

The logo for SAMDOSoft, featuring the word "SAMDOSOFT" in a bold, white, sans-serif font. A thin, dark diagonal line crosses through the letters "A", "M", "D", and "O". The text is centered within a solid black rectangular background.

# **SAMDOSOFT**

## **Perspective Trace**

### User Manual

[www.samdosoftware.com/perspective-trace](http://www.samdosoftware.com/perspective-trace)

# Contents

---

<b>General Overview .....</b>	<b>2</b>
<b>Solution Components Overview .....</b>	<b>2</b>
Debugger Script.....	2
Perspective Trace Application .....	3
Features .....	3
Controls .....	6
<b>Activation.....</b>	<b>7</b>
<b>Minimum Requirements – PC .....</b>	<b>7</b>
<b>Minimum Requirements – SAP .....</b>	<b>7</b>
<b>Required Permissions in SAP .....</b>	<b>7</b>
Option 1: Loading the Debugger Script from the txt file .....	8
Option 2: Loading the saved Debugger Script from the database .....	9
<b>Perspective Trace Workflow .....</b>	<b>10</b>
Creating a trace file .....	10
Advanced tips .....	13
Loading a trace file into the application .....	13

---

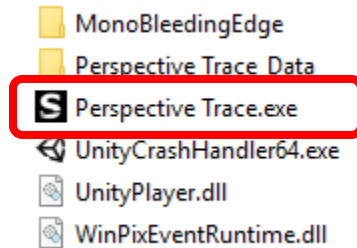


---



# Perspective Trace Application

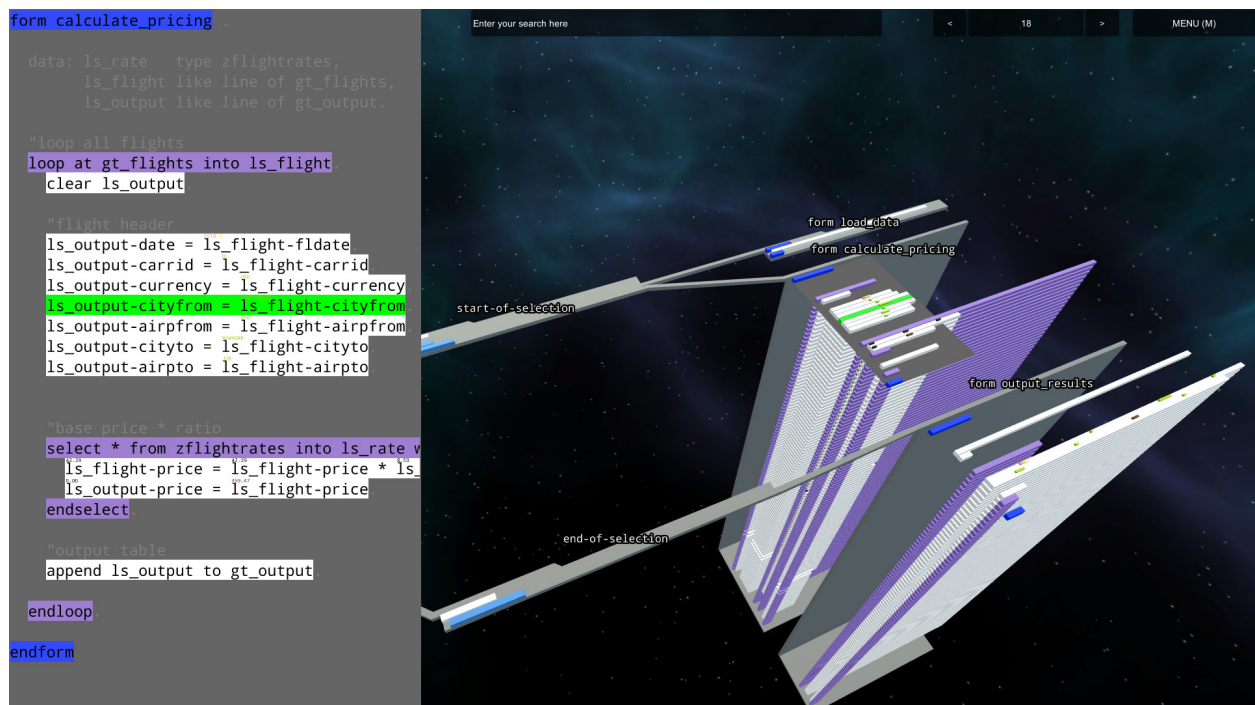
Perspective Trace is a standalone application responsible for visualization of the traced program flow. It offers several useful features.



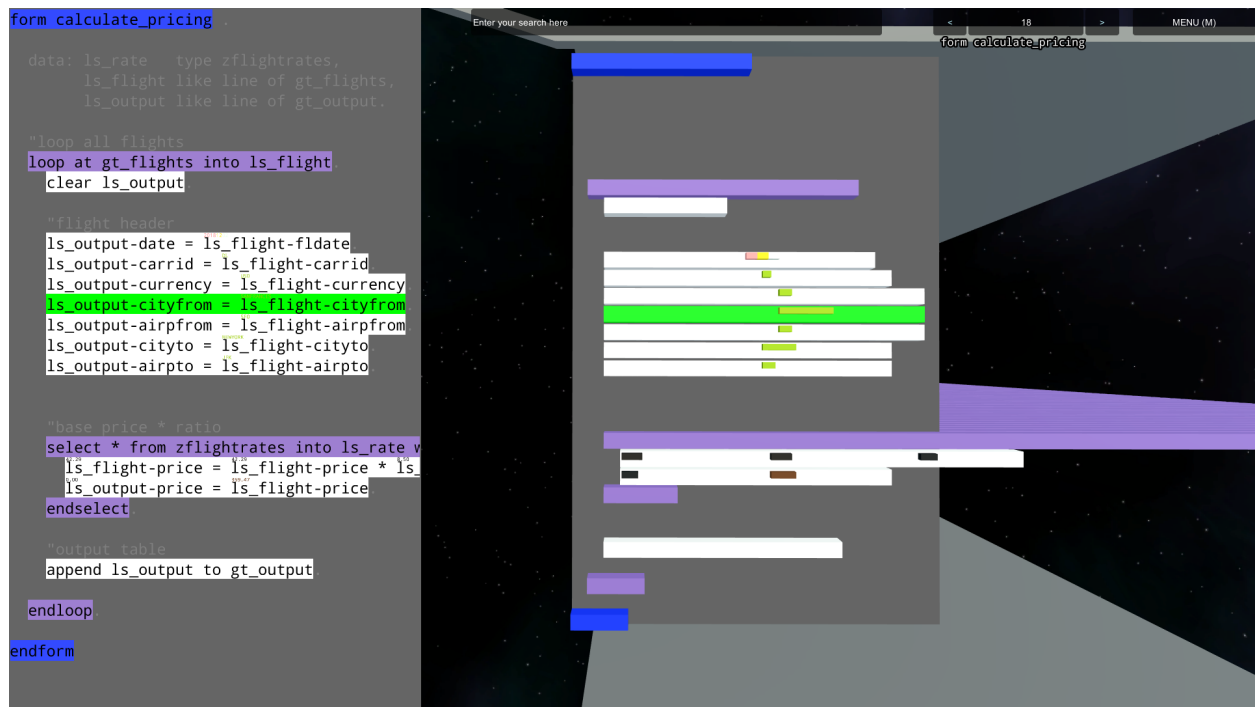
## Features

Perspective Trace screen is split in two windows – the 3D view and code explorer panel.

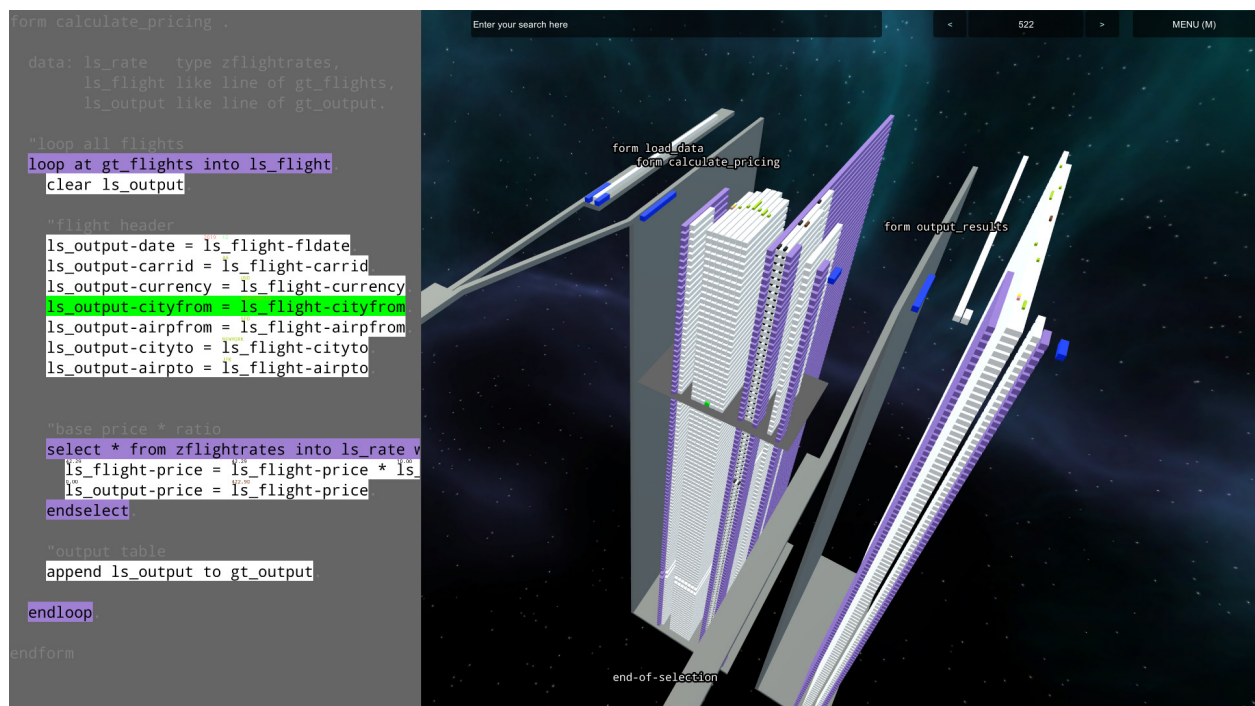
The 3D view (on the right side) is a map of executed statements. They are grouped into nodes (grey). Each node is a block of statements belonging to a single unit (a method, form, function, etc.). A link between nodes represents one unit calling another.



Left-side code explorer panel highlights the executed statements while still showing the full program source code including comments, declarations and unvisited statements.

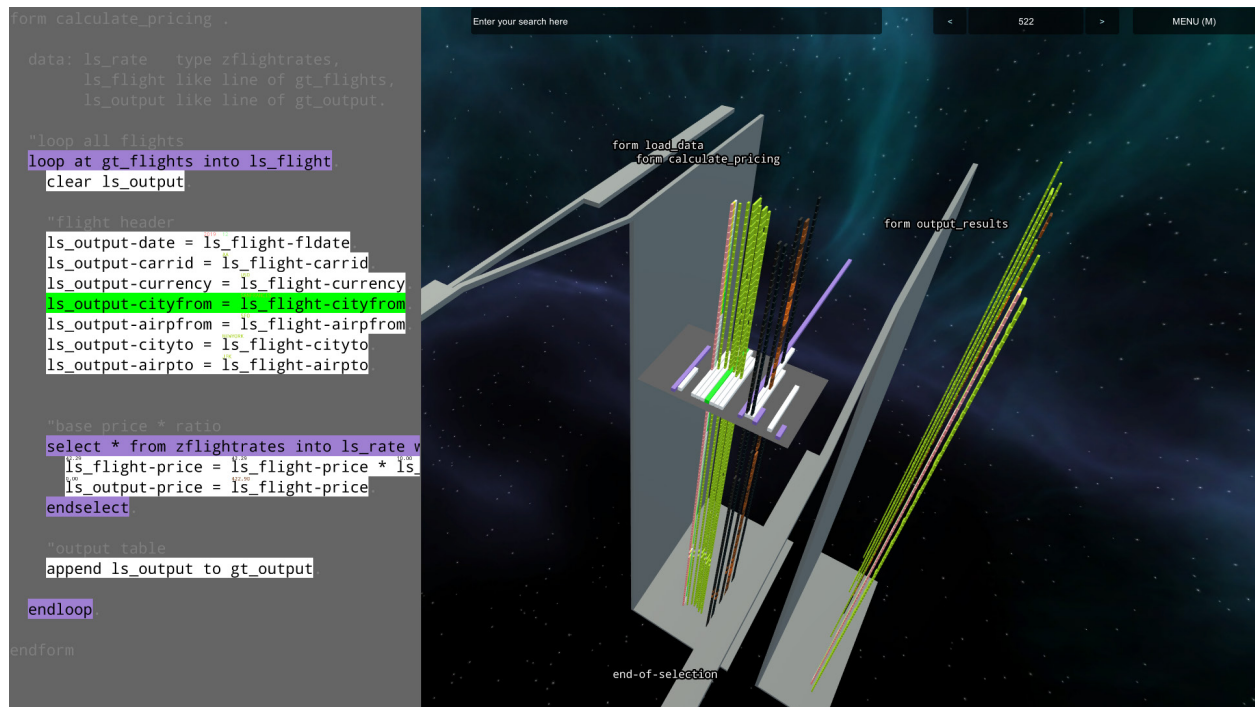


Left-side code explorer panel corresponds to right side 3D view when viewed from the top. The morphology of the code is preserved, shapes and indentation look exactly as in ABAP editor. This enables the user to remember all the unique patterns without introducing unnecessary abstraction.



The perspective projection allows visual inspection of a large number of steps and variables at once. The side view is ideal for examination of loops. Tens of thousands executed statements can be inspected at once.

Users can comfortably see the “big picture” and select the details they are interested in.



Great importance is given to the representation of variables. A special view removes the statements and shows just the variables. In the picture above, the intensity of orange is calculated according to the float value of the variable. Larger numbers are clearly distinguishable from others even when inspected from far away. Strings are shown in lime and dates use special colours for day, month and year.

## Controls

Shortcut	Description
W S A D Q E	Move in different directions
Hold right mouse button in 3D view	Look around
+ and - buttons in panels	Increase/Decrease font size
Mouse scroll up/down in 3D view	Zoom in/out
Mouse scroll up/down in left panel	Scroll in the source code
Hover (mouse over) in 3D view	Show text label for statements and variables
Click left mouse button in 3D view	Select a statement
Double click on a label or search result	Move to the clicked statement in 3D
Right/Left arrow	Jump to next/previous statement
O	Open a trace (zip) file
H	Reset camera view to initial position
1	Show/hide statements in 3D view
2	Show/hide variables in 3D view
3	Show/hide structural links in 3D view
4	Show/hide labels in 3D view
Hold middle mouse button	Orbit around selected statement

## Colours

Statements are coloured according to their type. Furthermore, different types of variables are coloured according to their content – value.

## Search

The search bar enables full text search in visited statements and traced variables. Right click on a search result to move to its location in the 3D view.



# Activation

---

Perspective Trace application requires activation with a valid license file. The application will generate a unique Device ID to identify the computer it runs on. Based on the Device ID, Samdosoft issues an appropriate license file with validity corresponding to the contract.

## Minimum Requirements – PC

---

Minimum requirements for the Perspective Trace application are a DX11-compatible graphics card and 8GB RAM running Windows 7 or newer.

An OSX version is available as well, please contact support for more details.

## Minimum Requirements – SAP

---

Debugger Script requires an SAP ECC with NetWeaver 7.0 EHP2 or higher – this is the version where the Debugger scripting functionality was introduced into the new ABAP debugger.

## Required Permissions in SAP

---

The user needs access to the debugger

Authorization object
S_DEVELOP objtype=DEBUG actvt=03 (display)

and a permission to download files to the front-end

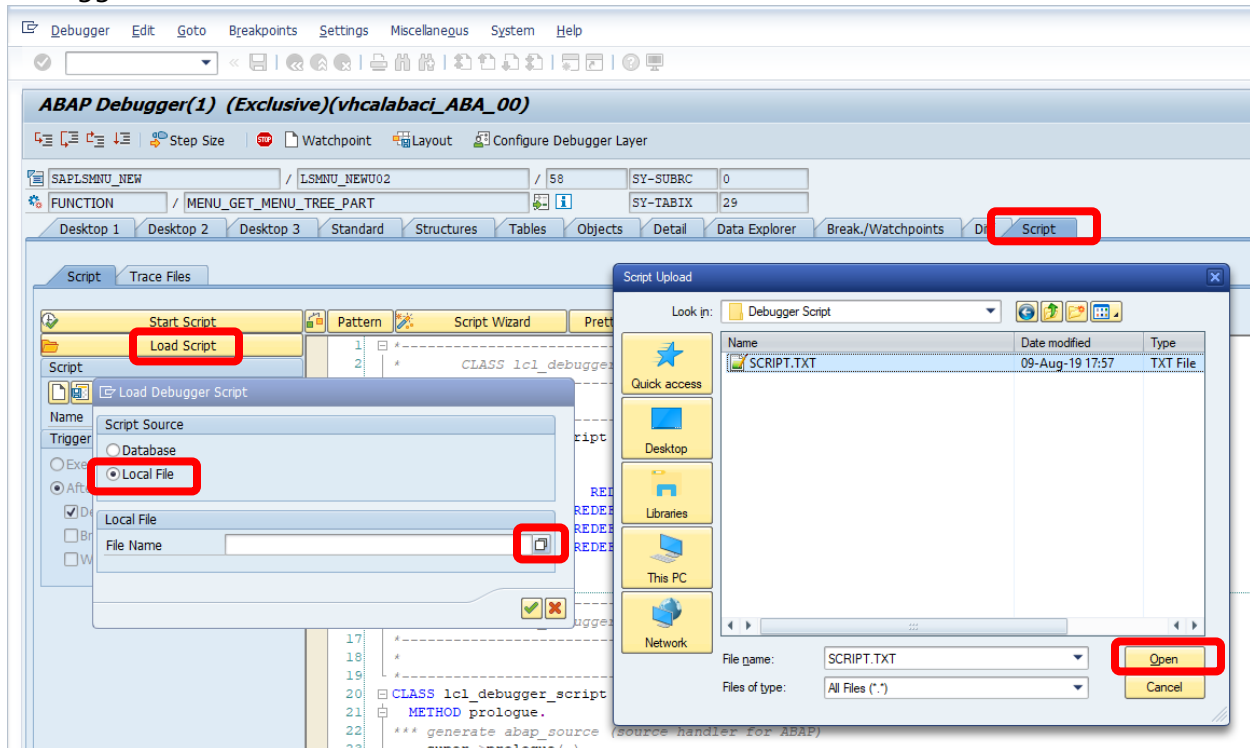
Authorization object
S_GUI actvt=61 / actvt=02

The remaining permissions depend on the desired script loading mode:



## Option 1: Loading the Debugger Script from the txt file

The picture below shows how to load the Debugger Script from a file into the new ABAP Debugger.



The following authorization object is required:

### Authorization object

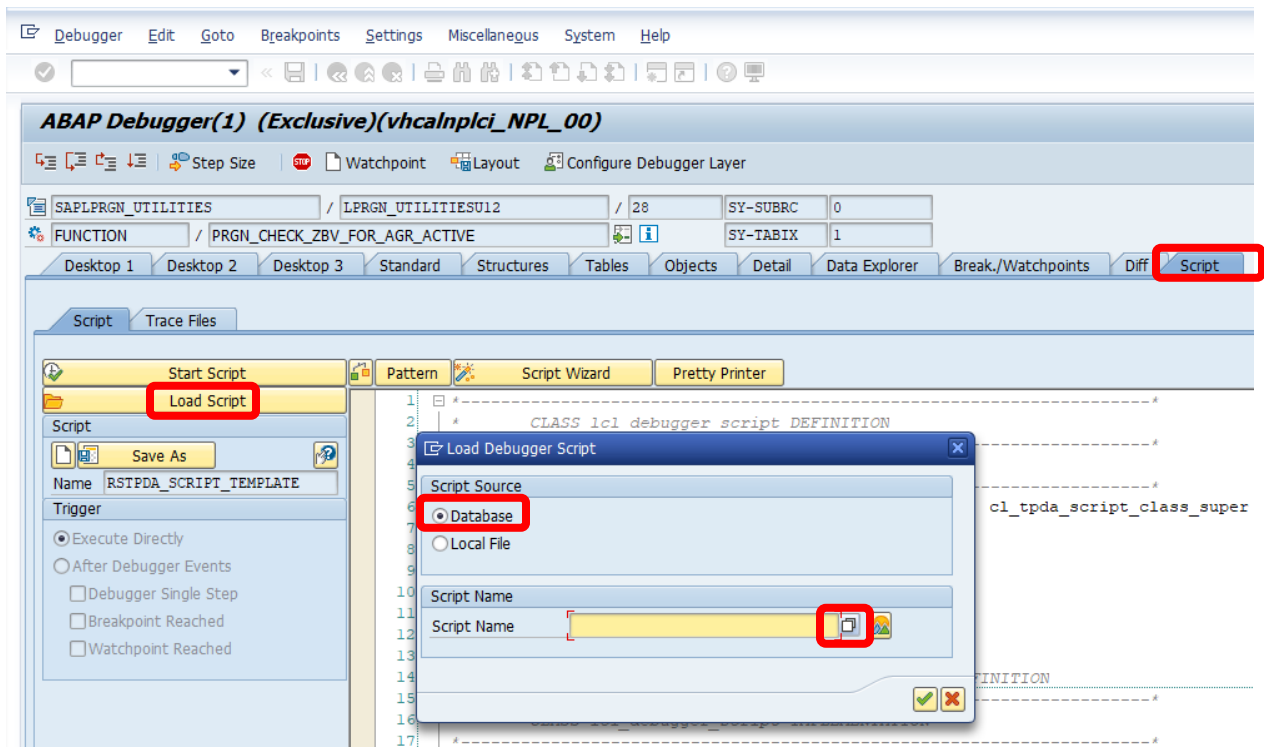
S\_DEVELOP objtype= PROG actvt=02 (change)

**Pros:** No transports, no developer key, no installation is necessary.

**Cons:** Broader permissions are needed.

## Option 2: Loading the saved Debugger Script from the database

The picture below shows how to load an already saved Debugger Script from the database into the new ABAP Debugger.



The following authorization object is required:

### Authorization object

S\_DEVELOP objtype= PROG actvt=03 (display)

**Pros:** Read-only access to the script prevents the user from modifying it.

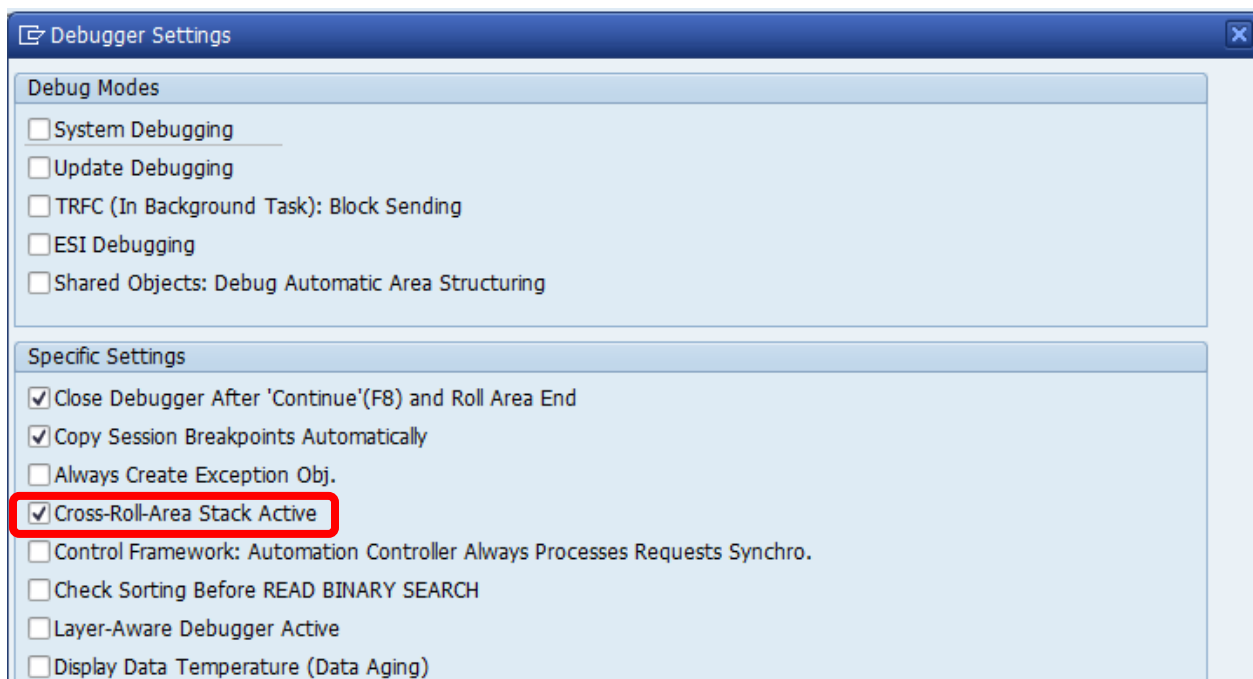
**Cons:** Saving the script to the database requires a developer key. Distribution to other systems may require transport.

# Perspective Trace Workflow

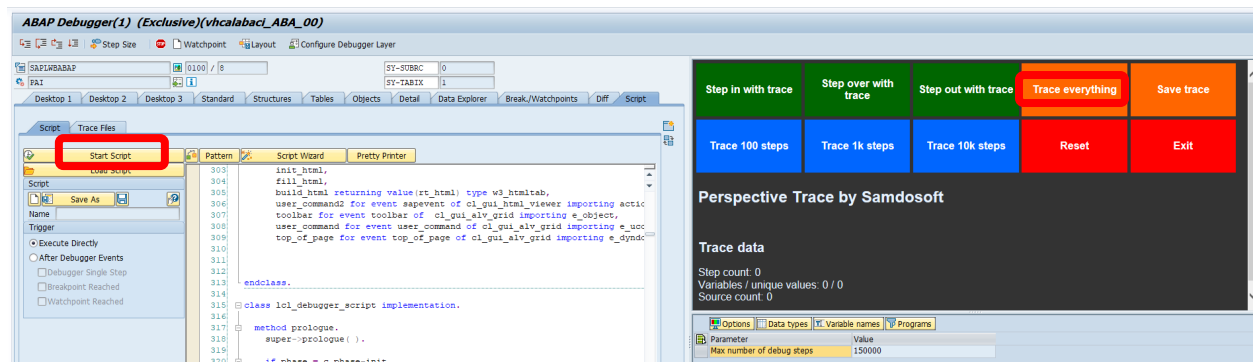
## Creating a trace file

Once the Debugger Script is loaded into the ABAP debugger, press the **Start Script** button. You can start tracing from any debugger position – for example, from a certain breakpoint or watchpoint activation. There are several configuration options for the tracing process. Some of them are deactivated and will be enabled in later versions.

First, enable Cross-Roll-Area stack in the Debugger settings. (SHIFT-F1 shortcut in the debugger)



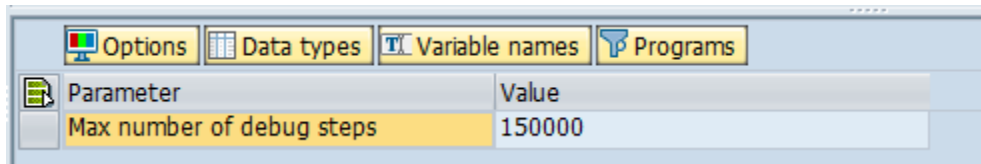
Now we are ready to start tracing.



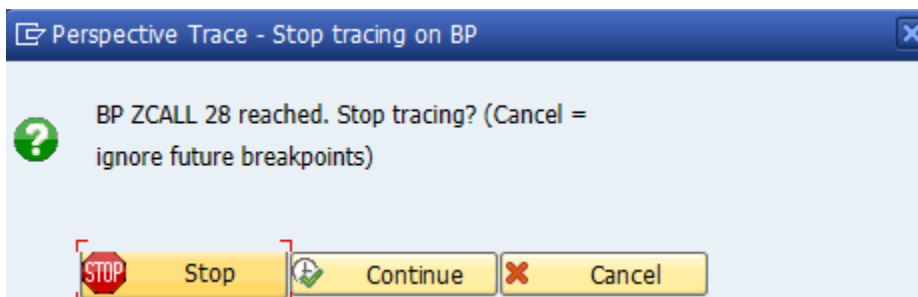
Press the **Start Script** button. From now on, the Debugger Script controls the ABAP debugger and displays various tracing options.

The **Trace everything** button starts a fully automatic tracing process. The script will **stop tracing** if **one of the following conditions is met**:

- the **maximum number of steps is reached** (defined in the options window)

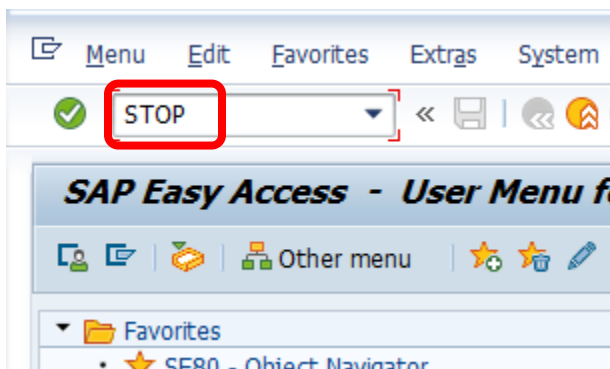


- a **breakpoint is reached and the user choses to stop the trace**



By choosing Cancel, further breakpoints are ignored and the script continues tracing.

- when the program being debugged reaches a screen and waits for user input, the **user inputs STOP into the transaction bar and confirms**



This may not work on some screens.

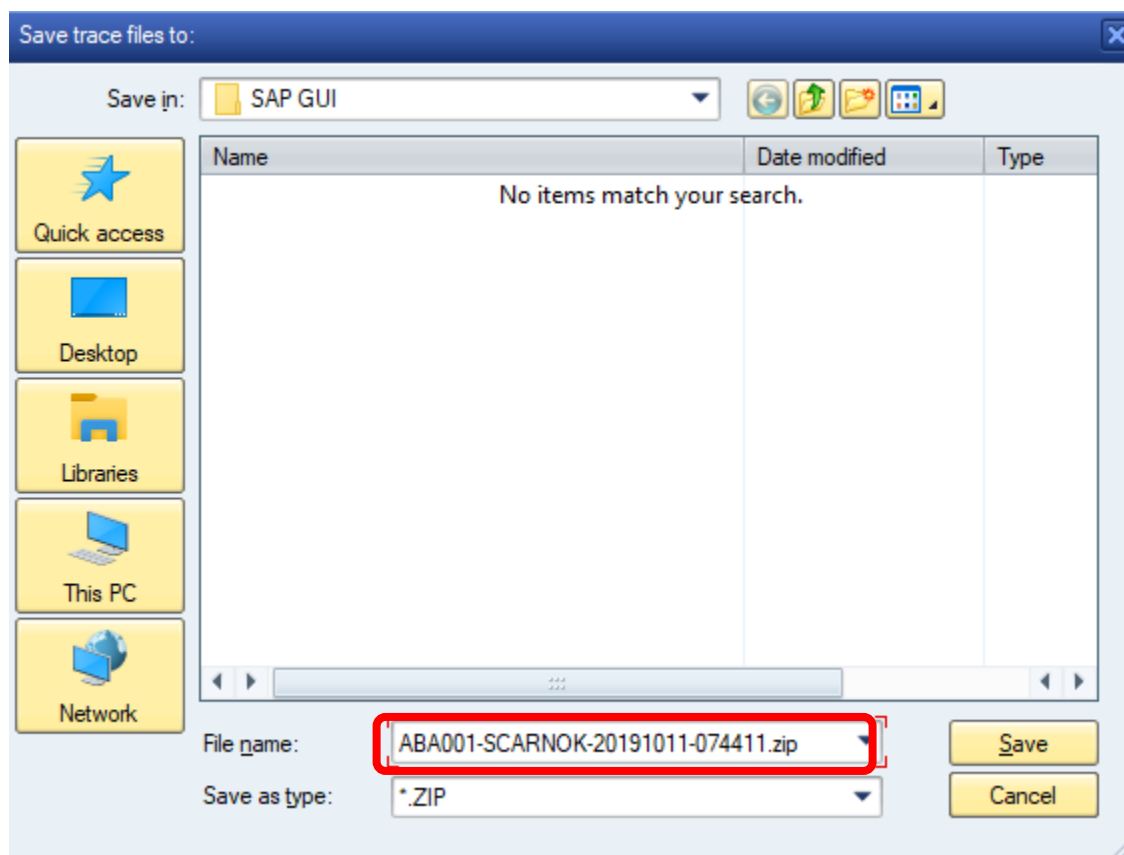
Apart from the fully automatic tracing process, there are several semi-automatic options.

These buttons are similar to classic debugging steps (**Step In, Step Over, Step Out, Continue**) but they perform the tracing while executing the desired step. The **Trace everything** button can be used as **Continue**.

Using this technique, user can trace selectively according to preference.

At any time, **Save trace** button can be used to finalize the traced data for export. The **save file dialog** window will appear. Choose where to save the trace file. The name of the file will be automatically generated in the format:

**SIDCLIENT-USER-DATE-TIME.zip**



Press **Reset** to clear already cumulated traced data. This will not reset the position of the debugger.

Press **Exit** to quit using the Debugger Script and return full control to ABAP debugger.

## Advanced tips

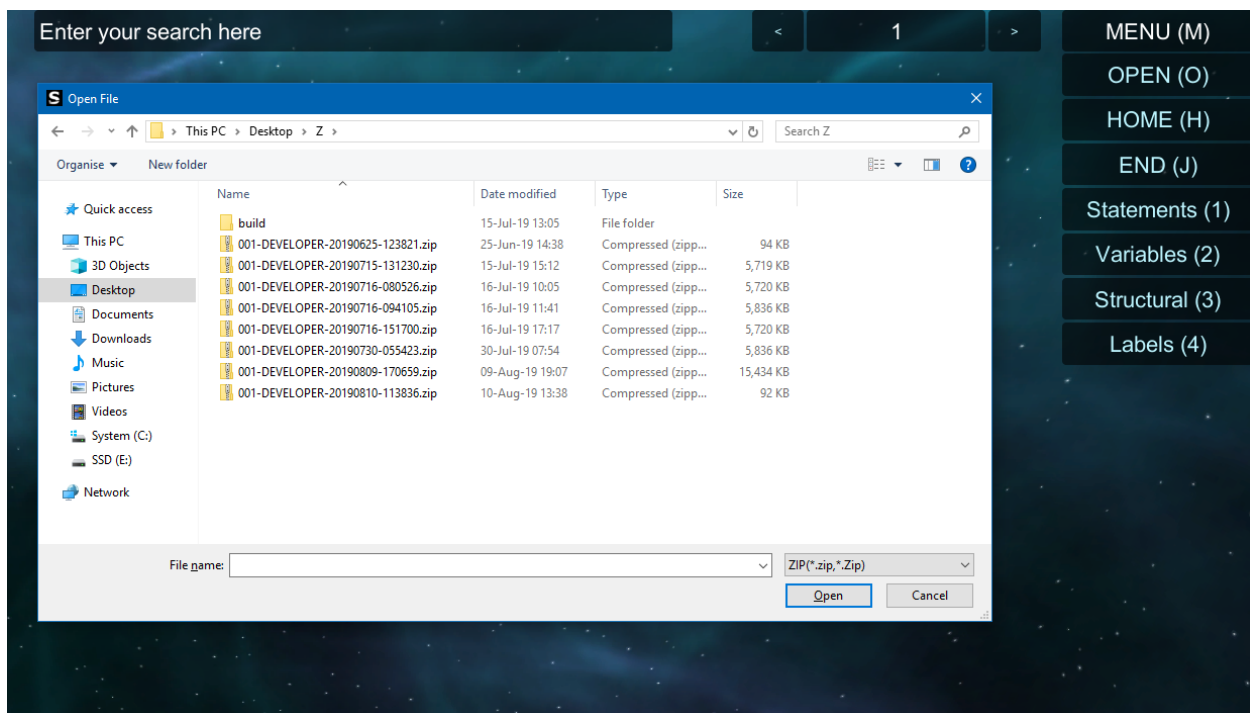
You can start the debugger in normal mode **/h** or system debugging mode **/hs** if necessary.

Software Layer-Aware Debugging (SLAD) can be used as well to limit the scope of visited statements and programs.

<https://blogs.sap.com/2010/07/27/layer-aware-debugging-in-nw-70-ehp2/>

## Loading a trace file into the application

Press **Open** (shortcut **O**) and select the zip file generated by the Debugger Script.



**Good luck and thank you for using Samdosoft Perspective Trace!**