

Was macht die SAP-Entwicklung von morgen?

On-Stack-Extensibility am Beispiel des neuen ABAP-Kochbuchs

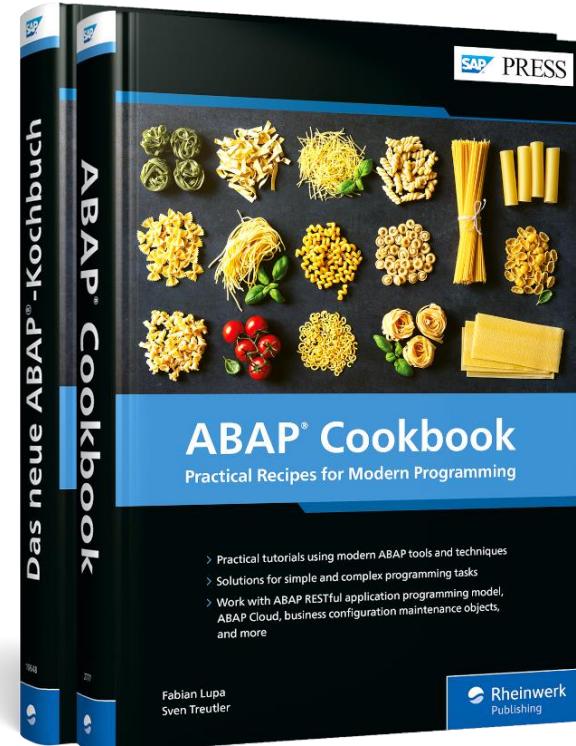


Das neue ABAP-Kochbuch



Fabian Lupa
Senior Software Engineer

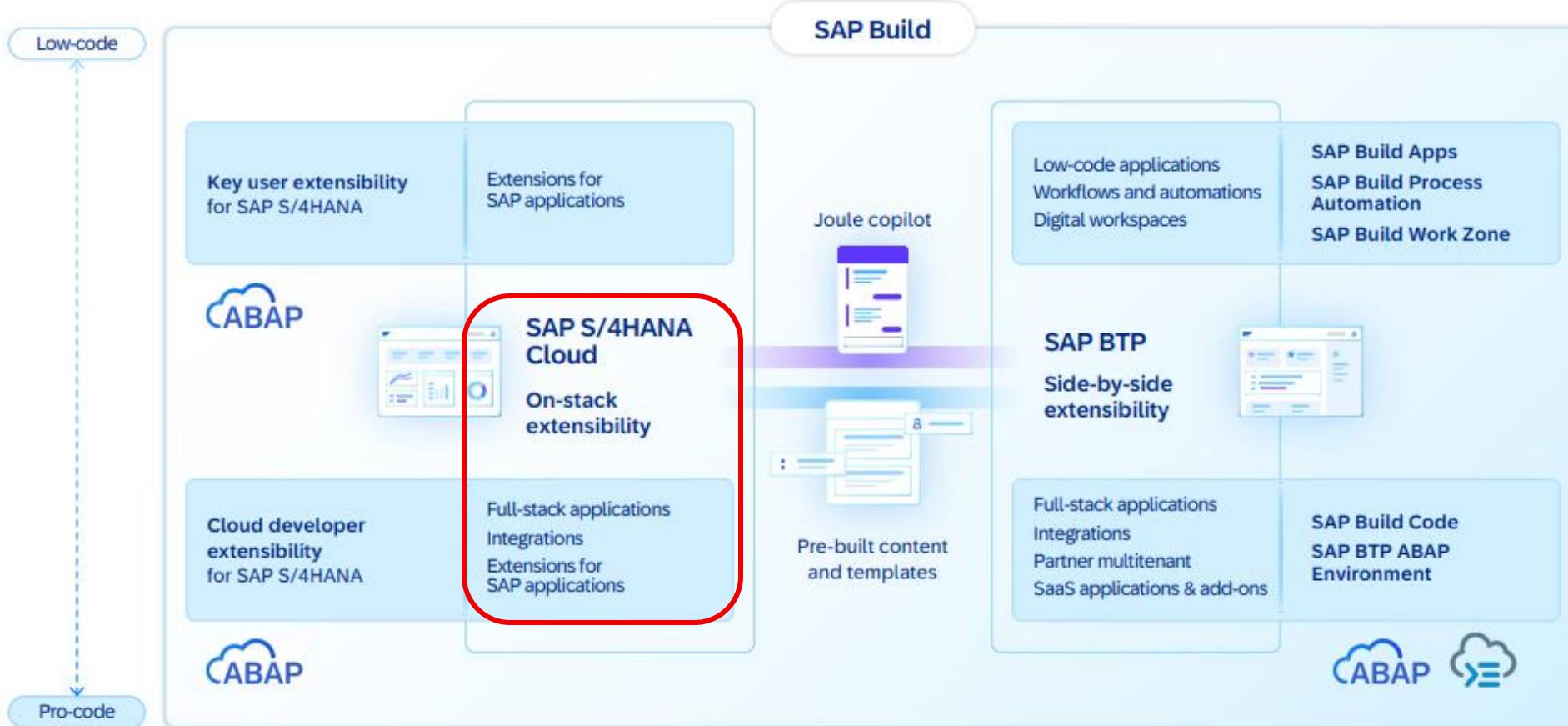
ABAP-Trainer
Practice Lead Development &
Extensibility



- > „Rezepte“ zur Lösung alltäglicher Programmierprobleme mit modernen Techniken
- > Durchgehendes Anwendungsbeispiel
- > Anwendbarkeit der Lösungen in allen modernen ABAP-Laufzeitumgebungen mittels ABAP Cloud: Public Cloud, Private Cloud und On-Premise

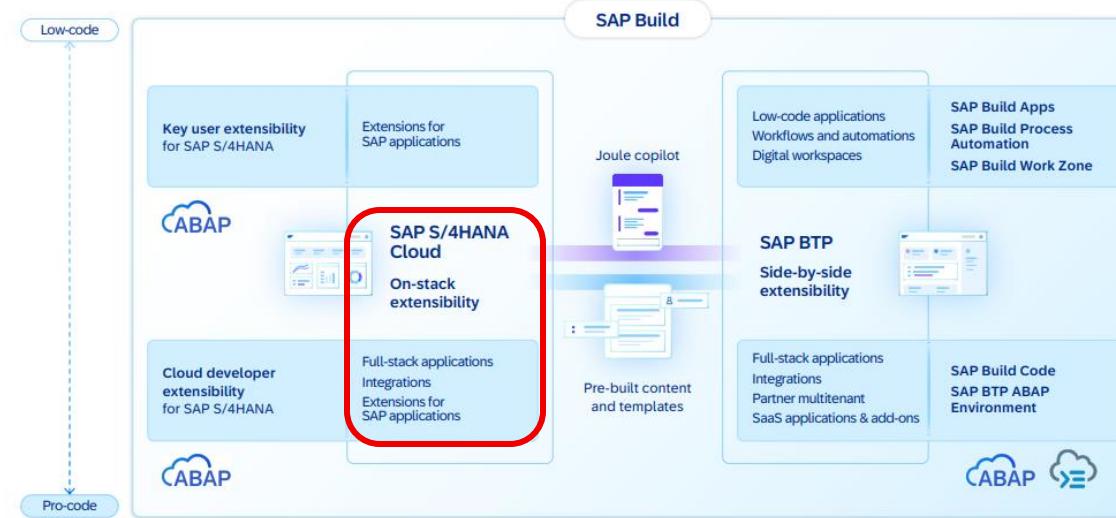


On-Stack-Extensibility in SAP S/4HANA (Cloud)



Bildquelle: SAP

On-Stack-Extensibility in SAP S/4HANA (Cloud)



Alleinstellungsmerkmale von SAP S/4HANA (On-Premise) / SAP S/4HANA Cloud Private Edition:

> Release-Stand

- Keine automatischen Upgrades
- Neues Release nur alle 2 Jahre
- ABAP-Version kann weit zurückliegen im Vergleich zu Public-Cloud-Umgebungen

> ABAP Cloud

- Verfügbar, aber optional, gesamter ABAP-Sprachumfang nutzbar
- Kombinationsmöglichkeiten mit Standard ABAP
- Abweichungen innerhalb von ABAP Cloud im Vergleich zu Public-Cloud-Umgebungen

> Clean Core

- Governance, Richtlinien und Tooling für sinnvollen Einsatz von nicht Public-Cloud-kompatiblen Techniken (Clean Core Level B bis D)

Laufzeitinformationen

GET TIME.

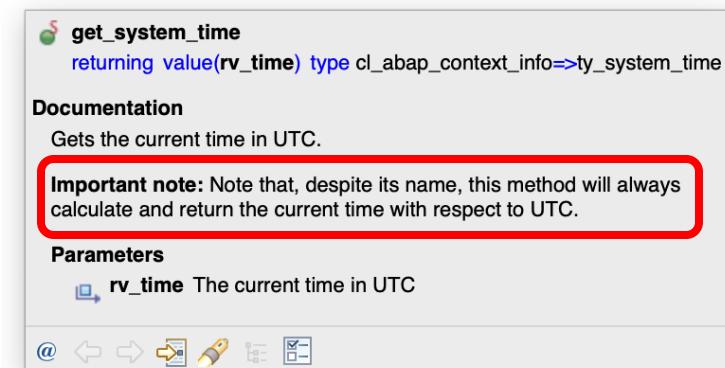
In der aktuellen ABAP-Sprachversion ist "GET" nicht erlaubt.

```
out->write( |⌚ Time information| ).  
out->write( |sy-datum: { sy-datum DATE = USER }| ).  
out->write( |sy-uzeit: { sy-uzeit TIME = USER }| ).
```

▼ Warnings (2 items)

- ⚠ Die veraltete Variante von "SY-DATUM" sollte in der aktuellen ABAP-Sprachversion nicht verwendet werden.
 - ⚠ Die veraltete Variante von "SY-UZEIT" sollte in der aktuellen ABAP-Sprachversion nicht verwendet werden.

```
out->write( |get_system_date: { cl_abap_context_info->get_system_date( ) DATE = USER }| ).  
out->write( |get_system_time: { cl_abap_context_info->get_system_time( ) TIME = USER }| ).
```





Laufzeitinformationen

☰ Convert ABAP code for cloud

...

FL

Heute 15:11

Hi, I got this coding in need to make it compatible with ABAP for Cloud Development in an on-premise system. Can you help me convert the code?
out->write(|sy-datum: { sy-datum DATE = USER }|).
out->write(|sy-uzeit: { sy-uzeit TIME = USER }|).

(...)

Cloud-Compatible ABAP Code for Date and Time Display

For ABAP Cloud Development, the original code needs to be modified to use cloud-compatible alternatives to the system fields sy-datum and sy-uzeit. Here's the updated version:

```
1 " Cloud-compatible approach using CL_ABAP_CONTEXT_INFO
2 DATA(current_date) = cl_abap_context_info=&gt;get_system_date( ).
3 DATA(current_time) = cl_abap_context_info=&gt;get_system_time( ).
4
5 out-&gt;write( |sy-datum: { current_date DATE = USER }| ).  
6 out-&gt;write( |sy-uzeit: { current_time TIME = USER }| ).
```

(...)

Key Changes for Cloud Compatibility:

1. Replace **sy-datum** with `cl_abap_context_info=>get_system_date()` [1] 
2. Replace **sy-uzeit** with `cl_abap_context_info=>get_system_time()` [2] 
3. Alternative: Use timestamp functions like `utclong_current()` for UTC-based time handling [3]



Laufzeitinformationen

1 How to get current system date in S/4 HANA cloud

✓ [Go to solution](#)

former_member284510
Explorer

on 2018 Jul 27 2:35 PM

0 Kudos 10,701

SAP MANAGED TAGS

SAP S/4HANA Cloud Public Edition Extensibility

Hello,

Help me out. As i'm new to S/4 HANA.

How to assign current system date in S/4 HANA cloud platform? As i used SY-DATUM but it's not working.

2023 Jun 19 2:38 PM

9 Kudos

Hello,

I think that using cl_abap_context_info is the easiest way to get this :

```
1 | cl_abap_context_info->get_system_date()
```

2 Smooth transition to ABAP for Cloud Development(Cheat sheet)

2023 Aug 15 9:51 AM

140 Kudos 86,788

SAP MANAGED TAGS

SAP BTP ABAP Environment SAP S/4HANA Cloud ABAP Environment ABAP Development ABAP Extensibility

ABAP RESTful Application Programming Model SAP Business Technology Platform

Content is now moved to GitHub. Future update will be performed at [Abap-for-Cloud-Development-CheatSheet](#).

More and more organizations are coming across the term ABAP on Cloud. Whether they are considering **Side-by-Side Extension** with SAP Business Technology Platform or **On-stack Extension** in S4 HANA, I'm certain that this is discussed in your strategy meeting at least once.

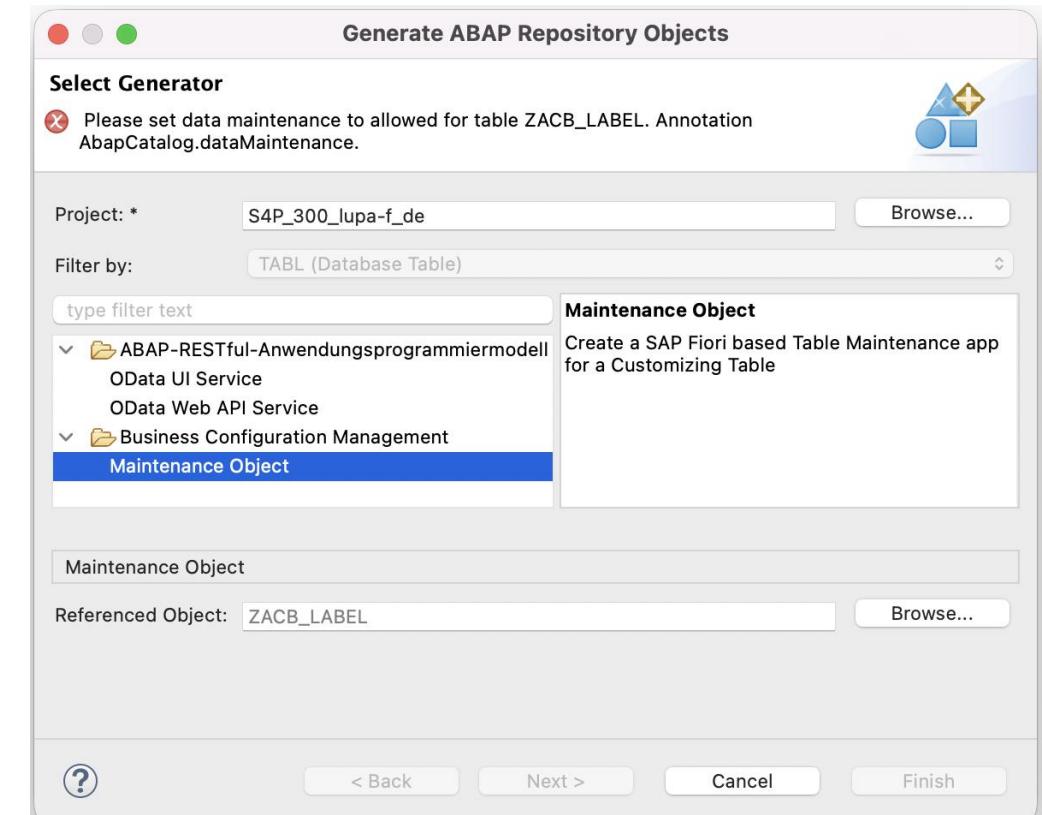
ABAP on Cloud(aka **ABAP for Cloud Development** language version) is similar yet different from **standard ABAP** in many ways. Even if your organization holds in-house ABAP developers, the transition path to ABAP on Cloud may not be easy. This is due to the fact that certain APIs(table, function module, tcode, etc.) exist in your system built with standard ABAP language are not functional in ABAP on Cloud. On the front-end level, the major factor of not being able to use SAP GUI changes certain ABAP based solutions. In addition, there are changes in ABAP syntax as well.

System date, time, timezone	
sy-datum	cl_abap_context_info->get_system_date()
sy-uzzeit	cl_abap_context_info->get_system_time()
sy-zonlo	cl_abap_context_info->get_user_time_zone()

Business Configuration

```
@EndUserText.label : 'Labels'  
@AbapCatalog.enhancement.category : #NOT_EXTENSIBLE  
@AbapCatalog.tableCategory : #TRANSPARENT  
@AbapCatalog.deliveryClass : #C  
@AbapCatalog.dataMaintenance : #RESTRICTED  
define table zacb_label {  
  
    key client          : abap.clnt not null;  
    key label_id        : zacb_label_id not null;  
    label_color         : zacb_label_color;  
    configdeprecationcode : config_deprecation_code;  
    last_changed_at    : abp_lastchange_tstmp;  
    local_last_changed_at : abp_locinst_lastchange_tstmp;
```

The screenshot shows the SAP Fiori application interface for the ZACB_LABEL table. At the top, there's a navigation bar with tabs: Transp.Tabelle, ZACB_LABEL, aktiv, Kurzbeschreibung, Labels, Eigenschaften, and Auslieferung und Pflege (which is selected). Below this, there's a section for 'Auslieferungsklasse' (Delivery Class) with a note: 'C Customizingtabelle, Pflege nur durch Kunden, kein SAP Import'. In the 'Data Browser/Tabellensicht-Pflege' section, there's a dropdown menu with several options: Anzeige/Pflege eingeschränkt erlaubt (selected), Anzeige/Pflege eingeschränkt erlaubt, D Nur Anzeige erlaubt, N Anzeige/Pflege nicht erlaubt, and X Anzeige/Pflege erlaubt.



Business Configuration

SAP Custom Business Configurations

Draft

Select Transport Show Documentation Display Change Logs

Label

Labels (1) Standard Copy Selected Entry Invalidate Selected Entry Deprecate Selected Entry Delete

Label ID	Label Text	Label Color	Validity
VEGAN	Vegan1	5	<input checked="" type="checkbox"/> Valid

Add Row Save Discard Draft

SAP Tabelle ZACB_LABEL ändern

CLIENT	300
LABEL ID	VEGAN
LABEL COLOR	1
CONFIGDEPRECATIONCODE	
LAST CHANGED AT	20.251.110.124.043,5637000
LOCAL LAST CHANGED AT	20.251.110.124.043,5637000
LABEL TEXT	

SAP Anzeige: 1 von 1 Treffern

Tabelle: ZACB_LABEL

CLIENT	LABEL_ID	LABEL_COLOR	CONFIGDEPRECATIONCODE	LAST_CHANGED_AT	LOCAL_LAST_CHANGED_AT	LABEL_TEXT
300	VEGAN	1		20.251.110.124.043,5637000	20.251.110.124.043,5637000	

Weitere Ungereimtheiten



Temporäre Pakete

- > Klassische temporäre Pakete mit \$-Präfix sind in Public-Cloud-Systemen nicht anlegbar, in Private Cloud / On-Premise hingegen schon, inklusive bei Nutzung von ABAP Cloud
- > Probleme:
 - > Softwarekomponente LOCAL vorbelegt und nicht änderbar
 - > Damit kein Zugriff per XCO oder CDS auf Entwicklungsobjekte:
 - Keine Domänenfestwerte
 - Keine Dictionary-Texte für Datenelemente
 - > Generiertes Coding in BC-Apps für Validierungen führt zu Laufzeitfehlern



APIs

- > Freigegebene APIs sind teilweise nicht überall freigegeben, selbst im Vergleich der Public SAP BTP ABAP Environment Trial-Umgebung und der Non-Trial-Umgebung (z. B. I_BusinessPartner, CL_MAIL_ADDRESS)
- > APIs unterscheiden sich in der Public Cloud und On-Premise, insbesondere bei Connectivity



abapGit

- > Der abapGit-Client in Public-Cloud-Systemen ist sehr eingeschränkt und führt bei neuen Objekttypen oft zu Abhängigkeitsproblemen

Restrictions and Errors

There are certain restrictions to the export and import functionality when using abapGit in the ABAP environment. This is due to the cloud-optimized set of supported ABAP object types that change with each release as new ABAP object types are added continuously. See [Released ABAP Object Types](#). Depending on your application, you may face the following issues when using abapGit in cloud systems:

- Export errors when pushing unsupported objects
- Import errors when pulling unsupported objects
- Activation errors when activating imported objects

Export Errors

If you try to export objects that are not supported, you receive a warning during the push operation, and the unreleased object is skipped.

The only work-around for this is to manually copy the missing ABAP objects to the target system. First, you have to create these objects using the same name in the target system. Afterwards, you either have to manually copy the code or reconfigure the objects in the target system.

Some objects are deliberately not exported because they are considered as compiled or generated so that they can be regenerated when activating their originating objects in the target system. For example, certain service definitions (SRVD) are generated from service consumption models (SRVC) so that they are not exported to git.

Import Errors

When importing objects with abapGit for ADT, you may get error messages in the import log.

Possible reasons can be:

- You are importing object types (for example from an on-premise system) that are not supported in the ABAP environment and the programming model of the ABAP environment. You may have to change your application in the cloud environment to work without these objects.
- You are importing objects with missing dependencies. These dependent objects can't be exported because they are not on the list of Released ABAP Object Types. You must copy/recreate these missing dependent objects and restart the import.
- You are importing objects using ABAP namespaces. Make sure these namespaces have been

Activation Errors

All objects imported with abapGit are imported in an inactive state, except the ones that do not offer an inactive state, to ensure that their consistency can be properly checked during activation. Hence, you have to activate all imported objects after importing them so that ABAP activation can verify that only released APIs have been used.

During the activation process, some objects may depend on other objects, which need to be activated beforehand. However, not all object types consider the objects they depend on during mass activation. For example:

- Data dictionary (DDic) and core data service (CDS) objects do not always resolve their dependencies in the first mass activation attempt so that you have to activate them in multiple cycles.
- Service bindings need to be activated after service definitions.

If activation issues occur, repeatedly try to activate objects in the right sequence. If that does not work, try reimporting and reactivating objects multiple times until all objects of your application are active.

Tip

If you face any issues with the official abapGit distribution of SAP, create an incident using component BC-CP-ABA.

For issues related to the community open-source distribution of abapGit, create an issue at <https://github.com/abapGit/abapGit/issues>.



SAP patrick_winkler
Product and Topic Expert

0 Kudos 601

SAP MANAGED TAGS

SAP BTP ABAP Environment ABAP Development ABAP Extensibility

...

2025 Mar 31 8:44 AM

Introduction

When you use the [ADT wizard](#) to generate a business configuration maintenance object (SMBC), a transport object (TOBJ) is generated by default and used in the behavior implementation.

While the [community version of abapGit](#) supports export/import of TOBJ, this is not the case in the pre-installed cloud version of abapGit.

This blog explains what you can do as a workaround when cloning a Git repository with TOBJ to an [ABAP platform system with preinstalled cloud version of abapGit](#).

This blog is relevant for

- [#SAP S/4HANA Cloud Public Edition](#)
- [#SAP BTP ABAP environment](#)

Further reading:

- [Related blog posts](#)
- Learn how you can use [ABAP technology](#) to develop innovative applications and business solutions across SAP's portfolio on [SAP Learning Site](#).

Workaround

The only way to create a TOBJ in BTP or SAP S/4HANA Cloud Public is to generate an SMBC object [using the ADT wizard](#).

1. Create a dummy table that meets the prerequisites of the wizard.
2. Run the wizard. In the "BC Administration" section of the configuration screen, change the name of the transport object to the name of the TOBJ of the Git repository.
3. Finish the wizard
4. Adjust the generated TOBJ to match the one from the Git repository.
5. Assign the generated TOBJ to the correct package
6. Delete all other generated repository objects including the dummy table

Problembehandlung

Der Repository Object Generator zeigt die Fehlermeldung "The use of ... is not permitted".

Hierbei handelt es sich um ein Problem in Kombination mit der ABAP-Sprachversion ABAP für Cloud-Entwicklung und lokalen \$-Paketen. Spielen Sie den SAP-Hinweis [3537063](#) in Ihr System ein.

Die Klasse CL_ABAP_BEHAVIOR_SAVER_FAILED ist nicht freigegeben.

Spielen Sie den SAP-Hinweis [3409039](#) in Ihr System ein.

Die Methoden zum Hinzufügen von Arbeitsblättern fehlen in der XCO-Komponente für Excel-Dateien.

Spielen Sie den SAP-Hinweis [3564421](#) in Ihr System ein.

Der Objekttyp TOBJ wird im SAP BTP ABAP Environment und in SAP S/4HANA Cloud Public Edition von der SAP-Version von abapGit nicht unterstützt.

Ein Workaround zur Anlage des in der Business Configuration Maintenance App verwendeten TOBJ -Objekts können Sie in folgendem Blog finden: [How to import a BC app based on RAP/SMBC/TOBJ with abapGit](#).

Die Objekte lassen sich im SAP BTP ABAP Environment oder in SAP S/4HANA Cloud Public Edition nicht aktivieren.

Details zur Fehleranalyse mit der SAP-Version von abapGit im SAP BTP ABAP Environment und in SAP S/4HANA Cloud Public Edition finden Sie im SAP Help Portal unter [Restrictions and Errors](#).

Die SAP-Version von abapGit zeigt einen Pull-Error in den ABAP Development Tools an.

Details zur Fehleranalyse mit der SAP-Version von abapGit im SAP BTP ABAP Environment und in SAP S/4HANA Cloud Public Edition finden Sie im SAP Help Portal unter [Restrictions and Errors](#).

Die SAP-Version von abapGit hat Objekte im SAP BTP ABAP Environment oder in SAP S/4HANA Cloud Public Edition mit der ABAP-Sprachversion Standard ABAP angelegt, die sich nicht bearbeiten lassen.

Hierbei handelt es sich um Bugs in der Implementierung der SAP-Version von abapGit. Öffnen Sie eine Meldung bei SAP in der Anwendungskomponente BC-CP-ABA-GIT zur Bereinigung/Bearbeitung der Objekte.

Die Klasse CL_MAIL_ADDRESS ist im SAP BTP ABAP Environment Trial nicht freigegeben.

Einige Entwicklungsobjekte sind generell in ABAP Cloud verfügbar und freigegeben, allerdings im Trial-System nicht verwendbar, und lösen daher Syntaxfehler aus. Zur Aktivierung müssen Sie im Trial-System die entsprechenden Code-Zeilen auskommentieren. In diesem Repository betrifft es die Klasse CL_MAIL_ADDRESS.

Ich erhalte den Laufzeitfehler BEHAVIOR_ILLEGAL_STATEMENT in der Business Configuration App

Die eingebaute Validierung in der Business Configuration Maintenance App nutzt zur Prüfung von Fremdschlüsseln die XCO-Bibliothek. Dieses ist jedoch nicht (mehr) in der Lage, Datenbanktabellen in temporären Paketen zu prüfen. Abhängig von der gewählten Installationsvariante und der Laufzeitumgebung schlägt die Validierungsprüfung daher mit einem internen Fehler fehl.

Als Workaround können Sie die Anwendung in einem transportpflichtigen Paket (z.B. ZC00KB00K) installieren, die Datenbanktabelle ZACB_LABEL über einen C1-Releasekontrakt mit Use in Cloud Development freigeben oder die Validierungsprüfung in der Methode VALIDATEDATACONSISTENCY im lokalen Klassen-Include der globalen Klasse ZBP_ACB_I_LABEL_S auskommentieren.

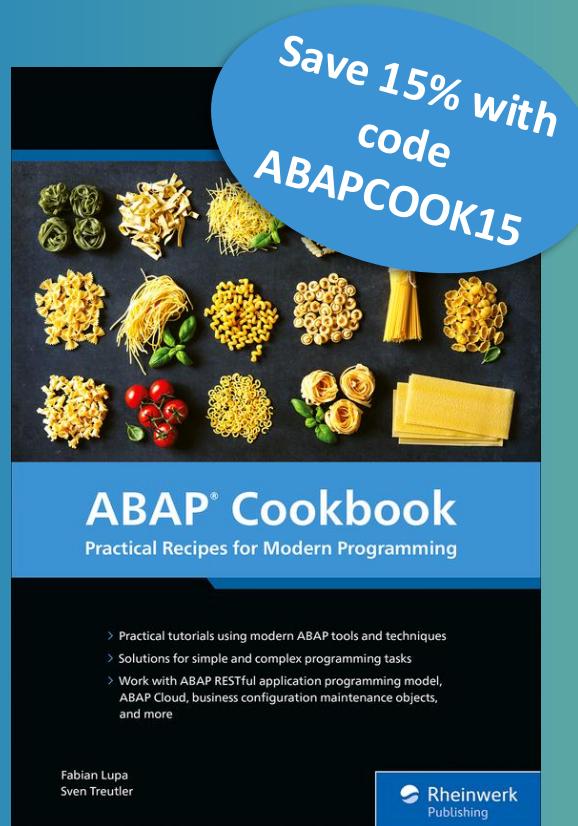
Details zum Fehler finden Sie im Hinweis [3671779 – Maintenance Object \(SMBC\) Wizard: Runtime error in consistency check](#).



Fabian Lupa

www.lupa.dev

   /fabianlupa



Save 15% with
code
ABAPCOOK15

ABAP® Cookbook
Practical Recipes for Modern Programming

> Practical tutorials using modern ABAP tools and techniques
> Solutions for simple and complex programming tasks
> Work with ABAP RESTful application programming model,
ABAP Cloud, business configuration maintenance objects,
and more

Fabian Lupa
Sven Treutler

Rheinwerk Publishing

sap-press.com/6198



SAP PRESS

Das neue ABAP®-Kochbuch
Praktische Programmierlösungen für Fortgeschrittene

> Nützliche Tutorials zu den wichtigsten Tools und Techniken
> Moderne Lösungen für Ihre täglichen Programmieraufgaben
> ABAP Cloud, ABAP RESTful Application Programming Model,
Business Configuration Maintenance Objects u.v.m.

Fabian Lupa
Sven Treutler

Rheinwerk Publishing

sap-press.de/6050

SAP Stammtisch Dortmund



SAP Stammtisch Dortmund / Community Meetup Dortmund

Treffen der lokalen SAP Community aus Dortmund und Umgebung

Zum nächsten Treffen 

Der SAP Stammtisch Dortmund ist ein regelmäßiges informelles Treffen der lokalen SAP Community. Eingeladen ist jeder, der Lust und Zeit hat, unabhängig vom fachlichen Hintergrund, von Entwickler, Berater, Projekt Manager, Basis Administrator bis hin zum Endanwender.

Im Mittelpunkt steht ein lockerer fachlicher Austausch rund um SAP-Themen über Unternehmensgrenzen hinweg.



sapstammtisch.github.io/Dortmund/