

Common Problem

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

What's on this page?

- [Implementation](#)
- [Dynamic Shares](#)
- [Embedded Records](#)
- [Table Maps](#)
 - [Outbound Table Maps](#)
 - [Inbound Table Map](#)
 - [Import Set](#)
- [Sample Output](#)

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. Please note this is only necessary if it applies to your use case.

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

[Go to top of page](#)

Embedded Records

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

These are the supported embedded records in Common Problem:

Attachment Embedded Record

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

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.)

Incident Record

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

Table Map Source Field	Definition
number	Record number of incident
opened_at	Date the incident is opened
category	Categorization of incident
caller_id_name	Name of the caller
assigned_to_name	Name of the user the incident is assigned to
short_description	Short description of incident
priority	Priority of incident
id	Record identifier of the incident
state	State of the incident
assignment_group_name	Name of the assignment group assigned to the incident

Problem Task Record

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

Table Map Source Field	Definition
cmdb_ci	Unique record identifier of the configuration item
assigned_to	User the problem is assigned to
state	State of the problem
short_description	Short description of problem
assignment_group	Unique record identifier of the assignment group record
number	Record number of problem
priority	Priority of problem

[Go to top of page](#)

Table Maps

Outbound Table Maps

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

Name	Type	Source table	Description
Problem to Common Problem	common_problem	Problem [problem]	Main body of the common_problem format
psp_attachment	embedded_attachment	sys_attachment	Map for building embedded attachments field

psp_incident	embedded_incident	Incident [incident]	Map for building embedded incidents
psp_problem_task	embedded_problem_task	Problem Task [problem_task]	Map for building embedded problem tasks

Inbound Table Map

In order to process messages of topic: siam and type: common_problem, 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)

<

≡

PSP Table Map
Common Incident to PSP Common Incident

Update

Delete

↑

↓

* Name

Common Incident to PSP Common Incider

Target table

PSP Common Incident [u_psp_com... ▼]

Topic

siam

Generate schema

☐

Type

common_incident

Direction

Inbound ▼

Import Set

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

- attachments
- incidents
- problem_tasks

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
- PerspectiumIncident
- PerspectiumProblemTask

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.

[Go to top of page](#)

Sample Output

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

```
<common_problem>
  <active />
  <activity_due/>
  <additional_assignee_list/>
  <approval />
  <approval_history/>
  <approval_set/>
  <assigned_to/>
  <assignment_group/>
  <attachments>
    <attachment>
      <data />
      <size_bytes />
      <file_name />
      <sys_id />
      <content_type />
    </attachment>
  </attachments>
```

```
<business_duration/>
<business_service/>
<calendar_duration/>
<closed_at/>
<closed_by/>
<close_notes/>
<cmdb_ci/>
<comments/>
<comments_and_work_notes/>
<company/>
<contact_type/>
<correlation_display/>
<correlation_id/>
<delivery_plan/>
<delivery_task/>
<description />
<due_date/>
<escalation />
<expected_start/>
<follow_up/>
<group_list/>
<impact />
<incidents>
  <incident>
    <number />
    <opened_at />
    <category />
    <caller_id_name />
    <assigned_to_name />
    <short_description />
    <priority />
    <id />
    <state />
    <assignment_group_name />
  </incident>
</incidents>
<knowledge />
<known_error />
<location/>
<made_sla />
<number />
<opened_at />
<opened_by />
<order/>
<parent/>
<priority />
<problem_state />
<problem_tasks>
  <problem_task>
    <cmdb_ci/>
    <assigned_to/>
    <state />
    <short_description />
    <assignment_group/>
    <number />
    <priority />
  </problem_task>
</problem_tasks>
<provider />
<reassignment_count />
<rejection_goto/>
<related_incidents />
<rfc/>
<short_description />
<sla_due/>
<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_around/>
<work_end/>
<work_notes/>
<work_notes_list/>
<work_start/>
</common_problem>
```

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

[Common Problem Sample - Inbound.xml.zip](#)

[Common Problem Sample - Outbound.xml.zip](#)

[Go to top of page](#)