Common Request

The **Common Request** document is an XML schema that contains default fields for mapping Request form values from one system to another. The common request format mirrors what one would see of the Request form with related lists expressed in embedded XML form.

What's on this page?

Implementation

The implementation of any common attachment format is symmetrical, meaning that the output, when consumed, should produce the same or similar records at the target. The implementation should also exhibit idempotent behavior, meaning when a document is consumed and processed repeatedly, the same result either appears or is ignored, because the results already exist.

Go to top of page

Dynamic Shares

For the correct outputs to be produced, you must create 2 Dynamic Shares - one to capture the **comments** and **work_notes** field values in a **before** c ontext, and the other as **async** mode to capture the related records in a delayed step.

In the **before** Dynamic Share, use the following code snippet to prevent unwanted outbound messages to be queued when **comments** or **work_notes** are not updated.

```
if (current.comments.nil() && current.work_notes.nil()) {
        ignore = true;
}
```

Go to top of page

Embedded Records

Embedded records are fields in the Common Request document that cannot be mapped by singular values. Instead, they are entirely different records **embedded** inside the Common Request document to be created on the subscribing system.

The following are examples of the embedded records in Common Request:

Approver Record

The **approver** field in the Common Request is an embedded record that, when populated, will create a new approver on the system receiving the Common Requested.

Table Map Source Field	Definition

created	Date the record was created on
approver_name	Name of the approver
assignment_group_name	Name of the assignment group
approver	Unique record identifier of the approver record
state	State of the approval
comments	Comments of the approver
approver_email	Email of the approver
assignment_group	Unique record identifier of the assignment group record

Attachment Embedded Record

The **attachment** field in the Common Request is an embedded record that, when populated, will create a new attachment on the system receiving the Common Request.

Table Map Source Field	Definition
data	Attachment data in an encoded string format
size_bytes	Measurement of how much the attachment data contains
file_name	Name of the attached file
sys_id	Unique record identifier of the attachment
content_type	Attachments content type (i.e jpeg, png, txt, etc.)

Group Approval Embedded Record

The **group approval** field in the Common Request is an embedded record that, when populated, will create a new group approval on the system receiving the Common Request.

Table Map Source Field	Definition
approval	Status of the catalog task's approval
approval_user	Unique record identifier of the approval user
assignment_group	Unique record identifier of the assignment group
short_description	Short description of the group approval

Recurring Price Record

The **recurring price** field in the Common Request is an embedded record that, when populated, will create a new recurring price on the system receiving the Common Request.

Table Map Source Field	Definition
recurring_frequency	Rate at which the recurring prices occurs
recurring_price	Price of the recurring prices

Requested Item Record

The **requested item** field in the Common Request is an embedded record that, when populated, will create a new requested item on the system receiving the Common Request.

Table Map Source Field	Definition
price	Price of the requested item
stage	Development of the requested item

quantity	Amount of requested item
assigned_to	User the requested item is assigned to
due_date	Date the requested item must be fulfilled
cat_item	Catalog item reference
number	Record number of the requested item
sc_catalog	Catalog reference

Go to top of page

Table Maps

Outbound Table Maps

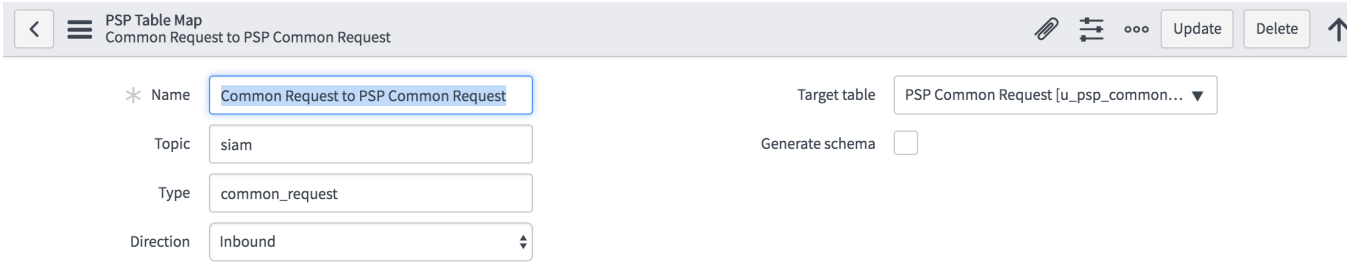
The following table maps construct the outbound messages to be queued. Specify the main **Request to Common Request** Table Map in your Dynamic Share as the root map.

Name	Type	Source table	Description
Request to Common Request	common_request	Request [sc_request]	Main body of the common_request format
psp_attachment	embedded_attachment	sys_attachment	Map for building embedded attachments field
psp_requested_item	embedded_requested_item	Requested Item [sc_req_item]	Map for building embedded requested items
psp_approver	embedded_approver	Approval [sysapproval_approver]	Map for building embedded approvers
psp_group_approval	embedded_group_approval	Group approval [sysapproval_group]	Map for building embedded group approvals
psp_recurring_price	embedded_recurring_price	Recurring Price [sc_recurring_rollup]	Map for building embedded recurring prices

NOTE: In Helium, the Catalog Task (**sc_task**) table map is included as part of the **Common Request** update set. To learn more about catalog tasks, see view and edit a catalog task.

Inbound Table Map

In order to process messages of topic: siam and type: common_request, you must create an inbound table map to target the import set table as follows. (This map should have been included in the provided update set already)



Import Set

The import set table is called **u_psp_common_request** and has a transform map called **PSP Common Request to Request**. This transform map has transform scripts that are responsible for parsing the embedded sections of common_request:

- attachments
- requested_items
- approvers
- group_approvals
- recurring_prices

NOTE: Select to deactivate each of these transform scripts if you choose not to parse these embedded objects into records. If you need to extend or add additional capabilities, instead of modifying the existing scripts, add new ones so that upgrades are possible.

These transform scripts use specific script includes to assist in parsing and ingesting the embedded objects into the correct artifacts. The script includes are

- PerspectiumAttachment

- PerspectiumRequestedItem
- PerspectiumApprover
- PerspectiumGroupApproval
- PerspectiumRecurringPrice

These script includes are included as part of the Common Endpoint update set that should be installed prior to installing the common object format update sets.

Variables

To access variables in your table maps scripts, you can use the variables object of the record. For example, to put all of a requested item's variables into the description field of the outbound table map record:

```
var variables = current.variables.getElements();
var output = "Variables: \r\n ";
for (var i = 0; i<variables.length; i++) {
    var question = variables[i].getQuestion();
    output = output + (question.getLabel() + ":" + question.getValue() + "\r\n")  ;
}

answer = current.description +"\r\n\r\n"+ output;
```

Go to top of page

Sample Output

A sample **Common Request** looks like this:

```
<common_request>
    <active />
    <activity_due />
    <additional_assignee_list />
    <approval />
    <approval_history />
    <approval_set />
        <approvers>
                <approver>
            <created/>
            <approver_name/>
            <assignment_group_name/>
            <approver/>
            <state/>
            <comments/>
            <approver_email/>
            <assignment_group/>
        </approver>
        </approvers>
    <assigned_to />
    <assignment_group />
    <attachments>
        <attachment>
                    <data />
                    <size_bytes />
                    <file_name />
                    <sys_id />
                    <content_type />
                </attachment>
    </attachments>
    <business_duration />
    <business_service />
    <calendar_duration />
    <calendar_stc />
    <closed_at />
    <closed_by />
    <close_notes />
    <cmdb_ci />
```

```
<comments />
<comments_and_work_notes />
<company />
<contact_type />
<correlation_display />
<correlation_id />
<delivery_address />
<delivery_plan />
<delivery_task />
<description />
<due_date />
<escalation />
<expected_start />
<follow_up />
<group_approvals>
             <group_approval>
        <short_description/>
        <assignment_group/>
        <approval_user/>
        <approval/>
    </group_approval>
    </group_approvals>
<group_list />
<impact />
<knowledge />
<location />
<made_sla />
<number />
<opened_at />
<opened_by />
<order />
<parent />
<price />
<priority />
<provider />
<reassignment_count />
<recurring_prices>
            <recurring_price>
                <recurring_frequency />
                <recurring_price />
            </recurring_price>
</recurring_prices>
<rejection_goto />
<requested_date />
<requested_for />
<requested_items>
    <requested_item>
                <price />
                <stage />
                <quantity />
                <assigned_to />
                <due_date />
                <cat_item />
                <number />
                <sc_catalog />
    </requested_item>
</requested_items>
<request_state />
<short_description />
<sla_due />
<special_instructions />
<stage />
<state />
<sys_class_name />
<sys_created_by />
<sys_created_on />
<sys_domain />
<sys_domain_path />
<sys_mod_count />
<sys_tags />
<sys_updated_by />
```

```
    <sys_updated_on />
    <time_worked />
    <upon_approval />
    <upon_reject />
    <urgency />
    <user_input />
    <variables />
    <watch_list />
    <wf_activity />
    <work_end />
    <work_notes />
    <work_notes_list />
    <work_start />
</common_request>
```

The following Perspectium Inbound/Outbound Message unloads can be un-gzipped and uploaded into your instance of ServiceNow to view and used for testing.

Common Request Sample - Inbound.xml.zip

Common Request Sample - Outbound.xml.zip

Go to top of page