n:n:n`**]** where `n` is a group of 4 hexadecimal digits. The industry standard for omitting zeros is also allowed when specifying IP addresses. Note that square brackets are not required for IPv4 addresses or if the port number is not included in the URL.

**Example IPv4 and IPv6 addresses**

192.0.2.1:4000

[2001:db8:0000:1050:0005:0600:300c:326b]:3004

[2001:db8:0:0:0:500:300a:326f]:1234

2001:db8::c3

If the URL starts with "http:" or "https:", then you must append a trailing forward-slash, "/", after the host name and port. For example:

```
http://domain_name:port/
```

```
http://domain_name:port/mydoc
```

Where:

- `domain_name` is the URL domain name or IP address.
- `port` is the port number to use.

## ASP tab fields

This table describes the function of each field on the **ASP** tab. The records that display on the **ASP** tab are sorted by the **Name** field.

| Field | Description |
|---|---|
| Name | (Required) The unique name for this ASP connection. The ASP name can be up to 64 alphanumeric characters in length.<br>**Note:** When you save a new ASP, this field becomes read only and may not be edited. |
| Description | (Required) A description of the ASP that can be up to 64 characters in length. |
| Destination URL | (Required) The destination URL to which requests are sent. Specify either the host name of the ASP or the IP address, and an optional port number. The destination URL can be up to 256 characters in length.<br>For more information about specifying the destination URL, see About Specifying URLs. |
| Protocol | The protocol field contains the protocol that should be used when interacting with the ASP. The protocol can be one of: XML, SOAP, HPSA, PIXML, or LDAP.<br>**Note:** You define which protocols are available to ASPs in the Mapping parameters configuration. |
| Connection | The type of the connection (normal/HTTP or encrypted/HTTPS).<br>**Note:** This field is only available for the SOAP and XML protocols. |
| Authenticate | Whether to check server authentication.<br>**Note:** This field is only available for HTTPS connectivity.<br>For more information about server authentication, see HTTP and HTTPS Connections. |

| Field | Description |
|---|---|
| PI User | The PI user for whom PI commands may be triggered by the DAP Send Request feature node. The PI logs in the specified PI user automatically when a PI command is first triggered.<br>**Note:** This field is only available for the PIXML protocol.<br>This field is populated by the **PI Users** tab in the PI Administration screen. For more information, see *PI User's Guide*. |
| Destination Auth User | The user name to authenticate with the ASP server using HTTP basic authentication.<br>**Note:** This field is only available for the SOAP, XML, and LDAP protocols. |
| Max Secondary Connections | This field contains the maximum number of secondary connections that can be created by the DAP interface. |

## ASP configuration

The table on the **ASP** tab in the Resources screen displays the ASPs that are currently available to the DAP. Follow these steps to edit or create an ASP.

For more information about the fields on this screen, see ASP tab fields.

1. On the ASP tab do one of the following:
   - If you want to create a new ASP, click **New**.
   - If you want to edit an existing ASP, select the ASP record in the table and click **Edit**.

   The ASP Configuration screen displays.
2. If you are creating a new ASP, enter the name of the ASP in the **Name** field.

   **Note:** When you save a new ASP, this field becomes read only and may not be edited.
3. Enter a description for the ASP in the **Description** field.
4. Select the protocol to use when interacting with the ASP from the **Protocol** list. The available protocols depend on the Mapping parameters configuration.

   The following table lists the additional fields that become enabled if you select one of the listed protocols.

| Protocols | Fields |
|---|---|
| XML, SOAP, and LDAP | Connection<br>Do HTTP Authentication? |
| PIXML | PI User |
| SLEE | Destination Interface |

5. Specify the destination URL to which requests should be sent in the **Destination URL** field. You can specify the host name of the ASP or the IP address, and an optional port number. For more information, see About Specifying URLs.

6. If you selected `SLEE` in the **Protocol** field, then specify the SLEE interface that the DAP should use when sending requests in the **Destination Interface** field.
7. If the protocol is XML or SOAP, select whether the connection should use HTTP or HTTPS.
8. If you selected an HTTPS connection and you want to ensure that it matches the server name in the HTTPS certificate, select the **Authenticate Server** check box.
9. If the protocol is PIXML, select the required PI user from the **PI User** list.
10. If required, specify the maximum number of secondary connections that can be created by the DAP interface in the **Max Secondary Connections** field.
11. If the protocol is XML or SOAP and you want to configure HTTP authentication then perform the following steps:
    1. Select the **Do HTTP Authentication?** check box.
    2. In the **Destination Auth User** and the **Destination Auth Password** fields, enter the user name and password to authenticate with, .
12. Click **Save**.

    The ASP is saved to the database.

    **Note:** When editing an ASP description, the **Save** button is disabled if the new description is the same as the original description.

**Related topic**

ASP

## Deleting an ASP

Follow these steps to delete an ASP from the service.

1. Select the ASP that you want to delete in the table on the **ASP** tab.
2. Click **Delete**.

   The Delete Confirmation prompt displays.

   **Note:** An error is raised if the ASP has any operations attached to it.
3. Click OK**.**

   The ASP record is removed from the database.

**Related topic**

ASP

# Operations

You use the **Operations** tab on the Resources screen to configure the operations that are available to the DAP application. These operations are used by the Send Request or DAP Request feature nodes.

## Operations tab fields

This table describes the function of each field in the **Operations** tab.

| Field | Description |
|---|---|
| Name | (Required) The name of the operation. The operation name can be up to 64 alphanumeric characters in length.<br>**Note:** When you save a new operation, this field becomes read only and may not be edited. |
| Description | (Required) A description of the operation. The description can be up to 64 characters in length. |
| ASP | The ASP associated with the operation.<br>This field is populated by the records configured on the ASP tab. |
| Timeout | The number of milliseconds that dapIF should wait for a response from the ASP. |
| Correlate | Indicates return parameters are expected. See Correlation. |

## Correlation

When the correlation comment `<!--CORRELATE-->`, or `<!--CORRELATE-ID-->` is specified within the request script, it is implied that return parameters will be sent back by the ASP. The return parameters can be stored within ACS in a user defined profile block and field.

The correlation tag is only supported in asynchronous connections. For more information about asynchronous connections, see Synchronous and asynchronous connections.

## Adding an operation

Follow these steps to add a new operation and the associated request and response templates.

For more information about the fields on the **Operations** tab, see Operations tab fields.

1. On the **Operations** tab, click **New**.

   The **Request** tab in the New Operation screen displays.
2. Type the name of the new operation in the **Name** field.

   **Note:** After you save an operation, the name is no longer editable.
3. Type the description for the operation template in the **Description** field.
4. Select the ASP to associate with the operation from the **ASP Name** list.
5. (Optional) Change the timeout specified for a response to the DAP interface from the ASP by selecting a different value in the **Timeout** field.
6. (Optional) Select the operation set for this operation from the **Operation Set** list. For more information, see Operation Sets.
7. If the ASP uses the SOAP protocol then you can optionally configure the following:
   - To allow the SOAP action in the HTTP header field to be overridden, enter the SOAP action override in the **SOAP Action** field. By default, the DAP sends the destination URL that is configured for the ASP.

- To send the SOAP Header tag select the **Send SOAP Header** check box. The DAP sends an empty SOAP Header tag that uses the form: `<soapenv:Header/>`. Select to send the SOAP Header tag only if the ASP is able to accept an empty SOAP Header tag.
- To define SOAP headers that will be used only by this template, enter the SOAP header definitions in the **SOAP Override header** field. The specified SOAP header overrides the standard XML and SOAP header tags, or the override header tags defined for all DAP operations in the `soapHeaderOverride` parameter. See soapHeaderOverride for details.

8. Select the **Time Sensitive** check box to discard SOAP requests after a certain period of time if the destination ASP cannot be reached. By default, this check box is unchecked.

   In the **Discard if Not Sent After (sec)** field, enter the total number of seconds which the request will be retried for before it is discarded.

   **Note**: The **Discard if Not Sent After (sec)** field is enabled only if the **Time Sensitive** check box is selected.

   If set, **Discard if Not Sent After (sec)** will take priority over discardPendingQueueRequestsAfterSeconds parameter.

9. In the **Request Template** section create the template for the request by entering a valid script in XML format. For more information about script formats, see Script Format.

10. Configure the variables listed in the **Request Parameters** area that have an error status and corresponding message in the **Error/Notice** area. A variable is any item within an element that is enclosed in double angle brackets <<>> and that has a $ prefix, or an empty element.

    For information about variable configuration, see Request Parameter Configuration. For more information about variables, see Variables.

11. Add and configure the response parameters that you require on the **Responses** tab. See Response Parameter Configuration.

12. Click **Save.**

    The **Save** button is available when there are no errors listed in the **Error/Notice** area.

**Related topic**

Operations

## Script Format

The script entered in the script text field of a template forms the body of the request template. The expected format is XML.

A parameter is defined either by using the $ prefix or by using an empty element:

```
<voicemail>
   <!--CORRELATE-->
   <msisdn>$1</msisdn>
   <language></Language>
   <date>$time$date</date>
</voicemail>
```

The parameters specified in the example are $1, language, $date and $time.

DAP requests sent through the SLEE have a limitation of 10 parameters per script. This limit is enforced at run time. A parameter can be up to 32 characters in length.

**Note:** The "$" character will not be treated as a parameter within the text of a tag if it is escaped, that is, "\$".

## Request Parameter Configuration

Follow these steps to configure operation request parameters.

1. Click on the parameter to configure.

   The parameter name is displayed in the **Name** field on the **Request** tab and the editable parameter fields are made available. The **Iterator For** panel is populated with all the request template elements.

2. In the **Description** field type a description for the parameter.

3. Select the feature node behavior for the parameter from the **Node Disposition** list.

   **Note:** The option you select determines whether or not you are required to configure a run-time parameter in a DAP feature node that sends XML requests.

   Select one of the following options:

   - `Hidden` – If the parameter will not be visible in the feature node configuration window and the parameter must be configured in this screen
   - `Text` – If the parameter will be completed in the feature node configuration window as a text field
   - `Profile Block –` If the parameter will be completed in the feature node configuration as a profile block location
   - `Either` – If the parameter will be completed in the feature node configuration either as a text field or as a profile block location
   - Transient – If you do not want to provide a run-time parameter and you also want to send the text defined in the XML template without any further processing.

4. **Note:** The Transient option enables the DAP to process XML requests containing attributes where the parameters are not recognized.

5. If you set **Node Dispostition** to `Hidden`, then specify a value for the parameter by configuring either, or both, the following fields:

   - (Optional) Enter the default value for the parameter in the **Default Value** field.
   - (Optional) Select the parameter location from the following lists: **Profile Field Type**, **Profile Block** and **Profile Field**.

6. **Note:** You can configure any missing profile fields in the ACS Configuration window in the ACS UI. For information about configuring profile fields, see *Advanced Control Services User's Guide*.

7. If you want to use this parameter as an iteration value for an element in a request template, scroll through the **Iterator For** list to find the element, and select the **Add** check box.

   The parameter is added to the request template element as an iteration.

8. Repeat steps **1** to **5** for each parameter that you need to configure.

**Related topic**

Operations

# Response Parameter Configuration

Follow these steps to add and configure any required response parameters.

1. Select the **Response** tab in the New Operation screen.
2. To add a new parameter, click **Add.**

   A new editable row is added to the table in the **Response Parameters** area.
3. Type the parameter name in the empty **Name** field.
4. Press Enter, or click on the **Name** field heading.
5. The status field displays `error`, and the parameter configuration fields are enabled. The error is also listed in the **Error/Notice** area.
6. Type a description for the parameter in the **Description** field.
7. Type a valid XPath search expression in the **Search** Expression field.
8. If this parameter must be returned, select the **Required** check box.
9. If this parameter can be edited in the feature nodes, select the **Node Editable** check box.
10. If the return of this parameter is an error condition, select the **Indicates Error** check box.

    **Note:** The **Required** and **Indicates Error** check boxes are mutually exclusive.
11. Select the profile field location for this parameter from the **Profile Field Type**, **Profile Block** and **Profile Field** lists.

    **Note:** If an expected profile field is missing, it can be added via the ACS Configuration screens. For information about configuring profile fields, see *Advanced Control Services User's Guide*.
12. Repeat steps 2 to 10 for all the response parameters that you want to add.
13. Click **Save**.

    **Note:** The **Save** button is enabled when all errors indicated in the **Error/Notice** area are resolved.

Related topic

Operations

# Editing an Operation

Follow these steps to edit an operation on the **Operations** tab.

For more information about the fields on the **Operations** tab, see Operations tab fields.

1. Select the operation that you want to edit on the **Operations** tab and click **Edit**.

   The **Request** tab for the operation displays in the Edit Operation screen. The **Name** field for the operation is read only.

2. (Optional) Change the description for the operation in the **Description** field.
3. (Optional) Select a different ASP to associate with the operation from the **ASP** list.
4. (Optional) Change the timeout specified for a response to the DAP interface from the ASP by selecting a different value in the **Timeout** field.
5. (Optional) Select a different operation set from the **Operation Set** list.
6. If the ASP uses the SOAP protocol then you can optionally configure the following:
   - To allow the SOAP action in the HTTP header field to be overridden, enter the SOAP action override in the **SOAP Action** field. By default, the DAP sends the destination URL that is configured for the ASP.
   - To send the SOAP Header tag select the **Send SOAP Header** check box. The DAP sends an empty SOAP Header tag that uses the form: `<soapenv:Header/>`. Select to send the SOAP Header tag only if the ASP is able to accept an empty SOAP Header tag.
   - To define SOAP headers that will be used only by this template, enter the SOAP header definitions in the **SOAP Override header** field. The specified SOAP header overrides the standard XML and SOAP header tags, or the override header tags defined for all DAP operations in the `soapHeaderOverride` parameter. See soapHeaderOverride for details.
7. (Optional) Change the **Request Template** script and update the configuration for any parameters listed in the **Error/Notice** area. See Request Parameter Configuration for more information.
8. If required, select the **Response** tab and update the parameters configured on the tab by adding, editing and deleting response parameters:
   - To add a new response parameter or edit an existing response parameter, see Response Parameter Configuration.
   - To delete a response parameter select the parameter you want to delete and click **Delete**.

   **Tip:** If you delete a parameter by mistake, then add a new parameter with the same name as the deleted parameter.
9. Click **Save**.

   **Note:** The **Save** button is enabled only when there are no errors listed in the **Error/Notice** panel.

**Related topic**

Operations

# Finding a DAP template

Follow these steps to search for a DAP operation.

1. On the **Operations** tab, click **Find**.

   **Result:** You see the Find Operations screen.
2. Enter the search criteria in one of the query fields using the drop down list and click **Find**.

   **Result:** When you click **Find** a query is triggered and the first 100 records matching the value in the query field will be returned. The results appear in the table on the Operations tab.

**Note:** If you select a value in more than one query field then the **Find** button will be disabled.

**Related topic**

Operations

## Copying DAP templates

Follow these steps to copy a defined DAP operation.

1. On the **Operations** tab, select the operation to copy.
2. Click **Copy**.

   **Result:** You see the Copy DAP Operation screen.
3. In the **Name** field, type a unique name for the new operation.
4. Click **Save**.

   **Result:** The operation will be saved to the database under the new name.

   **Note:** This operation will have exactly the same details as the original operation. See Editing an Operation to change the operation details.

**Related topic**

Operations

## Deleting an operation

Follow these steps to delete an Operation from the service.

1. In the table on the **Operations** tab, select the operation to delete.
2. Click **Delete**.

   **Result:** You see the Delete Confirmation prompt.

   **Note:** An error is raised if the operation exists in an active control plan.
3. Click **OK**.

   **Result:** The operation is removed from the database.

**Related topic**

Operations

# Operation Sets

Operation sets are used to limit the operations to selected users.

## Add or edit an operation set

The table in the **Resources** screen **Operation Sets** tab displays the operation sets that are currently available in the system.

Follow these steps to add a new operation set or edit an existing set.

1. On the **Operation Sets** tab:

- To create a new set, click **New**
- To edit an existing set, select the set, then click **Edit**

2. **Result:** You see the Operation Set Configuration screen.
3. In the **Name** field, enter or edit the name of the operation set.
4. Select the check box for all the listed users you want in this operation set.
5. Click **Save.**

**Related topic**

Operation Sets

## Deleting an operation set

Follow these steps to delete an operation set.

1. In the table on the **Operation Sets** tab, select the operation set to delete.
2. Click **Delete**.

   **Result:** You see the Delete Confirmation prompt.
3. Click **OK**.

   **Result:** The operation set is removed from the database.

**Related topic**

Operation Sets

# 2 Import WSDL

This chapter explains how to use the DAP Import WSDL screen.

This chapter contains the following topics.

## Import WSDL Screen

The Oracle Communications Convergent Charging Controller Data Access Pack DAP Import WSDL screen allows you to import and configure predefined web services from ASPs.

It contains these functions:

- Importing a WSDL file
- ASP configuration
- Operation Request Configuration
- Operation Response Configuration

### What is WSDL?

Web Services Description Language (WSDL) is a XML based language that provides a model for describing web services.

The reason for using WSDL is to import predefined web services from ASPs, thereby speeding up configuration of DAP messages.

For restrictions on what parts of the specification are supported, see *Convergent Charging Controller Data Access Pack PICS Guide*.

### Accessing the Import WSDL screen

Follow these steps to open the DAP Import WSDL screen.

1. Select the **Services** menu from the SMS main screen.
2. Select **DAP**.
3. Select **Import WSDL**.

   **Result:** You see the Import WSDL screen.

# Importing a WSDL file

Follow these steps to import a WSDL file.

1. On the Import WSDL screen, perform one of the following actions to select a file:

   - Enter the location and name of the **.wsdl** file in the **File/URL** field. The file location can be an URL. For information about specifing URLs, see About Specifying URLs.
   - Click **Browse** to find the file

   The file has the suffix of **.wsdl** (for example **DAP.wsdl**)

   **Result:** The **Import** button becomes available.

2. Click **Import**.

   **Result:** The WSDL file is imported, then:

   - The **Operation/Description** panel is populated from the imported file
   - The *No operation has been added* message is displayed in the Notice/Error box.

3. To add:

   - *All* operations, select the **Add** check box
   - *Selected* operations, select the check box for each of the required operations

   **Result:**

   - Any configuration requirements are displayed in the Notice/Error box.
   - The configuration fields under the **ASP**, **Request**, and **Response** tabs are populated with data from the WSDL and become available for editing.
   - The **Operation Set** field becomes available.

     **Warning:** If you select multiple operations, you must configure all of them before the **Save** button is available. If there are any kind of fatal system problems, you will need to re-do any unsaved configuration.

4. Select the **Operation Set** to use from the drop down list.

   **Note:** See Operation Sets for details on configuring this list.

5. Click on the operation to configure.
6. Configure the ASP as required. See ASP for more information.
7. Configure the request as required. See Operation Request Configuration for more information.
8. Configure the response as required. See Operation Response Configuration for more information.
9. Repeat steps 4 to 8 for each operation that you want to configure.
10. When there are no errors listed in the **Error/Notice** area, the **Save** button becomes available.

    Click **Save**.

    The ASP and the configured operations are created. You can view and edit them on the DAP Resources Screen.

# Operation Request Configuration

You use the **Request** tab on the Import WSDL screen to configure the request parameters that are within the imported operation script. The imported operations are used by the DAP Request feature node.

## Configuring requests

Follow these steps to configure operation request parameters.

1. In the **Request** tab, expand the parameter tree to see all the elements that can be configured (click **+** signs).

   **Note:** Each element is preceded by a C, a P, an O, or an E:

   - C – For a complex parameter that has more than one sub parameters.
   - P – For the last element down a branch. This is the parameter that can be configured.
   - O – For an optional parameter that may require configuration.
   - E – For an empty sub-parameter within a complex parameter. This is not configurable.

2. Select the parameters to configure by selecting the check box preceding the parameter.

   All the selected parameter names are colored red to indicate that the parameters require configuration, and the error reasons appear in the **Error/Notice** area. Sometimes all parameters under a selected complex parameter are automatically selected because they are all required.

   **Tip:** Hover over a parameter to get a tip if a box cannot be selected or deselected.

3. Click a parameter name to configure.

   The parameter name appears in the **Name** field, the editable parameter fields are made available, and the **Iterator For** table is populated with all the request elements.

4. If this parameter is to be used as a unique id, select the **Correlation ID** check box and then start configuring a new parameter.
5. Type a description for the parameter in the **Description** field.
6. Select the feature node behavior for the parameter from the **Node Disposition** list.

   **Note:** The option you select determines whether or not you are required to configure a run-time parameter in a DAP feature node that sends XML requests.

   Select one of the following options:

   - `Hidden` – If the parameter will not be visible in the feature node configuration window and the parameter must be configured in this screen.
   - `Text` – If the parameter will be completed in the feature node configuration window as a text field.
   - `Profile Block` – If the parameter will be completed in the feature node configuration as a profile block location.
   - `Either` – If the parameter will be completed in the feature node configuration either as a text field or as a profile block location.

- `Transient` – If you do not want to provide a run-time parameter and you also want to send the text defined in the XML template without any further processing.

**Note:** The Transient option enables the DAP to process XML requests containing attributes where the parameters are not recognized.

7. If you selected the `Hidden` disposition, perform one, or both, of the following steps:

   a. Type the default value for the parameter in the **Default Value** field.

   b. Select the location of the parameter from the **Profile Field Type**, **Profile Block,** and **Profile Field** lists.

   **Tip:** The **Xsd Type** field indicates what type of data is expected in the DAP message for this parameter.

   **Note:** If an expected profile field is missing, it can be added via the ACS Configuration screens. For information about configuring profile fields, see *Advanced Control Services User's Guide*.

8. If this parameter is being used as an iteration value for a request element, scroll to find the element in the **Iterator For** list and select the **Add** check box.

   The parameter will be added to the request template element as an iteration.

   **Tip:** A parameter can be an iterator for more than one element, however one element registers just one parameter as the iterator.

9. Repeat steps **3** to **8** for all the parameters that require configuration.

# Operation Response Configuration

You need to configure any responses required by the imported script. You perform this on the **Response** tab.

## Configuring responses

Follow these steps to configure all response parameters.

1. In the **Response** tab, expand the parameter tree to see all the elements that can be configured (click + signs).

   **Note:** Each element is preceded by a C, a P, an O, or an E:

   - C – For a complex parameter that has more than one sub parameters.
   - P – For the last element down a branch. This is the parameter that can be configured.
   - O – For an optional parameter that may require configuration.
   - E – For an empty sub-parameter within a complex parameter. This is not configurable.

2. Select the parameters to configure by selecting the check box that precedes each parameter.

   All the selected parameter names are colored red to indicate that the parameters require configuration.

3. Click a parameter name to configure.

The parameter name appears in the **Name** field, the editable parameter fields are made available

4. In the **Description** field, type what the response is expected to be.
5. If this parameter must be returned, select the **Required** check box.
6. If this parameter can be edited in feature nodes, select the **Node Editable** check box.
7. If the return of this parameter is an error condition, select the **Indicates Error** check box.

   **Note:** The Required and Indicates Error check boxes are mutually exclusive.
8. Select the profile type and location for the parameter from the **Profile Field Type**, **Profile Block** and **Profile Field** lists.

   **Note:** If an expected profile field is missing, it can be added via the ACS Configuration screens. For information about configuring profile fields, see *Advanced Control Services User's Guide*.

   **Result:** The last error message for this parameter will disappear from the Error/Notices panel.
9. Repeat steps 3 to 8 to configure all the response parameters.

3

# Data Access Pack Feature Nodes

This chapter describes the Oracle Communications Convergent Charging Controller Data Access Pack (DAP) feature nodes.

This chapter contains the following topics.

## DAP Feature Nodes List

This table lists the feature nodes available from the DAP palette group in the ACS Control Plan Editor and the fast key for each feature node in the list. You can use fast keys to search for feature nodes in the palette or the canvas.

| Node name | Node description |
|---|---|
| **Send Request** | The DAP Send Request node allows ACS to send DAP requests to a DAP interface for forwarding to an external ASP.<br>Fast key: DAP1 |
| **DAP Request** | The DAP Request feature node allows ACS to select, configure and then send DAP requests to a DAP interface for forwarding to an external ASP.<br>Fast key: DAP2 |
| **DAP VXML** | **Warning:** This feature node has now been deprecated and should no longer be used. This functionality has been replaced by the Diameter Charging Driver features.<br>This feature node plays an announcement based on a text string stored in a profile.<br>Fast key: DAPV |

**Note:** The Send Request node is deprecated and is included for backward compatibility to enable older control plans to compile and run correctly.

### Profile Blocks and DAP

In order to read from or write to the profile blocks, the control plan needs to have access to the profile. Profiles are generally handled by ACS, however, some components use the profile loaded by the service library which is handling the call. You can load a new profile using the Load Profile feature node.

**Note:** The DAP Send Request node has no service loader of its own, and therefore the profiles that can be written are dependent on the type of control plan used. For example, you can write to Application Specific 7 to move data between USSD and Send Request nodes as this is the temporary profile available within USSD GW control plans.

For more information about which profile is loaded by a service library, see the technical documentation for the component that the service library comes from.

For more information about profile blocks, see *ACS User's Guide*.

# DAP Request

The DAP Request feature node allows ACS to select, configure and then send DAP requests to a DAP interface for forwarding to an external ASP.

Request parameters can be configured in three places:

1. When importing WSDL files (**SMS > DAP > Import WSDL**), before saving the operation, all errors including parameter configuration values must be resolved.
2. After WSDL files have been imported and saved, they appear in the **SMS > DAP > Resources> Operations** dialog, where they may be edited.
3. At control plan creation time, within this node.

**Note:**

- If a parameter has been flagged as hidden either when importing or later editing the operation, this parameter will not be seen in the node parameters list. All required configuration is done outside of the node.
- The opposite is that any parameter not set to hidden, must be configured in the node.

## Node exits

This node has one entry and five exits. The number of exits cannot be changed.

| Exit | Cause | Description |
|---|---|---|
| 1 | Success | This branch taken for any of the following conditions:<br>If the DAP macro node instance receives an indication from the DAP interface indicating that the message has been successfully sent and the outstanding request does not expect parameters to be returned.<br>**Note:** Any unexpected returned parameters are ignored.<br>When all of the expected ASP response parameters have been copied into the stated designated fields or/and profile tag blocks. |
| 2 | Parameter Problem | This branch taken for any of the following reasons:<br>If there is a problem retrieving the value of a stated profile tag (for example, CCS productTypeID field) and no default value has been specified.<br>If the node could not convert the returned value to a profile tag.<br>If the value cannot be stored in the profile tag. |
| 3 | Mismatch | If the outstanding request expects response parameters to be returned from the ASP but the parameters returned from the ASP do not match what is expected. |
| 4 | ASP Failure | If the ASP returns a response that is configured as "Indicating Error". |

| Exit | Cause | Description |
|---|---|---|
| 5 | Error | This branch taken for any of the following conditions:<br>If there is a format problem with the retrieved parameter string.<br>If a DAP interface cannot be determined to send the request to.<br>If an unknown short code is specified for an ACS run time parameter.<br>If the DAP macro node instance receives an indication from the action handler or DAP interface indicating that the message has not been successfully sent.<br>If there is a format problem with the ASP response related parameters.<br>If an unknown short code is specified for an ACS run time parameter within the ASP response parameter information.<br>If an unknown profile application block code is stated within the ASP response parameter information.<br>If the macro node fails to retrieve profile block references. |

## Configuring the node

Follow these steps to configure the node request parameters.

1. Click the **Request** tab if not already displayed.
2. Select a parameter to configure from the **Request Parameters** list (click on the parameter name).

   **Result:** Any errors for this parameter are shown in the text box above the Exit Branches.

   **Note:** The various configuration fields become available for completion according to how the parameter was defined in the **SMS > DAP > Import WSDL** or **SMS > DAP > Resources> Operations** configuration screens:

   - Text - Only the **Description** and **Default Value** fields available.
   - Profileblock - Only the **Description**, **Profile Field Type**, **Profile Block** and **Prfile Field** fields available.
   - Either - all fields are available.
3. Enter a description for the parameter in the **Description** field.
4. If available, optionally enter a default value in the **Default Value** field.

5. If available, select the type and location of the value from the **Profile** drop down lists.

   **Note:** If an expected profile field is missing, it can be added via the ACS Configuration screens. See Profile Tag Details.

6. Repeat steps **2** to **5** for all the parameters in the list.

   **Note:** All errors for all parameters in both tabs must be resolved before the **Save** button is available.

7. Click **Save**.

## Configuring response tab - default

Follow these steps to configure the parameters

1. Click the **Response** tab if not already displayed.
2. Select the parameter to configure a response for from the **Parameter** drop down list.

   **Result:** Any configuration requirements for this parameter are shown in the text box above the Exit Branches.

   **Note:** The various configuration fields become available for completion according to how the parameter was defined in the **SMS > DAP > Import WSDL** or **SMS > DAP > Resources> Operations** configuration screens:

   - Text - Just the Description and Default Value fields available.
   - Profileblock - Just the Description Profile Field Type, Profile Block and Profile Field fields available.
   - Either - all fields are available.

3. Type a description for the parameter in the Description field.
4. If available, optionally type a default value in the **Default Value** field.
5. If available, select the type and location of the value from the **Profile** drop down lists.

   **Note:** If an expected profile field is missing, it can be added via the ACS Configuration screens. See Profile Tag Details.

6. Repeat steps **2** to **5** for all the parameters in the list.

   **Note:** All errors for all parameters in both tabs must be resolved before the Save button is available.

7. Click **Save**.

## View Operation Script

Follow these steps to see the script details behind the parameters this node will use.

1. Click **View**.

   **Result:** You see the Operation Overview screen.

2. Click **Close**.

# Send Request

**Warning:** This feature node has now been deprecated and should no longer be used. This feature node will still function in existing control plans. For new control plans please use the **DAP Request** feature node.

The DAP Send Request feature node allows ACS to send DAP requests to a DAP interface for forwarding to an external ASP.

Optionally following a send, the Send Request node can also read DAP responses received on the DAP interface sent by the ASP.

## Restrictions

A control plan may contain as many Send Request nodes as required.

## Node exits

This node has one entry and four exits. The number of exits cannot be changed.

| Exit | Cause | Description |
|---|---|---|
| 1 | Error | This exit is taken when the receiving system cannot be contacted, or there is a problem with sending the request (for example, when a NACK reply is received from an ASP).<br>**Note:** This branch will be taken at startup if slee_acs has started to process calls before the DAP interface is available. |
| 2 | Success | This exit is taken when either an ACK is received for the message sent, or if the message is successfully sent and the wait flag is not checked. |
| 3 | Mismatch | A mismatch has occurred between the configured and actual tags received in the ASP response. |
| 4 | Problem | If the associated value received within an ASP response cannot be stored within ACS or profile. For example, incorrect size or invalid storage location. |

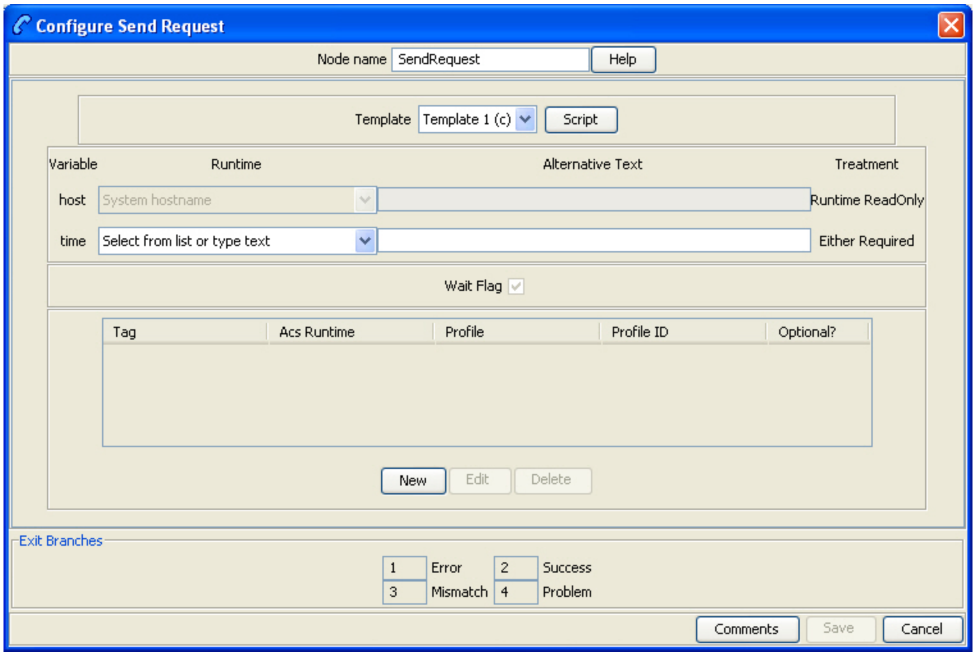
## Configuring the node

Follow these steps to configure the Send Request node.

1. Select the **Template** to be assigned to the feature node.

   **Note:** The templates available in the list are those that have been created using the DAP application screens. Template names suffixed with (c) indicates the template has been created with correlation.

   **Result:** The fields in the box below the template selection list will be populated with the data assigned for the selected template using the **Operations** tab on the DAP Resources screen. The area displays all the variables, but only allows you to select options that were made available by selecting a **Node Disposition** that is not `Hidden`, for each parameter.



Node name: SendRequest

Template: Template 1 (c)

| Variable | Runtime | Alternative Text | Treatment |
|---|---|---|---|
| host | System hostname | | Runtime ReadOnly |
| time | Select from list or type text | | Either Required |

| | | | |
|---|---|---|---|
| 1 | Error | 2 | Success |
| 3 | Mismatch | 4 | Problem |

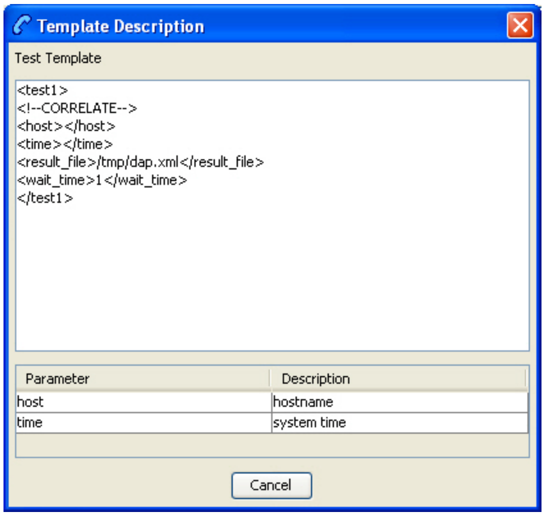
   The **Variable** field displays the parameter variable defined in the script for the template.

   The **Treatment** field displays the permissible treatment options for each variable. Must be one of;

   - Either Required. You must either select the **Runtime** or enter **Alternative Text**.
   - Text ReadOnly. Displays the alternative text supplied in the template.
   - Runtime ReadOnly. Displays the runtime supplied in the template.
   - Runtime Required. You must select the **Runtime**.
   - Text Required. You must enter **Alternative Text** for the parameter.

   **Note:Save** will become available once you have filled in all the required fields in this section.

2. Click **Script** to view the script used by the selected template in the Template Description screen. For more information, refer to Script Format.



Test Template

```
<test1>
<!--CORRELATE-->
<host></host>
<time></time>
<result_file>/tmp/dap.xml</result_file>
<wait_time>1</wait_time>
</test1>
```

| Parameter | Description |
|---|---|
| host | hostname |
| time | system time |

Click **Cancel** to close the screen.

3. If a **Runtime** field is active, you can change the selected runtime value that is to be associated with the variable. The parameter will be replaced by the dynamic value at runtime.
4. If you select the runtime of `Profile tag`, the screen will display the Profile Block group.

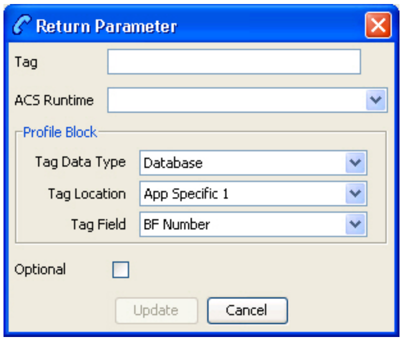
   Select the **Tag Data Type**, **Tag Location** and **Tag Field** from the lists.

   **Notes:**

   - The **Tag Field** is populated with the tags that have a Profile Tag Type of `String`, `Numeric String`, or `Limited Numeric String`, as specified on the **Profile Tag Details** tab of the ACS Configuration screen.
   - The Profile Blocks that can be used are dependent upon the other applications installed and the control plan you are using. For more information about how profile blocks are used, see Profile blocks.
   - If the selected template specifies a Profile Tag that cannot be defined in this screen, edit the selected template on the **Operations** tab on the DAP Resources screen, so that it is available.
5. You create up to 20 return parameters to associate with either an ACS Runtime Variable Part or in a user-defined profile block and ID.

   Click **New** to create a return parameter.

   **Result:** The Return Parameter screen appears.



Tag:

ACS Runtime:

Profile Block

| | |
|---|---|
| Tag Data Type | Database |
| Tag Location | App Specific 1 |
| Tag Field | BF Number |

Optional □

6. In the **Tag** text box, enter the name of the XML tag containing the return details.

   **Note:** This must match an existing XML tag that is returned by the ASP.

   **Example:** For an XML script returned from an ASP containing `<return_callingPartyNumber>441473289900</return_callingPartyNumber>`, the tag name to enter is `return_callingPartyNumber`.

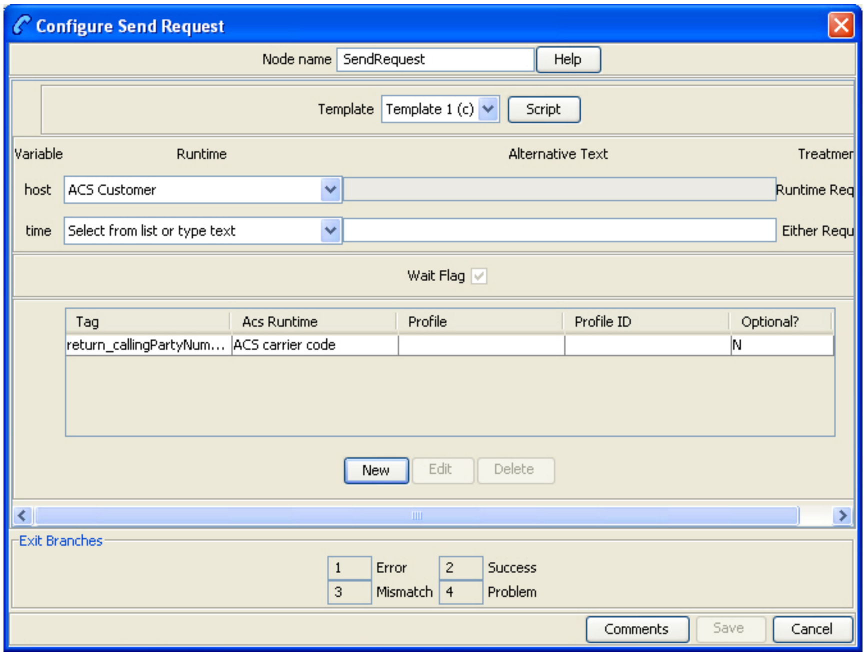
7. Select from either the **ACS Runtime** or the three fields in the **Profile Block** frame.

8. To make the return parameter optional, select the **Optional** check box.

   - If mandatory, this parameter must be returned from the ASP. If it is not, then it will go to exit branch 3 (Mismatch) and will log a syslog error naming the missing parameter.
   - If optional, the node will process the return parameter if it is returned from the ASP. If it is not, then it will still process successfully.

9. Click **Update** to save the return parameter.

   **Result:** The Configure Send Request screen shows the added return parameter.



Node name: SendRequest

Template: Template 1 (c)

| Variable | Runtime | Alternative Text | Treatmer |
|---|---|---|---|
| host | ACS Customer | | Runtime Req |
| time | Select from list or type text | | Either Requ |

Wait Flag ☑

| Tag | Acs Runtime | Profile | Profile ID | Optional? |
|---|---|---|---|---|
| return_callingPartyNum... | ACS carrier code | | | N |

Exit Branches

| | | | |
|---|---|---|---|
| 1 | Error | 2 | Success |
| 3 | Mismatch | 4 | Problem |

**Note:** To edit the return parameter, select the record in the table and click **Edit**.

10. The **Wait Flag** check box displays whether the system waits for a response from the ASP before returning a DAP response to the client.

    If correlation has:

    - Been defined for the template, the **Wait Flag** check box is selected and cannot be modified.
    - Not been defined, you can select the **Wait Flag** check box, if required.
11. Click **Save**.

# DAP VXML

This feature node plays an announcement based on a text string stored in a profile.

## Node exits

This feature node has one entry and three exits. The number of exits cannot be changed.

| Exit | Cause | Description |
|---|---|---|
| 1 | Success | The announcement was played successfully. |
| 2 | Abandon | The caller hung up during the announcement. |
| 3 | Error | General Error Handling. |

## Configuring the node

Follow these steps to configure the node.

**Note:** If an expected profile field is missing, it can be added via the ACS Configuration screens. See Profile Tag Details.

1. In the **Text to Play Location** field, select the profile block in which relevant information is stored.
2. In the **Text to Play Field** field, select the profile tag that identifies the information required by the node.
3. In the **Resource Name** field, enter the name of the media server which will provide the announcement.

   **Warning:** This string must be set to `uugw`.