SAP Ninja

Installation Guide

SAPlink Code Project

for

SAP Nuggets and Slinkees

Prepared by

Jim Cooke

www.sapninja.com
sapsplunk@gmail.com

Table of Contents

Background

This document is not intended as a replacement guide for the documentation provided by the SAPlink code project. It has been included in the SAP Ninja site to provide some additional support to people who wish to load the nuggets supplied on this site.

Register and Download SAPlink

In reality, this document merely supplies some screenshots to augment the official documentation which can be found at http://wiki.sdn.sap.com/wiki/display/ABAP/SAPlink+User+Documentation.

You will still need to visit and register with the code project at https://cw.sdn.sap.com/cw/groups/saplink to download "The Daily Build" and the following items from the code project:
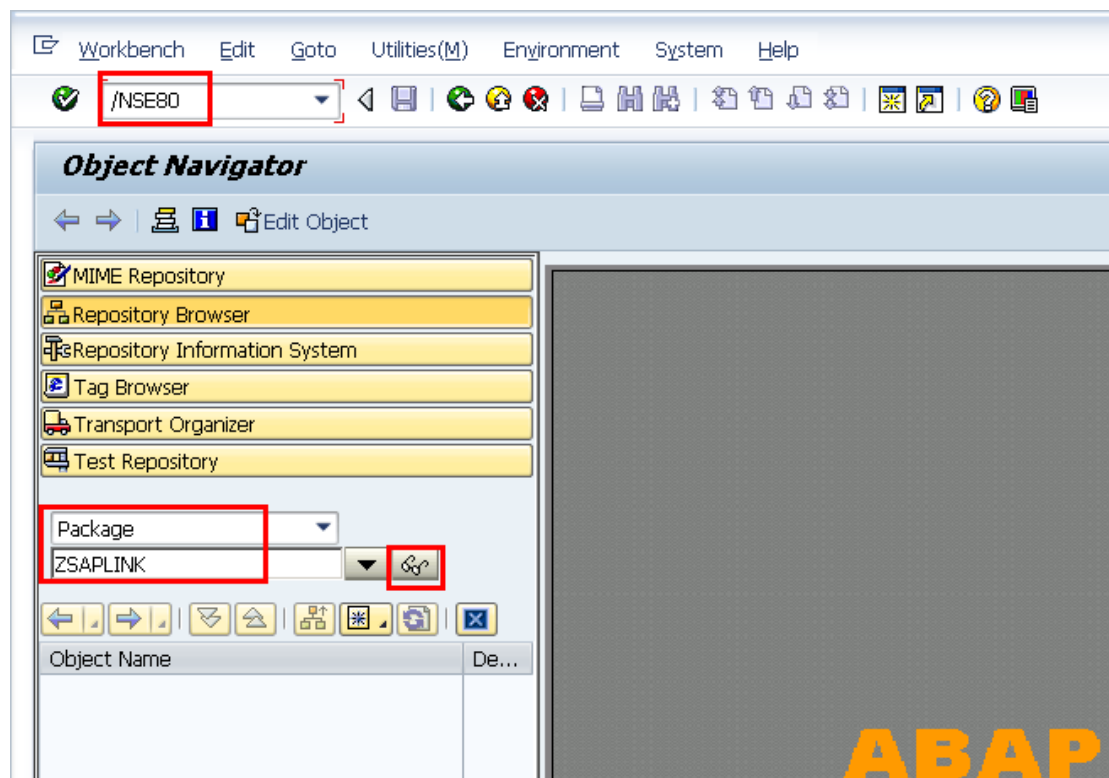
- Nugget for DDIC objects
- Nugget for function groups

Development Class

While you can just assign the SAPlink objects as local objects, it is better to do things properly and set up a new development class from transaction **SE80** and call it something like *ZSAPLINK*.
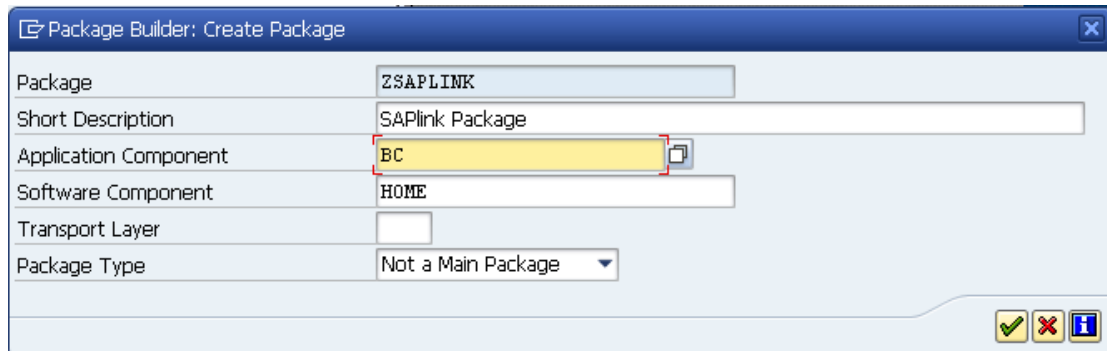
Start transaction **SE80**, choose *Package* as the object type and enter *ZSAPLINK*. Press on the *View* button.

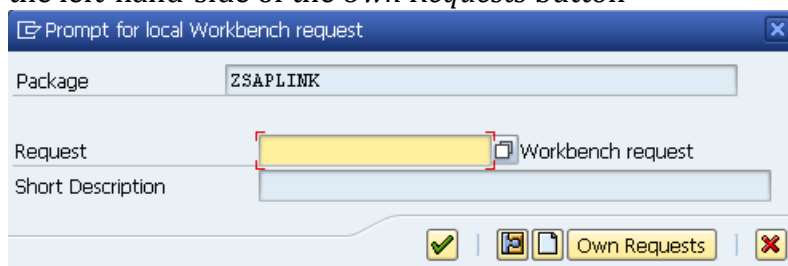At the prompt, choose *Yes* to create the package

Give the package a description like *SAPlink Package*, and assign it to the *BC* application component. If you intend on transporting these objects to other systems, you can enter a transport layer here, but this is unlikely to be needed because you will only ever really transport the objects that are created by SAPlink and not the SAPlink application itself.

Package Builder: Create Package	
Package	ZSAPLINK
Short Description	SAPlink Package
Application Component	BC
Software Component	HOME
Transport Layer	
Package Type	Not a Main Package

You will be prompted to create a transport request. Press the *Create* button on the left-hand-side of the *Own Requests* button
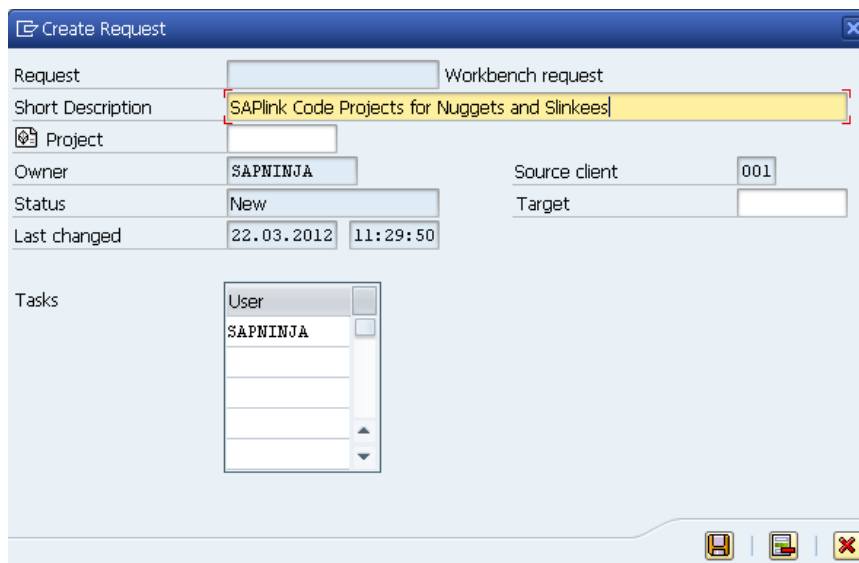
Prompt for local Workbench request	
Package	ZSAPLINK
Request	Workbench request
Short Description	

Own Requests

At the following screen, give a description for your transport request.

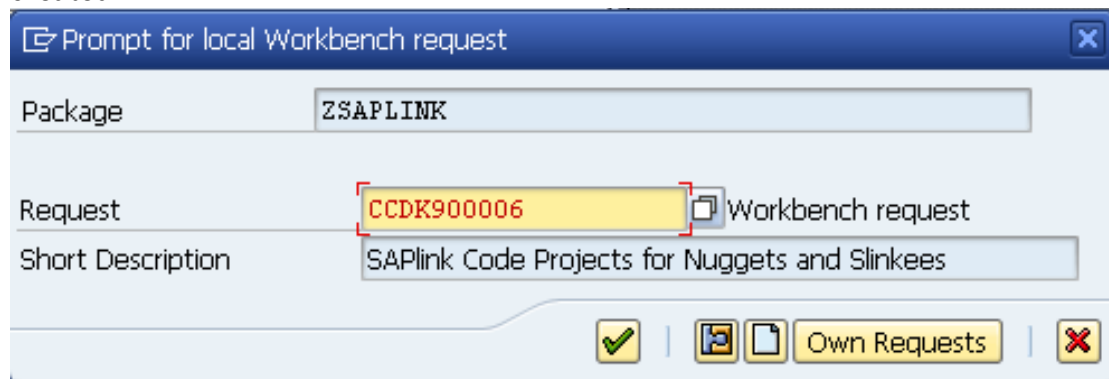| Create Request	
Request	Workbench request
Short Description	SAPlink Code Projects for Nuggets and Slinkees
Project	
Owner	SAPNINJA
Status	New
Last changed	22.03.2012 11:29:50

Source client 001
Target

Tasks	User
	SAPNINJA

Click the green tick at the following pop-up once your transport request has been created.
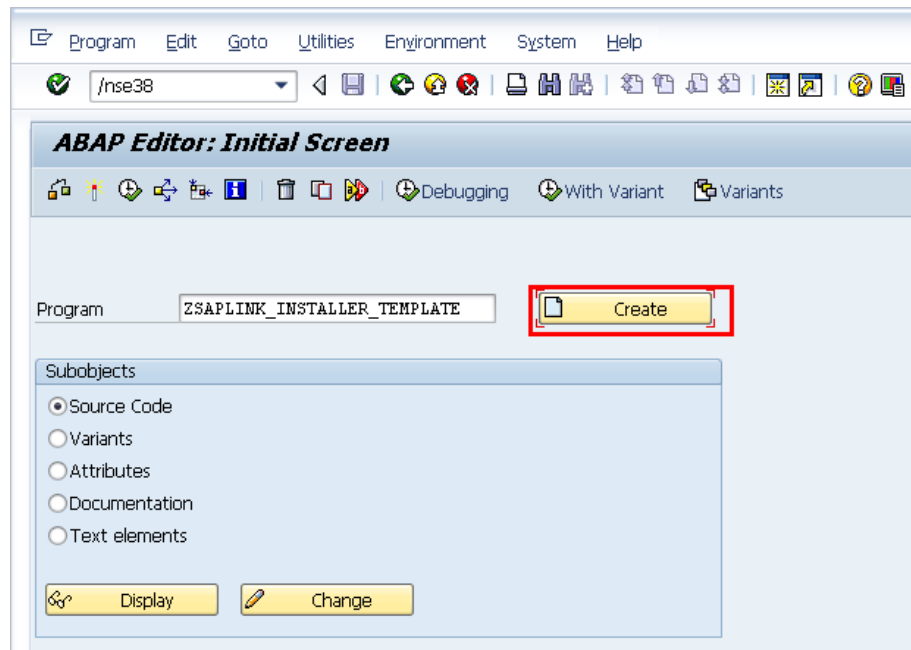

You have now finished creating the development package.

Create the SAPlink Installer Program

The SAPlink application is delivered as a nugget itself. You need to create a special program to upload this nugget. Start transaction *SE38*, enter a program name of *ZSAPLINK_INSTALLER_TEMPLATE* and press the *Create* button

At the next screen, give the program a title and set the attributes as shown below. Press the *Save* button when done.
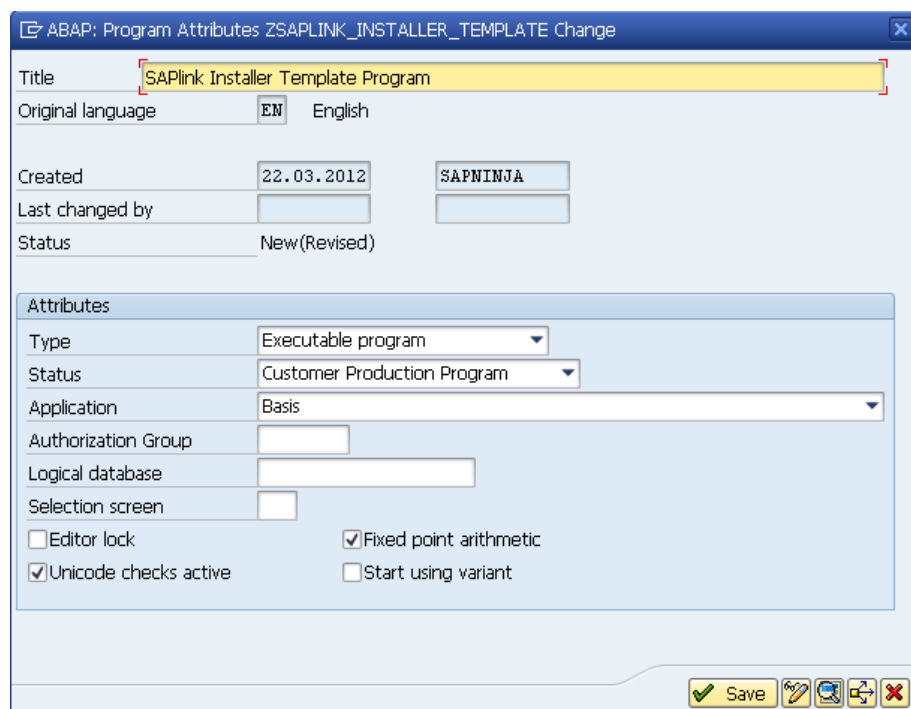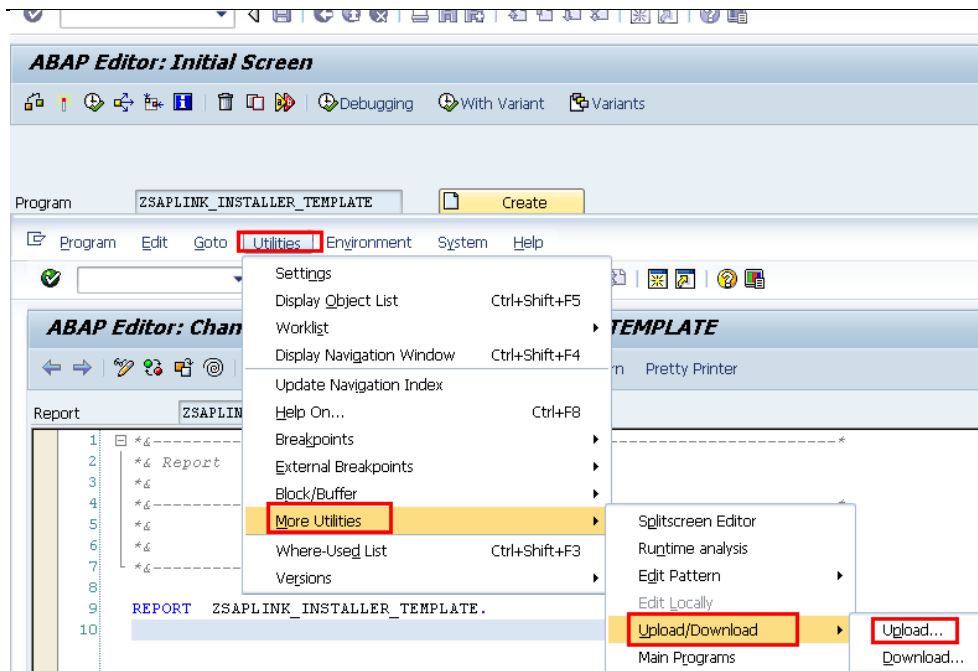
At the next screen, you get to assign the *ZSAPLINK_INSTALLER_TEMPLATE* program to the *ZSAPLINK* package.

You will be prompted for a transport request, so just use the one you created earlier.
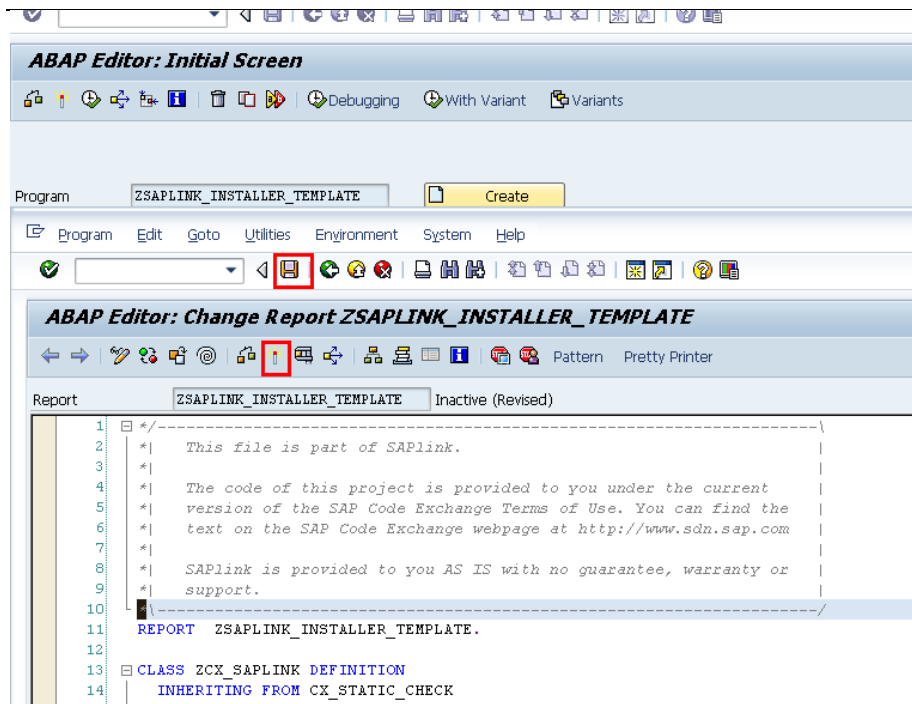

From within the program source editor, choose from the menu *Utilities -> More Utilities -> Upload/Download -> Upload*

When you unpacked the *Daily Download* bundle from the code project, one of the files extracted is called *zsaplink_installer.txt*. Choose this file for upload.
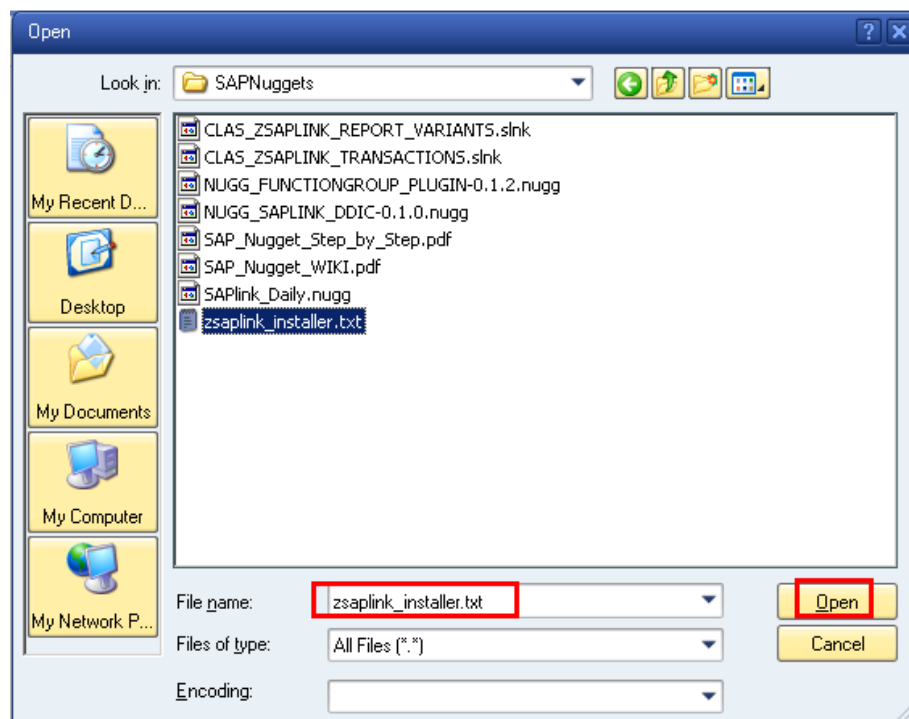

Once code has been imported, press the *Save* and *Activate* buttons as shown below.

Execute the SAPlink Installer

Now you just need the execute the code you imported. From transaction *SE38* or *SA38*, enter the program name *ZSAPLINK_INSTALLER_*TEMPLATE. Press the *Execute* button.   Enter the installation nugget that you unpacked earlier called *SAPlink_Daily.nugg* and press the *Execute* button.
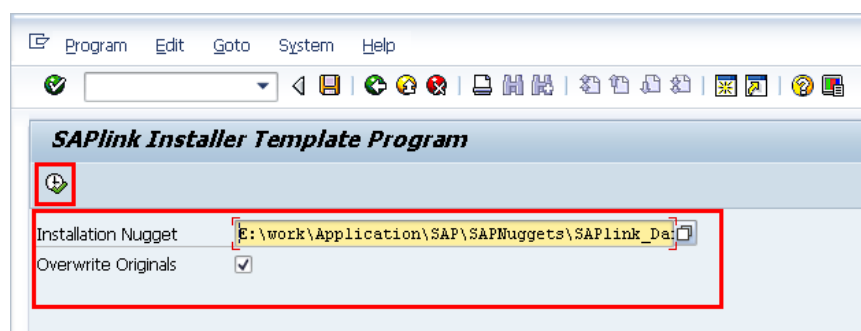
For each of the objects that gets imported, you will be prompted for a package name. Enter *ZSAPLINK* for each one as shown below.

Create Object Directory Entry

Object	R3TR	CLAS	ZCX_SAPLINK

Attributes

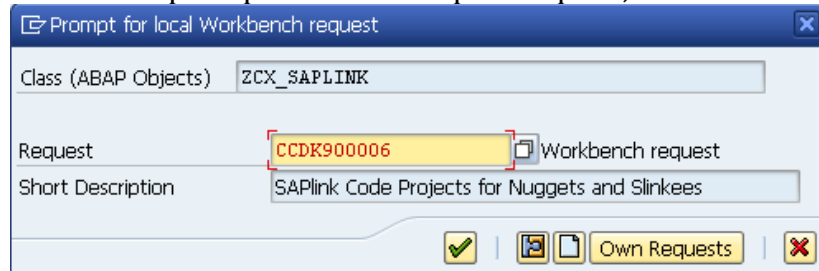|	
Package	ZSAPLINK
Person Responsible	SAPNINJA
Original System	CCD
Original language	EN English
Created On	

Local Object | Lock Overview

Open Source Solutions for the SAP Community

You will be prompted for a transport request, so use the one prepared earlier.

Once the program has loaded everything, you will get a listing of all the objects that were created as shown below.
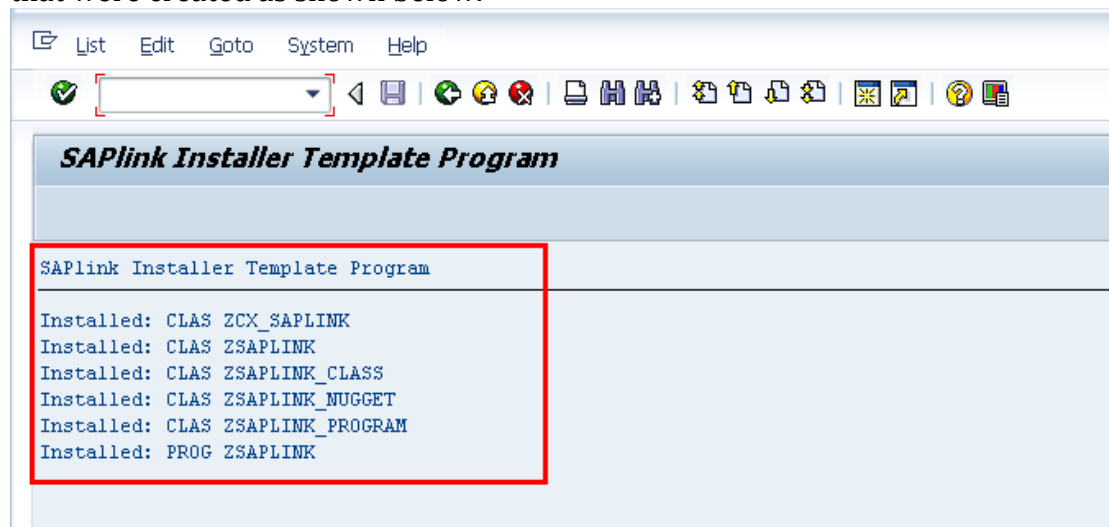
Start transaction *SE24* and enter the first object of type *CLAS* (from above) and enter the object name (eg. *ZCX_SAPLINK*). Press the *Activate* button.
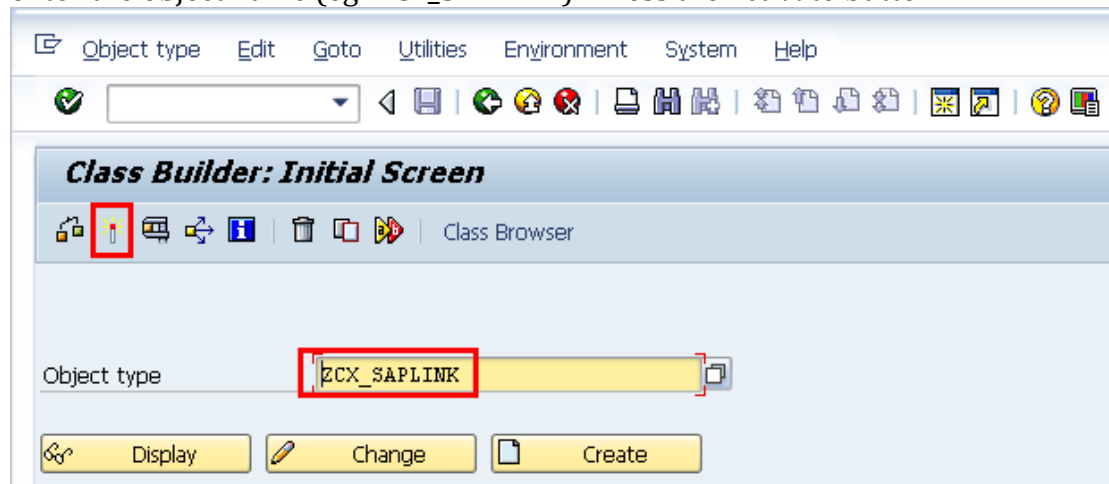
You will be presented with a list of all objects that need to be activated for this class. You accept the defaults and allow activation to take place. You need to repeat this for every object of type *CLAS* shown in the screenshot above after the installer template executed.

Alternatively, you can press the button *Whole Worklist* at the screen shown below and it will show all of your inactive classes. Select all objects and press *Enter* to activate the lot at once.
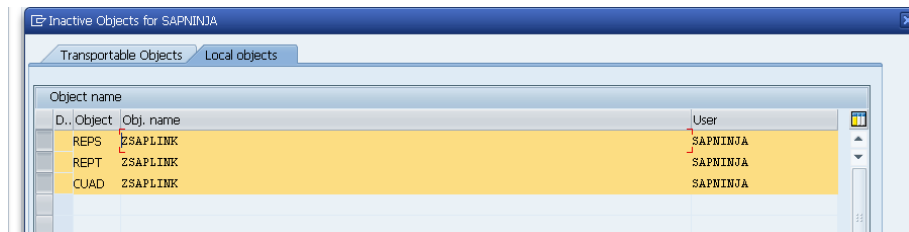
Import Plug-In Nuggets

Execute the main SAPlink program *ZSAPLINK* from *SA38*. You will be prompted to activate the program as shown below.

The next screen is shows the *ZSAPLINK* application in action:
1. Select the *Import Nugget* radio button
2. Enter a package name *ZSAPLINK* to own the objects
3. Enter the nugget file name for the DDIC plugins that you downloaded from the code project (*NUGG_SAPLINK_DDIC-0.1.0.nugg*)
4. Click the *Overwrite Originals* checkbox
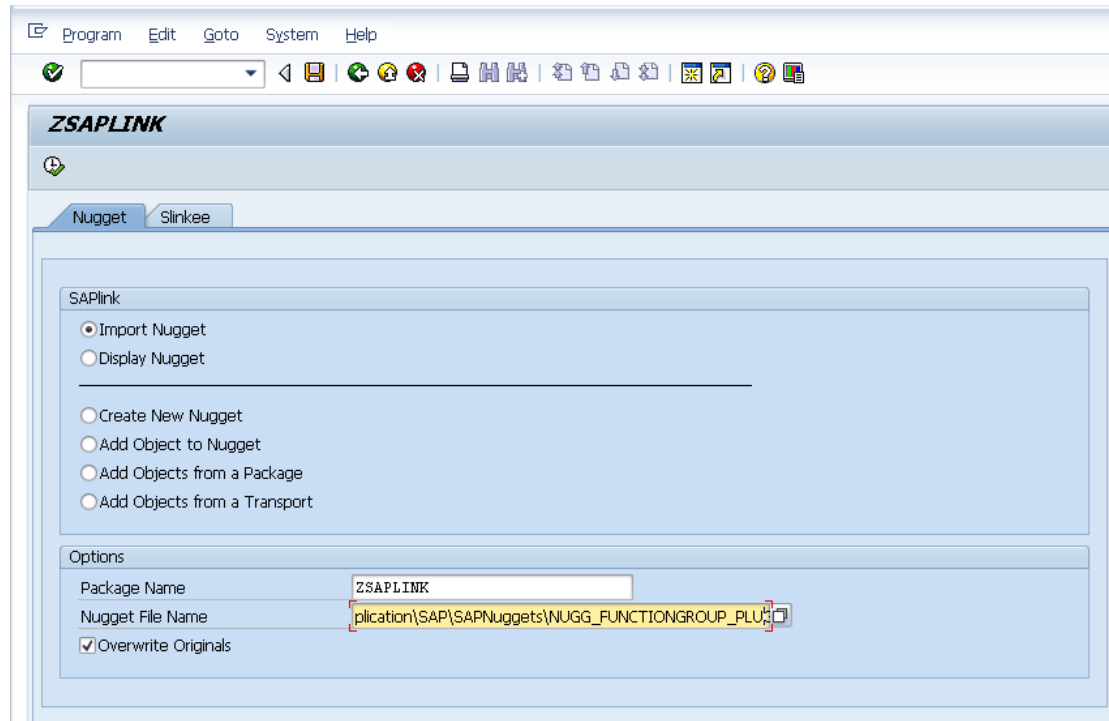5. Click on *Execute* to import the nugget.
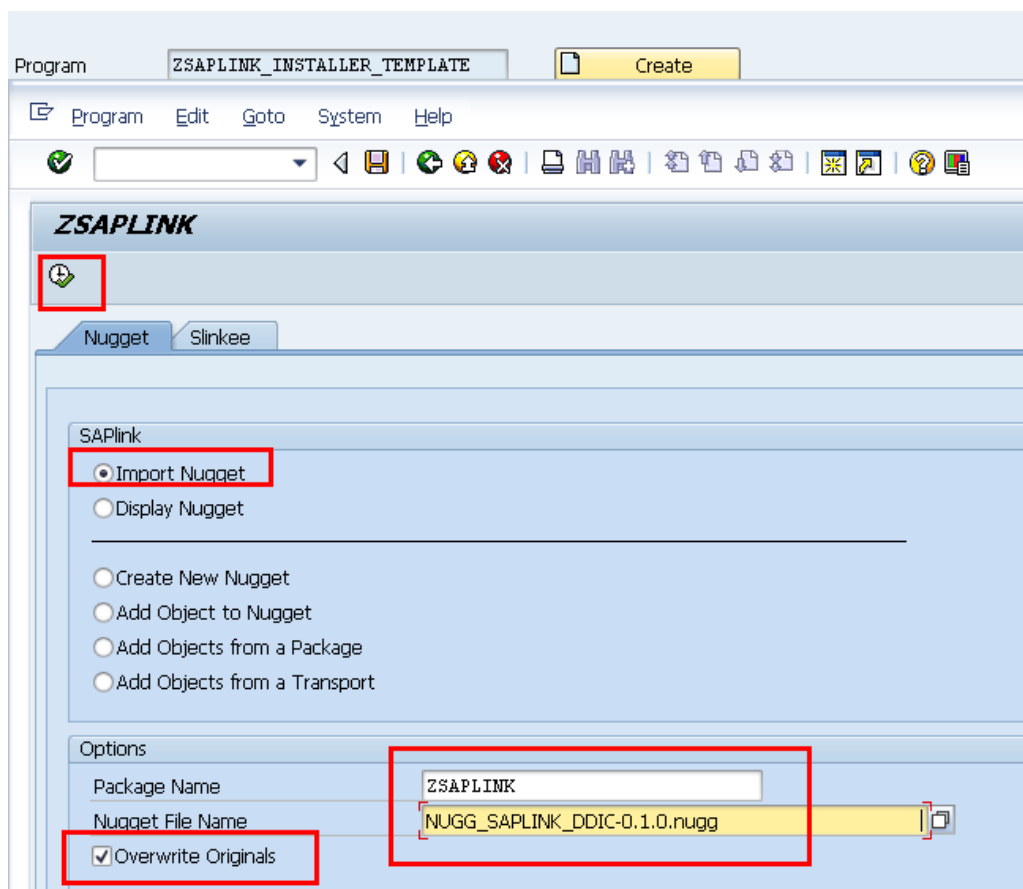
Enter the SAPlink transport request you created earlier when prompted.

Like before, the program will return with a list of objects (all *CLAS* types) that were imported. They all need to be active with transaction *SE24* as described above.
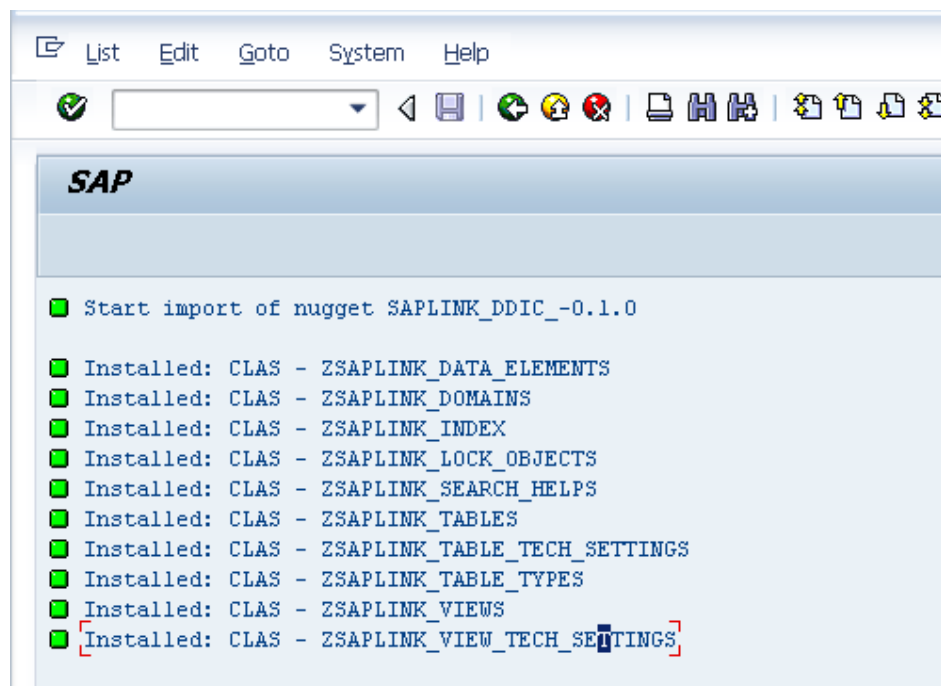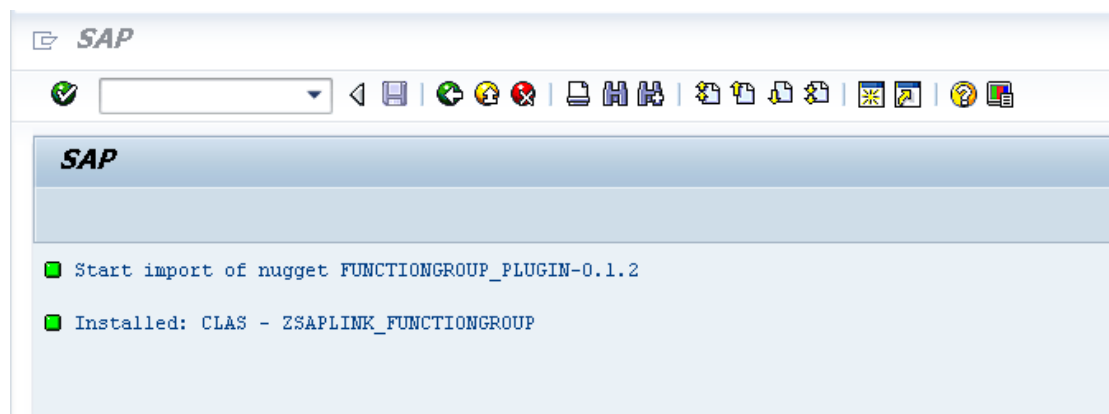
Repeat the steps above to import the function group nugget.

Again, there is a list of objects to be activated from *SE24*.

You now have the SAPlink tool installed correctly and are ready to start importing Ninja nuggets.