

# Manual Installation of SAP Extractor



## Contents:

- [Manual SAPEX Installation](#)
  - [Local vs Remote Kiuwan Analysis](#)
  - [SAPEX Installation Files](#)
  - [Common Installation steps for Local and Remote scenarios](#)
    - [1.Create ZKW\\_SAPEX package](#)
    - [2.Install support classes](#)
    - [3.Install source code extraction programs](#)
    - [4.Create OS commands \(ZKW\\_MKDIR and ZKW\\_RMDIR\)](#)
      - [ZKW\\_MKDIR](#)
      - [ZKW\\_RMDIR](#)
  - [Installation steps for Local scenario](#)
    - [1. Install Programs to execute Local Analysis](#)
    - [2. Create OS command to launch the analysis \(ZKW\\_KLA\\_AGENT\)](#)
  - [Installation steps for Remote scenario](#)
    - [Configure remote scripts](#)
    - [1. Install RFC-enabled function modules](#)
    - [2. Install SAP JCo Connector library](#)
      - [Full details for installing SAP JCo library](#)
    - [3.Configure SAPEX scripts in Kiuwan Local Analyzer](#)
  - [Test SAPEX installation](#)

## Related pages:

## Manual SAPEX Installation

### Local vs Remote Kiuwan Analysis

To analyze ABAP code with the Kiuwan Solutions, the source code and information from the SAP system need to be exported before they can be analyzed.



Once the source code is exported, the Kiuwan Solutions let you implement two different approaches on the **location where the ABAP code is analyzed**.

- To execute Kiuwan analyses within the SAP server (**local**), or
- To execute analyses from an external server (**remote**)

Depending on your approach, the installation takes different steps.

We will use the terms **local** and **remote** for specific installation steps, and **both** for common (mandatory) steps.

### SAPEX Installation Files



SAPEX installation files are contained in `sapex_abap_code.zip`, located in the `$(AGENT_HOME)/resources/abap` directory of your Kiuwan Local Analyzer installation.

Please visit [Kiuwan Local Analyzer](#) for further help on Kiuwan Local Analyzer.

The following table lists contents of `sapex_abap_code.zip`

| File             | Behaviour          | Notes                                |
|------------------|--------------------|--------------------------------------|
| <b>Programs:</b> |                    |                                      |
| ZKW_SAPEX_CODE   | Export source code | Exports to local or remote directory |

|  |   |   |
|--|---|---|
| ZKW_SAPEX_ME<br>TADATA                       | Export SAP information  | Exports to local or remote directory                                  |
| ZKW_ANALYSIS                                 | Analyzes source code (after optional previous export) in SAP system |   |
| <b>RFC-enabled<br/>Function<br/>modules:</b> |   |   |
| ZKW_SAPEX_CO<br>DE_RPC                       | Export source code  | Needed by <code>sapexCode.xml</code> remote script                    |
| ZKW_SAPEX_ME<br>TADATA_RPC                   | Export metadata   | Needed by <code>sapexMetadata.xml</code> remote script                |
| ZKW_SAPEX_LO<br>ADFILE_RPC                   | For downloading an exported file                                    | Needed by <code>sapexCode.xml</code> remote script                    |
| ZKW_SAPEX_RM<br>DIR_RPC                      | For removing directories where source code is exported              |   |
| ZKW_SAPEX_RM<br>DIR_CHECK                    | Security check for RMDIR calls                                      |   |
| <b>Classes:</b>                              |   |   |
| ZKW_CL_*,<br>ZKW_CX_*                        | Support classes   | Add them using class builder or Eclipse ADT                           |
| ZCL_IM_KW_BAD<br>I_REQ_CHECK                 | Implementation BAdI for CTS_REQUEST_CHECK                           | Sample BAdI for automated audit before release of a transport request |



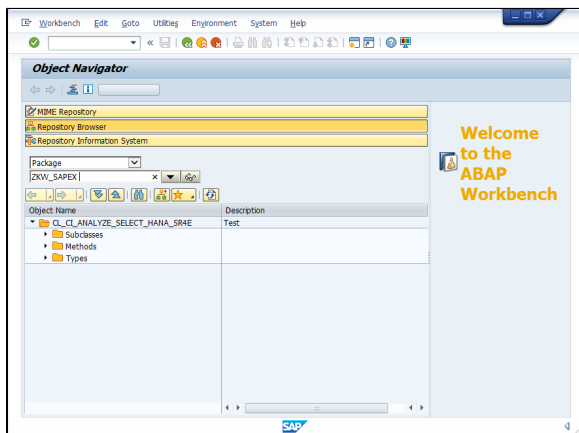
- Note 1: It is recommended to **create a package named ZKW\_SAPEX** to hold all entities created for SAPEX.
- Note 2: You may use ABAP Workbench / ABAP Editor or Eclipse-based ABAP Development Tool to create the SAPEX elements from the provided code.
- Note 3: **Remember to activate all ABAP elements installed, to enable the execution.**

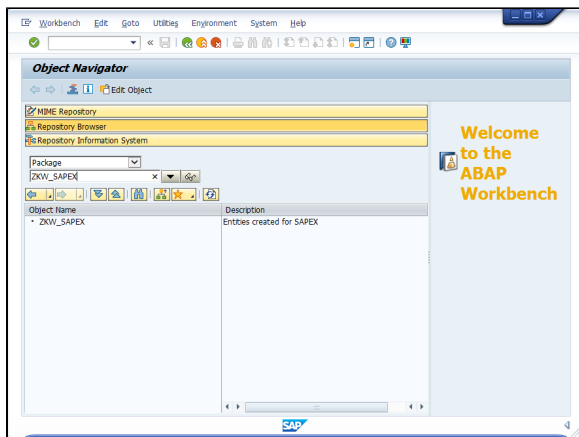
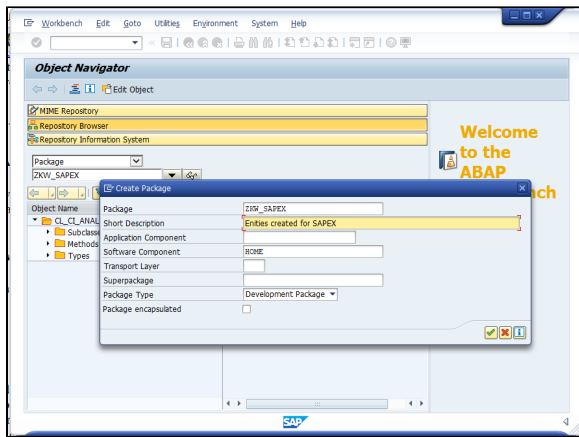
## Common Installation steps for Local and Remote scenarios

### 1.Create ZKW\_SAPEX package

It is recommended to create a package named **ZKW\_SAPEX** to hold all entities created for SAPEX.

To create the **ZKW\_SAPEX** package you can use the transaction **SE80** as follows:

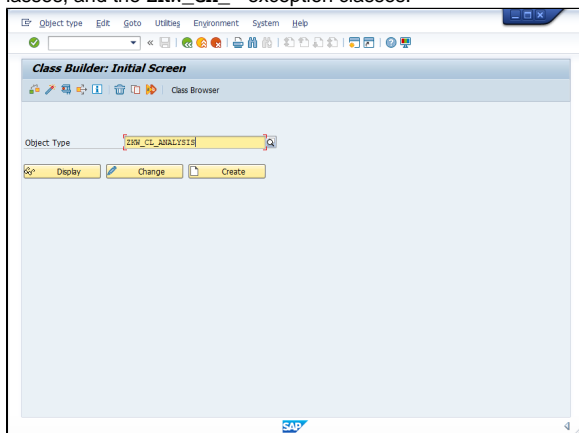


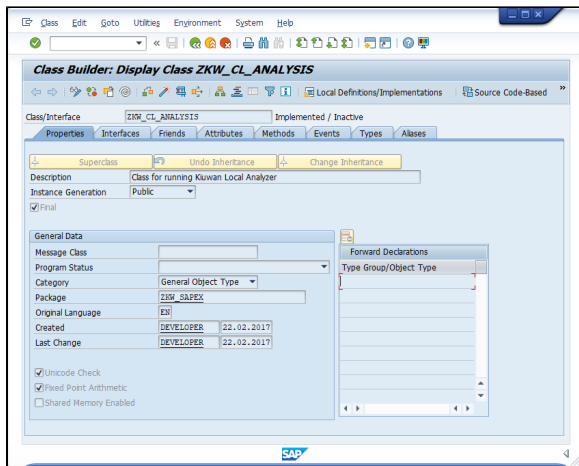


After creating the ZKW\_SAPEX package, you can follow the next steps.

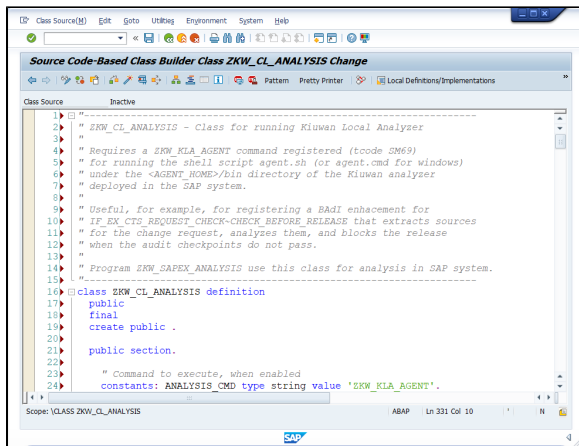
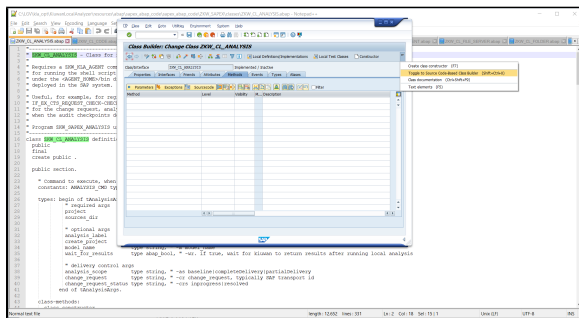
## 2.Install support classes

Using the Class Builder (transaction **SE24**) in source code mode, create and activate all the **ZKW\_CL\_\*** classes, and the **ZKW\_CX\_\*** exception classes.

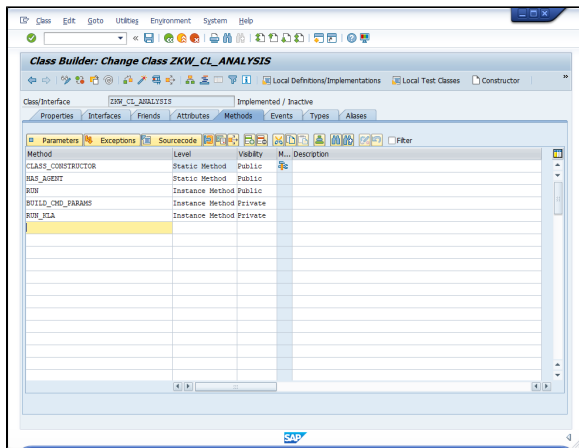




Toggle to **Source Code - Based Class Builder** and paste the content of the source file (substituting any previous content).

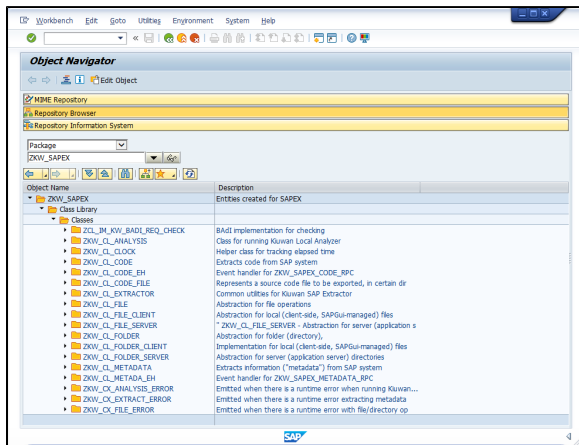


Switch back to **Form-Based Class Builder** to see the complete class.



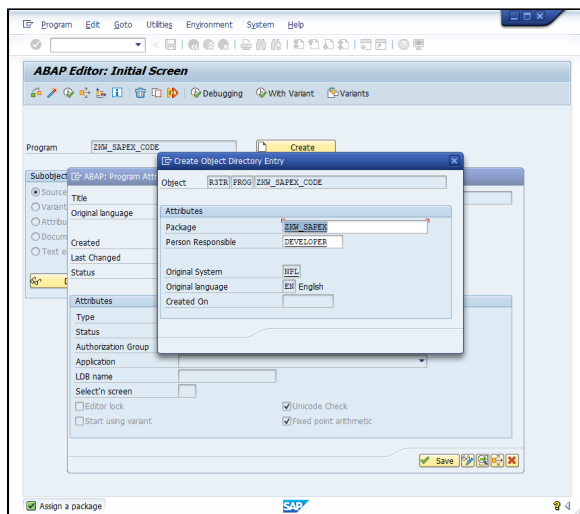
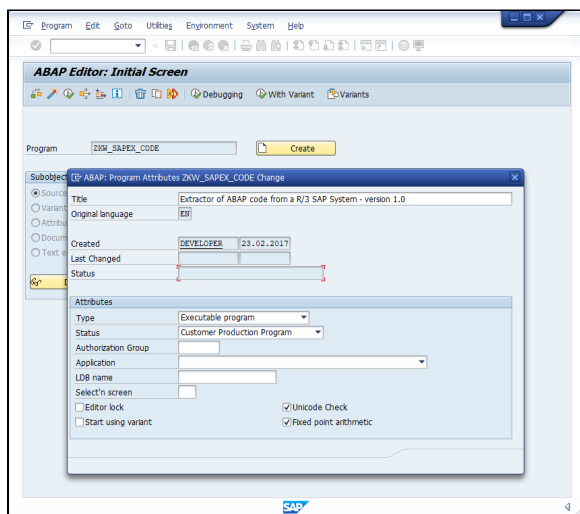
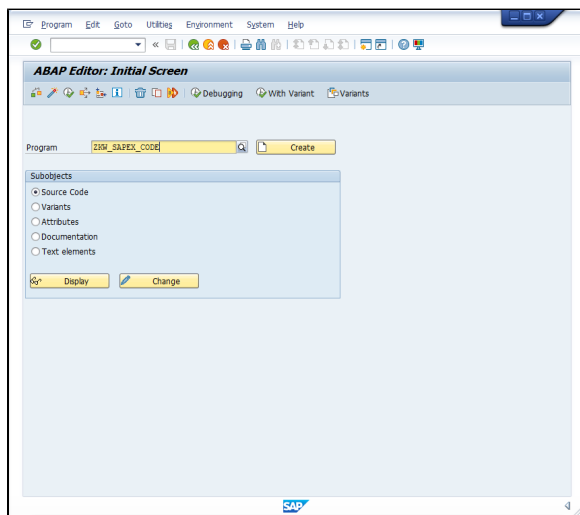
**Do not forget to **Activate** every class after creation.**

After importing all the classes you will see them under the **ZKW\_SAPEX** package:

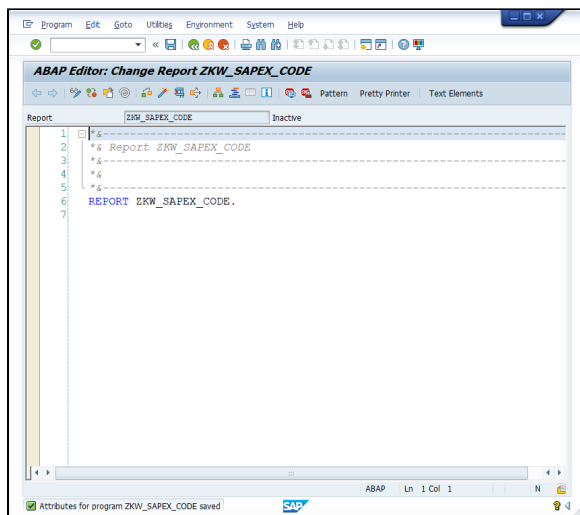


### 3.Install source code extraction programs

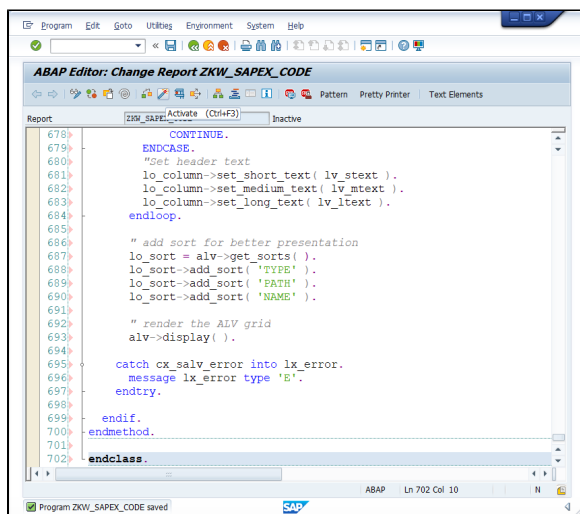
Using the ABAP Editor (transaction **SE38**), create the **ZKW\_SAPEX\_CODE** and **ZKW\_SAPEX\_METADATA** programs



Open the **Source** view.

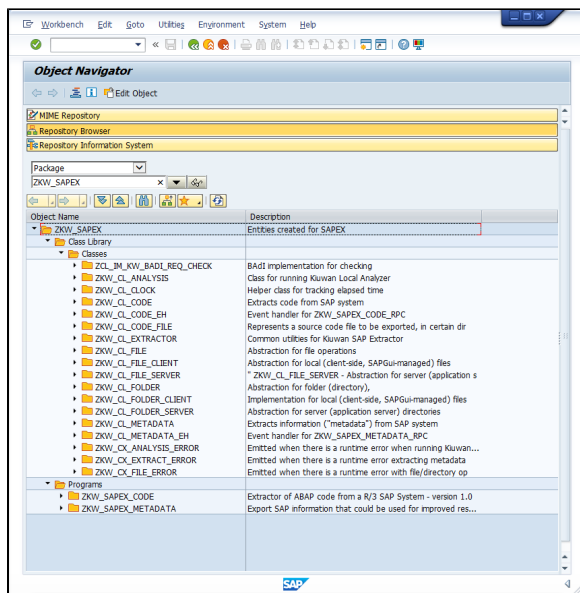


and paste the source code provided by SAPEX.



After saving, you will see all the SAPEX objects.

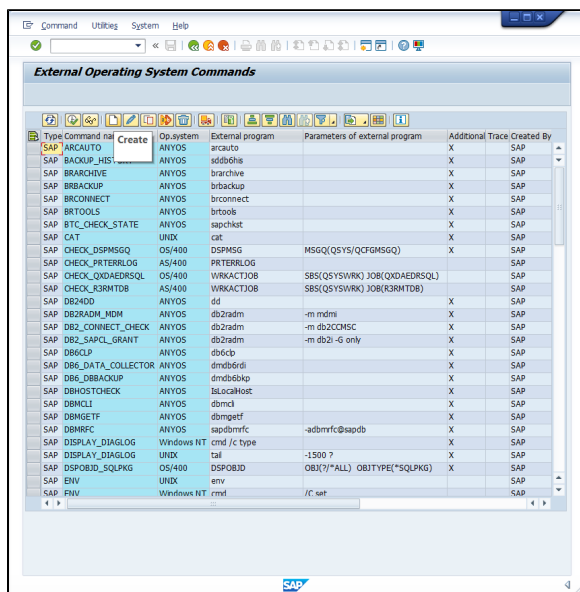
Do not forget to activate all the created objects.



## 4. Create OS commands (ZKW\_MKDIR and ZKW\_RMDIR)

Using transaction **SM69**, create the following OS commands:

- **ZKW\_MKDIR**
- **ZKW\_RMDIR**



### ZKW\_MKDIR

Used for creating directories in the SAP system:

- `mkdir -p ?` for Unix
- `cmd /c mkdir ?` for Windows

Note: The command configured must create all intermediate directories.



**Create an External Command**

Command

Command Name: ZKW\_RMDIR

Operating System: Linux

Type:

Create and Last Change

Created By:

Last Changed By:

Definition

Operating System Command: rmdir

Parameters for Operating System Command: -p ?

☒ Additional Parameters Allowed

☐ Trace

Check Module:

## ZKW\_RMDIR

Used for deleting a directory and all its contents in the SAP system:

- `rm -rf ?` for Unix
- `cmd /C rmdir /s /q ?` for Windows.

**Create an External Command**

Command

Command Name: ZKW\_RMDIR

Operating System: Linux

Type:

Create and Last Change

Created By:

Last Changed By:

Definition

Operating System Command: rm

Parameters for Operating System Command: -rf ?

☒ Additional Parameters Allowed

☐ Trace

Check Module: ZKW\_SAPEX\_RMDIR\_CHECK



To ensure that a proper `rmdir` command is selected, **register the ZKW\_SAPEX\_RMDIR\_CHECK** to avoid security issues.

This ensures that the command is one of the allowed RMDIR commands, and that the target directory is either sapex code or metadata directories, or a raw subdir under these directories.



**ZKW\_RMDIR performs a recursive deletion!**

To avoid unintended deletions, be sure the target directory does not contain directories out of the SAPEX code/metadata directories.

The result of creating the commands should be similar to the following:

| External Operating System Commands |              |                  |                          |                  |                       |     |
|------------------------------------|--------------|------------------|--------------------------|------------------|-----------------------|-----|
| Type                               | Command name | Operating system | Name of external program | Parameters       | Check module          | Adc |
| Custom...                          | ZKW_RMDIR    | Linux            | rm                       | -f ?             | ZKW_SAPEX_RMDIR_CHECK | X   |
| Custom...                          | ZKW_RMDIR    | WINDOWS NT       | cmd                      | /C rmdir /s /q ? | ZKW_SAPEX_RMDIR_CHECK | X   |
| Custom...                          | ZKW_MKDIR    | Linux            | mkdir                    | -p ?             |                       | X   |
| Custom...                          | ZKW_MKDIR    | WINDOWS NT       | cmd                      | /C mkdir ?       |                       | X   |

## Installation steps for Local scenario

### 1. Install Programs to execute Local Analysis

If you decide to use the local approach:

- Install [Kiuwan Local Analyzer](#) in the SAP system, and
- Using the ABAP Editor (transaction **SE38**), create and activate **ZKW\_ANALYSIS** program

It is recommended to use a <DIR\_HOME>/sape4k/kla directory for the Kiuwan Local Analyzer.

### 2. Create OS command to launch the analysis (ZKW\_KLA\_AGENT)

Used for launching the Kiuwan Local Analyzer script.

This OS command will be used by **ZKW\_ANALYSIS** program and **CTS\_REQUEST\_CHECK** BAdI implementation.

Use :

- <PATH\_TO\_KLA>/bin/agent.sh (Unix), or
- <PATH\_TO\_KLA>\bin\agent.cmd (Windows)

as the name of the **Operating System Command** parameter in transaction **SM69**.

| Change Command "ZKW_KLA_AGENT" for "Linux"  |  |
|---|--|
| <div> <div>Command</div> <div> <div>Command Name</div> <div>ZKW_KLA_AGENT</div> </div> <div> <div>Operating System</div> <div>Linux</div> </div> <div> <div>Type</div> <div>Customer</div> </div> </div>  |  |
| <div> <div>Create and Last Change</div> <div> <div>Created By</div> <div>DEVELOPER</div> </div> <div> <div>22.02.2017</div> <div>16:22:12</div> </div> <div> <div>Last Changed By</div> <div>DEVELOPER</div> </div> <div> <div>22.02.2017</div> <div>16:22:12</div> </div> </div>   |  |
| <div> <div>Definition</div> <div> <div>Operating System Command</div> <div>/usr/sap/NPL/D00/work/sape4k/klb/KiuwanLocalAnalyzer/bin/agent.sh</div> </div> <div> <div>Parameters for Operating System Command</div> <div></div> </div> <div> <div><input checked="" type="checkbox"/> Additional Parameters Allowed</div> <div><input type="checkbox"/> Trace</div> </div> <div> <div>Check Module</div> <div></div> </div> </div> |  |

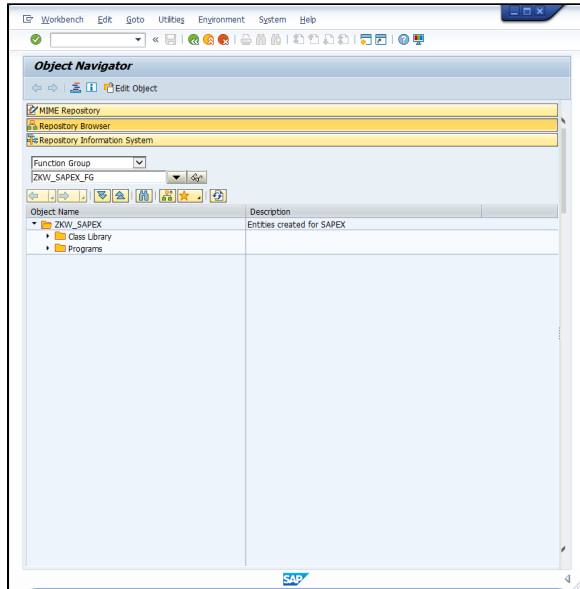
## Installation steps for Remote scenario

### Configure remote scripts

## 1. Install RFC-enabled function modules

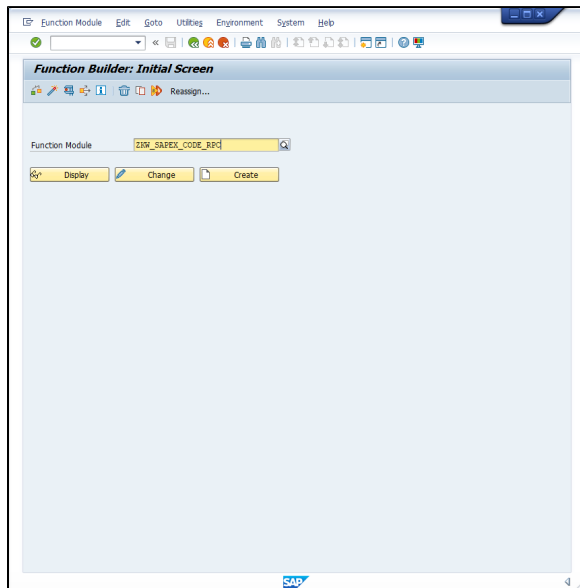
To run the SAPEX scripts from a remote host (where KLA is deployed), the `ZKW_SAPEX_*_RPC` function modules should be created using Function Builder (transaction **SE37**).

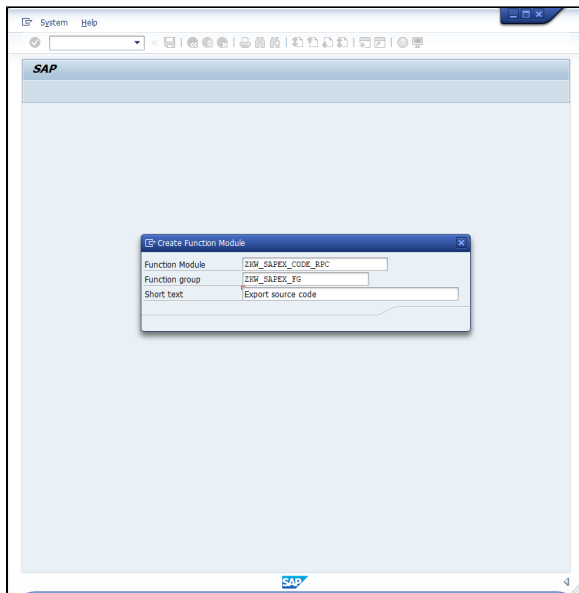
To create the `ZKW_SAPEX_FG` function group you can use transaction **SE80** as follows:  
Before creating the function modules, it is recommended to create the `ZKW_SAPEX_FG` function group.



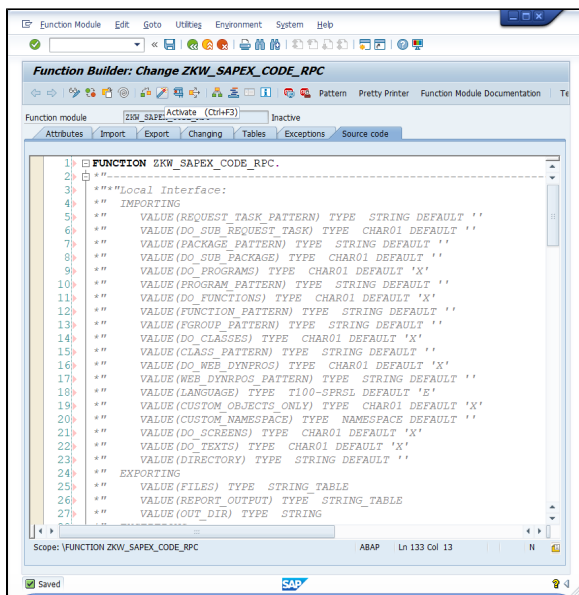
After creating the `ZKW_SAPEX_FG` function group, **create a Function Module for every `ZKW_SAPEX_*_RPC` function modules** included in `$(AGENT_HOME)/resources/abap/.../ZKW_SAPEX_FG/functions`

Use **Function Builder** (transaction **SE37**) as follows:

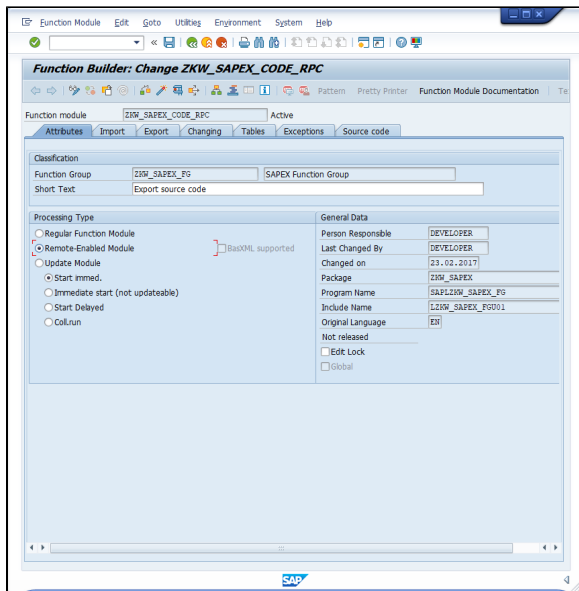




After created, go to the **Source Code** tab and paste the content of the associated file.



Make sure that all functions terminated with **\_RPC** are marked as RFC-enabled, as they will be called remotely from the SAPEX scripts.



Click **Save**.

## 2. Install SAP JCo Connector library

Remote scripts use SAP JCo Connector to remotely connect to the SAP server.

SAP JCo Connector must be installed in the remote containing the SAPEX remote scripts that will execute the Kiuwan Local Analyzer.

### Full details for installing SAP JCo library

Download SAP JCo from [SAP Service Marketplace Connectors](#) - **you need service-marketplace access for it**.

Once you have downloaded the corresponding package for your operating system, install it.

- **Windows:** There are different distribution packages for various JRE versions and hardware processors available:

| file                      | platform   |
|---------------------------|--|
| sapjco3-ntintel-3.0.8.zip | for a 32-bit JRE running on a 32- or 64-bit AMD or INTEL x86 or a 64-bit INTEL Itanium processor |
| sapjco3-ntia64-3.0.8.zip  | for a 64-bit JRE running on a 64-bit INTEL Itanium processor                                     |
| sapjco3-ntamd64-3.0.8.zip | for a 64-bit JRE running on a 64-bit AMD or INTEL x86 processor                                  |

Before installing JCo, please install the latest Microsoft Visual Studio 2005 C/C++ runtime libraries, as described in note 684106.

To install JCo for Windows unzip the appropriate distribution package into an arbitrary directory `sapjco3-install-path`.

Do not copy the `sapjco3.dll` in the `WINDIR\system32` nor into the `WINDIR\SysWOW64` directory. This will break the operability of other JCo versions that are already installed on the same system. Furthermore, you would risk that the current installation also would not work anymore if the `sapjco3.dll` gets replaced in the respective Windows system directory in the future.

- **Linux:** There are different distribution packages for various JRE versions and hardware processors available:

| file | platform |
|------|----------|
|------|----------|

|                               |  |
|-------------------------------|--|
| sapjco3-linuxintel-3.0.8.tgz  | for a 32-bit JRE running on a 32- or 64-bit AMD or INTEL x86 processor |
| sapjco3-linuxia64-3.0.8.tgz   | for a 64-bit JRE running on a 64-bit INTEL Itanium processor           |
| sapjco3-linuxx86_64-3.0.8.tgz | for a 64-bit JRE running on a 64-bit AMD or INTEL x86 processor        |
| sapjco3-linuxppc64-3.0.8.tgz  | for a 64-bit JRE running on a 64-bit PowerPC processor                 |
| sapjco3-linuxs390x-3.0.8.tgz  | for a 64-bit JRE running on a 64-bit IBM eServer zSeries processor     |

To install JCo for Linux, copy the appropriate distribution package into an arbitrary directory `sapjco3-install-path`. Next, change to the installation directory:

```
cd sapjco3-install-path
```

and extract the archive:

```
tar zxvf sapjco3-linux*3.0.8.tgz
```

### 3.Configure SAPEX scripts in Kiuwan Local Analyzer

The last step is to configure the scripts used for the scripts in KLA for remote extraction.



Scripts for extracting source code (*sapexCode.xml*) and metadata (*sapexMetadata.xml*) use following configuration files:

- *sapex.properties*
- *sapexCode.properties*
- *sapexMetadata.properties*

You can find configuration files at `$(AGENT_HOME)/conf/sapex` directory of your Kiuwan Local Analyzer installation directory.

As an alternative, you may use the extractor programs provided in the SAP system, and then either transfer the results to the system where the KLA is deployed, or directly run the KLA in the target SAP system (in particular, when an automated audit should be performed before releasing a transport request /task).

Please note that the extraction programs for code / metadata allows you to specify a local directory (in the user host) where exported contents will be saved.

If the extract operations will be performed manually, remote extraction scripts are not needed.

Please visit [SAP Extractor- Remote use - Appendix Configuration Files](#) for further information.

### Test SAPEX installation

Once completed, you may test the installation:

- You may run the [run the extraction programs in SAP](#) to check if programs and their dependencies are active.
- If the remote sapex was configured, [run command line scripts](#) to ensure that the connection properties and extraction filters work.