Context-based requests for Identity Manager

Written by Stephan Hausmann and Abdullah Ahmad

ONE IDENTITY by Quest

Introduction

The configuration of Context-based requests in Identity Manager comes up often for SAP ABAP systems, especially in combination with business Role requests.

Within this whitepaper, we will delve into this configuration, including how to define Context(s), how to make it available as part of the request and how to translate from the request with Context into a provisioning task.

Master-derived SAP role concept

Why are Context-based requests typical for SAP ABAP systems?

The concept of the master-derived SAP role is generally used when SAP has been implemented across many sites (e.g. company code, plant, sales organization and more) and when the authorization at object level remains the same across all sites. The only difference is in organizational values, such as in a company that has two employees in Materials Management with the same job description but who work in organizational units (company code 0005 and 2700). Both employees might require the same general access for something like Goods Movement Information but should be restricted to their respective organizational unit – the Context.

The authorization values are maintained in the master SAP role, and the SAP roles for the various sites are derived from the master SAP role. The organizational level values for different sites such as company code, plant, sales organization, etc. are maintained in the derived SAP roles. This makes it easier to maintain the SAP roles, as any changes to the authorization level only need to be made in the master SAP role, and the various derived SAP roles can be generated or regenerated from the master SAP role in a single step.

Context-based request

While the Master-derived SAP role concept simplifies the administration of SAP roles, it doesn't simplify the requests from an end user point of view.

But before we start, we want to define a few terms related to Roles that we are going to use:

SAP role	A role from a SAP ABAP system – either composite or single SAP role.
Role	A collection of SAP roles. This would often be called Business Role, but we try to distinguish between the Concept of a Role and the Business Role implementation in Identity Manager.
System Role	A System Role from Identity Manager – implements a Role.
Business Role	A Business Role from Identity Manager – implements a Role.

Table 1: Definition of the different Roles.

If the SAP roles are bundled into Roles, the Context needs also to be a part of the Role. Therefore, you would have to request a Role with the correct Context. In the example from the previous section, it would be "Goods Movement Information for company code 0005" and "Goods Movement Information for company code 2700" – two Roles with the Context. We are going to call this kind of Roles "Role with Context".

The concept of the Context-based request aims to reduce the number of requestable Roles for the end user and to streamline the request process. of Roles "Context-based Roles."

An example of a Context-based request would be a Role, such as “Goods Movement Information” that can be selected as part of the request with the correct Context – company code 0005 or company code 2700 in our case. We are going to call this kind of Roles “Context-based Roles.”

There may be different views on how an end user starts a request. It may be that the Role is the starting point, or it may be that the Context Type or Context is where the journey starts. So ideally, the modelling needs to allow more than one option to start a request.

The next screenshots (Figures 1 - 5) show what a request may look like from an end user point of view when starting with the Context Type and followed by Context and Role and demonstrate how to build it yourself – including the provisioning task.

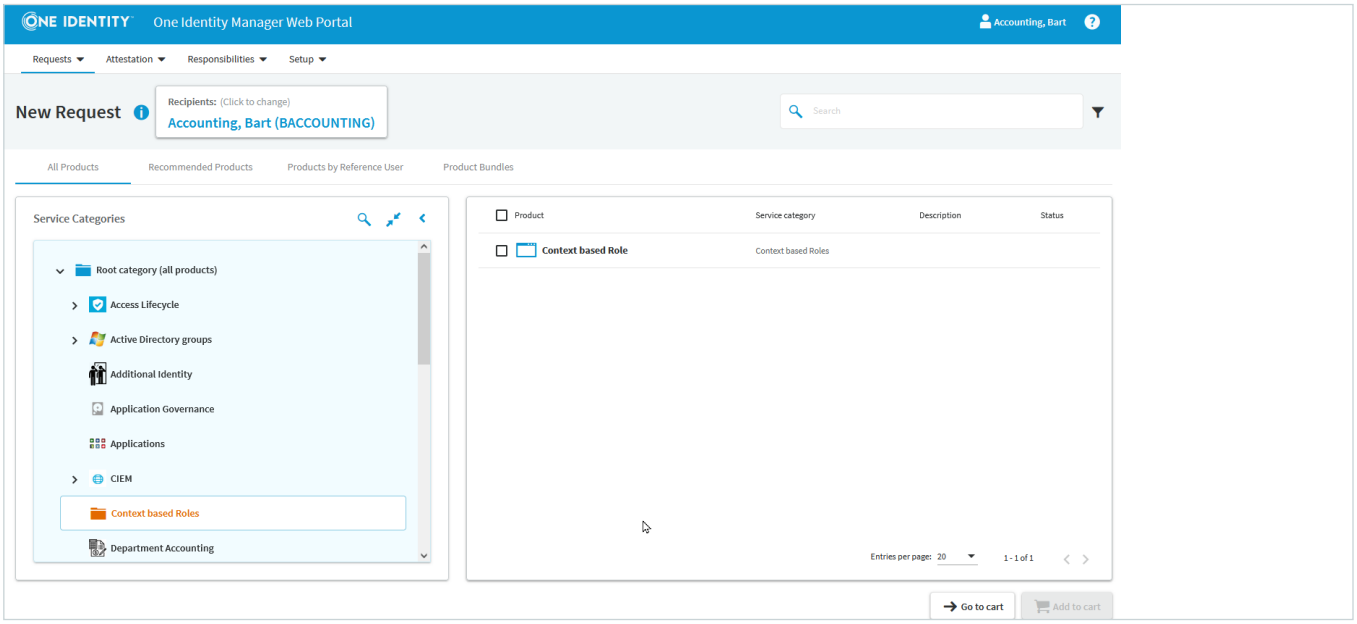

Figure 1. *: Select the Service Item for Context-based request*

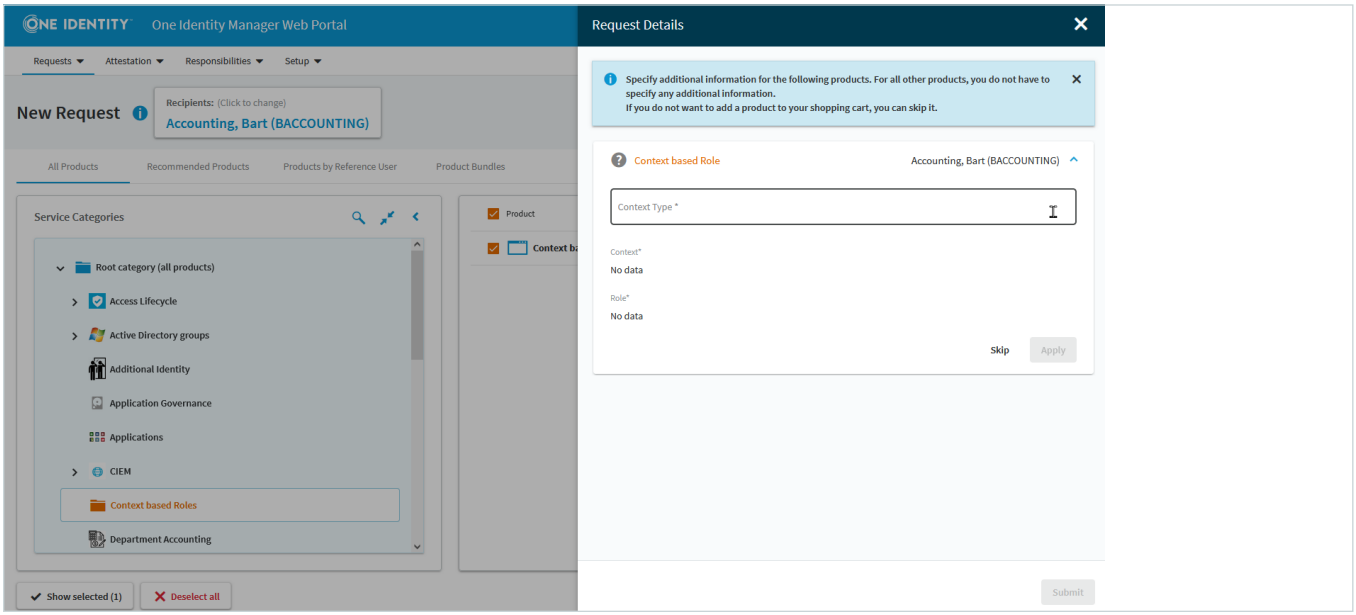

Figure 2. *: You must select the Context Type first.*

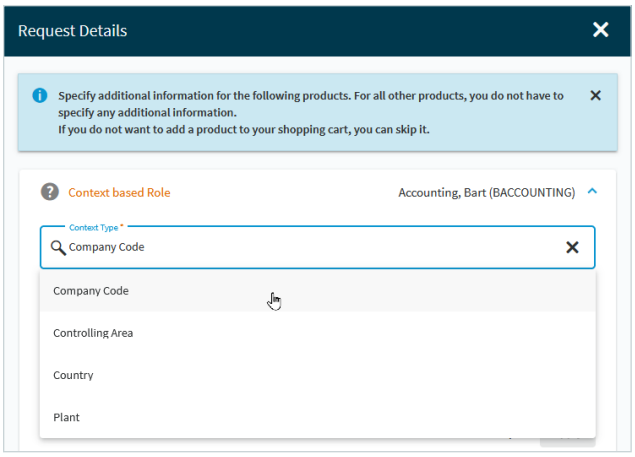

Figure 3. *: Select from the list of available Context Types.*

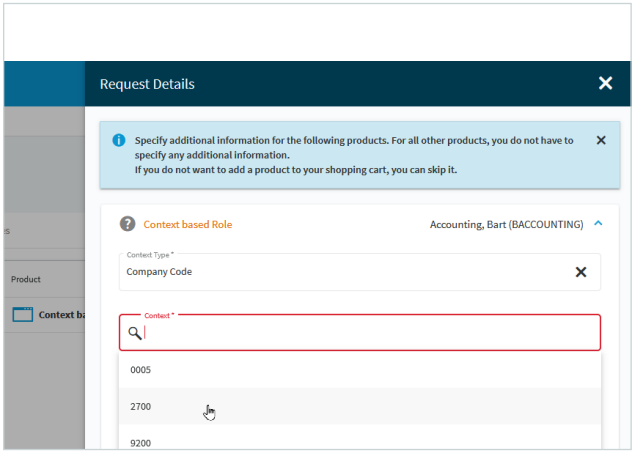

Figure 4. *: After selection of the Context Type, the Context is available for selection.*

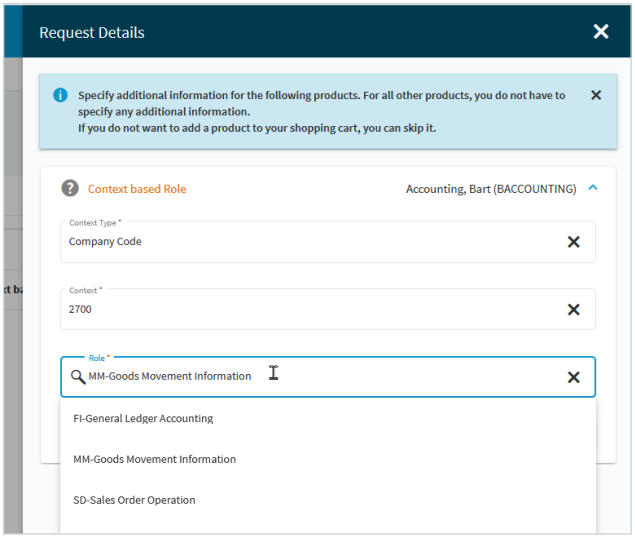

Figure 5. *: After selecting the Context, the Role can be selected.*

We will discuss a few available options here, and all examples are based on Identity Manager 9.2.

Data provided

Because we typically see this kind of requirement in connection with SAP ABAP systems, our examples will be based on SAP roles. The data you will get is usually of the format "Role – Context Type – Context – SAP role (Entitlement)."

It might look similar to the following example.

```
Role;ContextType;Context;Entitlement

SD-Sales Order Operation;Company
Code;2700;X:Y:SD:SALES _ ORD _ OPERAT _ :CH001

SD-Sales Order Operation;Company
Code;9200;X:Y:SD:SALES _ ORD _ OPERAT _ :HU001

SD-Sales Order Operation;Company
Code;0005;X:Y:SD:SALES _ ORD _ OPERAT _ :DE001

FI-General Ledger Accounting;Company
Code;2700;X:Y:FI:GL _ ACCOUNTING _ _ _ _ :CH001

FI-General Ledger Accounting;Company
Code;9200;X:Y:FI:GL _ ACCOUNTING _ _ _ _ :HU001

FI-General Ledger Accounting;Company
Code;0005;X:Y:FI:GL _ ACCOUNTING _ _ _ _ :DE001

MM-Goods Movement Information;Company
Code;2700;X:N:MM:IM _ GOODS _ MOV _ INFO:CH001

MM-Goods Movement Information;Company
Code;9200;X:N:MM:IM _ GOODS _ MOV _ INFO:HU001

MM-Goods Movement Information;Company
Code;0005;X:N:MM:IM _ GOODS _ MOV _ INFO:DE001
```

We see three Roles for the three different Contexts: company codes 0005, 9200 and 2700. We are using only one SAP role per Role for each Context on purpose to avoid showing too much data – in reality, there would be more than one SAP role for each Context of the Role.

Modelling the Context

SAP roles are already part of the Identity Manager model. After connecting and synchronizing a SAP ABAP system, the Context is the first thing we must consider.

Though there may be other options for modeling, like using attributes of objects, we are using Identity Manager Extended Properties. They are available for all relevant objects, can be assigned more than once and bring their own grouping mechanism for the Context Type.

Table 1 shows how the requirements map to the Identity Manager objects.

Requirement	Identity Manager object
Context Type	Property Group
Context	Extended Property

Table 2: Mapping Context Type and Context to Identity Manager Standard Objects.

An Extended Property can be assigned to a SAP role and to Roles and provides the Context Type Information via the Property Group. Figures 6 and 7 show how the Context is modelled in Identity Manager. Figure 8 shows how a Context can be assigned to a SAP role.

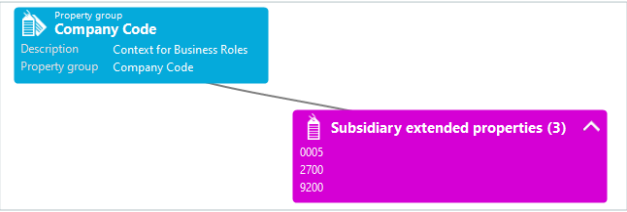

Figure 6. *: Property Group representing the Context Type (Company Code) and the assigned Extended Properties (0005, 2700, 9200).*

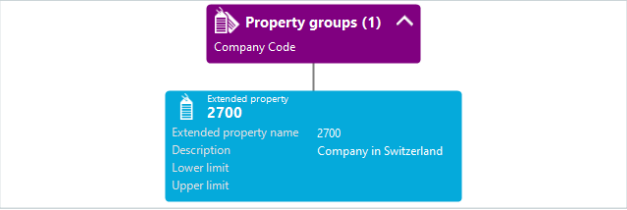

Figure 7. *: Extended Property representing the Context.*

Figure 8. *: SAP role with the assigned Context.*

Figures 9, 10 and 11 show the details of the objects. Most of the details are in the description, but we use one spare field to specify which objects are intended as Context for Context-based requests.

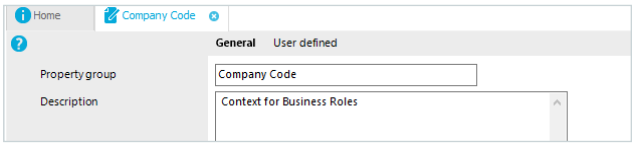

Figure 9. *: Details for a Property Group (1/2).*

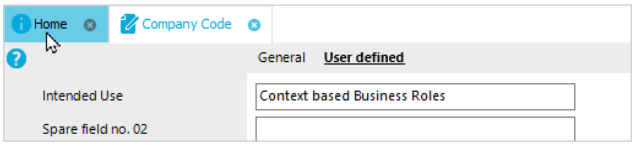

Figure 10. *: Details for a Property Group (2/2). We use a spare field (renamed to Intended Use) to be able to separate between Property Groups for Context-based requests and other usages.*

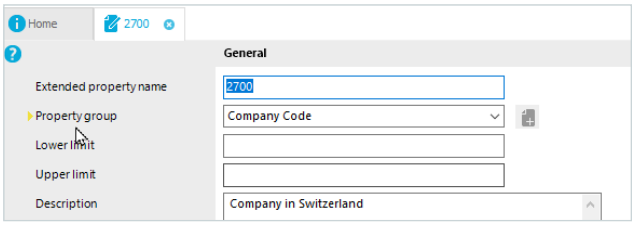

Figure 11. *: Details of an extended property.*

Modelling the Context for Roles

Where should you assign the Context? There are two options in Identity Manager.

Option A: Assigning the Context to SAP roles

In this scenario, you would assign the Context to the SAP roles (Figure 8). All SAP roles get assigned to the Context-based Role (Figure 12). When you request the Context-based Role and select the Context on the fly, a Role with Context with the derived SAP roles would be created (Figure 13) and assigned to the recipient.

From a maintenance point of view, modified data for the Context-based Roles would first require you to update the Context-based Role with all the assigned SAP roles and then to update the created Roles with Context accordingly.

Using this technique might be an advantage when not all combinations of Context-based Roles and Contexts are requested, as the number of created Roles with Context might remain low.

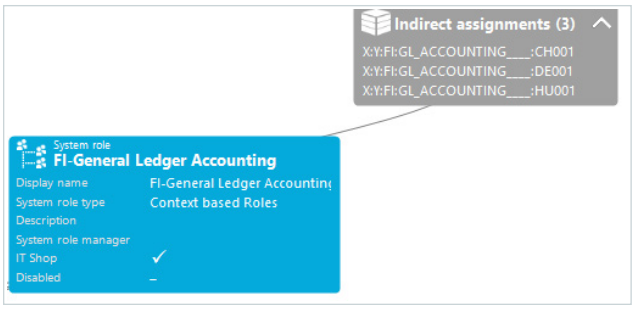

Figure 12. *: SAP roles assigned to the Context-based Role*

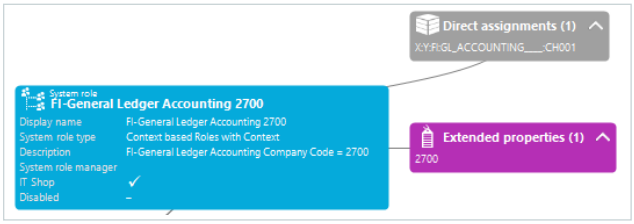

Figure 13. *: On the fly generated Role with Context*

Option B: Assigning the Context to the System Roles

In this scenario, for each Context, a Role with Context will be created upfront. The Context is assigned to the created System Roles and the SAP roles will be assigned to the created System Roles accordingly (Figure 14).

For requests, an “empty shell” and the ability to select the available Contexts will be needed. This means we are going to assign the possible System Roles with Context to the Roles serving as “empty shell” (Figure 15).

From the maintenance point of view, modified data will only require you to update the upfront created Roles. Initial loading and updates would typically be done using CSV imports using the Synchronization Editor or via API calls.

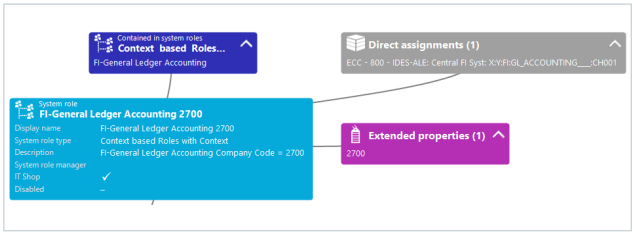

Figure 14. *: System Role with Context – and the reference to a shell.Role*

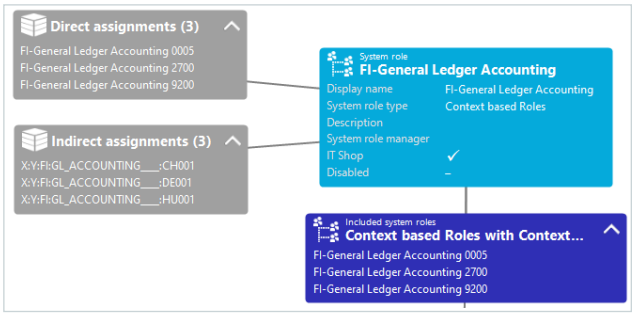

Figure 15. *: Context-based Role (serving as an “empty shell”) - with the assigned Roles with Context.*

There are probably more options, but we are going to use Option B for the “implementation.”

Which options exist to model the Roles?
The Context-based Role can be modelled in Identity Manager, as Business Roles or as System Roles.

Business Roles in Identity Manager

- have more options for management and ownership
- have more options for translations

System Roles in Identity Manager

- don't require an assignment resource

Each can have its own Role Class, which will be essential for modelling. Any other custom object might work too. We use System Roles for the "implementation."

What does the request look like?

From a user point of view, we have the following references that need to be considered for the request:

- **Role** – System Role
- **Context Type** – Property Group
- **Context** – Extended Property

The request can be started in different ways. You may start selecting the Role, then the Context Type - assuming there is more than one - and finally the Context. But starting with the Context Type and Context and selecting the Role at the end would work as well

Option A: One "Service Item" for everything with the parameters

- **Role**
- **Context Type**
- **Context**

The order of the parameters can be chosen based on the requirements – typically Role or Context Type first.

Option B.1: N "Service Items" according to the number of Roles with the parameters

- **Context Type**
- **Context**

Option B.2: M "Service Items" according to the number of Context Types with the parameters:

- **Context Type**
- **Context**

This option would allow you to search for description and name. Also, the IT Shop can be used to make some Roles only available to some requestors. For the implementation, we will use one Service Item.

Building the request in Identity Manager

From an Identity Manager point of view, we must configure the following objects and processes:

1. **Create** objects required for request and provisioning
2. **Define** request property with parameters
3. **Change** the initially requested Service Item

Create objects required for request and provisioning

As we are proceeding with the one Service Item option, you might think that we need to build only one Service Item upfront for the request. But since the initial Service Item is replaced after the request with a Service Item containing the System Role with Context, we must either create all Service items up front or build the Service Items as we go.

For our example, we are creating the Service Items upfront, but if you have a look at the ootb script "TSB_PersonWantsOrg_ HandleRequestWithParameters" in Identity Manager, you find an example how to create impromptu Context-based Service Items.

The screenshots show the Service Item to request initially (Figure 16), a Service Item for a System Role with context (Figure 17) and the details of that Service Item (Figure 18).

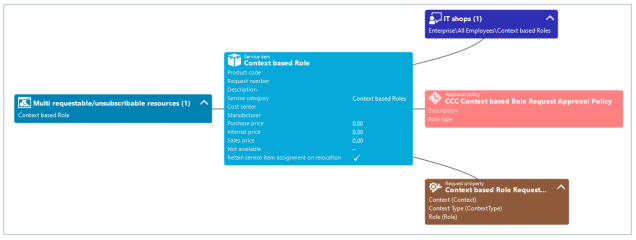

Figure 16. *: Service Item for the initial request - the approval workflow will take care to change the requested Service Item to the final Service Item.*

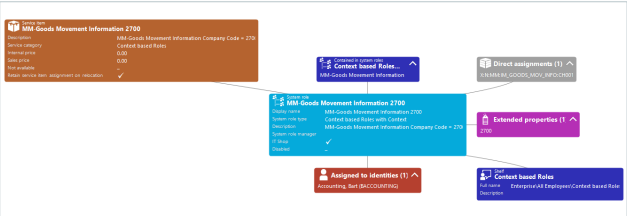

Figure 17. *: Service Item for a Role with Context.*

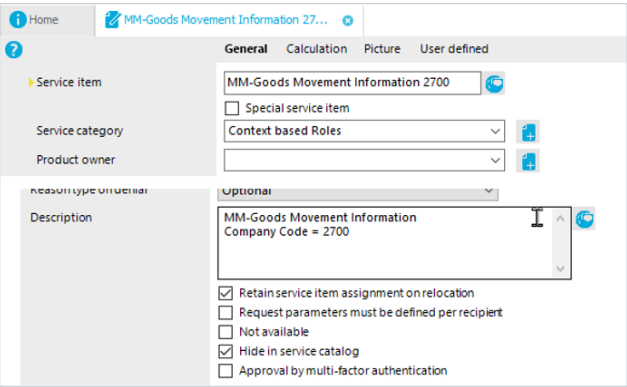

Figure 18. *: In addition to name and description, the flag "Hide in service catalog" is set for a System Role with Context.*

Once the final Service Item is assigned, it can be unsubscribed from or renewed like any other Service Item.

Define request property with parameters

We must create request properties and assign them to the Service Item used for the initial request. As we are using one Service Item with three parameters, the order of the parameters can still be chosen – either start with the Context Type or with the Role. We will start by selecting the Context Type and, depending on that parameter, you can next select the Context and finally the Role available for that Context. The details of the Request Property are shown in Figures 19 to 23.

You may have to add permissions for your user to view the existing ExtendedAttribute (= Extended Property) and ExtendedAttributeGroup (= Property Group) tables.

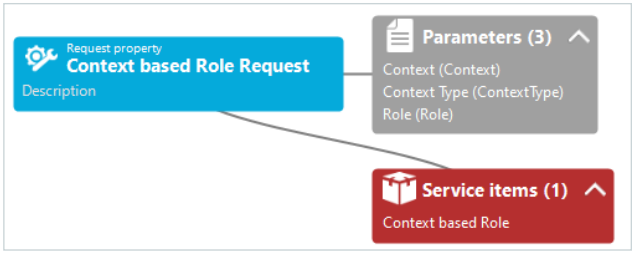

Figure 19. *: Overview of the Request Property.*

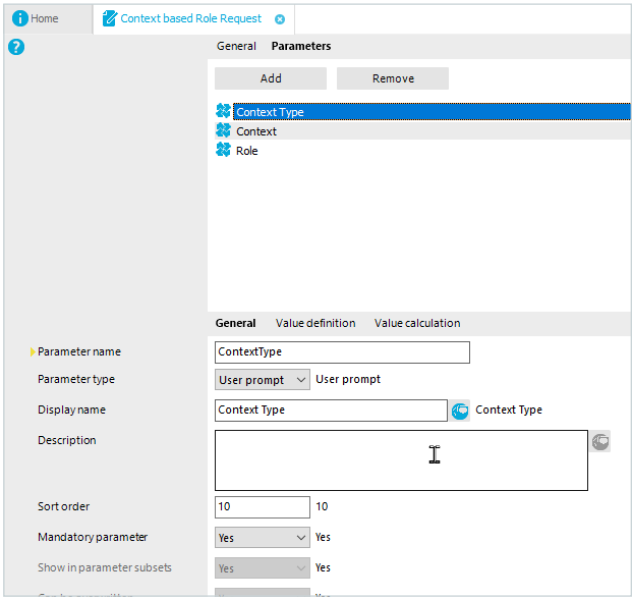

Figure 20. *: The parameters will be "User prompt".*

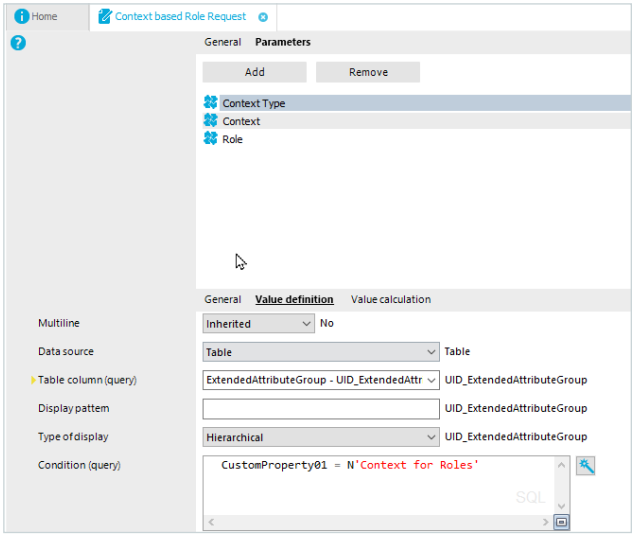

Figure 21. *: First parameter is Context Type, and we filter on CustomProperty01 the ExtendedAttributeGroups (= Property Groups) that are intended for the Context-based Roles.*

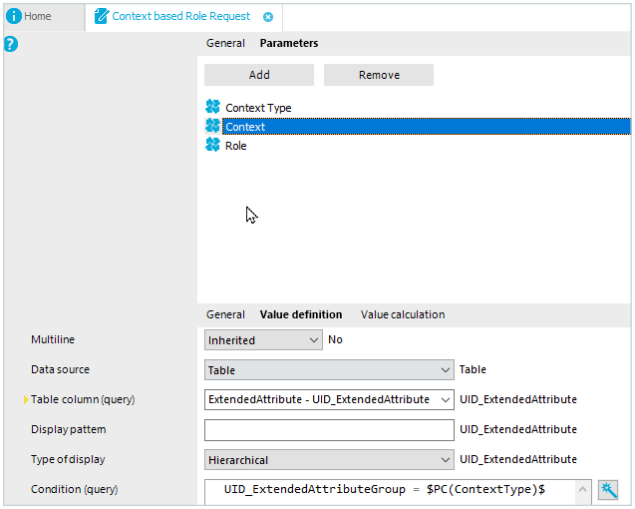

Figure 22. *: We filter the Context by the selected Context Type.*

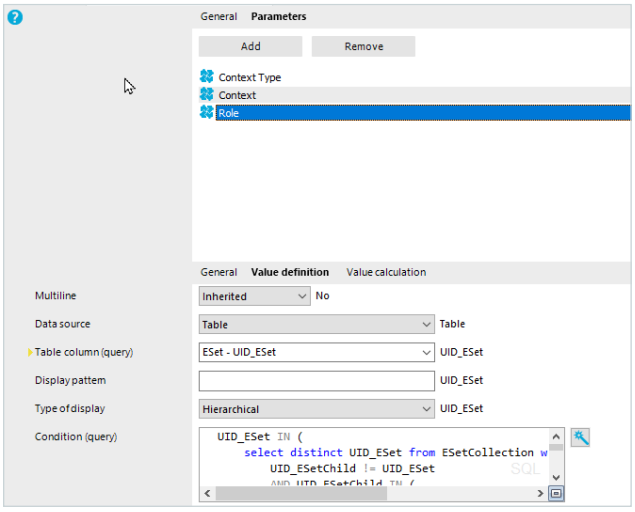

Figure 23. *: And finally, we filter the possible Context-based Roles by the selected Context. You find the SQL in Appendix A.1.Roles by the selected Context. You find the SQL in Appendix A.1.*

Change the initially requested Service Item

We must change the initially requested Service Item by a Service Item with the selected Context. For that, we use an "EX" step in the approval workflow (Figure 24, Figure 25) and start a process chain to change the Service Item.

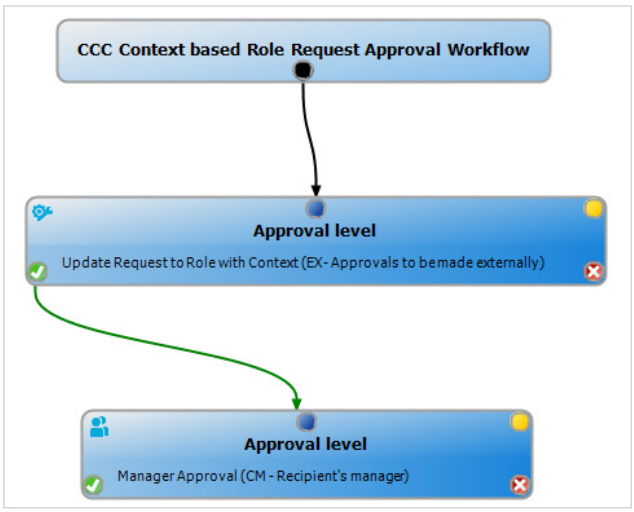

Figure 24. *: Approval Workflow with a step to update the Service Item.*

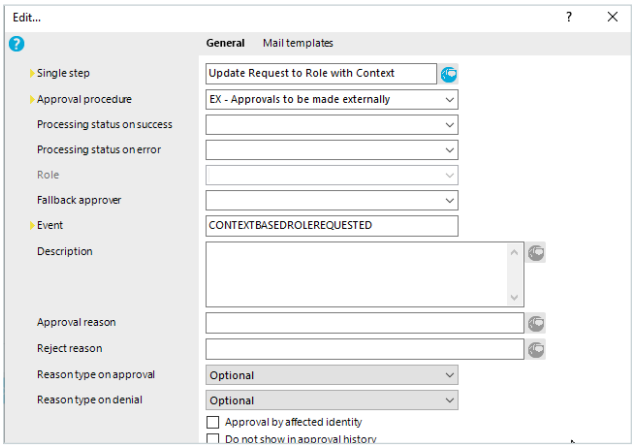

Figure 25. *: Details of the approval step that fires the event for a process chain.*

As the Approval Workflows fires an Event "CONTEXTBASEDROLEREQUESTED" on the PersonWantsOrg table, we need to define the process chain for that table and the defined event.

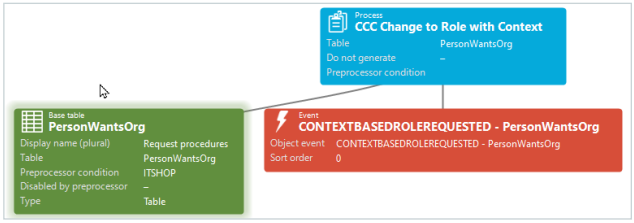

Figure 26. *: Overview of the process chain.*

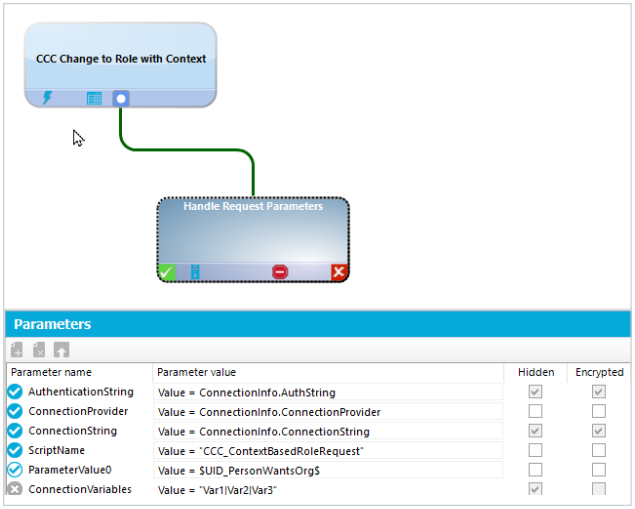

Figure 27. *: Details of the process chain, calling the script "CCC_ContectBasedRoleRequest". You find the script in Appendix A.2.*

Now we can have a look at the result. We have seen already in Figures 1-5 what the request looks like from an end user point of view. After the request is submitted, the following processes run in the background (Figures 28-29).

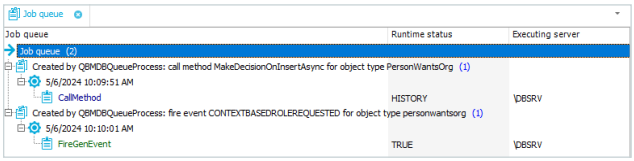

Figure 28. *: Approval process started and fires the defined Event.*

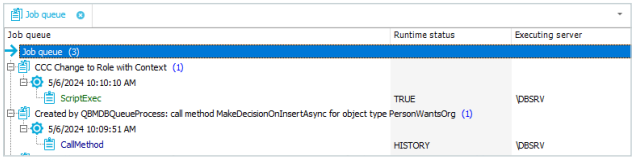

Figure 29. *: The process chain to exchange the Service Item is running.*

For the approval, the initially requested Service Item and the selected parameters are shown.

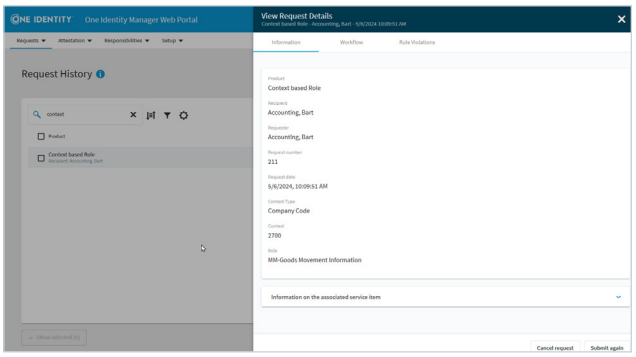

Figure 30. *: Manager approval.*

When the requester reviews the process, the Service Item has been exchanged with the Service Item with Context. Renewal and un-subscription are available for the replaced Service Item (Figure 31).

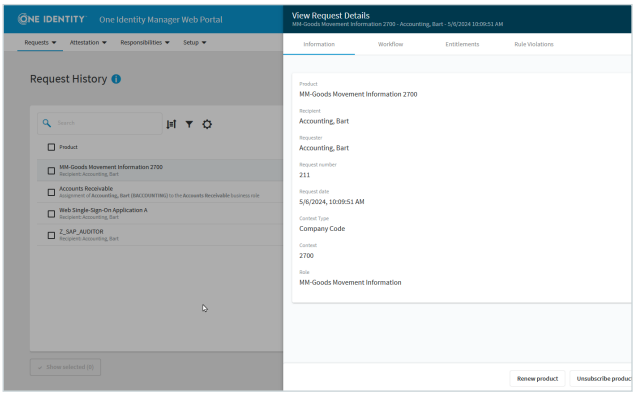

Figure 31. *: The initial Service item has been replaced with the Service item with Context, but the original request parameters are also shown.*

Enhancing the request

To provide additional information during the request, you can add information to "Display pattern(long)" for ExtendedAttribute and ExtendedAttributeGroup. We added the Description in our example. This will be displayed in addition for the drop-down lists (Figure 34).

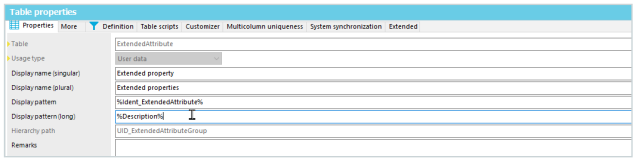

Figure 32. *: Display pattern (long) with added description for ExtendedAttribute (=Extended Property).*

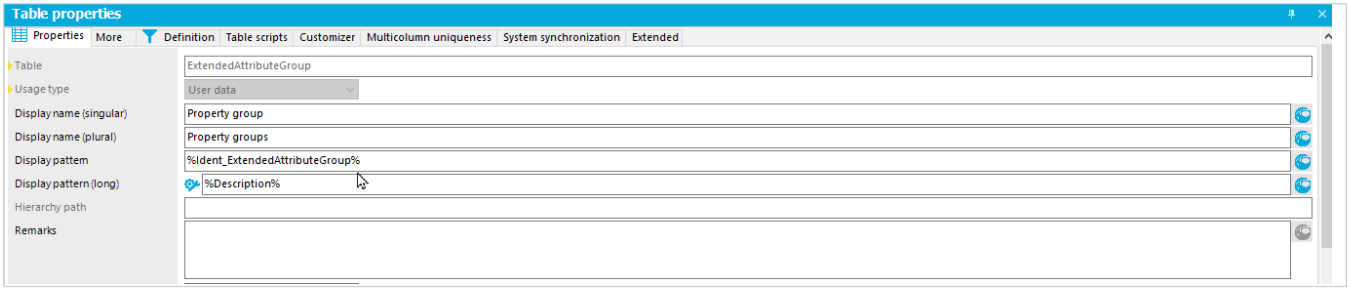

Figure 33. *: Display pattern (long) with added description for ExtendedAttributeGroup (=Property Group).*

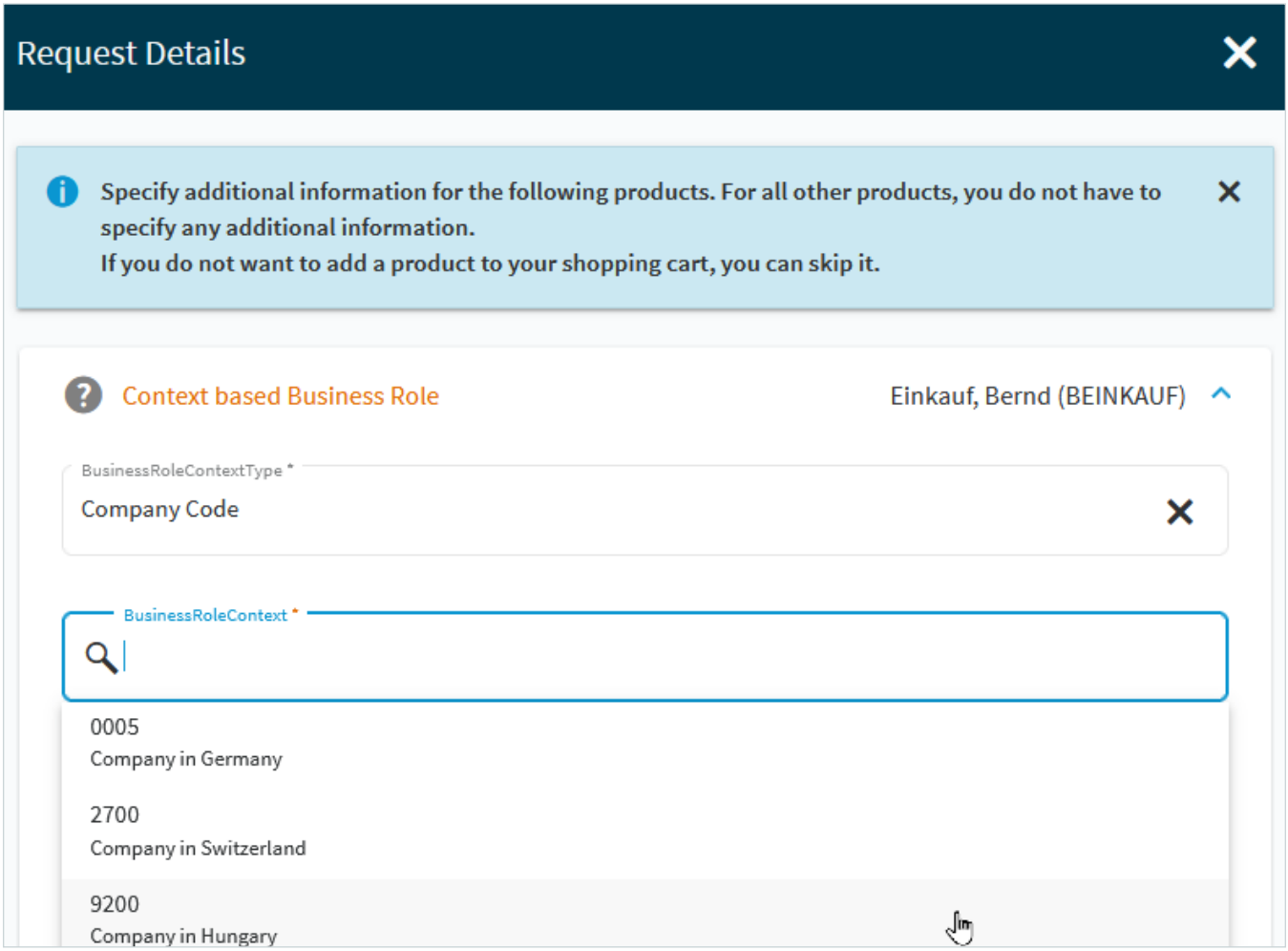

Figure 34. *: The description is shown in addition for the selection of the Extended Property.*

Summary

In this whitepaper, you have seen how Identity Manager can be configured to allow Context-based requests for SAP ABAP systems. Combining several techniques and features of Identity Manager makes it easy to implement business requirements on top of the existing platform.