

Local use - Resolved Deliveries



Contents:

- [Resolved deliveries - automated audit before release of a transport order](#)
- [Registration of SAPEX Implementation](#)
- [Activation of SAPEX Implementation](#)
- [Automatic analysis before releasing a Change Request](#)

Resolved deliveries - automated audit before release of a transport order



A transport order for a **Change Request/Task** in SAP Change Transport System (CTS) could be released using the transaction **SE10** (among others).

SAP provides an enhancement to hook the release action with custom code to check the contents of the transport order. You may register an implementation for it, and activate it.

The implementation may leverage Kiuwan audits for checking that the source code in the transport order to be released satisfies quality / security constraints.

The Kiuwan Local Analyzer should be deployed in the SAP system, with the **ZKW_KLA_AGENT** OS command registered.

See [Install programs](#) for details.

This uses the Classic BAdI **CTS_REQUEST_CHECK**

SAPEX provides an implementation for it that leverages Kiuwan audits to check that the source code in the transport order to be released satisfies quality / security constraints.

Such an implementation will:

1. Extract source code for objects in the transport order.
2. Run the analysis with Kiuwan Local Analyzer.
3. Check that the audit checkpoints passes.
4. Cancel the release when the audit does not pass.

To use it, you must follow the two steps:

1. [Registration of SAPEX Implementation](#)
2. [Activation of SAPEX Implementation](#)

Registration of SAPEX Implementation



If you have installed SAPEX as a Transport Order, you can omit this step (because it was already done when importing the Transport Order) and go directly to the activation.

A candidate implementation for the **CTS_REQUEST_CHECK** is provided in the **ZCL_IM_KW_BADI_REQ_CHECK**.

For registration,

1. Open transaction **SE19**.
2. In the **Create implementation** section, choose under **Classic BAdI** the **CTS_REQUEST_CHECK**.

BAdI Builder: Initial Screen for Implementations

Edit Implementation

☒ New BAdI
Enhancement Implementation

☐ Classic BAdI
Implementation

Create Implementation

☐ New BAdI
Enhancement Spot

☒ Classic BAdI
BAdI Name

3. Click **Create**.
4. Give **ZKW_BADI_REQ_CHECK** an implementation name (so the generated implementation class will be named **ZCL_IM_KW_BADI_REQ_CHECK**)

Business Add-In Builder: Create Implementation

Definition Name

Implementation Name

5. Replace the source with the one in the provided **ZCL_IM_KW_BADI_REQ_CHECK.abap** codes.
6. Click **save**.

Activation of SAPEX Implementation

1. Open transaction **SE19**.
2. Choose **Edit Implementation** and **Classic BAdI Implementation**, specifying **ZKW_BADI_REQ_CHECK**.

BAdI Builder: Initial Screen for Implementations

Edit Implementation

☐ New BAdI
Enhancement Implementation

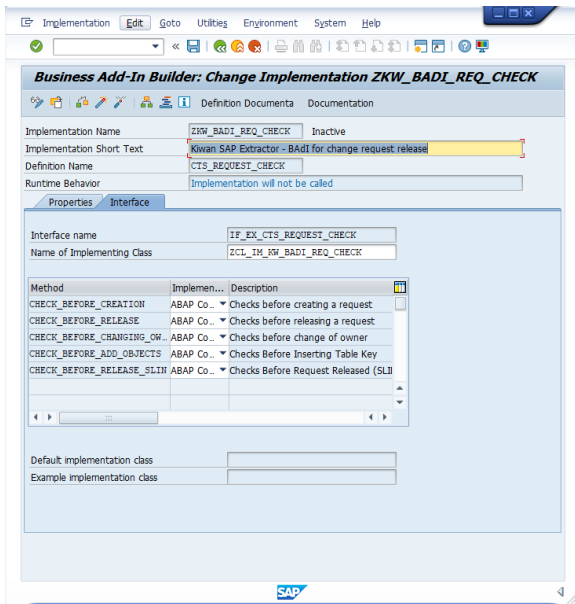
☒ Classic BAdI
Implementation

Create Implementation

☒ New BAdI
Enhancement Spot

☐ Classic BAdI
BAdI Name

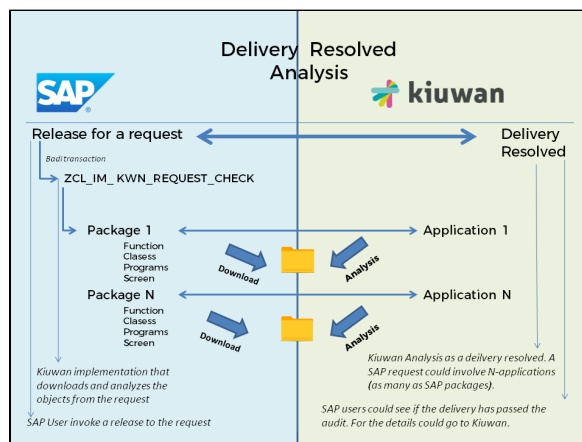
3. Click **Change**



4. Click the **Activate** icon

Automatic analysis before releasing a Change Request

Once activated, the automatic Kiwan analysis is triggered as soon as a CR is to be released. The process can be depicted as follows:



1. Source code associated to CR is downloaded
 - a. Download directory is `<DIR_HOME>/sape4k/code/<CR_ID>` (you can use the AL11 transaction to inspect the DIR_HOME value)
 - i. in linux, typically `/usr/sap/<SAP_SID>/D<Instance_Number>/work`
 - ii. in windows, change `/` by `<drive>:\`
 - b. Under `<DIR_HOME>/sape4k/code/<CR_ID>`, there will be a directory for every SAP Package involved in the CR
 - c. **Important:** Only Active objects will be exported and analyzed
2. For every SAP package contained into the Request/Task, a Kiwan analysis is executed:
 - a. using SAP package name as Kiwan application name (`-n <SAP_package_name>`, see [Kiwan Local Analyzer CLI - Command Line Interface](#))
 - b. using `<DIR_HOME>/sape4k/code/<CR_ID>/<SAP_Package_Name>` as source dir (`-s <DIR_HOME>/sape4k/code/<CR_ID>/<SAP_Package_Name>`)
 - c. using CR description plus SAP package as a Kiwan analysis label (`-l <CR description-Analysis label>`)
 - d. the analysis is executed as a **Partial Delivery** (`-as partialDelivery`)
 - e. the analysis will use Request/Task ID as Kiwan Change Request name (`-cr <CR_ID>`)
 - f. the delivery will be marked in Kiwan as **Resolved** (`-crs resolved`)
 - g. the analysis will wait for the results (`-wr`) from the Kiwan cloud (i.e. it will be blocked until the cloud analysis finishes)
3. The process will analyze every SAP package unless an unsuccessful return code is received (`ret code != 0`)

- a. In case of an unsuccessful analysis (ret code != 0), the release does not proceed and a window is displayed with the message
- b. Bear in mind that a special case is when the Audit associated with the analysis of a package fails. This condition will stop the release process.

