



# Guidance for Extensibility in SAP S/4 HANA Cloud, single tenant edition

PUBLIC



## CONTENTS

1.	BACKGROUND –SAP S/4 HANA EXTENSIBILITY IN GENERAL.....	3
2.	SCOPE .....	3
3.	OVERVIEW.....	3
4.	CLASSIFICATION.....	3
5.	ENHANCEMENTS .....	5
5.1.	Business Logic Enhancement.....	5
5.2.	Data Model Enhancement .....	6
5.3.	User Interface Enhancement .....	8
6.	CUSTOM DEVELOPMENT .....	9
6.1.	Apps .....	9
6.2.	Workflow.....	9
6.3.	Output .....	10
6.4.	Programs .....	12
7.	CONFIGURATION .....	13
8.	RESTRICTED ABAP.....	14

## 1. BACKGROUND – SAPS/4 HANA EXTENSIBILITY IN GENERAL

SAP Whitepaper '[SAP S/4 HANA Extensibility for Customers and Partners](#)' explains that extensibility in SAP S/4 HANA may be categorised in these ways:

- **Side-by-side extensibility**, using SAP Cloud Platform (SCP)
- In-app extensibility – built in the same system (or software stack) as the enhanced application. This category is further subdivided into:
  - **Key user extensibility** – using apps, which are described in more detail in the [In-App Extensibility Overview](#).
  - **Classic extensibility** – using development tools and techniques (e.g. transaction SE80, Eclipse, BAdIs) which were also previously available in SAP Business Suite.

The whitepaper above discusses the relative merits of these three approaches in detail, but broadly speaking side-by-side and key user extensibility are preferred. Classic extensibility is not possible at all in S/4 HANA Cloud (multi-tenant).

## 2. SCOPE

This document specifically addresses extensibility in the context of SAP S/4 HANA Cloud, single tenant edition. With single-tenant edition, classic extensibility *is* still possible, but with some restrictions compared with SAP S/4 HANA. This document assumes the reader is already familiar with classic extensibility techniques and terminology.

## 3. OVERVIEW

ERP Cloud solutions like SAP S/4HANA Cloud and SAP S/4HANA Cloud, single tenant edition have the advantage of standard processes with multiple updates per year. Customers can innovate continuously in incremental steps. In order to facilitate smooth, frequent updates (applied by SAP) in single-tenant edition:

- In general, Modifications are not allowed, as these would require customer-specific analysis during each update. This document clarifies in detail exactly what this restriction entails.
- Side-by-side and key user extensibility are preferred over classic extensibility, as they avoid dependencies on non-whitelisted SAP objects, which may require customer regression testing following an update.
- Also note that prior to an update, classic extensions must be activated and in released transports;

Additionally, many customers choose the SAP S/4HANA Cloud, single tenant edition first, with the SAP S/4HANA Cloud on their implementation roadmap. Side-by-side and key user extensions could potentially be replicated in any future S/4 HANA Cloud (multi-tenant) implementation.

## 4. CLASSIFICATION

The remainder of the document outlines common extension scenarios which are split between:

- Enhancements – changing something that was provided as standard
- Custom Development – building something new (in some cases using a standard template as a starting point).
- Configuration

For each scenario the possible extension techniques are listed in a suggested order of preference. The various techniques are assigned a color coding as follows:

Green

Possible in all S/4 HANA solutions.

These extension techniques are expected to have no impact on upgrades.

Amber	Allowed in SAP S/4 HANA Cloud, single-tenant edition, but not in SAP S/4 HANA Cloud. These classic extensibility techniques would not prevent an upgrade but may need regression testing afterwards. The reason being, they may reuse standard SAP ABAP objects such as tables, data elements, function modules or classes. If those standard objects are changed or replaced as part of the upgrade, then the custom extension may no longer work as it was originally designed or may no longer compile at all. These risks could be mitigated if customers voluntarily use the more restrictive form of ABAP, also used in key user extensibility apps. See section Restricted ABAP for more details. Or if this is not sufficiently flexible, customers could nevertheless try to use only whitelisted objects and APIs in their custom developments.
Red	Not allowed in SAP S/4 HANA Cloud, single tenant edition; not possible in SAP S/4 HANA Cloud (multi-tenant). These extension techniques typically involve directly changing standard SAP code or some other object. Such changes may require confirmation or adjustment during the upgrade process, which requires customer knowledge and judgement and could not be completed automatically by SAP.

## 5. ENHANCEMENTS

For all types of enhancements, customers should first consider carefully if the enhancement can be avoided altogether?

- Verify that SAP S/4 HANA functionality can't already handle the business requirement in some different way.
- Is it possible to fit the business process to the standard functionality?

### 5.1. Business Logic Enhancement

Order of Preference	Technique
1	Key user extensibility: use 'Custom Fields and Logic' app to implement a BAdI, using restricted ABAP.
2	Side-by-side extension may be used in combination with 'Custom Fields and Logic', to implement more sophisticated logic involving an external service. The technique is used in <a href="#">this example</a> to call an address validation microservice. Note that side-by-side extension does not provide any additional extension points for standard apps, and any extension is still limited by the input and output parameters provided in the BAdI interface.
3	Use a classic extensibility technique with a stable enhancement point, that does not require a modification key. E.g. - ABAP BAdIs (SE18/SE19) - AMDP BAdIs to enhance standard SQL Script procedures - SMOD/CMOD enhancements - BTEs (Finance)
4	Use a classic extensibility technique with a stable enhancement point, but that requires a modification key. E.g. - User Exits - VOFM routines such as pricing formulas
5	Implicit enhancements, or enhancement spots. These do not require a modification key, but otherwise are much like Modifications. They enable customers to change any SAP code at the start or end of any coding block. In an upgrade, these would need to be processed in SPAU_ENH – the risk is that the enhancement point may no longer exist, or may no longer have access to the same data.
6	Modifications In this standard SAP code is directly changed without restriction. The customer must first obtain an SSCR Modification Key, so each change is registered with SAP. In an upgrade, these would need to be processed in SPAU – the risk is that the enhancement point may no longer exist, or may no longer have access to the same data.

## 5.2. Data Model Enhancement

### 5.2.1. Add Custom Fields to a standard SAP business object

Order of Preference	Technique
1	Key user extensibility: use 'Custom Fields and Logic' app to define additional custom fields for a given Business Context. The app can extend not only the underlying database tables, but also any associated CDS views and oData services.
2	<i>Side-by-side extension would generally be the next recommendation, but it's not applicable in this case, as it doesn't offer any additional way to add custom fields to a standard Business Object.</i>
3	Use predefined Customizing Includes (CI_*). These are predefined Include structures present in some standard SAP tables, in which custom fields may be added. The fields may be defined through customizing (hence the name), for example in Finance the CI_COBL 'coding block' structure. It may also be possible to add fields to the includes directly. All custom field names must be in the customer namespace (e.g. starting 'ZZ') to avoid any clashes with SAP fields which may be added in the future.
4	Use an Append Structure. Additional fields may be added to standard SAP tables modification-free using an Append Structure. These are not predefined - they may be added to any table. Fields should again be in the customer namespace (see above). Typically, such a change would be made along with associated UI and Business Logic enhancements. In some cases, SAP have provided 'How to' guides describing all the steps required. Append structures may also be used to attach search helps or define foreign keys for existing standard fields.
5	Extension Includes These may be used to add custom fields to an app's oData service, if the app cannot be changed through 'Custom Fields and Logic'. Technically these are provided as DDIC Includes in standard structures, which can be extended using an Append Structure.
6	Direct table modification is not allowed, but regardless should never be necessary, as fields can always be added using an Append Structure.



### 5.2.2. Custom CDS Views

These may be created e.g. for custom embedded analytics, to provide an oData service, or to encapsulate and push down to HANA complex selection logic.

Order of Preference	Technique
1	Key user extensibility: use 'Custom CDS Views' app. This allows a custom view to be created using data from other published views. Quantities/Values may be aggregated. Input parameters and filters may be defined.
2	Classic extensibility: a CDS view may be defined using the ABAP Development Tools in Eclipse. In this case the CDS Data Definition Language is coded directly and therefore more sophisticated selection logic may be implemented.
3	Side-by-side extensibility could mean: 1. Replicating data tables to HANA in the SAP Cloud Platform (e.g. <a href="#">using SLT</a> ), and building a CDS view on top of them there. 2. Extracting data out to a separate BW system or other data warehouse, to build views on it there.

### 5.3. User Interface Enhancement

UI Technology	Order of Preference	Technique
All	1	<p>Personalization.</p> <p>For some UI elements users can make changes themselves. For example, for tables, it's typically possible to change the fields displayed, the order of the columns or the overall sort order. If the requirements are user-specific, then consider any personalization options first.</p>
	2	<p>Configuration / Settings.</p> <p>To apply changes more widely, there may be associated configuration or settings relating to the app. These may control for example what fields are displayed, read only or mandatory. It's worth checking if such configuration is available ahead of using other techniques.</p> <p>If the app being changed is available in SAP S/4 HANA Cloud (multi-tenant) then the related settings may be available there also, however this is not guaranteed.</p>
SAPUI5 Fiori Apps	3	<p>Run Time Adaptation (RTA).</p> <p>Use to make simple changes like hiding or adding fields or rearranging fields.</p>
	4	<p>SAP Web IDE - hide controls.</p> <p>Suitable if you only need to hide some UI controls, but the app does not support Run Time Adaptation.</p> <p>Note that using SAP Web IDE to change the UI of a standard app is not supported in S/4 HANA Cloud (multi-tenant).</p>
	5	<p>SAP Web IDE - extension points</p> <p>Use pre-defined extension points to change the app. These are documented in the Fiori Apps Library.</p>
	6	<p>SAP Web IDE - replace views.</p> <p>If the available extension points are not sufficient, then the entire view may be replaced with a custom one. The custom view may be created as a copy of the standard one.</p> <p>The disadvantage of this approach is that any improvements to the standard screen supplied through updates or SAP Notes will not automatically be applied to the custom one – if needed they would have to be applied manually.</p>
Web Dynpro Apps, including Floorplan Manager apps	3	SAP Screen Personas or Web Dynpro Customizing

Web UI Apps. E.g. SAP S/4 HANA for Customer Management; SAP Solution Manager, SAP ChaRM	3	SAP Screen Personas or Web UI Configuration Tool
SAPGUI Transactions	3	SAP Screen Personas
	4	Classic extensibility techniques such as CMOD enhancements or BAdIs. This technique may be needed to add custom fields to the UI, if this cannot be done using the Custom Fields and Logic app.

## 6. CUSTOM DEVELOPMENT

### 6.1. Apps

Order of Preference	Technique
1	Key user extensibility: use 'Custom Business Objects' app to implement any custom data model required, then generate the maintenance app, and code determination and validation logic as required using restricted ABAP.
2	Use 'Custom Business Objects' app as above, and then use the SAP Web IDE (available in SAP Cloud Platform, SCP) to further enhance the UI.
3	Side-by-side extensibility: build the app in the SCP. ABAP is on the roadmap for the SCP, but currently business logic must be written in another language such as Java. SAP S/4 HANA Cloud SDK may be used to integrate with SAP S/4 HANA Cloud, Single tenant edition where needed, e.g. calling oData services and BAPIs.
4	Classic extensibility using the BOPF framework. This is the same application framework underlying the Custom Business Objects app, but by working directly in this framework with classic ABAP, it's possible that a more sophisticated application may be built.
5	Classic extensibility using the BOL / GenIL, when developing Web UI based apps e.g. relating to SAP S/4 HANA for Customer Management; SAP Solution Manager, SAP ChaRM
6	Web Dynpro or Floorplan Manager apps without the BOPF. Possible if there's some barrier to using BOPF to define the business object.
7	SAPGUI Dynpro Transactions. Generally, this is no longer recommended for new applications - it should be possible to use one of the other approaches instead.

### 6.2. Workflow

Order of Preference	Technique



1	Key user extensibility: use the various Manage Workflow apps, relating to pre-defined business processes such as purchasing approval. These apps support the definition of workflows with multiple steps, recipient determination, and preconditions, for example based on value.
2a	Side-by-side extensibility: to create an entirely custom workflow process, use the SCP. It may be possible to use Business Events raised from SAP S/4-HANA as a trigger. Using the SCP, more complex workflow processes are possible which could involve multiple systems.
2b	Classic extensibility may still be more suitable than a side-by-side approach in two scenarios: 1) You may need more sophisticated workflow logic than the Manage Workflow apps support. With classic extensibility you may still reuse a standard workflow template as a starting point. 2) It may be that a suitable standard workflow exists already for the required business process, and could be adapted, but is not yet supported in the Fiori apps.

### 6.3. Output

#### 6.3.1. Print Forms

Order of Preference	Technique
1	Key user extensibility: use Maintain Form Templates app in combination with Adobe LiveCycle Designer. Data is supplied through an oData service, separate forms exist for the document headers/footers, and for the body content. LiveCycle Designer is a fully-featured, form design tool. For more complex form logic some scripting may be needed, which could be JavaScript.
2	Classic extensibility: older print form technologies are also supported in S/4 HANA on premise and single tenant edition, including 'classic' Adobe Forms (transaction SFP). However, it is recommended to use one consistent approach for all print forms, which should be the Maintain Forms Template app.
3	<i>Side-by-side extensibility - not applicable in this case. At the time of writing there's no SCP service to design print forms.</i>

#### 6.3.2. Emails

Order of Preference	Technique



1	In-App Extensibility: use Maintain Email Templates app. Create the required email template based on one pre delivered by SAP.
2	Prior to SAP S/4 HANA, there was no single and consistent approach to determine email texts. In some cases, it may be possible through configuration; in others an enhancement may need to be coded.



#### 6.4. Programs

Sub Type	Order of Preference	Technique
Programs - background jobs	1	<p>Classic extensibility - ABAP</p> <p>These are programs that should be scheduled to run regularly in the background without user intervention, usually to perform some technical activity or long-running update process.</p>
<i>Programs - background jobs</i>	2	<p><i>Side-by-side - currently there's no recommended way to implement and schedule a background job with the SCP.</i></p>
Programs - reports		<p><i>It's still technically possible, but should no longer be necessary, to write a report in ABAP. Recommended approach is to define a data source using a CDS view, then use one of the S/4 HANA embedded analytics options. For more sophisticated reporting the SAP Analytics Cloud or BW/4HANA could be used.</i></p>



## 7. CONFIGURATION

Extension Type	Order of Preference	Technique
Custom Configuration table	1	<p>Classic extensibility - define the table and generate table maintenance in the usual way, with the ability to transport the data. The table maintenance may be added to the IMG (transaction SPRO).</p> <p>Note that in SAP S/4 HANA Cloud, users do not have access to the IMG.</p>
	2	<p>Key user extensibility - 'Custom Business Objects' app may be used to define custom tables, however this is designed for master and transactional data, not configuration. There is no option in this app to make the table contents transportable. However, it may be used to create custom tables to hold non-transportable settings, which perhaps depend on master data which differs in each environment.</p>
Add/Change/Delete standard entries in standard table with no table maintenance	1	<p><i>Do not do this, unless performed as instructed in an SAP Note. At upgrade, custom content is likely to be overwritten and the table entries will revert to standard.</i></p>

## 8. RESTRICTED ABAP

Key user extensibility apps 'Custom Fields and Logic', 'Custom Business Objects' and 'Custom Re-usable Elements' include an in-app ABAP editor for coding e.g. data validations, substitutions or determinations. In this editor only a restricted form of ABAP is possible.

Supported language features in restricted ABAP are:

- Basic expressions, control and flow statements, variables and internal tables
- String, math, data & time operations
- Read and write access to the interface of the enhancement option
- Read and write access to white-listed SAP APIs (classes)
- Read access (full SQL select support) to white-listed SAP CDS views.

The following ABAP features are not allowed to ensure the robustness, security and data consistency target:

- Any database operation except selects from released views, for example the commit work statement is not allowed
- Access to files and other I/O commands
- Creating new tasks (parallel processing)
- Dynamic programming
- Code generation
- To simplify the syntax, obsolete ABAP statements are removed from the syntax

The complete list of allowed ABAP statements is available in the documentation:

<https://cp.hana.ondemand.com/dps/d/preview/1a93686c176845f0832a2a73221dd90b/1511%20500/en-US/frameset.htm?dd39f11e977b41b49a5421399f4850f5.html>

When implementing classic extensions (e.g. in Eclipse or SE80), it may be possible to choose the ABAP Language Version:

[https://help.sap.com/doc/abapdocu\\_752\\_index\\_htm/7.52/en-US/abenabap\\_versions.htm](https://help.sap.com/doc/abapdocu_752_index_htm/7.52/en-US/abenabap_versions.htm)

This means that customers could voluntarily use the same restricted ABAP from the key user extensibility apps, in their classic extensions. Whether this will be possible or appropriate should be considered on a case-by-case basis.

It may be useful to note, from the SAP Help link above:

- The program ABAP\_DOCU\_VERSION\_WHITELIST shows the repository objects that are allowed for the restricted ABAP language versions.
- The program DEMO\_ABAP\_VERSIONS makes it possible to check ABAP source code using the syntax rules of the different ABAP language versions.

[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2017 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See <http://www.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.