

How to configure incident management to align with CSDM; a leading practice guide

ServiceNow has made several changes to how incidents integrate with CSDM. This lays the foundations for Service Portfolio Management, Digital Portfolio Management, and ability to tie Events to Business Services. Ensuring first line agents capture and record incidents so that they align with CSDM will reduce data entry time and maximize the capabilities of ServiceNow platform.

Summary of Recommendations:

We recommend 5 configuration and operational changes to maximize the potential of CSDM in the incident process

1. Get started with CSDM by creating *Service offerings*, linked to *Services*
2. Consider three use cases for incident creation: agents, IT staff, system-generated
3. Configure your incident form so that *Services* are automatically derived from service offerings
4. Populate the *Service Offering* first when creating new incidents
5. Auto-populate the *Service*. Examples are included below.

Following those recommended changes, the form should look similar to below. Representative data is shown in the form to show how this might also be populated in practice: -

| | | | |
|---------------------|-------------------------------------------------|------------------|------------------------------|
| Number | INC0010032 | Channel | Phone |
| * Caller | Abel Tuter | State | New |
| Category | Operational | Impact | 2 - Medium |
| Subcategory | Database Related | Urgency | 1 - High |
| Service | Application Mgmt Services | Priority | 2 - High |
| Service offering | Dynamics Administration | Assignment group | Microsoft App Support Admins |
| Configuration item | Microsoft Dynamics Prod | Assigned to | |
| * Short description | Database is missing key attributes for Accounts | | |

1. Get started with CSDM by creating service offerings, linked to Services

Mapping services is key to taking advantage of the recommended changes and can feel like a big task, but it can be broken down into key steps that reflect your organizational needs and maturity. There is no one single model that suits all organizations, think of CSDM as a compliance framework for the model that suits your organization.

1. Set up some service offerings, using our [CSDM workshop](#) and [Service Portfolio Management Workshop](#) as guidance
 - this can be application or service focused dependent on what fits your organization best
 - ensure you stay in alignment with CSDM
2. Where you have Discovery, it is recommended to populate and have these classes as Principal CI's
 - Application Service, Server, Computer, Network Gear, Data Center, Database, PDU, UPS
 - further detail see [CMDB process guide](#)
3. For further guidance see:
 - [What is a Service and Service Offering](#)
 - [Define and map out your business services](#)
 - [CSDM workshop](#)
 - [CMDB workshop](#)
 - [CSDM Data model examples](#)

Simple example from the CSDM workshop:-

The screenshot shows an incident record in ServiceNow. The incident title is "Database is missing key attributes for Accounts". The incident details include: Number (INC0010021), Employee Name (Adam Ringle), Category (Operational), Subcategory (Database related), Service (Application Mgmt Services), Service offering (Dynamics Administration), and Configuration item (Microsoft Dynamics Prod). The incident is assigned to the "Microsoft App Support Admins" group.

Callouts explain the relationships:

- The Parent Service of the Service Offering:** Points to "Application Mgmt Services".
- The Offering to "administrate" the Application Service:** Points to "Dynamics Administration".
- Often the Application Service is the CI:** Points to "Microsoft Dynamics Prod".
- Populated based on the selection of the CI or the Service Offering. Foundational Data that was referenced on the Service Offering or CI:** Points to the "Assignment group" field, which is "Microsoft App Support Admins".

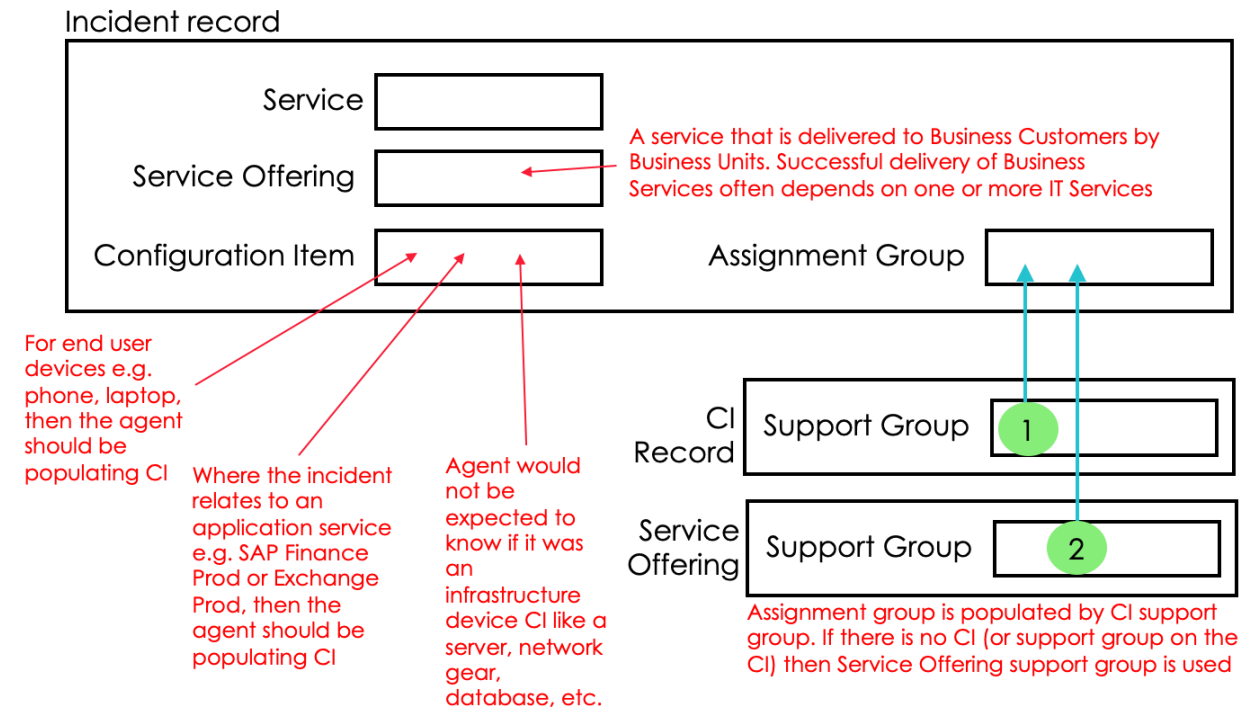
2. Consider three use cases for incident creation: agents, IT staff, system-generated

There are 3 main perspectives to consider when raising an incident. The use case will have a significant effect on the approach taken to populate the fields and whether that is manual entry or driven by the system.

| User Perspective | IT Staff | System Perspective |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| End users do not have a concept of business services or business service offerings. They will often refer to an application that has been offered to them e.g., Outlook, Workday or device e.g., Printer | Users in the support groups (e.g. network, middleware, database) raising incidents either for systems they own or systems affecting those | System generated incidents <ul style="list-style-type: none"> • alert and event management systems • outages • Caused by change |

Here we will focus on then **user perspective** as this is likely to be an interaction with first line agents to capture and record incidents, though the others should be considered (see [CSDM Workshop](#) for further detail).

Out of the box it is possible to populate 3 fields (Service, Service Offering and Configuration Item) with data intended for one of the other fields. E.g. Service entered into CI field. This leads to inconsistent data entry impacting quality of service.



3. Configure your incident form so that Services are automatically derived from Service Offerings

These simple changes to the incident form will reduce data inaccuracies and inconsistency, making incident reporting more reliable:

- a) Make the Service field read-only
- b) Configure the form to auto-populate the Service once the Service Offering is populated.
- c) Add filtering on CI classes to display those that relate to a typical IT environment

The screenshot shows three input fields in a form. The first field is labeled 'Service' and contains the text 'SAP Materials Management'. The second field is labeled 'Service offering' and contains 'SAP Materials Management Express'. The third field is labeled 'Configuration item' and contains 'SAP LoadBal02'. Each field has a search icon to its right. To the right of the 'Service' field is a dropdown icon. To the right of the 'Service offering' field is an information icon. To the right of the 'Configuration item' field are both a dropdown icon and an information icon.

The configuration changes required to achieve this are enumerated here:

| Field | Type | Change | Outcome |
|-------------------------------------|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------|
| Service (business_service) | Client Script | 1. New client script to update service based on service offering 2. Disable 'Clear Offering on changing Service' client script | Removes the need for the agent to enter value into the field |
| Service (business_service) | UI Policy | New UI policy to make business_service field read only | Forces Service Offering to be filled in as primary field |
| Service offering (service_offering) | Dictionary Override | Deselect 'Override reference qualifier' on the incident management table for service_offering | Allows service offering to be selected independent of service |
| Service offering (service_offering) | Client Script | Copy "Clear Offering on changing Service" to new script "Clear service on changing offering" | Ensures service and service offering are consistent |
| Configuration item (cmdb_ci) | Reference Specification | Reference Specification should be set to simple reference qualifier with conditions Class is Application Service, Server, Computer, Network Gear, Data Center, Database, PDU, UPS *if Principal CI's are configured this can be skipped | These classes represent the core required to support incident management |

Configuration item
(cmdb_ci)

Dependent Field

Remove company

Allows CI's to be selected when no company is set on the CI record

Following those recommended changes the form should look similar to below. Representative data is shown in the form to show how this might also be populated in practice :-

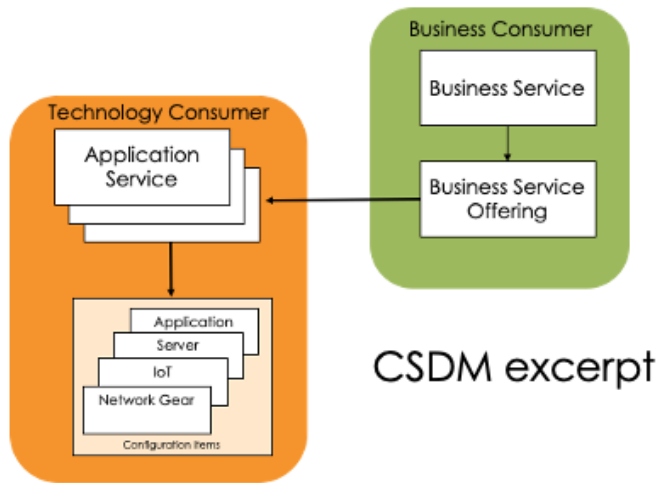
| | | | |
|---------------------|-------------------------------------------------|------------------|------------------------------|
| Number | INC0010032 | Channel | Phone |
| * Caller | Abel Tuter | State | New |
| Category | Operational | Impact | 2 - Medium |
| Subcategory | Database Related | Urgency | 1 - High |
| Service | Application Mgmt Services | Priority | 2 - High |
| Service offering | Dynamics Administration | Assignment group | Microsoft App Support Admins |
| Configuration item | Microsoft Dynamics Prod | Assigned to | |
| * Short description | Database is missing key attributes for Accounts | | |

* form layout may differ slightly for workspace

The fields on the Incident form directly related to CMDB/CSDM are :

- Service
- Service Offering
- Configuration Item

These fields relate to each other as described in the [CSDM Whitepaper](#)



CSDM excerpt

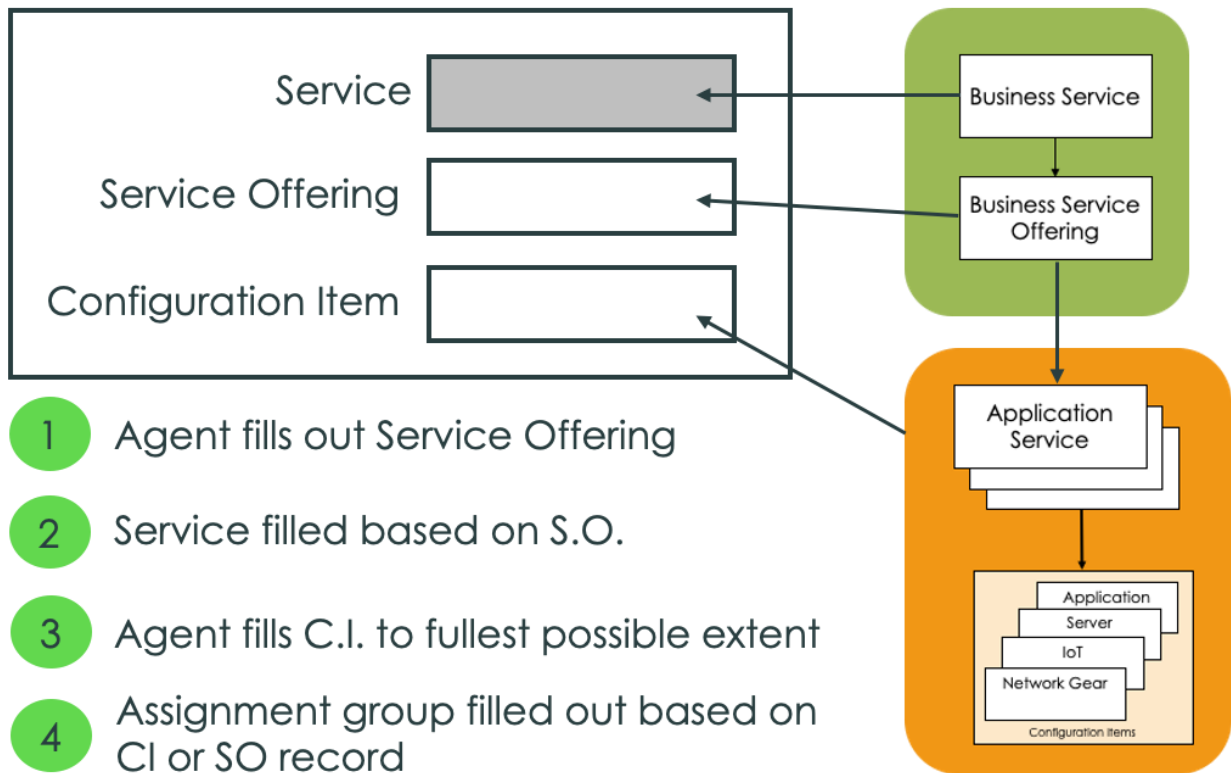
4. Populate the *Service Offering* first when creating incidents

The hierarchy described in CSDM needs to be reflected and asserted within the incident form. The critical field on the record is the **service offering**. Taking a one up / one down perspective, populating this provides filtering for

- Business Service
- Application Service / CI

This reduces the potential for level of error in populating these fields and time required. The diagram below shows how the modified form and CSDM relate to each other and the required actions to populate.

Incident record



5. Auto-populate the Service

There are a range of scenarios that apply to when an incident is created, Three of the most common to incident creation for end users are illustrated below:-

- End user device issue
- Cloud Native (SaaS) Application
- Infrastructure / Cloud Hosted App

Scenario 1

End user device issue

Service Offering is entered by agent

Service auto-populated from Service Offering and read-only

CI selection from those associated to service offering

The agent should fill in the service offering first for consistency and then fill in the CI.

Changing the Service Offering should update the Service

Incident record

| | |
|--------------------|-------------------|
| Service | End User Services |
| Service Offering | Adobe Acrobat |
| Configuration Item | ThinkStation C20 |

Scenario 2

Cloud Native (SaaS) Application

Service Offering is entered by agent

Service auto-populated from Service Offering and read-only

CI not populated for cloud native environments

With cloud native environments the value of CI is limited value as the service offering represents the key artifact

Incident record

| | |
|--------------------|-------------------|
| Service | End User Services |
| Service Offering | Salesforce |
| Configuration Item | |

Scenario 3

Infrastructure / Cloud Hosted App

Service Offering is entered by agent

Service auto-populated from Service Offering and read-only

CI populated post initial incident by technical team, which may impact Service Offering

Upon initial raising of the incident the Infrastructure CI is unlikely to be known unless monitoring systems have previously raised an event / outage.

Child incident can be raised for the Technical Service Offering

Incident record

| | |
|--------------------|-----------------------------|
| Service | Manufacturing Mgmt Services |
| Service Offering | Inventory Mgmt |
| Configuration Item | DB_Server_07 |

Addendum : Example Scripts

Clear service on changing offering

| | | | |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-------------------------------------|
| Name | <input type="text" value="Clear service on changing offering"/> | Application | <input type="text" value="Global"/> |
| Table | <input type="text" value="Incident [incident]"/> | Active | <input checked="" type="checkbox"/> |
| UI Type | <input type="text" value="All"/> | Inherited | <input type="checkbox"/> |
| Type | <input type="text" value="onChange"/> | Global | <input checked="" type="checkbox"/> |
| Field name | <input type="text" value="Service offering"/> | | |
| Description | <input type="text" value="Client Script to clear the Service field when the Service Offering field value is changed"/> | | |
| Messages | <input type="text"/> | | |
| Script | <pre> 1 function onChange(control, oldValue, newValue, isLoading, isTemplate) { 2 if (isLoading) 3 return; 4 5 g_form.clearValue('business_service'); 6 } </pre> | | |
| Isolate script | <input checked="" type="checkbox"/> | | |

Make "Clear Offering on changing Service" to Active = false

Clear Offering on changing Service false Incident [incident] Global

| | | | |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------|-------------------------------------|
| Name | <input type="text" value="update service based on service offering"/> | Application | <input type="text" value="Global"/> |
| Table | <input type="text" value="Incident [incident]"/> | Active | <input checked="" type="checkbox"/> |
| UI Type | <input type="text" value="Desktop"/> | Inherited | <input type="checkbox"/> |
| Type | <input type="text" value="onChange"/> | Global | <input checked="" type="checkbox"/> |
| Field name | <input type="text" value="Service offering"/> | | |
| Description | <input type="text"/> | | |
| Messages | <input type="text"/> | | |
| Script | <pre> 1 function onChange(control, oldValue, newValue, isLoading, isTemplate) { 2 if (isLoading newValue === '') { 3 return; 4 } 5 var service_offering = g_form.getReference('service_offering', setBusinessService); 6 7 } 8 9 function setBusinessService(set_service_offering) { //reference is passed into callback as first arguments 10 11 g_form.setValue('business_service', set_service_offering, parent); 12 13 } </pre> | | |

```
function onChange(control, oldValue, newValue, isLoading, isTemplate) {
  if (isLoading || newValue === "") {
    return;
  }
  var service_offering = g_form.getReference('service_offering', setBusinessService);
}

function setBusinessService(set_service_offering) { //reference is passed into callback as first arguments

  g_form.setValue('business_service',set_service_offering.parent);
}
```