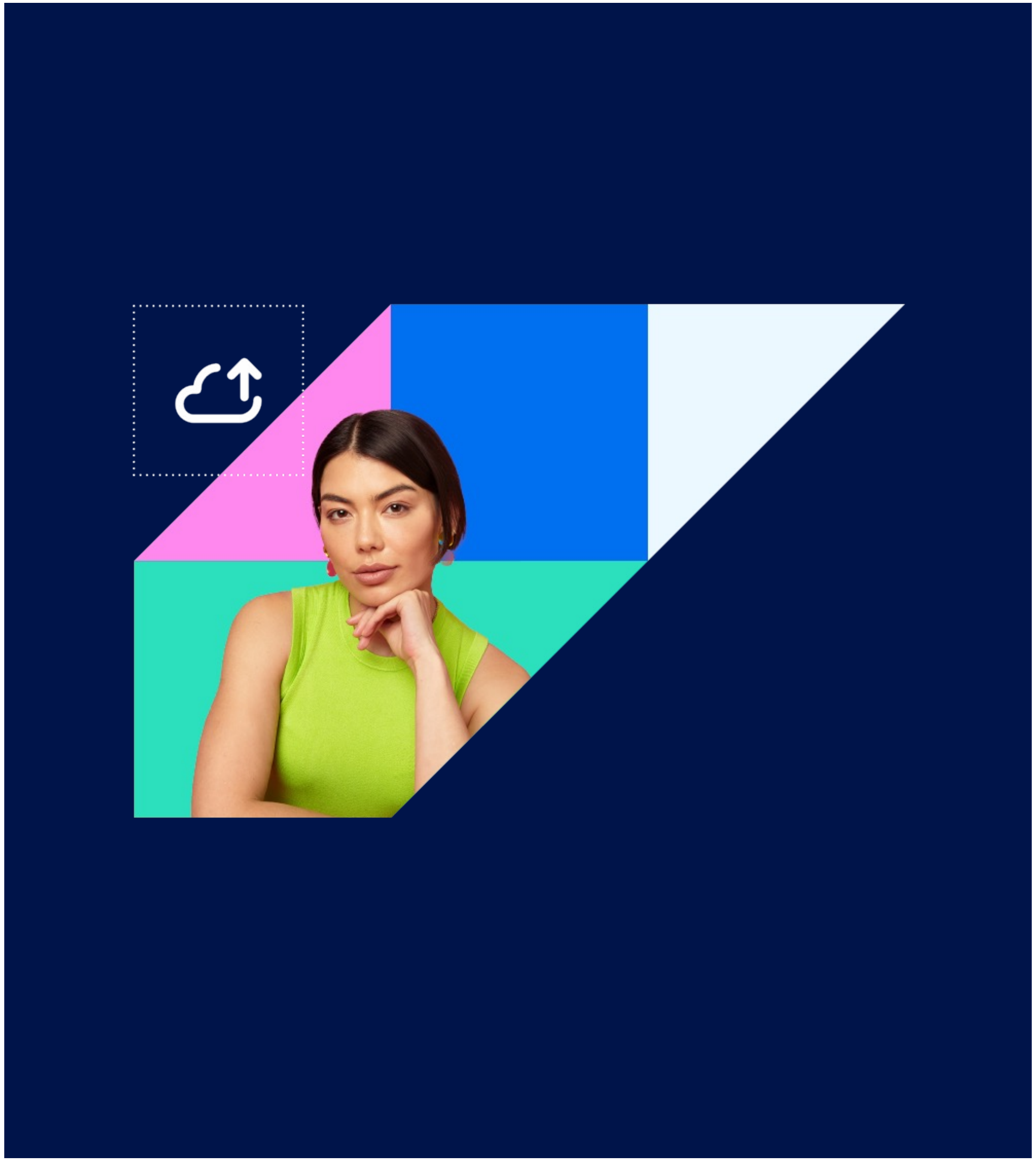
Implementing Extension Fields using Key user and Developer Extensibility – Do's and Don’ts

Focus on S/4HANA Cloud Private Edition and On-Premise

Karsten Schaser, Daniel Wachs

SAP BTP ABAP

PUBLIC



Presenters

Daniel Wachs

Architect
Key User and Developer Extensibility

Karsten Schaser

Development Expert
Key User Extensibility

Agenda

SAP S/4HANA Extensibility Overview

Key User vs. Developer

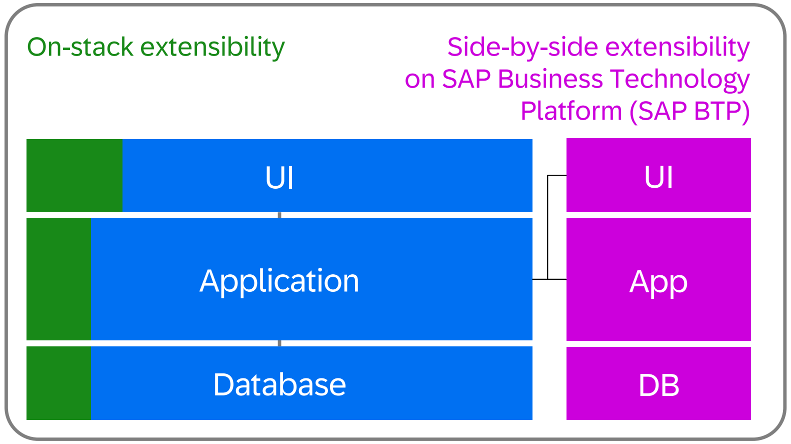
Finding the right extension points

Do’s and Don’ts

Further Information

SAP S/4HANA Extensibility Overview

Decoupled and lifecycle-stable extensibility in SAP S/4HANA


On-stack extensibility
Side-by-side extensibility on SAP Business Technology Platform (SAP BTP)
UI
Application
Database
UI
App
DB

Decoupled, lifecycle-stable extensions

Decoupling and lifecycle stability

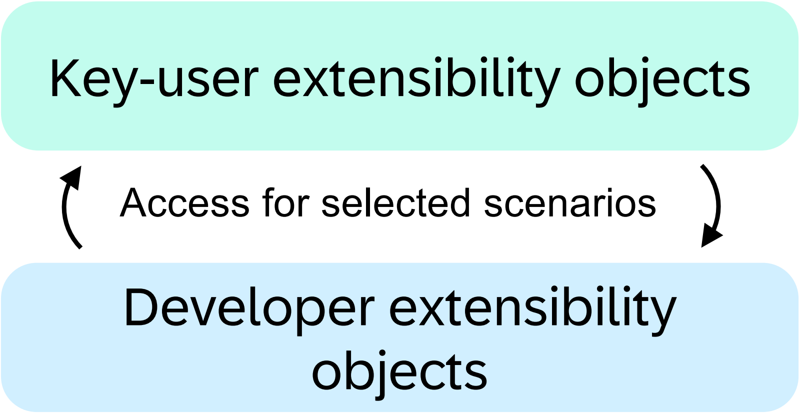
Challenge

- Software is always patched and upgraded based on a fixed schedule and new innovations must be introduced easily.

Solution

- Only extensibility that does not block SAP software updates and continues to work after an update without manual steps is allowed
- Custom extensions are lifecycle stable - that is, the software lifecycle of extensions is decoupled from the systems of records (clean core):
  - Modification free
  - Released APIs, events, and extension points used
  - Upgrade stable

Developer and key-user extensibility: Layering


Key-user extensibility objects
Access for selected scenarios
Developer extensibility objects

Key-user objects

- Are fully managed by key user extensibility tools
- Can be displayed and debugged, but cannot be edited with ABAP development tools
- Are separated by name range, software component, and ABAP package

Access between key user and developer extensibility objects is restricted
(see documentation).

Exception: Key User Custom Fields – full interoperability

Blog Post: Layering of Key User Extensibility and Developer Extensibility

Key User vs. Developer

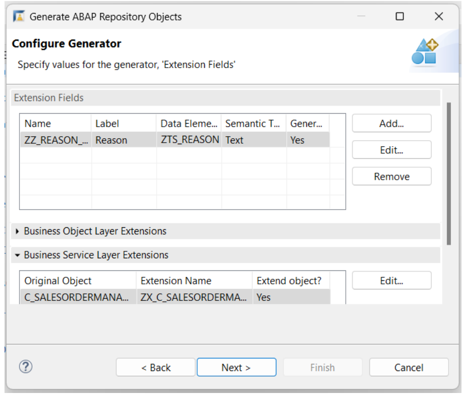
Developer and key-user extensibility: Extension fields as an example

Developer extensibility

Extension field
(ABAP development tools – editors and wizard)

Demo

Create extension field
with ABAP development tools wizard



Generate ABAP Repository Objects

Configure Generator

Specify values for the generator, 'Extension Fields'

Extension Fields

Name	Label	Data Eleme...	Semantic T...	Gener...
ZZ_REASON_...	Reason	ZTS_REASON	Text	Yes

Business Object Layer Extensions

Business Service Layer Extensions

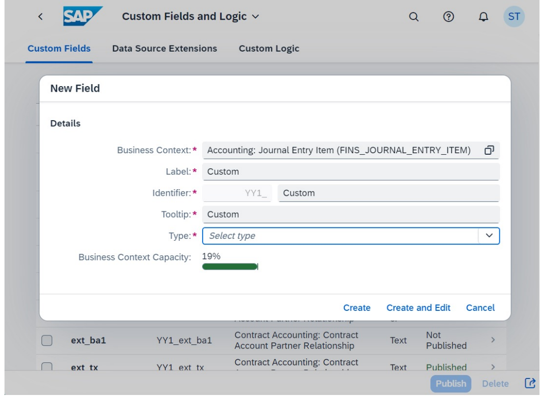
Original Object	Extension Name	Extend object?
C_SALESORDERMANA...	ZX_C_SALESORDERMA...	Yes

Key-user extensibility

Extension field
(Custom Fields SAP Fiori app)

Demo

Create extension field
with custom fields SAP Fiori app



Custom Fields and Logic

New Field

Details

Business Context:* Accounting: Journal Entry Item (FINS_JOURNAL_ENTRY_ITEM)

Label:* Custom

Identifier:* YY1_ Custom

Tooltip:* Custom

Type:* Select type

Business Context Capacity: 19%

□	ext_ba1	YY1_ext_ba1	Contract Accounting: Contract Account Partner Relationship	Text	Not Published
□	ext_tx	YY1_ext_tx	Contract Accounting: Contract	Text	Published

Key User vs. Developer Extensibility Custom Fields: When to choose what…

Recommendation is to go follow the following sequence and choose the subsequent option only if the previous option is not possible:

1. Use the Custom Fields app (key user extensibility tool)
   - Prerequisite: respective application is enabled for key user custom fields
   - Does also support "legacy" technology like SAP GUI, "old" SEGW implemented OData services
   - or SAP S/4HANA (on-premise/private cloud), legacy fields can be enabled so that they can be used with the Custom Fields app, see transaction SCFD_EUI
2. Use ABAP Development Tools (ADT) with ABAP language version ABAP for Cloud Development (developer extensibility)
   - Prerequisite: respective application is enabled for developer extensibility custom fields (only possible for RAP based applications)
3. Use ADT or SAPGUI tools with ABAP language version Standard (classic extensibility), e.g. if the corresponding SAP application is not enabled for the key user extensibility or developer extensibility
   - Custom fields can be added to CDS views (extend view) and for table/structure appends with developer tools
   - Adding custom fields to the transactional logic of an application depends on the SAP application. We recommend that you read the extensibility documentation that is provided for the application.
   - Adding custom fields to OData services that were built with technologies before RAP, depends on the implementation details of the OData service. See for example, the blog: How to redefine RDS based OData services? | SAP Blogs

See also Blog

(Custom Fields in S/4HANA: Key User versus Developer Extensibility)

Finding the right extension points

How to find out what to extend…business context, OData service, CDS View…

1. For User Interfaces, always start with the Fiori Apps Library (https://fioriappslibrary.hana.ondemand.com/sap/fix/externalViewer/)

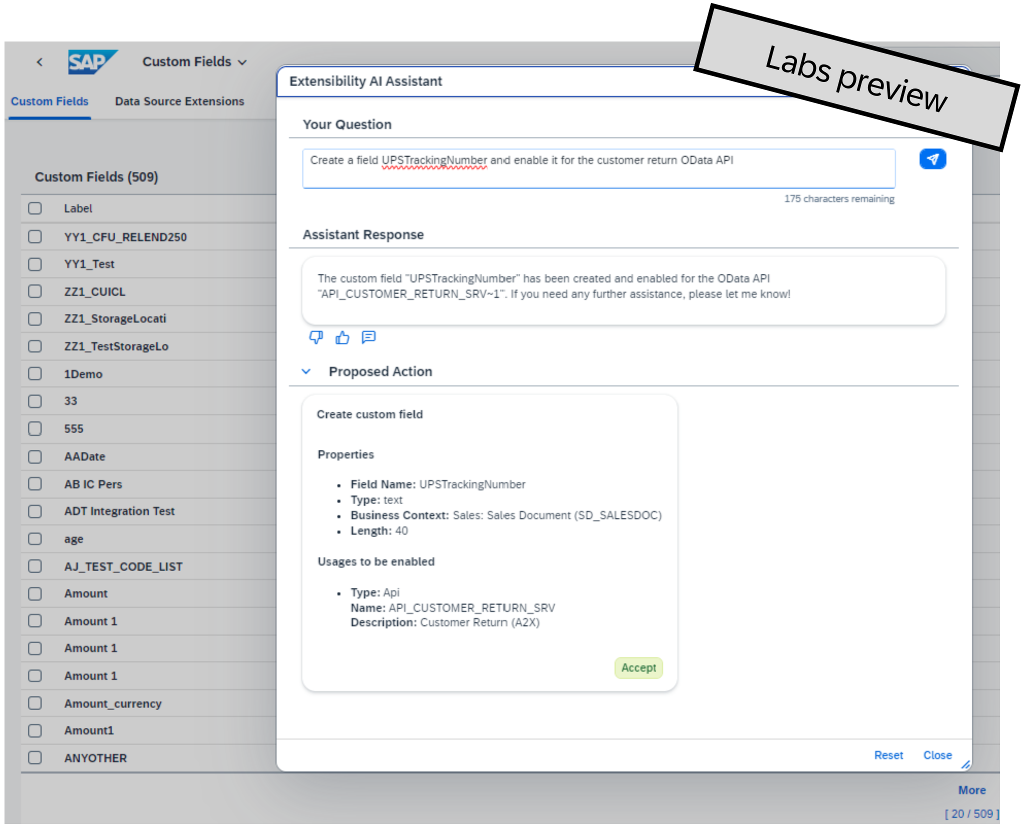
- There find the app the customer wants to extend – the application type attribute tells you whether it is a Fiori UI/GUI UI
- For Fiori UIs:
  - In the implementation information tab and there in the configuration section you can find the OData service
  - SEGW based OData Services (example Plan Customer Projects)
    - With the OData service information, you can have a look in transaction SCFD _REGISTRY to find out whether the OData service is extensible and which business contexts are exposed in which OData entities based on which CDS views
  - Service Binding based OData Services: (example Manage Sales Orders – Version 2)
    - With the OData service information (in OData V4 this is the service group name) in ADT look for a service binding with that exact same name
    - From the service binding navigate to the service definition that the service binding is exposing
    - Within the service definition all exposed CDS views are listed…with the entity under which they are exposed…with this you can identify your CDS view and you can now have a look in transaction SCFD _REGISTRY to find out whether the CDS view is extensible and which business contexts are exposed
- For GUI UIs (example VA01):
  - The Fiori ID is identical with the transaction code -> in transaction SCFD _REGISTRY you can have a look at the SAP GUI Context tab and search there for the respective SAP GUI transaction…
  - if key user custom fields are supported you will find a SAP GUI context entry referencing the SAP GUI transaction in SCFD_EUI
  - In some cases, an KDT document exists (with the same name as the SAP GUI context) which describes where exactly the field will be shown
  - The customer can “rearrange” the location of the field using SAP Screen Personas (blog…)

How to find out what to extend…business context, OData service, CDS View…

2. For APIs – both remote and local, start with the SAP Business Accelerator Hub (https://api.sap.com)

- Select the relevant product, e.g. SAP S/4HANA Cloud Private Edition
- For remote APIs
  - Select API Tab and then choose the relevant technology (Odata V4/V2, SOAP)
  - Search for the API you are interested in and open it
  - Find the Extensibility part in the Overview Section of the API
  - Extensibility information for Key User Field Extensibility is displayed (e.g. with the name of the relevant business contexts)
- For local APIs select the On Stack Extensibilty Tab
  - Here you can find CDS Views, RAP BO Interfaces as well as BAdIs
  - CDS Views:
    - Contains Extensibility Information for key user and developer extensibility
    - For details on key user extensibility open SAP GUI transaction SCFD_REGISTRY for the view
    - For details on developer extensibility, open the view in ABAP Development Tools for Eclipse
  - RAP BO Interfaces
    - For details on extensibility, open the API State information in ABAP Development Tools for Eclipse
  - BAdI
    - Information about availability for key user and developer tools is displayed

How to find out what to extend…
the Extensibility AI Assistant will make this a lot easier in the future…



□ Label
□ YY1_CFU_RELEND250
□ YY1_Test
□ ZZ1_CUICL
□ ZZ1_StorageLocati
□ ZZ1_TestStorageLo
□ 1Demo
□ 33
□ 555
□ AADate
□ AB IC Pers
□ ADT Integration Test
□ age
□ AJ_TEST_CODE_LIST
□ Amount
□ Amount 1
□ Amount 1
□ Amount 1
□ Amount_currency
□ Amount1
□ ANYOTHER

Extensibility AI Assistant

Your Question

Create a field UPSTrackingNumber and enable it for the customer return OData API

175 characters remaining

Assistant Response

The custom field "UPSTrackingNumber" has been created and enabled for the OData API "API_CUSTOMER_RETURN_SRV~1". If you need any further assistance, please let me know!

Proposed Action

Create custom field

Properties

- **Field Name:** UPSTrackingNumber
- **Type:** text
- **Business Context:** Sales: Sales Document (SD_SALESDOC)
- **Length:** 40

Usages to be enabled

- **Type:** Api
  Name: API_CUSTOMER_RETURN_SRV
  Description: Customer Return (A2X)

Motivation

Extending applications in SAP S/4HANA requires knowledge about the business processes, including the available objects and business configuration.

Solution

An extensibility AI assistant to help customers find the right extensible object

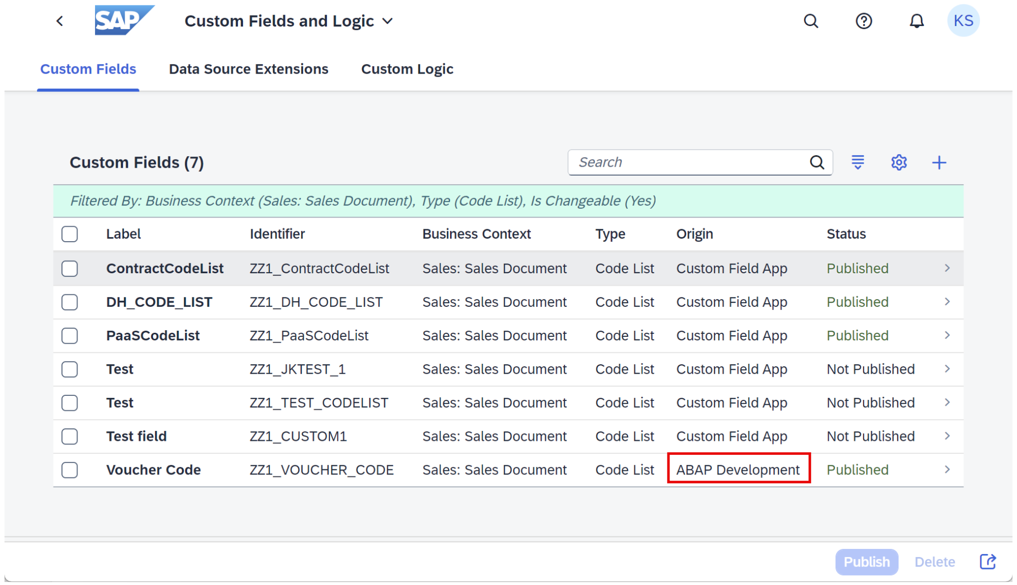
Devtoberfest Session

https://community.sap.com/t5/devtoberfest/what-software-developers-should-know-about-key-user-extensibility/ev-p/13806665

Do’s and Don’ts

ABAP Managed Fields (AMF) – SCFD_EUI
Demo

- **Transaction SCFD_EUI**
  - **Enable manually created fields for use in Fiori app *Custom Fields***
  - **Sometimes also called *ABAP Managed Fields* or in short *AMF***
  - **Enabled fields will be shown with *Origin* as *ABAP Development* in Fiori app *Custom Fields***



Custom Fields (7)

Filtered By: Business Context (Sales: Sales Document), Type (Code List), Is Changeable (Yes)

□	Label	Identifier	Business Context	Type	Origin	Status
□	ContractCodeList	ZZ1_ContractCodeList	Sales: Sales Document	Code List	Custom Field App	Published
□	DH_CODE_LIST	ZZ1_DH_CODE_LIST	Sales: Sales Document	Code List	Custom Field App	Published
□	PaaSCodeList	ZZ1_PaaSCodeList	Sales: Sales Document	Code List	Custom Field App	Published
□	Test	ZZ1_JKTEST_1	Sales: Sales Document	Code List	Custom Field App	Not Published
□	Test	ZZ1_TEST_CODELIST	Sales: Sales Document	Code List	Custom Field App	Not Published
□	Test field	ZZ1_CUSTOM1	Sales: Sales Document	Code List	Custom Field App	Not Published
□	Voucher Code	ZZ1_VOUCHER_CODE	Sales: Sales Document	Code List	ABAP Development	Published

ABAP Managed Fields (AMF) – SCFD_EUI

When to use

In general: do NOT use ABAP managed fields!

→ Their software logistics are more fragile then "normal" key user custom fields

The only exceptions are:

- Existing field: migration to S/4 HANA from a previous Suite on HANA system
  - Take over their extension fields into the key user app *Custom Fields* → prevent data loss
  - Re-use name of existing extension field → prevent adoption effort
- New field: special requirements regarding data element
  - Check table → enable validation also in generic SAPGUI transactions
  - Conversion exit

ABAP Managed Fields (AMF) – SCFD_EUI

Do's and don'ts

- **Do's**
  - Only use AMF, if really required
  - Add suffix also for append fields to be enabled with SCFD_EUI (see SAP note 2981484)
  - Follow Clean Core principle (linked soon in blog SAP S/4HANA Extensibility: All You Need to Know)

- **Don'ts**
  - Delete append fields via SE11 before disabling the Custom Field via SCFD_EUI
  - Modify/delete the value help view used in AMF field of type List
  - Delete the data element used in AMF field
  - Continue with publishing/deletion of further custom fields, if a previous publishing/deletion failed

ABAP Managed Fields (AMF) – SCFD_EUI

Further documentation/notes

- Documentation built-in in transaction SCFD_EUI
- SAP Help Portal: Enabling Custom Database Fields for Usage in SAP Fiori Applications
- General handling: SAP note 2981484
- Enable ABAP managed fields for database tables with proxy views: SAP note 2463257
- Re-creating a custom field: SAP note 3327309

Software logistics aspects – Private cloud/onPremise

Dos

- **Use apps for transport handling (also refer to note 2660797)**
  - **Register Extension for transport to**
    - **assign key user items to a package**
    - **assign key user items to a transport**
  - **Configure Software Packages to**
    - **Configure automatic transport/task handling for packages**
    - **Assign transports to packages**

- **Make sure that**
  - job user is provided in maintenance view ATO_PK_JOB_USER (SM30)
  - job user has sufficient authorization for publishing
  - job user is valid (exists, within validity range)
  - operation (save/publish/delete) is performed in the correct client

Software logistics aspects – Private cloud/onPremise

Don'ts

- Develop key user custom fields in several dev systems
  - Implemented checks are system specific → no data of other systems is considered
  - Generated artifact names depend on custom field name → clash of object names possible
- Edit key user custom fields in several systems
  - Custom field and its generated artifacts have a dedicated source system → i.e. risk system inconsistency in case of custom field deletion, if generated objects exist only locally

Software logistics aspects – Private cloud/onPremise

Report CFD_REGENERATE_IN_ONPREMISE

- Purpose
  - Show regeneration need of custom fields
  - Perform regeneration of custom fields to adapt application/technology changes
- Additional Checks (added via note 3487987)
  - Transport configuration (user not assigned to transport, transport does not fit to package e.g. because of unequal transport layer, ...)

- Special scenario: Customer wants to test an upgrade
  - Sometimes done in cloned dev system
  - Possible issues
    - TADIR entries of custom fields and their secondary objects have not been set to the current system (use Note 3484189)

Questions & Answers and general Dos & Don’ts

Customers may want to recreate a custom field with same name

- As long as the transport with the field creation has not been released one can delete and recreate the field (even when choosing an incompatible field type)
- As soon as the transport with the field creation has been released one can delete and recreate the field only as long as the field type chosen is compatible (see also note 3328541)
- As soon as the transport with the field creation has been released and field type needs to be changed incompatibly: one has to release the transport containing the field deletion and import it in subsequent systems → Prevent corrupt data

Customers want to utilize OData redefinition approach in order to add key user custom fields to OData Service Entities

- This does not work and is not supported (see constraint 34 in note 1574568):

  “34. Redefinition of SEGW projects with _FLX classes is not supported”

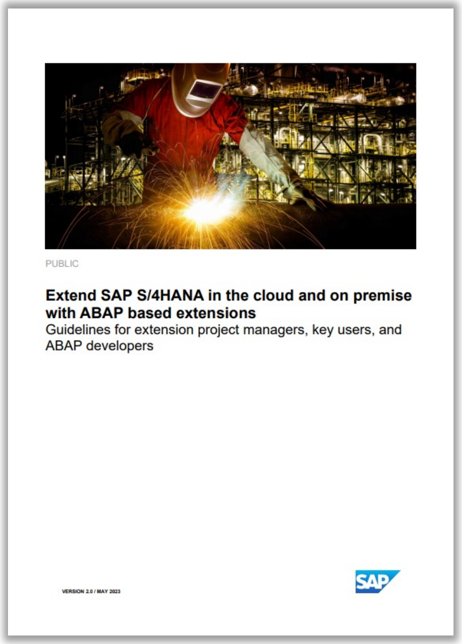
Further Information

Key User Custom Field Blog Posts

- Key User Custom Fields: When to use what type of code list
- SAP S/4HANA Key User Extensibility powered by Embedded Steampunk: Custom Field with ABAP implemented Value Help
- SAP S/4HANA Key User Extensibility: Custom field with context-dependent value helpEnrich SAP S/4HANA Key User Custom fields with Intent-Based Navigation
- Custom field type “Code list based on CDS view”: Finding/defining the right value help view and the right Semantic Object and Semantic Object Parameter for Intent-Based Navigation
- Custom Fields in S/4HANA: Key User versus Developer Extensibility
- SAP Screen Personas: How to position key user custom fields exactly where you want them to be

- Layering of Key User Extensibility and Developer Extensibility
- Custom Fields in S/4HANA: Key User versus Developer Extensibility

Important ABAP Cloud development model: related guides

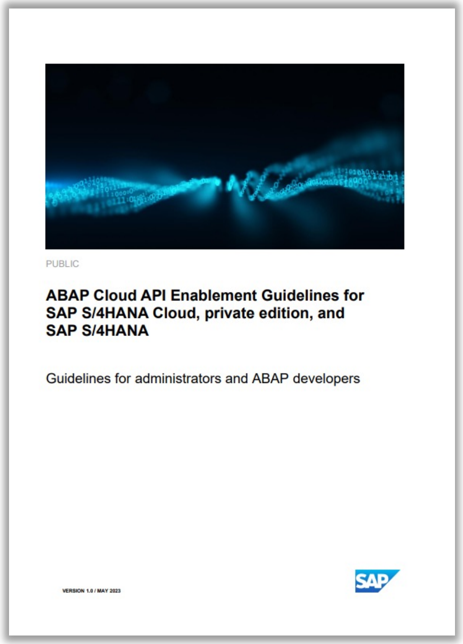


PUBLIC

Extend SAP S/4HANA in the cloud and on premise with ABAP based extensions
Guidelines for extension project managers, key users, and ABAP developers

VERSION 2.0 / MAY 2023

[Overview, comparison of extensibility options, introduction to ABAP Cloud, and three-tier extensibility model](#)



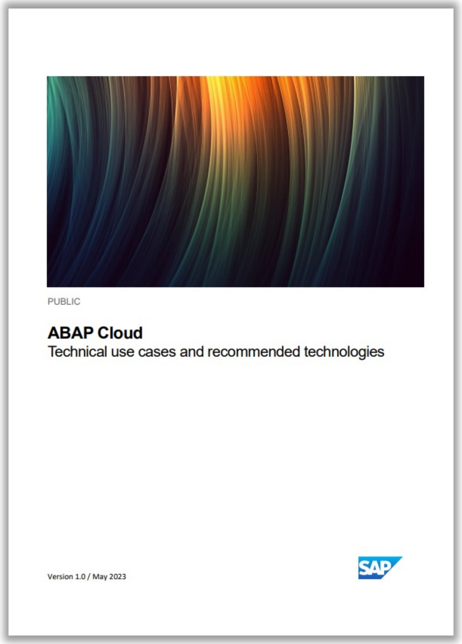
PUBLIC

ABAP Cloud API Enablement Guidelines for SAP S/4HANA Cloud, private edition, and SAP S/4HANA

Guidelines for administrators and ABAP developers

VERSION 1.0 / MAY 2023

[How to leverage tier 2 of the three-tier extensibility model](#)



PUBLIC

ABAP Cloud
Technical use cases and recommended technologies

Version 1.0 / May 2023

[How the old world of tier-3 and classic ABAP maps to the new, tier-1 and ABAP Cloud development model world](#)

Further information and blog posts linked at https://community.sap.com/topics/s4hana-cloud-abap-environment and https://community.sap.com/topics/abap.

Extensibility of SAP S/4HANA – Helpful links

Blog posts

- Key User Extensibility Tools of SAP S/4HANA last update 2021
- SAP S/4HANA Extensibility Use Case Overview (2016)
- SAP S/4HANA Extensibility: All You Need to Know last update 2024
- Building Low Code Extensions with Key User Extensi... - SAP Community 2024

SAP S/4HANA Cloud Public Edition – Documentation

- SAP S/4HANA Cloud: Extend and integrate your SAP S/4HANA Cloud solutions-> Extensibility
- SAP Fiori – Extensibility documentation: Personalizing and adapting apps
- Inside the user assistance documents, see the videos with a tour on the available extensibility tools:
  - Creating custom fields
  - Creating customer applications

SAP S/4HANA and SAP S/4HANA Cloud Private Edition – Feature description

- https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE > Product Assistance (English) > SAP S/4HANA Enterprise Technology > ABAP Platform > Developing on the ABAP Platform > Development Concepts and Tools > Extensibility
- SAP Fiori – Extensibility documentation: Personalizing and Adapting Apps
- In the SAP Fiori app library (SAP Fiori apps library) you can find the information on the extensibility for SAP Fiori apps. Open the **App Details** and go to **Implementation Information, Extensibility**.

/thank you

Contact information:

daniel.wachs@sap.com
karsten.schaser@sap.com