HOW-TO GUIDE | PUBLIC

2020-02-12

Custom Code Migration Guide for SAP S/4HANA 1909

Feature Package Stack 01

THE BEST RUN **SAP**

Content

1 Custom Code Migration Guide for SAP S/4HANA 1909

This guide focuses on the custom code related process from checking custom code remotely, analyzing the findings, and finally adapting the custom code to SAP S/4HANA 1909.

Read this guide carefully to get instructions on how to adapt your custom code to SAP S/4HANA 1909.

1.1 System Conversion

Custom Code Migration is part of the system conversion process from the classic SAP Business Suite running on any database to the SAP S/4HANA system. In the context of this system conversion, custom ABAP source code needs to be adapted, since a lot of SAP code within SAP S/4HANA was simplified and in some cases changed in a non-compatible way.
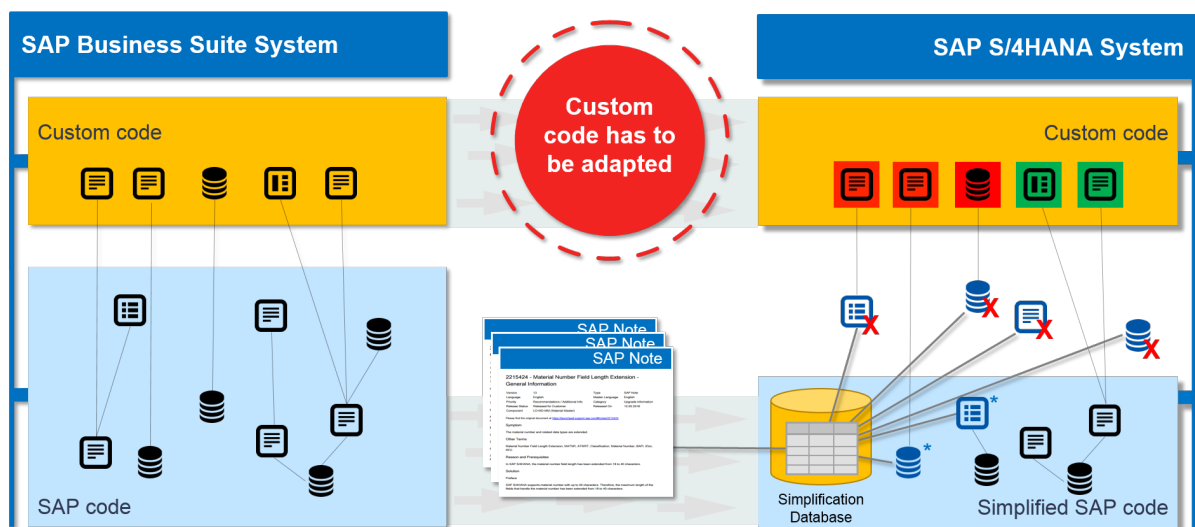
Some of your custom code objects are not valid anymore and either do not perform as expected or produce syntax errors or dumps (red objects in the picture). You almost certainly have other objects that do perform as expected and do not need to be changed (green objects in the picture).

SAP provides tools, based on the Simplification Database, that detect any custom code that needs to be adapted to SAP S/4HANA. The Simplification Database is a database table in the SAP S/4HANA system that contains all Simplification Items that refer to SAP objects simplified in SAP S/4HANA. Each simplification item describes changed or removed SAP objects and refers to a dedicated SAP Note that describes the impact of the change and how the related custom code can be adapted.

1.2   Process

The custom code migration process for the system conversion to SAP S/4HANA describes the tools and necessary activities to migrate your custom code. The process consists of preparatory analysis (custom code scoping and custom code analysis) and the adaptation of the custom code (custom code adaptation) after the technical system conversion.

The following figure provides you an overview of the phases, the activities, and the tools involved in this process.
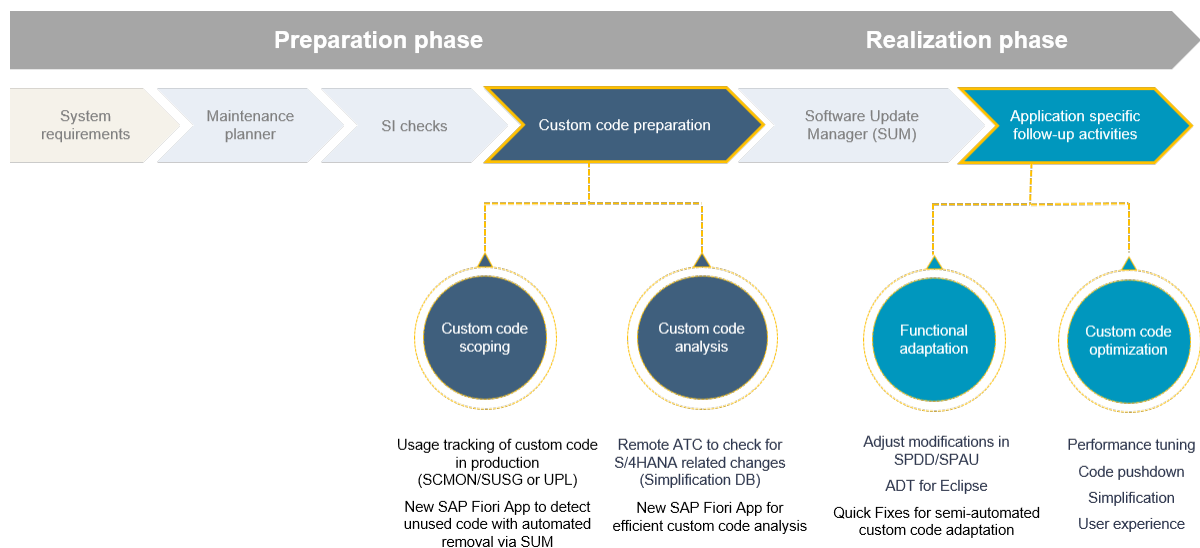
Overview of the conversion process that is divided into the preparation phase and the realization phase

For more information, see SAP S/4HANA System Conversion – Custom code adaptation process 🖉

Custom Code Scoping

Custom code scoping enables you to determine the ABAP custom code that is frequently used and needs to be taken over to SAP S/4HANA. Unused custom code can be deleted using deletion transport requests during the system conversion. This means, scoping enables you to reduce the number of custom code to be migrated and to minimize the efforts of the custom code adaptation.

> **i Note**
>
> Scoping is only supported using the `Custom Code Migration` app.

> **→ Recommendation**
>
> SAP recommends to scope your custom code in order to delete unused custom code and to minimize your efforts.
>
> If you want to benefit from scoping, you need to monitor your productive system landscape in advance. To do this, SAP recommends
>
> - Using the ABAP Call Monitor (transaction `SCMON`) and
> - Transaction `SUSG` to collect usage data in production system for at least one year.

For more information, see

- ABAP Call Monitor (SCMON) – Analyze usage of your code
- Aggregate usage data in your production system with SUSG transaction

Custom Code Analysis

Custom code analysis enables you to identify the custom code which has to be adapted in the context of the system conversion to SAP S/4HANA.

You have the following possibilities to run the custom code analysis:

- Using the `Custom Code Migration` app to
  - Run the SAP S/4HANA checks to analyze which custom code needs to be adapted.
  - Get statistical information about the findings in your development objects that need to be adapted.
- Using the classic **SAP GUI** to run the SAP S/4HANA checks based on the ABAP Test Cockpit (ATC).

Both options provide the relevant checks in a central check system. The central check system can remotely execute these checks on any ABAP system ((`SAP_BASIS` ≥ 7.00 where the custom code to be analyzed is stored.

Custom Code Adaptation (Functional Adaptation)

After the system conversion, you need to perform the functional adaptations in your custom code.

In the system conversion, you need to adapt any modifications related to ABAP Dictionary objects using transaction `SPDD`.

After you performed the system conversion to SAP S/4HANA with Software Update Manager (SUM), you need to adapt any modifications related to repository objects using transactions `SPAU` and `SPAU_ENH`.

After this, we recommend that you run ABAP Test Cockpit (ATC) with SAP S/4HANA checks in ABAP Development Tools (ADT). The result is a list of ATC findings that are related to SAP S/4HANA simplifications and refer to SAP Notes which describe how to solve the issues. Based on these ATC findings, you can start adapting your custom code using quick fixes in a semi-automated way.

→ Recommendation

SAP recommends you to run ABAP Test Cockpit (ATC) with SAP S/4HANA checks in *ABAP Development Tools (ADT)*. Here, the check result is provided as a list containing the ATC findings that relate to SAP S/4HANA simplifications and refer to the SAP Notes which describe how to solve the issues. Based on these ATC findings and specific quick fixes, you can start adapting your custom code on a semi-automated way.

i Note

You also need to adapt some of your custom code if you upgrade from a lower release to SAP S/4HANA 1909.

Performance Tuning

Finally, you need to check which business processes, for example performance of critical database queries, need to be optimized in SAP HANA. To do this, you need to check which SQL statements can be optimized.

The relevant performance data for all SQL statements executed in your productive system is displayed in *SQL Monitor*. Here, you can find the most expensive business processes and the frequently most used SQL statements.

2 Custom Code Analysis Before System Conversion

Overview

The possibilities how to run the custom code analysis depends on whether your central check system is an on-premise system or based on SAP Cloud Platform ABAP Environment.

Options

	SAP Fiori Custom Code Migration App	Remote ATC with SAP GUI	SAP Fiori Custom Code Migration App in SAP Cloud Platform
Required system	Central ATC check system **(SAP S/4HANA 1909)**	Central ATC check system **(SAP_BASIS 7.52)**	SAP Cloud Platform, ABAP Environment
System location	Customer landscape	Customer landscape	SAP Cloud Platform
Remote connectivity	Through RFC	Through RFC	Through RFC and SAP Cloud Connector
Analyze SAP S/4HANA findings	✓	✓	✓
Simplification information in ATC result	✓	✓	✓
Filter results by scope and Quick Fix availability	✓	✗	✓
Define custom code migration scope based on usage data	✓	✗	✓
Remove unused code during system conversion via SUM	✓	✗	✓

Options to Run the Custom Code Analysis

Follow **one** of the subsequent options how to run the custom code analysis:

1. If your central check system is based on **SAP S/4HANA 1909**, you can use the `Custom Code Migration` app to define the scope of your custom code that needs be migrated to SAP S/4HANA and to analyze custom code for SAP S/4HANA custom code checks. This SAP Fiori app acts as central check system on the latest version of SAP S/4HANA. This ensures that you run the latest version of the SAP S/4HANA custom code checks.

> → Tip
>
> If you choose this option, see Using SAP S/4HANA 1909 as Central Check System [page 8] for more information how to run the custom code scoping and custom code analysis.

2. If your central check system is based on **SAP NetWeaver Application Server for ABAP 7.52**, you can use an ABAP Test Cockpit (ATC) central check system instead. The latter analyzes custom code for SAP S/4HANA custom code checks.
With this approach you can neither scope your custom code based on usage data nor filter the analysis results by scope or quick fix availability.

> → Tip
>
> If you choose this option, see chapter Using SAP NetWeaver Application Server for ABAP 7.52 as Central Check System [page 22] for more information how to run the custom code analysis.

3. If your central check system is based on SAP Cloud Platform ABAP Environment, you can use the `Custom Code Migration` app to define the scope of your custom code that needs to be migrated to SAP S/4HANA and to analyze custom code for SAP S/4HANA custom code checks. The SAP Cloud Platform ABAP Environment will analyze your custom code in your on-premise system remotely through Cloud Connector. The SAP Cloud Platform ABAP Environment is updated four times a year and contains the latest features of the `Custom Code Migration` app.

> → Tip
>
> If you choose this option, see Using SAP Cloud Platform ABAP Environment as Central Check System [page 35] for more information how to run the custom code scoping and custom code analysis.

2.1 Using SAP S/4HANA 1909 as Central Check System

If your central check system is based on **SAP S/4HANA 1909**, you use the `Custom Code Migration` app to scope and analyze custom code in your on-premise SAP system for SAP S/4HANA custom code checks. This ensures that you run the analysis on base of the latest ATC check versions.

The `Custom Code Migration` app enables you to scope and analyze custom code that needs to be migrated from an SAP Business Suite system to SAP S/4HANA 1909.

To evaluate the development objects to be adopted, this SAP Fiori app performs the SAP S/4HANA custom code checks.
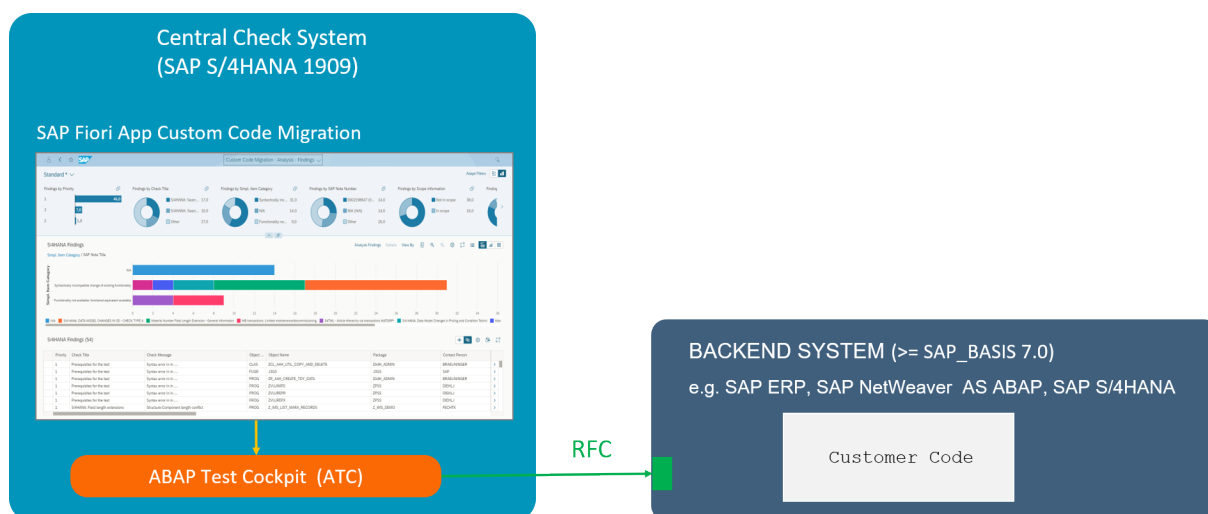
Setup of the remote custom code adaption of the check results investigated using the Custom Code Migration app

> **i** Note
>
> If you have already setup an SAP S/4HANA 1909 sandbox system, this system can also be used for the custom code analysis.

2.1.1 Getting Started

2.1.1.1 System Requirements

To perform the custom code analysis with the `Custom Code Migration` app, you must set up a SAP S/4HANA 1909 system.

SAP System	Software Component
Central Check System	`SAP_BASIS 7.54`
Checked System	SAP Business Suite system based on `SAP_BASIS 7.00` or higher

2.1.1.2 Configuring the User

In the **checked system**, the RFC user needs the following authorizations:

Name of Authorization Object	Activity	Type of RFC Object	Name of RFC Object	Description
S_RFC	16 (Execute)	FUGR	SABP_COMP_PROCS_E, SCA_REMOTE_DATA_ACCESS	Authorization for: • Configuring Object Providers • Running SAP S/4HANA checks
		FUNC	REPOSITORY_ENVIRONMENT_ALL, RFC_GET_NAMETAB, FUNCTION_EXISTS, RFCPING	
S_DEVELOP	03 (Display)	*	*	

In the **Central Check System**, you need the following user to use transaction `ATC` to perform custom code checks:

User Role	Description
SAP_SATC_ADMIN	Authorization for setting up ABAP Test Cockpit (ATC) for central quality checking

In addition, you need the following authorization object for importing the Simplification Database into the Central Check System:

Name of Authorization Object	Name of the Authorization Field	Value of the Authorization Field	Description
S_YCM	SYCM_AREA	SDB	Authorization for importing the Simplification Database
	ACTVT	UL	

2.1.1.3 Applying SAP Notes

Apply the following SAP Notes in the **checked system**:

- 2485231 - Remote ATC Checks of Modifications and Enhancements
- 2270689 - RFC Extractor for performing static checks
- 2190065 - ATC/CI: Remote Code Analysis - Object Provider Stub

Apply all the relevant SAP Notes that are mentioned in the following SAP Notes in the **Central Checked System**:

- 2436688🌧 - Recommended SAP Notes for using SAP S/4HANA custom code checks in ATC
- 2364916🌧 - Recommended SAP Notes for using ATC to perform remote analysis

2.1.1.4   Configuring the Connection Between Central Check System and Checked System

To perform the custom code analysis, you need to configure the connection between the Central Check System and the checked system.

2.1.1.4.1   Configuring RFC Connection Between the Central Check System and the Checked System

The `Custom Code Migration` app performs the SAP S/4HANA custom code checks in the checked system in which the custom code to be analyzed is stored.

Prerequisites

The RFC user in the checked system requires the authorizations that are specified in SAP Note 2672703🌧.

The RFC connection must be usable without a logon dialog.

Context

You want to configure the required connection from the back-end system of this SAP Fiori app to the checked system.

For more information, see Establishing a Connection Using a Destination (SM59)
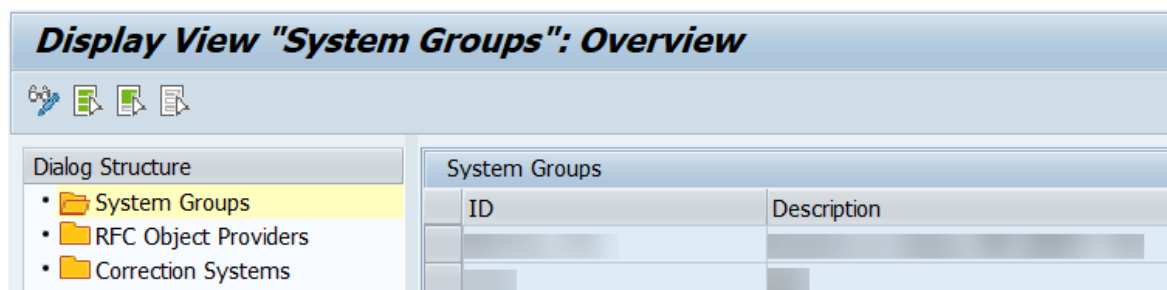
Procedure

Use transaction `SM59` to create RFC connections for each checked system.

2.1.1.4.2 Creating System Groups

A system group is a logical unit that groups multiple SAP systems. Every Object Provider must be assigned to a system group and therefore you need to create a system group before you configure an Object Provider.

Procedure

1. Run transaction `ATC`.

2. In the navigation pane, expand the node ▌▶ *ATC Administration* ❯ *Setup* ▌ and double-click *Object Providers*.

3. Double-click the item *System Groups* for selection.
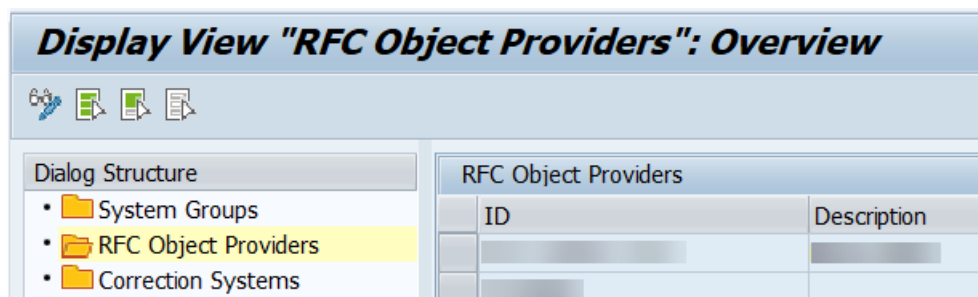

4. Switch to change mode (`F6`).

5. Choose the *New Entries* button in the toolbar.

6. Enter an *ID* and short *Description* for the new system group you want to add.

7. Save the new entry.

2.1.1.4.3 Configuring Object Providers

An Object Provider defines the RFC connection to be used for analysis in a remote SAP system. While a check run is being executed, the ATC framework uses this RFC connection to the checked system to extract a model from the custom code.

Procedure

1. In the *RFC Object Providers* view, double-click the item *RFC Object Providers* for selection.

2. Switch to change mode (F6).
3. Choose the *New Entries* button in the toolbar.
4. Specify the following entries for the Object Provider you want to create:

Fields	Description
`<ID>`	ID that specifies the Object Provider **i Note** This ID is used when configuring an ATC run series.
`<Description>`	Short text that specifies the Object Provider
`<System Group>`	The system group to which the Object Provider belongs
`<SAP System>`	ID of the remote system, to which you want to connect using the Object Provider
`<RFC Destination>`	Valid RFC destination for RFC connection to the checked system **i Note** This RFC connection must be usable without a logon dialog. The RFC user in the checked system requires the authorizations that are specified in SAP Note 2672703 .

5. Save the new entry.

2.1.1.5 Implementing the Custom Code Migration App

Perform the following tasks to implement the Custom Code Migration app.

Prerequisites

Before implementing the Custom Code Migration app, ensure that you have followed the steps for Implementing General Functions for the Key User.

Front-End Server: Enabling App for Access in SAP Fiori Launchpad

1. In transaction PFCG, add the business catalog SAP_BASIS_TCR_T (SAP: Application Services) to the user roles that will use the Custom Code Migration app.

> **i Note**
>
> This can be either an existing or a new user role.

2. Assign the user role to all users that will use the `Custom Code Migration` app.

The app is now available in SAP Fiori launchpad for all users that have assigned the changed user role.

> **→ Tip**
>
> Each user can add the app to its SAP Fiori launchpad by using the app finder in their user settings.

Front-End Server: Activating OData Services

1. In transaction `/IWFND/MAINT_SERVICE`, add the following technical service names as services:

Component	External Technical Name
Custom code migration projects	`SYCM_APS_C_PROJECT_CDS`
Analysis of SAP S/4HANA custom code check findings	`SYCM_APS_C_ATC_FIND_ALP_CDS`
Custom code scoping by request entry points	`SYCM_APS_C_SCP_BY_EP_CDS`
Custom code scoping by packages	`SYCM_APS_C_SCP_BY_PK_CDS`

For more information about activating OData services, see http://help.sap.com/s4hana `<Your on-premise edition>` ▶ *Product Documentation* ❯ *UI Technology Guide* ❯ *Implementation Task on the Front-End Server* ❯ *Activate OData Services* ❯ http://help.sap.com/fiori_implementation ▶ *Implementation* ❯ *SAP Fiori: App Implementation* ❯.

Back-End Server: Assigning Authorizations

1. Assign the user role `SAP_BC_YCM_APS` to the users who will work with the `Custom Code Migration` app.

2.1.2 Preparing the Custom Code Analysis

The following steps are relevant for the custom code analysis:

2.1.2.1 Importing the Simplification Database

To check your custom code against the SAP S/4HANA simplifications, you need to import the Simplification Database into the Central Check System.

2.1.2.1.1 Downloading the Simplification Database

SAP provides content for the Simplification Database of an SAP product in the SAP Support Portal. The content of the Simplification Database can be downloaded as a ZIP file. This ZIP file then has to be uploaded to the Central Check System.

Procedure

1. Open the SAP Software Download Center .
2. Choose *Software Downloads* and search for `CCMSIDB`.
3. Choose the relevant ZIP file and choose *Download Basket*.

   > **i** Note
   >
   > To find the most recent content of the Simplification Database provided for SAP S/4HANA, see also SAP Note 2241080 .

Results

The ZIP file is downloaded and saved on your selected drive. It contains the content of the Simplification Database.

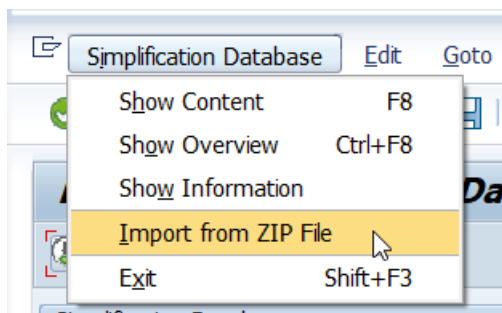## 2.1.2.1.2 Importing the Simplification Database

To add the content of the Simplification Database to your system for further analysis, you need to upload the downloaded ZIP file to the Central Check System.

Procedure

1. Log on to the Central Check System.
2. Run transaction SYCM.

   The *Display Simplification Database Content* view is opened.

3. Choose ▶ *Simplification Database* ❭ *Import from ZIP File* ❭ from the menu bar.

4. Choose the ZIP file with the downloaded Simplification Database from your drive and confirm with *Open*.

Results

The Simplification Database is uploaded to the Central Check System and is now available for analysis in transaction SYCM.

2.1.2.1.3 Displaying the Content of the Simplification Database

You can display the content of the Simplification Database to get an overview of all changes or specific changes and the corresponding SAP objects that are simplified in SAP S/4HANA.

Procedure

To get a list of all Simplification Items, choose `Ctrl` + `F8` or the *Overview* button.

> **i Note**
>
> To limit the number of Simplification Items displayed, enter the relevant filter criteria in transaction *SYCM* and choose Execute.
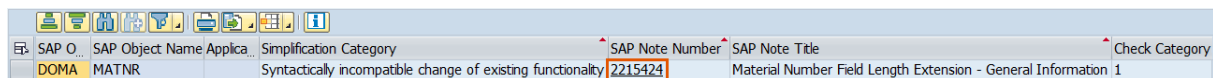
Results

The *Simplification Database Content* view is opened. From here you can order the displayed list, for example by their *SAP Object Type*, the assigned *Simplification Category*, or the relevant *SAP Note Number*.

From the overview, you can display the list of objects contained in a Simplification Item.

Example

To find out whether a certain object is related to a Simplification Item, enter the name of the object (for example **MATNR**) in the *Object Name* field and execute (`F8`). Then you can navigate to the corresponding SAP Note to get more information.

2.1.3 Custom Code Analysis with SAP Fiori App (Recommended)

The `Custom Code Migration` app enables you to scope and analyze custom code that needs to be migrated from an SAP Business Suite system to SAP S/4HANA 1909.

2.1.3.1 Creating a Custom Code Migration Project

A custom code migration project defines the scope of custom code which you want to analyze and migrate to SAP S/4HANA.

Procedure

1. Start the tile *Custom Code Migration* from the SAP Fiori launchpad.
2. Choose **+** to create a new custom code migration project.
3. Enter a *Project Description* for your new project and the *RFC Destination* to the SAP system which contains your custom code.

4. Define the *Target Release* of the back-end and the *Material Number Length*.

> **i Note**
>
> Make sure, that the checkboxes for *Extended Amount Length*, *Extended Stock/Requirement Segment* and *Extended Season/Theme/Collection* are enabled, if you use them.

5. To define the *Destination*, choose the object provider.

6. Confirm your input with *Save*.

2.1.3.2 Defining the Scope of Your Custom Code Migration Project

The Custom Code Migration app supports you with identifying your unused custom code based on your collected usage data. With the scope of a custom code migration project, you can specify which of your custom code needs to be converted to SAP S/4HANA.

To define the scope of your custom code migration project, you can either use your usage data collected with transaction SUSG or your usage data collected in your Solution Manager (this requires Solution Manager 7.2 support package 8 or higher).

> **i Note**
>
> If you didn't aggregate your usage data so far, you should activate the aggregation of your usage data in transaction SUSG in your production system now. To do so, start transaction SUSG in your production system and choose *Activate* to activate the aggregation of your usage data.
>
> Ideally, the usage data you add to your custom code migration project should cover at least one year of usage information, so that also usage data of quarter and year ending functionality is considered.

2.1.3.2.1 Adding Usage Data to your Custom Code Migration Project

To define the scope of your custom code migration project, you can use your usage data collected with transaction SUSG.

Procedure

1. Start transaction SUSG in your production system.

   The *Usage Data: Aggregation State* view is opened.

2. Choose *Create Snapshot* to create a snapshot of your usage data.

3. Recommended: Transfer your usage data from your production system to your checked system.

      a. Choose *Manage Snapshots*.

      b. Choose *Download to File* to export your snapshot as a file.

      c. Choose *Manage Snapshots* in transaction `SUSG` in your checked system.

      d. Choose *Upload from File* to import your usage data from a file.

4. Add the snapshot of your usage data to your custom code migration project in the Custom Code Migration app.

      a. Choose *Edit*.

      b. In the *Usage Data* frame, choose ➕.

      c. Choose ⬑ beneath the *Usage Description* field.

      d. Search for your usage data you created as a snapshot.

> **i Note**
>
> When you transferred your usage data to your checked system, your usage data is displayed automatically in the *Items* overview.

      e. Select your usage data to add it to your custom code migration project.

> **i Note**
>
> When you add usage data to your custom code migration project, a default scope is calculated. The default scope includes all used objects, all objects which are statically referenced by the used objects, and objects for which no usage data is available such as database tables or data elements.

5. Choose *Scope* to display the scope information.

    Here you can see, how many objects are in and out of scope.

> **→ Tip**
>
> Choose *View By* to display detailed scope information or your scope sorted by object name or object type.

2.1.3.2.2   Changing the Scope of Your Custom Code Migration Project

In the Custom Code Migration app, you can change the scope on the following levels:

- **Request Entry Points**
  Request entry points are all applications or services which are used in your system like programs, transactions, URLs, etc. For each request entry point, the used objects are recorded.
  In the initial scope, all used request entry points are added to the scope. You can remove request entry points from the scope by processing the following steps:
  1. Choose *Scope*.
  2. Choose *Scope Request Entry Point*.
  3. Select the request entry point you want to remove from the scope.
  4. Choose *Remove from Scope*.
- **Packages**

You can add or remove whole packages to/from the scope by processing the following steps:

1. Choose *Scope*.
2. Choose *Scope Packages*.
3. Select the package you want to add to or remove from the scope.
4. Choose *Add to Scope* or *Remove from Scope*.

- **Objects**
  You can add or remove single objects to/from the scope by processing the following steps:

  1. Choose *Scope*.
  2. Choose *Scope Packages*.
  3. Navigate to the package of the objects.
  4. Select the object you want to add to or remove from the scope either in the list "Not Scoped Objects" or in the list "Scoped Objects".
  5. Choose *Add to Scope* or *Remove from Scope*.

When is an object in scope or out of scope?

Objects are in scope, if …

- … they are added to the scope explicitly (by object),
- … their package is added to scope explicitly,
- … they are used by at least one request entry point which is in scope,
- … they are referenced by any other object which is in scope,
- or their object type has no usage data.

Objects are out of scope, if …

- … they are removed from scope explicitly (by object),
- or they are not used by any request entry point.

Example

When do you need to change the scope of your custom code migration project?

- **Scenario 1**
  A transaction which is used in your Business Suite system shall not be used any longer, because you want to change your business processed in SAP S/4HANA.
  Choose the filter *Scope by Request Entry Points* for your transaction and remove the request entry point from your scope.
- **Scenario 2**
  A new functionality is in development but has not been used in your production system yet. Therefore, no usage data has been collected for this application and it has not been added to your scope automatically. Choose the filter *Scope by Package* for all packages containing your developments for the new functionality and add them to your scope.

2.1.3.2.3 Removing Unused Custom Code During System Conversion

When you have specified your scope in your custom code migration project, you can create a transport request in the checked system. This transport request contains all objects which will not be migrated during the system conversion.

Procedure

Choose *Save Scope as Transport Request* to create the transport request.

The transport request is also shown in the field *Request/Task* in your custom code migration project.

Next Steps

For more information, see chapter "Integration of Customer Transports" in the corresponding upgrade guide for SAP HANA DB Software Update Manager 2.0.

2.1.3.3 Analyzing the Findings

When you create a custom code migration project, the SAP S/4HANA custom code checks are performed automatically. The Custom Code Migration app then gives you an analytical representation of SAP S/4HANA custom code check findings.

Procedure

1. Choose *Analysis* to get an overview of the results, as soon as the first analysis has been finished.
2. Choose *Analyze Findings* to get a detailed analysis.
3. In the *Custom Code Migration - Analysis - Findings* view, you can specify various filters, for example:
   - *Simplification Item Category*: The simplification item category specifies whether findings are related to functionality which is not available any more or has been changed in an incompatible way.
   - *SAP Note Number*: Lets you analyze findings of a specific Simplification Item (for example field length extension of material number)
   - *Scope Information*: Lets you filter the findings to show only findings for objects that are in scope.
   - *Quick Fix Availability*: Lets you analyze which findings can be solved by a Quick Fix and which findings have to be solved manually.
4. Choose ⚙ to change the settings for the chart.

   To show the Quick Fix availability per SAP Note title, you must specify the following settings for the chart:

- ○ SAP Note title (as Category)
- ○ Findings (as Axis 1)
- ○ Quick Fix Availability (as Series)

Results

You can now continue with chapter .

2.2 Using SAP NetWeaver Application Server for ABAP 7.52 as Central Check System

Performing the Custom Code Analysis

If the central check system is based on SAP NetWeaver Application Server for ABAP 7.52, your custom code needs to be checked remotely against the SAP S/4HANA simplification items in the Simplification Database using ABAP Test Cockpit (ATC) in the classic SAP GUI. The result is a list of findings where your custom code does **not** comply with the scope and data structure of SAP S/4HANA. At this point, you can estimate the effort required to adapt custom code to be migrated to SAP S/4HANA.
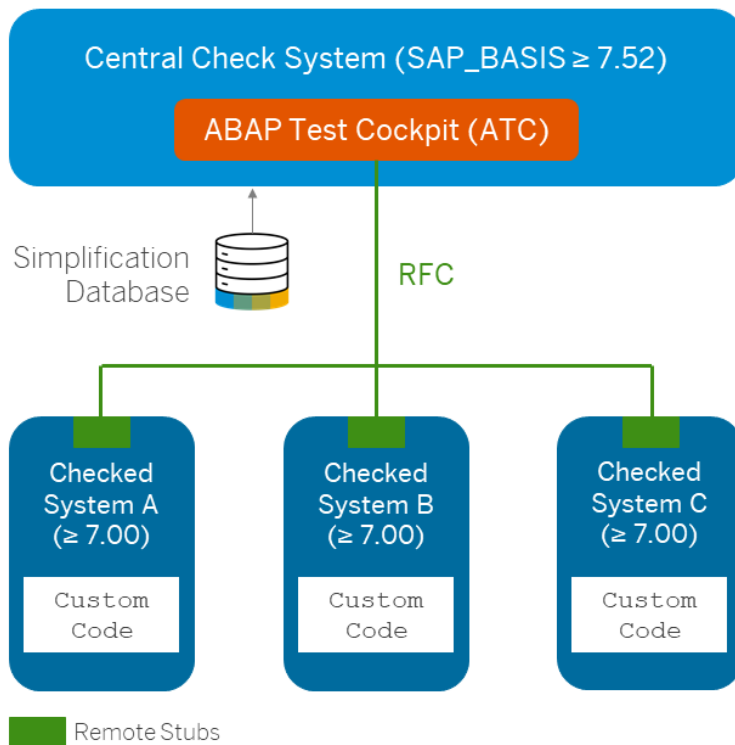
> **i** Note
>
> The Central Check System can be used to check one or more SAP Business Suite systems.

Enabling the Custom Code Analysis

To perform the custom code analysis, you need to set up a centralized SAP NetWeaver AS for ABAP 7.52 system as the Central Check System within your SAP system landscape.

For remote access, the Central Check System needs RFC destinations for each relevant system that you want to check using ATC. When executed, the Central Check System accesses the checked system using Remote Stubs and the RFC connection. These Remote Stubs are an interface between the Central Check System and the checked system and return a model from custom code that needs to be checked.

Central Check System (SAP_BASIS ≥ 7.52)

ABAP Test Cockpit (ATC)

Simplification Database

RFC

Checked System A (≥ 7.00)

Custom Code

Checked System B (≥ 7.00)

Custom Code

Checked System C (≥ 7.00)

Custom Code

Remote Stubs

2.2.1 Getting Started

2.2.1.1 System Requirements

To analyze your custom code in SAP GUI, you need to meet the following system requirements:

SAP System	Software Component
Central Check System	SAP_BASIS 7.52 or higher
Checked System	SAP Business Suite system based on SAP_BASIS 7.00 or higher

2.2.1.2 Configuring the User

In the **checked system**, the RFC user needs the following authorizations:

Name of Authorization Object	Activity	Type of RFC Object	Name of RFC Object	Description
S_RFC	16 (Execute)	FUGR	SABP_COMP_PROCS _E, SCA_REMOTE_DATA _ACCESS	Authorization for: • Configuring Object Providers • Running SAP S/4HANA checks
		FUNC	REPOSITORY_ENVI RONMENT_ALL, RFC_GET_NAMETAB, FUNCTION_EXISTS, RFCPING	
S_DEVELOP	03 (Display)	*	*	

In the **Central Check System**, you need the following user to use transaction `ATC` to perform custom code checks:

User Role	Description
SAP_SATC_ADMIN	Authorization for setting up ABAP Test Cockpit (ATC) for central quality checking

In addition, you need the following authorization object for importing the Simplification Database into the Central Check System:

Name of Authorization Object	Name of the Authorization Field	Value of the Authorization Field	Description
S_YCM	SYCM_AREA	SDB	Authorization for importing the Simplification Database
	ACTVT	UL	

2.2.1.3 Applying SAP Notes

Apply the following SAP Notes in the **checked system**:

- 2485231 - Remote ATC Checks of Modifications and Enhancements
- 2270689 - RFC Extractor for performing static checks
- 2190065 - ATC/CI: Remote Code Analysis - Object Provider Stub

Apply all the relevant SAP Notes that are mentioned in the following SAP Notes in the **Central Checked System**:

- 2436688 - Recommended SAP Notes for using SAP S/4HANA custom code checks in ATC
- 2364916 - Recommended SAP Notes for using ATC to perform remote analysis

2.2.1.4 Configuring the Connection Between Central Check System and Checked System

To perform the custom code analysis, you need to configure the connection between the Central Check System and the checked system.

2.2.1.4.1 Configuring RFC Connection Between the Central Check System and the Checked System

The `Custom Code Migration` app performs the SAP S/4HANA custom code checks in the checked system in which the custom code to be analyzed is stored.

Prerequisites

The RFC user in the checked system requires the authorizations that are specified in SAP Note 2672703.

The RFC connection must be usable without a logon dialog.

Context

You want to configure the required connection from the back-end system of this SAP Fiori app to the checked system.

For more information, see Establishing a Connection Using a Destination (SM59)
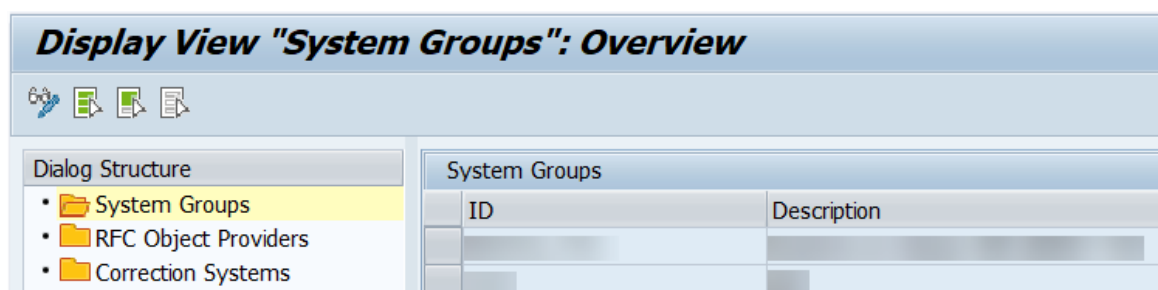
Procedure

Use transaction `SM59` to create RFC connections for each checked system.

2.2.1.4.2 Creating System Groups

A system group is a logical unit that groups multiple SAP systems. Every Object Provider must be assigned to a system group and therefore you need to create a system group before you configure an Object Provider.

Procedure

1. Run transaction `ATC`.

2. In the navigation pane, expand the node ▌▶ *ATC Administration* ❯ *Setup* ▌ and double-click *Object Providers*.

3. Double-click the item *System Groups* for selection.
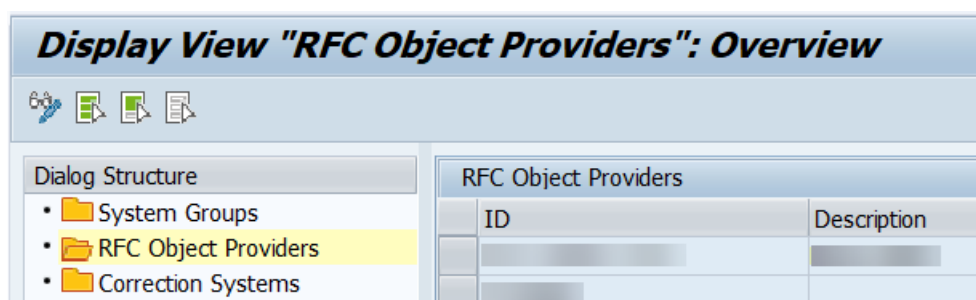

4. Switch to change mode (`F6`).

5. Choose the *New Entries* button in the toolbar.

6. Enter an *ID* and short *Description* for the new system group you want to add.

7. Save the new entry.

2.2.1.4.3 Configuring Object Providers

An Object Provider defines the RFC connection to be used for analysis in a remote SAP system. While a check run is being executed, the ATC framework uses this RFC connection to the checked system to extract a model from the custom code.

Procedure

1. In the *RFC Object Providers* view, double-click the item *RFC Object Providers* for selection.

2. Switch to change mode (F6).

3. Choose the *New Entries* button in the toolbar.

4. Specify the following entries for the Object Provider you want to create:

Fields	Description
<ID>	ID that specifies the Object Provider i Note This ID is used when configuring an ATC run series.
<Description>	Short text that specifies the Object Provider
<System Group>	The system group to which the Object Provider belongs
<SAP System>	ID of the remote system, to which you want to connect using the Object Provider
<RFC Destination>	Valid RFC destination for RFC connection to the checked system i Note This RFC connection must be usable without a logon dialog. The RFC user in the checked system requires the authorizations that are specified in SAP Note 2672703 .

5. Save the new entry.

2.2.2 Preparing the Custom Code Analysis

The following steps are relevant for the custom code analysis:

2.2.2.1 Importing the Simplification Database

To check your custom code against the SAP S/4HANA simplifications, you need to import the Simplification Database into the Central Check System.

2.2.2.1.1 Downloading the Simplification Database

SAP provides content for the Simplification Database of an SAP product in the SAP Support Portal. The content of the Simplification Database can be downloaded as a ZIP file. This ZIP file then has to be uploaded to the Central Check System.

Procedure

1. Open the SAP Software Download Center .
2. Choose *Software Downloads* and search for `CCMSIDB`.
3. Choose the relevant ZIP file and choose *Download Basket*.

> **i Note**
>
> To find the most recent content of the Simplification Database provided for SAP S/4HANA, see also SAP Note 2241080 .
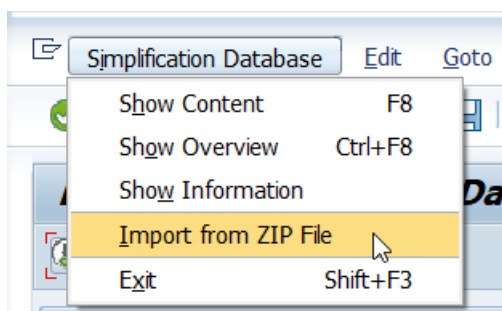
Results

The ZIP file is downloaded and saved on your selected drive. It contains the content of the Simplification Database.

2.2.2.1.2 Importing the Simplification Database

To add the content of the Simplification Database to your system for further analysis, you need to upload the downloaded ZIP file to the Central Check System.

Procedure

1. Log on to the Central Check System.
2. Run transaction `SYCM`.

   The *Display Simplification Database Content* view is opened.

3. Choose ▌ *Simplification Database* ❯ *Import from ZIP File* ❱ from the menu bar.

4. Choose the ZIP file with the downloaded Simplification Database from your drive and confirm with *Open*.

Results

The Simplification Database is uploaded to the Central Check System and is now available for analysis in transaction `SYCM`.

2.2.2.1.3 Displaying the Content of the Simplification Database

You can display the content of the Simplification Database to get an overview of all changes or specific changes and the corresponding SAP objects that are simplified in SAP S/4HANA.

Procedure

To get a list of all Simplification Items, choose `Ctrl` + `F8` or the *Overview* button.

> **i Note**
>
> To limit the number of Simplification Items displayed, enter the relevant filter criteria in transaction *SYCM* and choose Execute.

Results

The *Simplification Database Content* view is opened. From here you can order the displayed list, for example by their *SAP Object Type*, the assigned *Simplification Category*, or the relevant *SAP Note Number*.

From the overview, you can display the list of objects contained in a Simplification Item.

Example

To find out whether a certain object is related to a Simplification Item, enter the name of the object (for example **MATNR**) in the *Object Name* field and execute (`F8`). Then you can navigate to the corresponding SAP Note to get more information.

	SAP O...	SAP Object Name	Applica...	Simplification Category	SAP Note Number	SAP Note Title	Check Category
	DOMA	MATNR		Syntactically incompatible change of existing functionality	2215424	Material Number Field Length Extension - General Information	1

2.2.3 Custom Code Analysis in SAP GUI

If the central check system is based on SAP NetWeaver Application Server for ABAP 7.52, your custom code needs to be checked remotely against the SAP S/4HANA simplification items in the Simplification Database using ABAP Test Cockpit (ATC) in the classic SAP GUI

## 2.2.3.1 Setting the Role as Central Check System

The system role needs to be specified as Central Check System for remote SAP S/4HANA checks.

Procedure

1. Run transaction `ATC`.

   The *ABAP Test Cockpit Overview* screen appears.

2. In the navigation pane, expand the node ▶ *ATC Administration* ❯ *Setup* ❯ and double-click *System Role*.
3. Switch to change mode (`F6`).
4. In the *Change System Role* view, choose *ATC Checks Using Object Providers*.
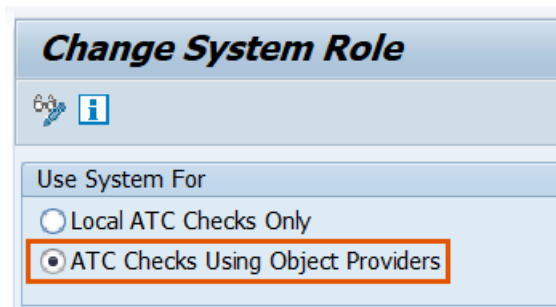
5. Save the new settings and return to the *ABAP Test Cockpit Overview* screen.

2.2.3.2 Running Remote SAP S/4HANA Checks

In the Central Check System, you can perform SAP S/4HANA checks to analyze development objects in a checked system remotely. In this step, you can estimate the effort required to adapt your ABAP source code to SAP S/4HANA-related changes.

2.2.3.2.1 Configuring a Remote ATC Run Series

Procedure

1. Run transaction `ATC`.
2. In the navigation pane, expand the nodes ▶ *ATC Administration* ❯ *Runs* ❯ and double-click *Schedule Runs*. The screen that appears displays a list of existing run series.
3. Choose the *Create* button in the toolbar.
4. Enter the name for the series you want to create and confirm.
5. Specify the following entries for the run series you want to create:

Field	Description
`<Description>`	Enter a short description for the ATC run series you want to configure.
	→ Tip You can use the template provided by the *Description* field. When you execute the run series, the built-in variables are filled with data for the system, the day of the week, the calendar week, and for the year. However, you can also add further texts to these variables, rearrange them, or replace the built-in variables with your own text.

Field	Description
`<Check Variant>`	Enter the check variant `S4HANA_READINESS_1909` **i Note** This product-specific global check variant checks only the simplifications relating to SAP S/4HANA 1909. **→ Tip** If you want to use a material number with less than 40 characters in SAP S/4HANA, you can copy the check variant `S4HANA_READINESS` and specify your desired length of the material number for the check in transaction `SCI` for the *S/4HANA Field Length Extension*. Here, also make sure that the *Extended Amount Length*, *Extended Stock/Requirement Segment* or *Extended Season/Theme/Collection* are enabled, if you use them. Note that you will get more findings to be adapted in your custom code if you use material numbers with more than 18 characters.
`<Object Provider>`	Enter the ID of the relevant object provider that represents the remote system you want to check.
`<Objects to Check>`	Specify the set of development objects you want to check in the checked system. On the *Checkable Namespaces* tab, the option *By Query* lets you specify objects by name (for example, by package name), by transport layer, or by component. The option *By Object Set* lets you specify an object set you defined in Code Inspector in the checked system. **→ Tip** You can use the value help to choose the packages or the object set in the remote system. On the *Modified Objects* tab, you can specify the modified source code objects you want to check.

6. *Save* the configuration.

2.2.3.2.2 Scheduling a Remote ATC Run Series

Procedure

1. In the list of run series, select the run series in question and click the *Schedule* button in the toolbar.
2. In the dialog that appears, adapt the settings for the *Life Span* of the series and for *Execution* to your needs. Otherwise, leave these settings unchanged.

   > **i Note**
   >
   > The *Life Span* defines how long (in days) the ATC result is kept in the system. After this, the ATC result is automatically deleted in the system.

3. Choose *Execute* (F8).

   > **i Note**
   >
   > You can use the option *Execute in Background* (F9) to schedule a run series at regular intervals.

2.2.3.2.3 Monitoring an ATC Run Series

In this step, you can monitor the status of the remote ATC run series.

Procedure

1. Run transaction ATC.
2. In the navigation pane, expand the nodes ▌ *ATC Administration* ❯ *Runs* ▌ and double-click *Monitor and Control Runs*.
3. Leave the run series field empty and execute (F8).

   The *ATC Run Monitor* view is displayed. The view shows the status of check runs as running, finished, or failed.

2.2.3.2.4 Analyzing ATC Run Results

Procedure

1. Run transaction ATC.
2. In the navigation pane, expand the nodes ▌ *ATC Administration* ❯ *Runs* ▌ and double-click *Manage Results*.

3. Leave the run series field empty and choose *Execute* (F8).

> **i Note**
>
> By default, the ATC run results of the last 10 days are displayed. If you want to see the ATC run results from a different period of time, change the settings in *Schedule Data*.

The *ATC Manage Results* view is displayed. The view shows a list of ATC run results from your requested period of time.

4. In the *ATC Manage Results* view, select a run series and choose *Display* to inspect the results.

A list of all ATC findings is displayed in transaction SE80.

5. Choose *Statistics View*.

6. Choose *Choose Statistic* to group the ATC findings (for example, by SAP Note Number).
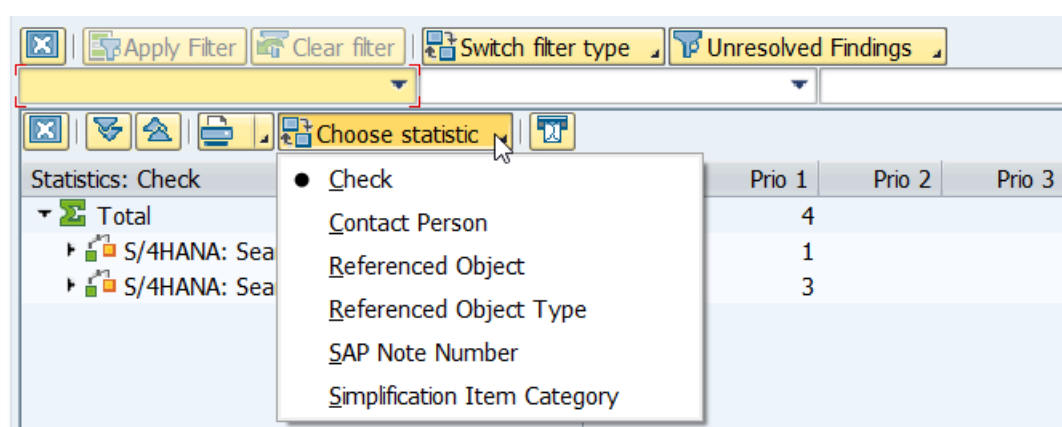
The ATC findings are grouped by the selected statistics.

7. Double-click a statistics group to display the list of ATC findings assigned to the selected statistics.

8. Double-click an ATC finding to open detailed information. Here you can find the SAP Note number referring to a dedicated SAP Note and the referenced object.

9. Click the object name to navigate to the source code to see where the incompatible code occurs.

Results

You can now continue with chapter Custom Code Adaptation After System Conversion [page 53].

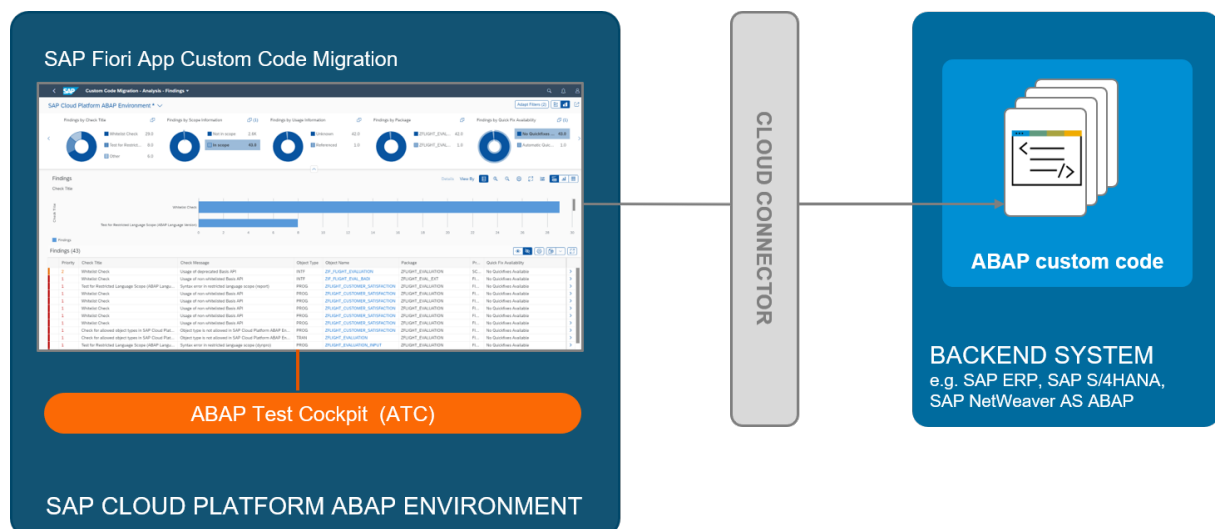## 2.3 Using SAP Cloud Platform ABAP Environment as Central Check System

If you use **SAP Cloud Platform ABAP Environment** as central check system, you can use the `Custom Code Migration` app to scope and analyze custom code in an on-premise SAP system for SAP S/4HANA. In this case, you are using the latest ATC checks.

You want to bring your custom code from an SAP Business Suite system to SAP S/4HANA. The Central Check System, where the analysis for possible adaptations takes place, is based on SAP Cloud Platform ABAP Environment. Note that here are always the latest ATC checks are provided.

To do this, the `Custom Code Migration` app enables you to find out the relevant parts in your source code and the effort for converting your customer code SAP S/4HANA, on-premise edition.

To use this SAP Fiori app, your SAP administrator needs to enable it and provide access for the relevant users. In addition, you also need to establish an RFC connection for the cloud connector. This enables you to access the on-premise system from ABAP environment remotely.

The connection between the ABAP environment as Central Check System to the on-premise system as Checked System is established on base of an RFC connection that is configured as destination and a communication arrangement in the *Cloud Connector*.

Remote infrastructure using an RFC connection between an on-premise SAP system and SAP Cloud Platform ABAP Environment across the Cloud Connector

2.3.1 Getting Started

2.3.1.1   Creating Subaccounts in the Global Account

A subaccount enables you to run applications and services in the global account of your ABAP environment.

Prerequisites

You need SAP administrator permission to create subaccounts.

Context

You want to establish RFC connection to your on-premise system to be checked.

Procedure

To do this, you need to create a subaccount for the Neo environment and the Cloud Foundry environment as follows:

1. As an SAP administrator, select the *Global Accounts* navigation pane and open the relevant global account in the *SAP Cloud Platform Cockpit*.
2. Select the *Subaccounts* navigation pane and create a new subaccount in the Cloud Foundry environment and the Neo environment. To do this, choose the *New Subaccount* button from the header toolbar each time.
3. In the dialog, enter the following data:
   a. Enter the *Display Name*.
   b. [Optional:] Provide a *Description*.
   c. To create a subaccount for the Neo environment, select the *Neo* checkbox.

      To create a subaccount for Cloud Foundry environment, deselect the latter checkbox.

      In accordance to your selection, the relevant fields are display.
   d. Choose the *Provider* and *Region* of your system connection.
   e. If you choose *Cloud Foundry*, enter the *Subdomain* to use a custom domain for your company.
   f. [Optional:] If required, choose the following checkboxes:
      - *Use for production*
      - *Enable beta features*
      - *Copy settings from an existing subaccount*
4. Choose *Create* to finish the creation process.

Results

For each environment, a new subaccount appears in the global account page and displays the details.

The subaccount for the ...

- `Neo environment` enables you to connect your cloud connector which contains the RFC connection(s) to your on-premise system(s).
- `Cloud Foundry` environment enables you to create a space and a service instance.

2.3.1.2 Configuring and Administrating the Cloud Connector

The *Cloud Connector* enables the connection between the SAP Cloud Platform ABAP Environment and on-premise systems in a secured network.

To enable connection from the ABAP environment to the on-premise system, proceed as follows:

2.3.1.2.1 Connecting the Neo Environment Subaccount to the Cloud Connector

Prerequisites

You have created a subaccount for the Neo environment.

Context

You want to add and to connect your subaccount for the Neo environment to the *Cloud Connector*.

For more information, see

- Managing Subaccounts
- Add Members to Your Neo Subaccount

Procedure

1. In the *Cloud Connector Administration*, select the *Connector* navigation pane.

> **i Note**
>
> Check if the ID of the *Subaccount* that you have created for the *Neo environment* in the *SAP Cloud Platform Cockpit* is entered in the title header.

2. Choose the *Add Subaccount* button in the title toolbar.

3. In the *Add Subaccount* wizard, enter the *Region* of your server.

4. Enter the ID of the *Subaccount* that you have created for the Neo environment in the *SAP Cloud Platform Cockpit*.

5. [Optional:] Enter the *Display Name* how to label the subaccount in the *Cloud Connector*.

6. To add the users as members to a subaccount in the Neo environment, enter the ID of the *Subaccount User*.

7. Enter the *Password* of the subaccount user.

8. [Optional:] To overwrite the default location ID of the data center in the Neo environment, enter the relevant *Location ID*.

9. [Optional:] Provide a *Description* for further details.

10. To add your subaccount to the *Cloud Connector*, choose *Save*.

Results

A new entry for the Neo subaccount is added in the *Subaccount Dashboard* table.

2.3.1.2.2 Configuring the Access Control (RFC) for the Remote Connection

The resource accessible represents the on-premise system that can be accessed through the *Cloud Connector*.

Context

You want to enable the connection for SAP Cloud Platform ABAP Environment to an on-premise system.

For more information, see Configure Access Control (RFC)

Procedure

To do this, you need to define access authorization by using the resource accessible (a. k. a. access control list or ACL).

1. In the *Cloud Connector*, select the ▌▶ *[Display Name of the Neo Subaccount]* ▶ *Cloud to On-Premise* ▶ navigation pane.

2. To add a new system mapping, choose ✚ from the section toolbar of the *Mapping Virtual To Internal System* table.

3. In the *Add System Mapping* wizard, provide the following data:

   a. Choose *ABAP System* from the *Back-end Type* drop-down list box.

   b. Choose *Next*.

   c. Choose *RFC* as *Protocol* from the drop-down listbox.

   d. Choose *Next*.

   e. Choose the *Without load balancing (application server and instance number)* radio button.

   f. Choose *Next*.

   g. Enter the following connection details of the on-premise system:

      ○ Virtual name of the *Application Server* ABAP.

      ○ *Instance Number* of your Application Server ABAP.

      ○ Name of the *SAProuter*.

   h. Choose *Next*.

   i. Enter the *Virtual Message Server*.

   j. Choose *Next*.

4. To trigger creation, choose *Create*.

   You have added the relevant on-premise system as resource accessible.

   Now, you want to define the permitted function modules as resources for a particular back-end system.

   To do this, proceed as follows:

5. In the *Cloud To On-Premise* navigation pane, select the relevant system mapping in the *Mapping Virtual to Internal System* section.

6. To define which function modules are executable in the on-premise, choose ⬆ *Import resources belonging to a scenario* from the section toolbar in the *Resources Of ...* section.

   > **i** Note
   >
   > To get the relevant import file for the Custom Code Migration scenario, see SAP Note 2861842 ☁ - Custom Code Migration in SAP Cloud Platform ABAP Environment: Set up SAP Cloud Connector.

7. *Browse* the *Scenario File* and choose *Select*.

Results

You have defined the on-premise system(s) that are accessible from the ABAP environment.

Now, you can call the defined function modules in the on-premise system.

2.3.1.3   Setting Up Destinations To Enable RFC Connection

To set up the RFC connection from the ABAP environment to your on-premise systems to be checked, proceed as follows:

2.3.1.3.1   Creating a Space

A space is a working area for a single application in the Cloud Foundry. The resources defined in the global account will be allocated to a space.

Prerequisites

- You have a `Cloud Foundry` organization in your `Cloud Foundry` subaccount.
- You have the `Org Manager` role in that organization.

Context

You want to define a working area for the `Custom Code Migration` app.

Procedure

To do this, you need to create a space as follows:

1. In the *SAP Cloud Platform Cockpit*, select the *Global Accounts* navigation pane and navigate to the subaccount that contains the relevant `Cloud Foundry` organization.

   The *Overview* page of the Cloud Foundry subaccount is opened.

2. In the *Spaces* section, choose the *Create Space* button from section toolbar.

3. Enter a *Space Name* and choose the roles that you want to assign to your ID.

4. *Save* your changes.

Results

A new space is added to the *Spaces* section.

You can use the new space as your working environment for the `Custom Code Migration` app in the Cloud Foundry. Herein, you will create a (destination) service instance to access the on-premise system.

2.3.1.3.2 Creating a Destination Service Instance

A (destination) service instance enables you to access a destination.

Prerequisites

You have access as a Cloud Foundry administrator to the SAP Cloud Platform Cockpit.

Context

You want to define the applications in the ABAP environment that are needed for outbound connectivity. To do this, you need to create a destination service instance and a service key in the created space for the destination service.

Procedure

To create a destination service instance, proceed as follows:

1. In the *SAP Cloud Platform Cockpit*, navigate to the global account of the SAP Cloud Platform Cloud Foundry environment.
2. Choose the tile for the Cloud Foundry subaccount.
3. On the *Cloud Foundry* tile, choose the link in the *Spaces* field.
4. Choose the tile for the space of the ABAP environment.
5. Choose ▶ *Services* ❯ *Service Marketplace* ❯ navigation pane.
6. Choose the tile for the *Destination* service.
7. In the navigation area, choose *Instances*.
8. Choose *New Instance* from the title toolbar.
9. Choose *Next* until the *Confirm* section.

   The selection in between are optional.
10. Define and enter an *Instance Name*.
11. Choose *Finish*.

Results

A new service instance is added.

2.3.1.3.3 Creating a Service Key for the Destination Service

A service key is used to generate credentials to communicate directly with a service instance.

Prerequisites

You have access as a Cloud Foundry administration user to the SAP Cloud Platform.

Context

You want to define the applications in the ABAP environment that are needed for outbound connectivity. To do this, you need to create a destination service instance and a service key in the created space for the destination service

Procedure

To create a service key for the destination service, proceed as follows:

1. In the *SAP Cloud Platform Cockpit*, go to the global account of the SAP Cloud Platform Cloud Foundry environment.
2. Open the Cloud Foundry subaccount.
3. Select the *Spaces* navigation pane.
4. Choose the tile for the space of the ABAP environment.
5. Choose ▶ *Services* ❯ *Service Instance* ❯ navigation pane.
6. Form the *Service Instances* navigation pane, choose the (destination) service instance which you have created for the destination.
7. Choose *Service Keys* navigation pane.
8. If no entry exists for the (destination) service instance, choose *Create Service Key*.
   a. Enter the *Name* for the service key.

   > ℹ Note
   >
   > Keep the *Configuration Parameters (JSON)* text box empty. The service key will be generated by the system.

   b. Choose *Save*.
9. Copy the created service key to the clipboard for later reuse.

Results

A new service key is generated in JSON format and added.

This service key is needed for the `SAP CP CF Destination Service Integration` communication scenario

2.3.1.3.4 Add an RFC Destination to Your (Destination) Service Instance

An RFC destination enables RFC connection between the ABAP environment and to the on-premise system across the Cloud Connector.

Context

You want to enable communication from your ABAP environment to the on-premise system for the `Custom Code Migration` app.

Procedure

To do this, you have to configure a (destination) service instance of type RFC as follows:

1. In the *SAP Cloud Platform Cockpit*, navigate to the global account of the SAP Cloud Platform Cloud Foundry environment.
2. Open the Cloud Foundry subaccount.
3. Select the *Spaces* navigation pane, and choose the tile for the space of the ABAP environment.
4. Choose ▶ *Services* ❯ *Service Instance* ❯ navigation pane.
5. Form the *Service Instances* navigation pane, choose the (destination) service instance which you have created for the destination.
6. Select the *Destinations* navigation pane and choose *New Destination* from the section toolbar.

   The *Destination Configuration* section is opened.
7. Enter the *Name* of the (destination) service instance.
8. Select *RFC* from the *Type* drop-down listbox
9. [Optional:] If you have overwritten the default location ID of the data center in the Neo environment, enter the relevant *Location ID*.
10. Enter the *User* to access the on-premise ABAP system.
11. Enter the *Passwort* of the user.
12. In the *Additional Properties* section, choose *New Property*.

    A drop-down listbox and an empty field is added.

13. Choose *jco.client.ashos* from the drop-down listbox and enter the system ID of the on-premise system.

14. Choose *jco.client.client* from the drop-down listbox and enter the value of the client that is used to logon to the on-premise system.

15. Choose *jco.client.sysnr* from the drop-down listbox and enter the instance number of the on-premise system.

16. [Optional]: Choose *Check Connection* to test if the connection to your on-premise system can be established.

2.3.1.4 Administratinging User Assignments and Maintaining Communication Arrangements

To assign the `Custom Code Migration` app to the users as well as to configure the connectivity between the ABAP environment and the Cloud Connector, proceed as follows:

- Administrating User Assignments [page 44]
- Maintain Communication Arrangements [page 45]

2.3.1.4.1 Administrating User Assignments

The user assignment describes the relevant roles that a user requires to access SAP Cloud Platform ABAP Environment using the `Custom Code Migration` app.

Context

You want to enable business users to access the `Custom Code Migration` app.

For more information see, How to Create a Business Role from a Template.

Procedure

To do this, create the `Project Manager - IT` business role using the `SAP_BR_IT_PROJECT_MANAGER` template and assign one or more business user(s) to it as follows:

1. In the SAP Fiori launchpad, navigate to the *Identity and Access Management* section and open the `Maintain Business Roles` app.

2. On the initial screen, select *Create from Template* from the bottom of the page.

   The *Create Business Role from Template* wizard is opened.

3. Enter *SAP_BR_IT_PROJECT_MANAGER* as the underlying *Template*.

4. Enter *BR_IT_PROJECT_MANAGER* as the name of the *New Business Role ID*.

5. Enter a *New Business Role Description*.

6. Confirm the creation with *OK*.

   The new business role is created and opened in the *Maintain Business Role* page.

7. Select *Maintain Restrictions* from the title toolbar.

   The relevant subpage is opened.

8. To enable write access for this business role, choose *Unrestricted* from the drop-down list in the *Write, Read, Value Help* tab.

9. Navigate back to the main page of your new business role.

10. Open the *Assigned Business Users* tab and add the relevant users using the *Search* field.

11. *Save* the changes from the bottom.

Results

The business role is created and added. The assigned users now have access to the `Custom Code Migration` app.

2.3.1.4.2 Maintain Communication Arrangements

A communication arrangement describes a communication scenario with a remote system during configuration time. It provides the necessary metadata for service configuration.

Prerequisites

To create these communication arrangements, you need the `Administrator` and the `Project Manager – IT` role.

`SAP_COM_0464` can only be created when communication arrangements `SAP_COM_0200` and `SAP_COM_0276` have already been created.

Context

You want to enable communication from your ABAP environment to your on-premise systems using Remote Function Calls (RFC).

For more information, see Integrating On-Premise Systems

Procedure

To do this, you need to create the SAP Cloud Connector Integration (`SAP_COM_0200`), SAP CP CF Destination Service Integration (`SAP_COM_0276`), and SAP Custom Code Migration Integration (`SAP_COM_0464` communication arrangements in your ABAP environment:

1. In the *SAP Cloud Platform Cockpit*, create a destination to your on-premise system in the service instance.
2. Log on to the *SAP Fiori launchpad* of your ABAP environment.
3. In the *Communication Management* section, select the *Communication Arrangement* tile.
4. Create a new communication arrangement using the `SAP_COM_0464` scenario. To do this, choose *New*.

   If not yet available or defined, create a new communication system for the communication arrangement to define an endpoint for your checked system.

   a. In the *New Communication System* dialog, enter the *System ID* and *System Name* of the checked system.
   b. Choose *Create*.
   c. In the *General* tab, turn on the slider for the *Destination Service*.
   d. Select the service instance that is defined in communication arrangement `SAP_COM_0276` as *Instance*.
   e. Enter the corresponding destination to your on-premise system that you have defined in the service instance in your *SAP Cloud Platform Cockpit* as *Name*.

   > **i Note**
   >
   > You have to enter exact the same *Name* of the destination as you have created it. Otherwise, the connection cannot be established.

   f. Confirm with *Save*.
5. Select the communication system which you have created for your communication arrangement and confirm with *Save*.

Results

Now, you can use the communication arrangement as *Destination* in the `Custom Code Migration` app to establish the connection to your on-premise system.

2.3.1.5 Installing Remote Stubs

During ATC execution, the central check system accesses remotely the systems in your landscape through so-called remote stubs using RFC connection. So, remote stubs serve as interface between the central check system and on-premise systems. They return a model from custom code which needs to be checked.

Prerequisites

In the on-premise system, implement SAP Note 2599695 (Custom Code Migration Fiori App: Remote Stubs for the Checked System) in advance.

Context

You want to analyze your custom code in your on-premise system using the `Custom Code Migration` app.

Procedure

To do this, you need to install several remote stubs in the checked system as follows:

1. In the on-premise system, perform transaction `SNOTE`.
2. Download SAP Note 2599695 .

   The *Note Assistant: Note Download* dialog is opened.
3. To implement the remote stubs, select and confirm the displayed SAP Note.

Results

The `Custom Code Migration` app now has access to the on-premise system. You can now start the custom code analysis.

2.3.2 Custom Code Analysis with SAP Fiori App (Recommended)

The `Custom Code Migration` app enables you to scope and analyze custom code that needs to be migrated from an SAP Business Suite system to SAP S/4HANA 1909.

2.3.2.1 Creating a Custom Code Migration Project

A custom code migration project defines the scope of custom code which you want to analyze and migrate to SAP S/4HANA.

Procedure

1. Start the tile *Custom Code Migration* from the SAP Fiori launchpad.
2. Choose **+** to create a new custom code migration project.
3. Enter a *Project Description* for your new project and the *RFC Destination* to the SAP system which contains your custom code.
4. Define the *Target Release* of the back-end and the *Material Number Length*.

   > **i** Note
   >
   > Make sure, that the checkboxes for *Extended Amount Length*, *Extended Stock/Requirement Segment* and *Extended Season/Theme/Collection* are enabled, if you use them.

5. To define the *Destination*, choose the object provider.
6. Confirm your input with *Save*.

2.3.2.2 Defining the Scope of Your Custom Code Migration Project

The Custom Code Migration app supports you with identifying your unused custom code based on your collected usage data. With the scope of a custom code migration project, you can specify which of your custom code needs to be converted to SAP S/4HANA.

To define the scope of your custom code migration project, you can either use your usage data collected with transaction SUSG or your usage data collected in your Solution Manager (this requires Solution Manager 7.2 support package 8 or higher).

> **i** Note
>
> If you didn't aggregate your usage data so far, you should activate the aggregation of your usage data in transaction `SUSG` in your production system now. To do so, start transaction `SUSG` in your production system and choose *Activate* to activate the aggregation of your usage data.

> Ideally, the usage data you add to your custom code migration project should cover at least one year of usage information, so that also usage data of quarter and year ending functionality is considered.

2.3.2.2.1   Adding Usage Data to your Custom Code Migration Project

To define the scope of your custom code migration project, you can use your usage data collected with transaction `SUSG`.

Procedure

1. Start transaction `SUSG` in your production system.

   The *Usage Data: Aggregation State* view is opened.
2. Choose *Create Snapshot* to create a snapshot of your usage data.
3. Recommended: Transfer your usage data from your production system to your checked system.
   a. Choose *Manage Snapshots*.
   b. Choose *Download to File* to export your snapshot as a file.
   c. Choose *Manage Snapshots* in transaction `SUSG` in your checked system.
   d. Choose *Upload from File* to import your usage data from a file.
4. Add the snapshot of your usage data to your custom code migration project in the Custom Code Migration app.
   a. Choose *Edit*.
   b. In the *Usage Data* frame, choose ✚.
   c. Choose ⬓ beneath the *Usage Description* field.
   d. Search for your usage data you created as a snapshot.

   > i Note
   >
   > When you transferred your usage data to your checked system, your usage data is displayed automatically in the *Items* overview.

   e. Select your usage data to add it to your custom code migration project.

   > i Note
   >
   > When you add usage data to your custom code migration project, a default scope is calculated. The default scope includes all used objects, all objects which are statically referenced by the used objects, and objects for which no usage data is available such as database tables or data elements.

5. Choose *Scope* to display the scope information.

   Here you can see, how many objects are in and out of scope.

> **→ Tip**
>
> Choose *View By* to display detailed scope information or your scope sorted by object name or object type.

2.3.2.2.2 Changing the Scope of Your Custom Code Migration Project

In the Custom Code Migration app, you can change the scope on the following levels:

- **Request Entry Points**
  Request entry points are all applications or services which are used in your system like programs, transactions, URLs, etc. For each request entry point, the used objects are recorded.
  In the initial scope, all used request entry points are added to the scope. You can remove request entry points from the scope by processing the following steps:
  1. Choose *Scope*.
  2. Choose *Scope Request Entry Point*.
  3. Select the request entry point you want to remove from the scope.
  4. Choose *Remove from Scope*.
- **Packages**
  You can add or remove whole packages to/from the scope by processing the following steps:
  1. Choose *Scope*.
  2. Choose *Scope Packages*.
  3. Select the package you want to add to or remove from the scope.
  4. Choose *Add to Scope* or *Remove from Scope*.
- **Objects**
  You can add or remove single objects to/from the scope by processing the following steps:
  1. Choose *Scope*.
  2. Choose *Scope Packages*.
  3. Navigate to the package of the objects.
  4. Select the object you want to add to or remove from the scope either in the list "Not Scoped Objects" or in the list "Scoped Objects".
  5. Choose *Add to Scope* or *Remove from Scope*.

When is an object in scope or out of scope?

Objects are in scope, if …

- … they are added to the scope explicitly (by object),
- … their package is added to scope explicitly,
- … they are used by at least one request entry point which is in scope,
- … they are referenced by any other object which is in scope,
- or their object type has no usage data.

Objects are out of scope, if …

- … they are removed from scope explicitly (by object),
- or they are not used by any request entry point.

Example

When do you need to change the scope of your custom code migration project?

- Scenario 1
  A transaction which is used in your Business Suite system shall not be used any longer, because you want to change your business processed in SAP S/4HANA.
  Choose the filter *Scope by Request Entry Points* for your transaction and remove the request entry point from your scope.
- Scenario 2
  A new functionality is in development but has not been used in your production system yet. Therefore, no usage data has been collected for this application and it has not been added to your scope automatically. Choose the filter *Scope by Package* for all packages containing your developments for the new functionality and add them to your scope.

2.3.2.2.3   Removing Unused Custom Code During System Conversion

When you have specified your scope in your custom code migration project, you can create a transport request in the checked system. This transport request contains all objects which will not be migrated during the system conversion.

Procedure

Choose *Save Scope as Transport Request* to create the transport request.

The transport request is also shown in the field *Request/Task* in your custom code migration project.

Next Steps

For more information, see chapter "Integration of Customer Transports" in the corresponding upgrade guide for SAP HANA DB Software Update Manager 2.0.

2.3.2.3 Analyzing the Findings

When you create a custom code migration project, the SAP S/4HANA custom code checks are performed automatically. The Custom Code Migration app then gives you an analytical representation of SAP S/4HANA custom code check findings.

Procedure

1. Choose *Analysis* to get an overview of the results, as soon as the first analysis has been finished.
2. Choose *Analyze Findings* to get a detailed analysis.
3. In the *Custom Code Migration - Analysis - Findings* view, you can specify various filters, for example:
   - *Simplification Item Category*: The simplification item category specifies whether findings are related to functionality which is not available any more or has been changed in an incompatible way.
   - *SAP Note Number*: Lets you analyze findings of a specific Simplification Item (for example field length extension of material number)
   - *Scope Information*: Lets you filter the findings to show only findings for objects that are in scope.
   - *Quick Fix Availability*: Lets you analyze which findings can be solved by a Quick Fix and which findings have to be solved manually.
4. Choose ⚙ to change the settings for the chart.

   To show the Quick Fix availability per SAP Note title, you must specify the following settings for the chart:
   - SAP Note title (as Category)
   - Findings (as Axis 1)
   - Quick Fix Availability (as Series)

Results

You can now continue with chapter Custom Code Adaptation After System Conversion [page 53].

3 Custom Code Adaptation After System Conversion

After Software Update Manager (SUM) has performed the technical conversion, you can start adapting your custom code.

3.1 Running Transactions SPDD, SPAU, and SPAU_ENH

SAP provides the adjustment tools `SPDD`, `SPAU`, and `SPAU_ENH`, which enable you to reimplement any modifications related to ABAP Dictionary objects and development objects (such as programs, function modules, screens, interfaces, and documentation) in system upgrades.

To adapt these modifications, see https://help.sap.com/viewer/p/SAP_S4HANA_ON-PREMISE ▌▷ *Product Assistance* ❯ *English* ❯ *Enterprise Technology* ❯ *ABAP Platform* ❯ *Developing on the ABAP Platform* ❯ *Development Concepts and Tools* ❯ *Application Development on AS ABAP* ❯ *Customer-Specific ABAP Development* ❯ *Changing the SAP Standard (BC)* ❯ *Installing Upgrades, Support Packages, and Enhancement Packages* ▌ and SAP Note 2168190🖝 for more information.

> **i Note**
>
> Transaction `SPDD` is performed during the system conversion whereas all other adaptations take place after the conversion.

3.2 Running Local SAP S/4HANA Checks

To check for ATC findings in the converted system, you need to configure a local ATC run series that checks the requested development objects.

3.2.1 Importing the Simplification Database

To detect custom code which needs to be adapted to SAP S/4HANA, SAP provides tools based on the Simplification Database.

Procedure

Import the Simplification Database. For more information, see Importing the Simplification Database [page 15].

3.2.2 Configuring Local ATC Run Series

Procedure

1. Log on to your converted system.
2. Run transaction `ATC`.

   The *ABAP Test Cockpit Overview* screen appears.

3. In the navigation pane, expand the nodes ▶ *ATC Administration* ❯ *Runs* ❯ and double-click the *Schedule Runs* entry.

   The screen that appears displays a list of existing run series.

4. Choose the *Create* button in the toolbar.
5. Enter a name for the new series and confirm.
6. Specify the following entries for the new run series:

Field	Description
`<Description>`	Enter a short description for the ATC run series you want to configure.
	→ Tip
	You can use the template provided by the *Description* field. When you execute the run series, the built-in variables are filled with data for the system, the day of the week, the calendar week, and the year. However, you can also add further texts to these variables, rearrange them, or replace the built-in variables with your own text.
`<Check Variant>`	Enter the check variant `S4HANA_READINESS`.
	i Note
	This check variant provides checks for SAP S/4HANA readiness (for example, searches for DB operations or field length extensions).
	→ Tip
	If you want to use a material number with less than 40 characters in SAP S/4HANA, you can copy the check variant `S4HANA_READINESS` and specify your desired length of the material number for the check in transaction `SCI` for the *S/4HANA Field Length Extension*. Here, also make sure that the *Extended Amount Length*, *Extended Stock/Requirement Segment* or *Extended Season/Theme/Collection* are enabled, if you use them.
	Note that you will get more findings to be adapted in your custom code if you use material numbers with more than 18 characters.

Field	Description
`<Objects to Check>`	Specify the set of development objects you want to check.
	On the *Checkable Namespaces* tab, the option *By Query* lets you specify objects by name (such as by package name), by transport layer, or by component. The option *By Object Set* lets you specify an object set that you have defined in Code Inspector in the checked system.
	→ Tip You can use the value help to choose the packages or the object set in the remote system.
	On the *Modified Objects* tab, you can specify the modified source code objects you want to check.

7. Save the configuration.

3.2.3 Scheduling Local ATC Run Series

Procedure

1. In the list of run series, select the run series in question and click the *Schedule* button in the toolbar.
2. In the dialog that appears, choose *Set to Active Result*.

3. Adapt the settings for the *Life Span* of the series and for *Execution* to your needs. Otherwise, leave these settings unchanged.

> **i Note**
>
> The *Life Span* defines how long (in days) the ATC result is kept in the system. After this, the ATC result is automatically deleted in the system.

4. Choose *Execute in Background* (F9) to schedule the run series in regular time intervals.

> → Recommendation
>
> SAP recommends to schedule the local ATC run series in regular time intervals since new ATC findings
> could emerge during custom code adaptation.

3.3 Adapting Custom Code in ADT

After a local ATC run series has checked the requested development objects, you can start adapting the
custom code by fixing the ATC findings of the ATC run result with ABAP Development Tools (ADT).
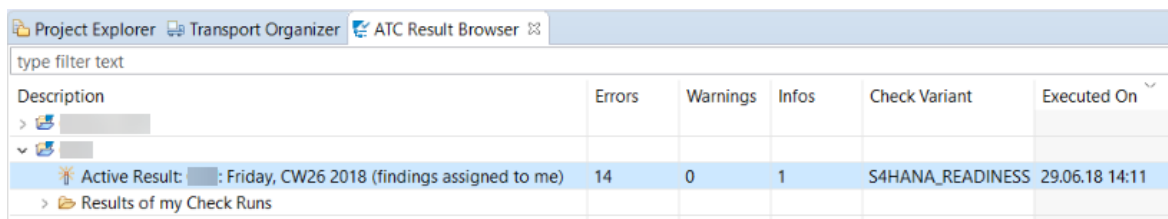
3.3.1 Displaying Active Results

Prerequisites

You installed the latest version of ABAP Development Tools (ADT) in Eclipse. In Eclipse, you created an *ABAP Project* which is connected to your converted SAP S/4HANA system. For more information, see the *Help Contents* in ADT.

Procedure

1. In ADT, log on to your converted system.
2. Open the *ATC Result Browser* view and select your converted system.

3. Select the *Active Result* to display the list of ATC findings.

> i Note
>
> By default, all ATC findings are filtered by your user name and only ATC findings belonging to your user
> are displayed. To display the ATC findings of all users, open the context menu of the *ABAP Project* in the
> *ATC Result Browser* view, choose *Change User Filter*, and enter * as the user name.

4. Choose ▷ *Group By* ❯ *Check* ☐ to display the ATC findings sorted by the different checks of the check
   variant `S4HANA_READINESS`.

> → Tip
>
> Choose *Configure Columns* to add or remove the columns you want to display or hide in the details of the result. Subsequently, you can filter the active result according to all values in the visible columns.

5. Select the check group *S/4HANA: Search for S/4 related syntax errors* and choose *Recheck* in the context menu.

3.3.2 Optional: Changing the Contact Person of ATC Findings

If you want to assign ATC findings of an ATC run result to a certain developer, you can do that by changing the contact person of the ATC findings.

Prerequisites

You need the authorization object `S_Q_GOVERN` (`ACTVT` = 03 and `ATC_OTYPGO` = 02) to change the contact person.

In addition, you need one of the following authorization objects:

- `S_Q_GOVERN` (`ACTVT` = 01 and `ATC_OTYPGO` = 01) or
- `S_DEVELOP` (`ACTVT` = 02)

> i Note
>
> You can change the contact person only for local check runs.

Procedure

1. In the *ATC Result Browser* view, select the *Active Result* to display the list of ATC findings.
2. Select all ATC findings in question and choose *Change Contact Person* in the context menu.
3. In the *Change Contact Person* view in the field *New Contact Person*, enter the user name of the developer to whom you want to assign the ATC findings.

   The ATC findings are now assigned to the determined developer.
4. Select the *Active Result* and choose *Refresh* in the context menu.

3.3.3 Fixing ATC Findings

Procedure

1. Open the *ATC Problems* view.

   The view shows a *Worklist* of ATC findings resulting from the recheck on the check group *S/4HANA: Search for S/4 related syntax errors*.
2. Select an ATC finding to display further information about what and where the statements with problems are and what you can do to fix the ATC finding.
3. Double-click an ATC finding to jump to the source code at the position where the ATC found a statement with a problem.
4. Fix each ATC finding in the *Worklist* as described in the *Details* view and the assigned SAP Note.

   > i Note
   >
   > You can also fix ATC findings by using Quick Fixes. See Applying Quick Fixes [page 60] for more information.

3.3.3.1 Applying Quick Fixes

You can fix certain ATC findings with Quick Fixes. These Quick Fixes provide functions that enable you to resolve errors and warnings without adapting your source code manually.

Context

In the context of the custom code migration, Quick Fixes are available for the following checks and its corresponding simplification items:

Check	Simplification Item	Referenced Object
S/4HANA: Field length extensions	2215852 - Material Number Field Length Extension: Code Adaptions	for example MATNR
	2610650 - Amount Field Length Extension: Code Adaptations	
S/4HANA: Search for database operations	2198647 - S/4 HANA: Data Model Changes in SD	VBFA, VBUK, VBUP
	2220005 - S/4 HANA: Data Model Changes in Pricing and Condition Technique	KONV
	2431747 - General Ledger: Incompatible changes in S/4HANA compared to classic ERP releases	BSEG
S/4HANA: Search for usages of simplified objects	2198647 - S/4 HANA: Data Model Changes in SD	VBTYP
S/4HANA: Search for S/4 related syntax errors	2198647 - S/4 HANA: Data Model Changes in SD	VBTYP
Search problematic statements for result of SELECT/OPEN CURSOR without ORDER BY		

> **i Note**
>
> Please note, that not all ATC findings resulting from these checks can be fixed with Quick Fixes.
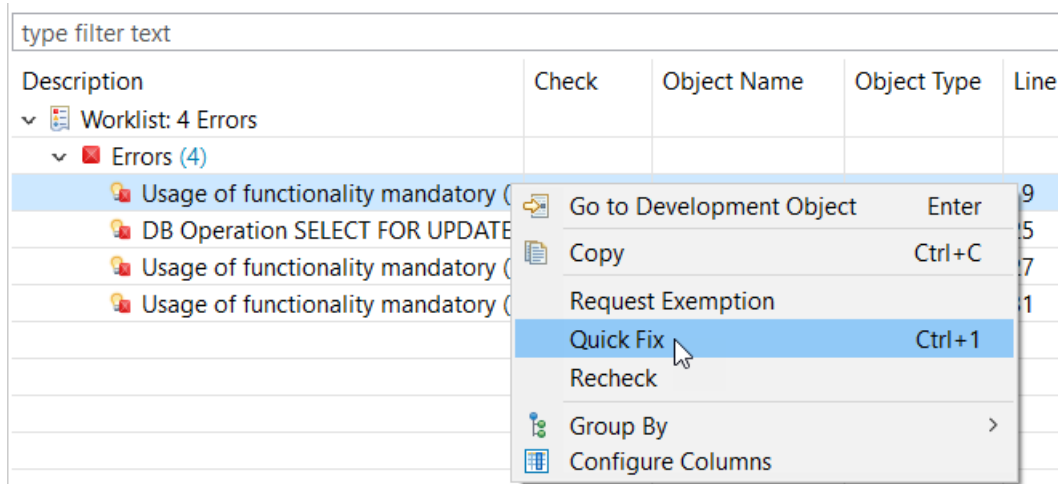
Procedure

> **i Note**
>
> ATC findings that can be fixed with a Quick Fix are displayed with a lightbulb icon 💡.

1. Select an ATC finding with a lightbulb icon in the *ATC Problems* view.

2. Right-click the ATC finding and choose *Quick Fix* (`Ctrl` + `1`).

The *Quick Fix* view opens.

3. Select the displayed Quick Fix and choose *Finish*.

> **→ Recommendation**
>
> If there is more than one Quick Fix available for an ATC finding, we recommend that you select the first Quick Fix displayed.
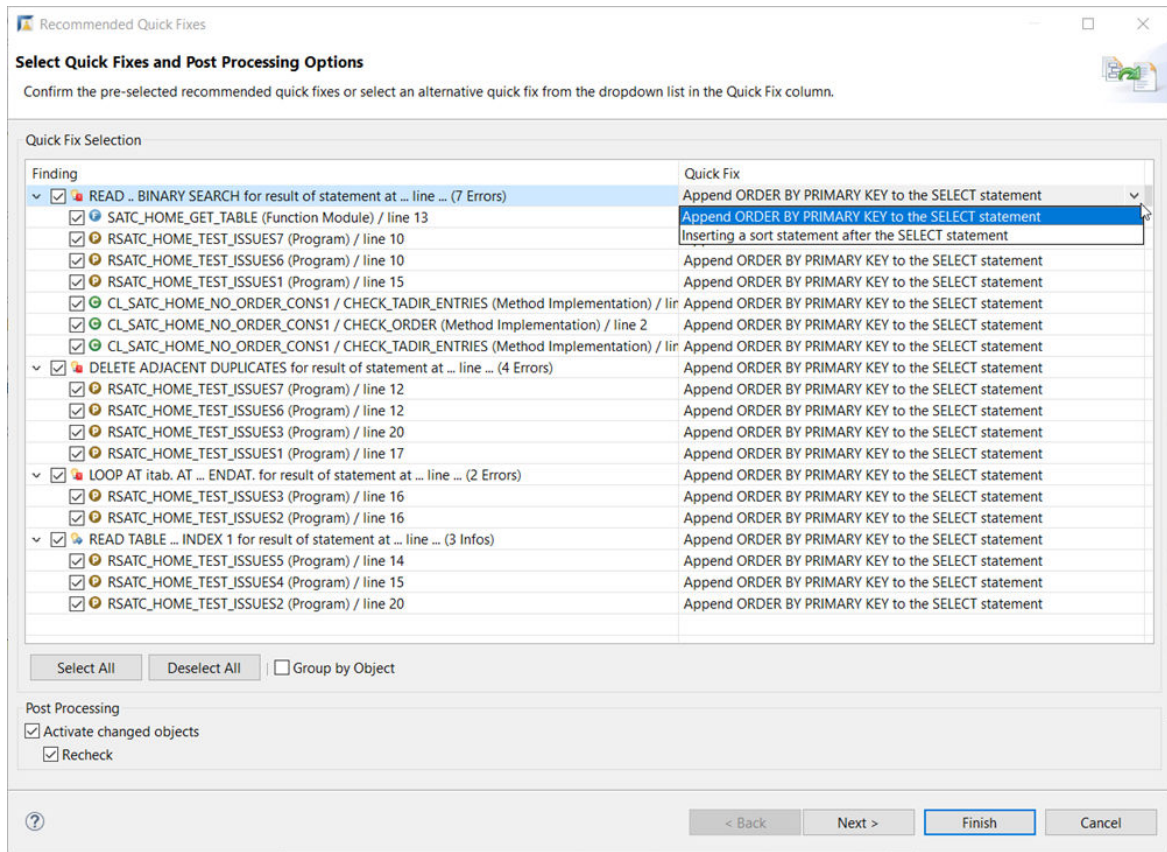
3.3.3.1.1 Applying Recommended Quick Fixes for Multiple ATC Findings

You can also fix multiple ATC findings at once with the *Recommended Quick Fixes* wizard.

Procedure

1. Select multiple ATC findings in the *ATC Problems* view.
2. Open the context menu and choose *Recommended Quick Fixes...*.

   The *Recommended Quick Fixes* wizard is opened.

3. In the *Finding* column in the *Quick Fix Selection* frame, select the ATC findings and the affected objects. By default, all ATC findings and their affected objects are selected.

> → Tip
>
> If you want to display all affected objects and their respective ATC findings, choose *Group by Object*.

4. In the *Quick Fix* column in the *Quick Fix Selection* frame, the recommended Quick Fixes for the ATC findings are displayed by default. Select a Quick Fix to open a dropdown list with alternative Quick Fixes.

> → Recommendation
>
> We recommend applying the Quick Fixes displayed by default.

5. In the *Post Processing* frame, you can specify that the changed objects are activated after you apply the Quick Fixes. If this option is selected, you can specify that the selected ATC findings are rechecked after you finish the wizard.

> i Note
>
> If you do not select any post processing options, the initial ATC result is displayed after you finish the wizard. In this case, you have to activate and recheck the ATC findings manually.

6. Choose *Next*.
7. Select a transport request if required.
8. Choose *Next*.

9. Review the changes. Here, a comparison editor is displayed where you can review the refactored source code. The code line where the source code has been refactored is highlighted.
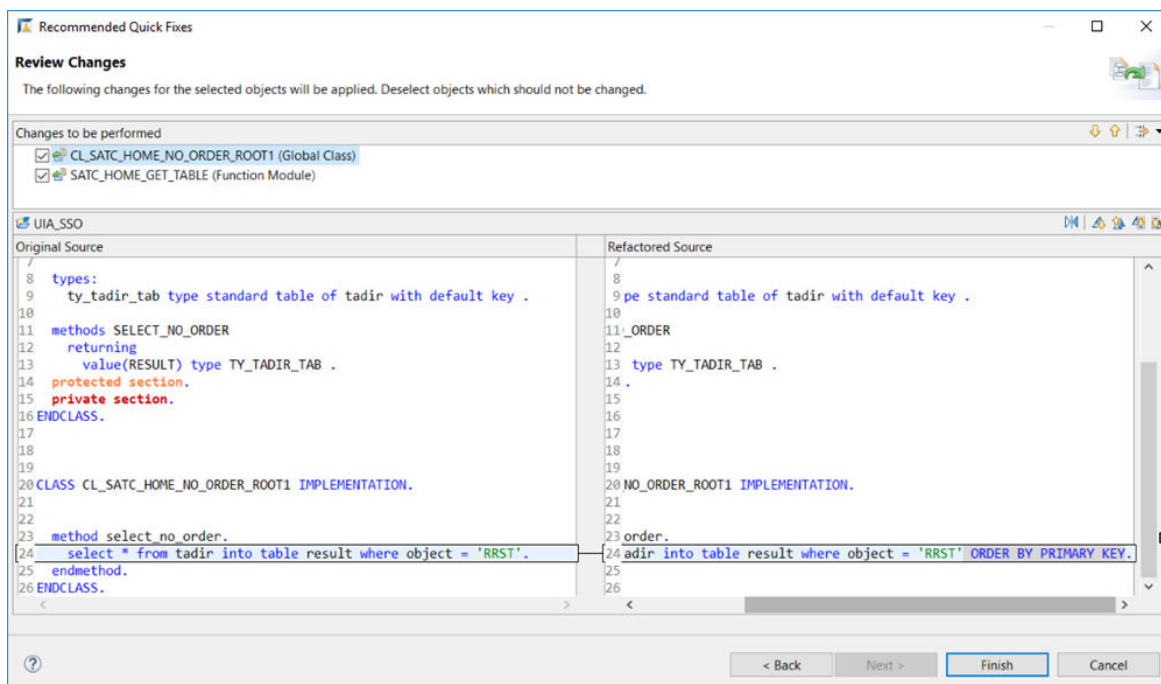
In this example, an `order by primary key` statement was added to the source code to fix the ATC finding.

10. Choose *Finish*.

Results

You applied Quick Fixes for multiple ATC findings at once.

3.3.3.2 Using Pseudo Comments

Pseudo comments are one way of suppressing ATC findings. Sometimes it is the case that certain ATC findings cannot be fixed (so-called *false positives*). In this case and especially in the context of the Custom Code

Adaptation to SAP S/4HANA, we recommend that you use pseudo comments to suppress the specific ATC finding.

Procedure

> **i Note**
>
> If SAP S/4HANA-related pseudo comments are available for an ATC finding, they can easily be applied as a Quick Fix.

Apply the Quick Fix. For more information, see Applying Quick Fixes [page 60].
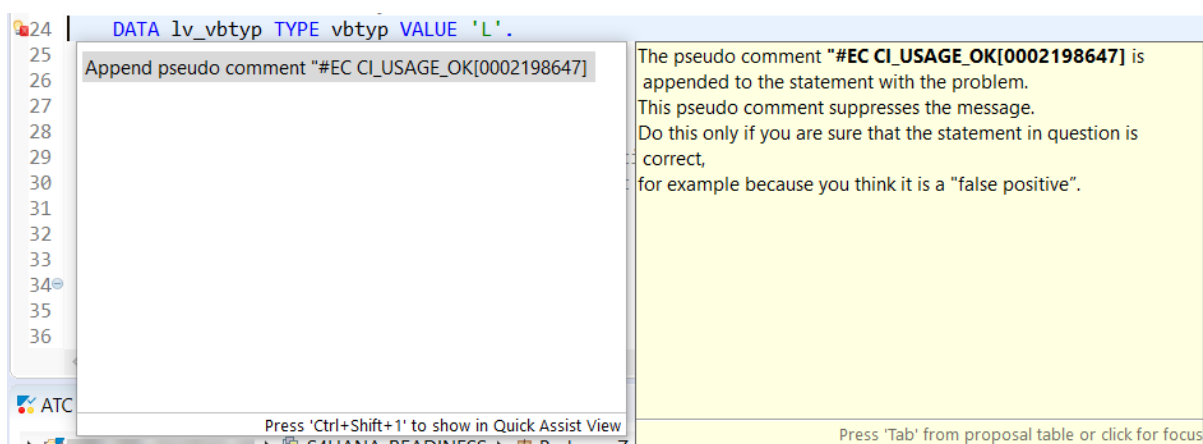
The pseudo comment is appended at the end of the code line in question.

> **→ Remember**
>
> The pseudo comment appended does not fix the ATC finding. It only suppresses the ATC finding, so it does not appear anymore after a recheck.

3.3.4 Rechecking the Active Result

Once you have fixed the ATC findings of the check group *S/4HANA: Search for S/4 related syntax errors*, you need to recheck the active result and fix all other ATC findings.

Procedure

1. Open the *ATC Result Browser* view.
2. Right-click *Active Result* and choose *Recheck*.
3. Open the *ATC Problems* view.

   The view shows a *Worklist* of ATC findings resulting from the recheck on the active result.

4. Fix the ATC findings in the *Worklist*.

5. Recheck the *Active Result* in *ATC Result Browser* view again to validate that the ATC findings are fixed.

> **i Note**
>
> Repeat the steps 4 and 5 if there are still ATC findings in the *Worklist* after the recheck.

3.4 Rerunning Local SAP S/4HANA Checks

Rerun the local ATC run series once all the ATC findings are fixed, since new ATC findings could have emerged during the adaptation of your custom code.

4 What's Next?

There are a few more things you can do after the custom code migration:

Testing Applications

Test your applications to check whether your programs run on SAP S/4HANA.

> **i Note**
>
> ATC is not able to find all potential issues (for example, dynamic coding is not covered by static code checks).

Regular ATC Checks with `S4HANA_READINESS`

Add the SAP S/4HANA checks with check variant `S4HANA_READINESS` to your regular ATC checks.

Runtime Checks

Run the Runtime Check Monitor in transaction `SRTCM` in the new productive system and activate the checks *Empty table in FOR ALL ENTRIES clause* and *Missing ORDER BY or SORT after SELECT*. Schedule the check runs on a regular basis and correct the additional findings found by the runtime check.

Performance Optimizations

Switch on SQL Monitor in the productive system. Sort the SQL Monitor results by *execution time* and optimize the top 10-20 SQLs in your custom code which affect relevant business processes. After performance optimizations reach the productive system, this process needs to be repeated 2-3 times to achieve the best results.

For more information, see ABAP SQL Monitor Implementation Guide and Best Practices

Optimizing Your Code for SAP S/4HANA

After the successful migration, you can now start to think about optimizing your business processes by using code push down techniques of SAP S/4HANA, like the Core Data Services (CDS) and ABAP-managed database procedures (AMDP) implemented by the native HANA language SQL script.

Enabling In-App Help Content Using SAP Enable Now Web Assistant

You can access in-app help from the UIs of an app that provides you context-sensitive help information for UI elements and guided tours.

To get access to the in-app help, the SAP Enable Now Web Assistant is a web-based and the embedded help system (for example, the SAP Help Portal) are required.

> ### i Note
>
> In your on-premise system, you need to configure the Web Assistant before you can benefit from it.
>
> In the context of ABAP environment, Web Assistant is enabled per default.

For further information, see:

- SAP Enable Now Web Assistant Overview for SAP S/4HANA
- SAP Enable Now Web Assistant Integration

5 Glossary

The following terms are used within the context of this Custom Code Migration guide:

Term	Definition
(ABAP) Custom Code	Describes the development objects and ABAP source code that has been developed by customers or SAP partners.
ABAP Test Cockpit (ATC)	Tool for performing static and dynamic quality checking of ABAP code and associated repository objects both remotely and locally.
ATC Finding	Message with supporting information and functionality that alerts a developer to a problem with ABAP code or some other object in ABAP Repository.
Central Check System	Backend system that provides and performs the SAP S/4HANA custom code checks for one or multiple SAP systems remotely.
Checked System	System with custom code that gets checked remotely by the Central Check System.
Conversion / migration	Describes the process for bringing custom code from an existing Enterprise Resource Planning (ERP) system to an SAP S/4HANA system.
Simplification Database	Database table in the SAP S/4HANA system that contains all the Simplification Items referring to SAP objects simplified in SAP S/4HANA.
Simplification Item	Describes the adaptation of an SAP product that has a technical impact on custom code.
Transformation	Describes the process for bringing custom code from an existing on-premise Enterprise Resource Planning (ERP) system to SAP Cloud Platform ABAP Environment.

Important Disclaimers and Legal Information

Hyperlinks

Some links are classified by an icon and/or a mouseover text. These links provide additional information.
About the icons:

- Links with the icon 🡕 : You are entering a Web site that is not hosted by SAP. By using such links, you agree (unless expressly stated otherwise in your agreements with SAP) to this:

    - The content of the linked-to site is not SAP documentation. You may not infer any product claims against SAP based on this information.
    - SAP does not agree or disagree with the content on the linked-to site, nor does SAP warrant the availability and correctness. SAP shall not be liable for any damages caused by the use of such content unless damages have been caused by SAP's gross negligence or willful misconduct.

- Links with the icon 🔗: You are leaving the documentation for that particular SAP product or service and are entering a SAP-hosted Web site. By using such links, you agree that (unless expressly stated otherwise in your agreements with SAP) you may not infer any product claims against SAP based on this information.

Beta and Other Experimental Features

Experimental features are not part of the officially delivered scope that SAP guarantees for future releases. This means that experimental features may be changed by SAP at any time for any reason without notice. Experimental features are not for productive use. You may not demonstrate, test, examine, evaluate or otherwise use the experimental features in a live operating environment or with data that has not been sufficiently backed up.
The purpose of experimental features is to get feedback early on, allowing customers and partners to influence the future product accordingly. By providing your feedback (e.g. in the SAP Community), you accept that intellectual property rights of the contributions or derivative works shall remain the exclusive property of SAP.

Example Code

Any software coding and/or code snippets are examples. They are not for productive use. The example code is only intended to better explain and visualize the syntax and phrasing rules. SAP does not warrant the correctness and completeness of the example code. SAP shall not be liable for errors or damages caused by the use of example code unless damages have been caused by SAP's gross negligence or willful misconduct.

Gender-Related Language

We try not to use gender-specific word forms and formulations. As appropriate for context and readability, SAP may use masculine word forms to refer to all genders.

Videos Hosted on External Platforms

Some videos may point to third-party video hosting platforms. SAP cannot guarantee the future availability of videos stored on these platforms. Furthermore, any advertisements or other content hosted on these platforms (for example, suggested videos or by navigating to other videos hosted on the same site), are not within the control or responsibility of SAP.

THE BEST RUN SAP