SAPlink:Docs:UserDoc

From Skunkwerkz

Table of contents

Documentation SAPlink Version

- 0.1.0
- 0.1.1
- 0.1.2

Online version of documentation available in docs (*http://saplink.googlecode.com/svn/branches/SAPlink/*) folder of each release at the SAPlink project page http://code.google.com/p/saplink/

Install

- *Known ABAP basis versions compatible with SAPlink include:*
  - 6.20
  - 6.40(Netweaver 2004)
  - 7.0(Netweaver 2004s)

Initial

1. Download the latest version of the SAPlink installer zip file from the SAPlink Project page (*http://code.google.com/p/saplink/*) which contains the following files:
    1. SAPlink_installer.txt - hopefully the last program you will ever have to copy and paste
    2. NUGG_SAPLINK.nugg - nugget containing necessary objects to install
    3. ReadMe.pdf - PDF document with contributors and these instructions
    4. COPYING - text document with GNU GENERAL PUBLIC LICENSE
2. Extract contents of the SAPlink installer zip file
3. Copy, paste, save, and activate the contents of the SAPlink_installer.txt file into a new program ZSAPLINK_INSTALLER onto the target SAP system
4. Execute newly activated program ZSAPLINK_INSTALLER
5. Use the extracted file NUGG_SAPLINK_INSTALL.nugg for the selection parameter "Installation Nugget"
6. WARNING: Using checkbox "Overwrite Originals", will overwrite any existing version of the SAPlink objects that were previously installed onto the system. Only use when reinstalling or installing a major release where a brand new install is required. Otherwise, follow the normal procedures for upgrade below.
7. Execute
8. Newly installed objects need to be activated manually, as all objects are installed as local & inactive
    1. Run transaction code SE38
    2. On initial screen of SE38, enter program name ZSAPLINK into selection parameter "Program" and hit activate command to show worklist
    3. On worklist screen, select "Whole Worklist" button on bottom to show all inactive objects in your worklist *IMPORTANT NOTE: You must click the button labeled "Whole Worklist" to show and select all SAPlink related objects. There are multiple objects that are imported during the initial install and must all be activated for SAPlink to run properly*
    4. Select all newly installed SAPlink related objects (*ZSAPLINK*) and activate
    5. Once activation for all objects is complete, SAPlink can be executed via program ZSAPLINK

Plugins

- To see a list of installed plugins, use the F4 value help in the selection parameter field for "Object Type" from ZSAPLINK->Object Tab->Export Object->Object Type. Additional plugins can be downloaded from the SAPlink Project page (*http://code.google.com/p/saplink/*)
  and imported/installed using SAPlink. A plugin can be delivered via a Slinkee or a Nugget. Follow normal usage instructions below for the plugin file accordingly.
- If no plugin is available for the object type needed, we need your help. If you are a gracious soul and willing to give back to the community as a plugin developer, go to the SAPlink Group (*http://groups.google.com/group/saplink*)
  and let us know. Developer documentation is available on the SAPlink documentation page.

Upgrade

1. Download the latest version of SAPlink or any of the SAPlink plugins from the SAPlink Project page (*http://code.google.com/p/saplink/*)
2. Unless otherwise stated, you can use your existing version of SAPlink to upgrade itself or any of its plugins. Use the downloaded slinkee or nugget file according to the normal usage instructions below

Usage

- To execute SAPlink, use program ZSAPLINK. *If you do not know which SAP transaction to use to run this program from, you don't need to be running SAPlink anyway. Please go find and bribe a developer to help you before proceeding.*
- There are two types of files that can be imported or exported using SAPlink, either a Slinkee file (.slnk) or a Nugget file (.nugg). Both filetypes contain XML representation of the objects to be installed. They can be read using any XML reader, a text editor, or some compatible browsers (confirmed using Firefox).

Slinkee

A Slinkee file(.slnk) is used for single objects to be imported or exported. Slinkees are very quick and easy to use & recommended for single object collaboration. They can be imported and exported from the "Object" tab

Import

1. Select radio button "Import Object"
2. Use the Slinkee file containing the object to be imported in the selection parameter "SAPlink File Name"
3. WARNING: Using checkbox "Overwrite Originals", will overwrite any existing object with the same type and name on the target system. Be sure you know
4. Execute
5. Newly installed objects need to be activated manually, as all objects are installed as local & inactive

Export

1. Select radio button "Export Object"
2. Enter the name of the object to be exported for the selection parameter "Object Name". *Your object name should be somewhat unique. Remember that you will be sharing your objects with other developers and they probably already have a program called ZFoo, ZMyTest, or ZINeedARaise. An object that is not unique for that object type, will have to be overwritten in order to be installed on the target system*
3. In the "Object Type" selection parameter, use the F4 value help selection to
   1. See a list of installed plugins and their compatible object types
   2. Enter the R3TR type of the object to be exported for the selection parameter "Object Type"
4. Execute and save file to local machine when prompted. *To meet standards for object collaboration, it is recommended, yet not technically required to use the suggested default file name*

Nugget

SAPlink nuggets is the SAPlink packaging system that enables multiple objects to be imported/exported in one handy dandy file(.nugg). Nuggets can be imported, created, and displayed from the "Nugget" tab

Import

1. Select radio button "Import Nugget"
2. Use the Nugget file containing the objects to be imported in the selection parameter "Nugget File Name"
3. WARNING: Using checkbox "Overwrite Originals", will overwrite any existing object with the same type and name on the target system. Be sure you know
4. Execute
5. Newly installed objects need to be activated manually, as all objects are installed as local & inactive
6. To see a list of all objects in a nugget, use the option to "Display Objects in Nugget"

Create New Nugget

1. Select radio button "Create Empty Nugget"
2. Type the Name of your Nugget in, this should be as unique as possible as upcoming releases may include versioning. *If you do not give your Nugget a unique name you will have problems later. Avoid things like TEST, FOO, or BAR as these are probably not unique. By the same token don't name it 9283kasdfasdbfasdfksfdf9823498jfsaa as this is useless and makes no sense.*
3. Press Execute, SAPlink will ask you where to put the Nugget file
4. You should now see your empty Nugget file in the target directory
5. You can now start adding Objects to your Nugget

Exporting/Adding Objects to Existing Nugget

1. Select radio button "Add Object to Nugget"
2. Type the Object Name
3. Pick Object Type from the value help of installed plugins
4. Pick the Nugget File you wish to add this object to
5. Press Execute
6. If everthing goes well you will see a confirmation message of the amount of bytes saved on your local machine

Display Objects in Nugget

- Use this while building or before installing a Nugget to see what you've got in there, simply point it at the File Name of a Nugget on your local machine to see the list

Bug Reporting/Feature Requests

Please submit all bug reports and feature requests at the SAPlink Group (*http://groups.google.com/group/saplink*) For more details of fixed bugs visit the SAPlink code page issues list http://code.google.com/p/saplink/issues/list

0.1.2

Bug fixes

Program ZAPLINK

- Filename changed to allow for lower case to work with OS with case sensitive filenames
- Removed erroneous "Import Successful" message after export of single object
- Added messages for required parameters on selection screen to avoid short dumps

Class Plugin (ZSAPLINK_CLASS)

- Major fixes to class plugin
  - multiple level inheritance
  - interface usage
  - friends
  - types
  - local types
  - macros
  - implementations
- Exception class bug error for SOTR_CONC types with no text maintained

Program Plugin (ZSAPLINK_PROGRAM)

- Bug fixes for text pool install for multiple or non-login languages
- Fixed for export not to include ABAP generated selection subscreens as dynpros

Enhancements

Program ZSAPLINK

- Value help added for object name after object type/plugin is specified

Class Plugin (ZSAPLINK_CLASS)

- Full support of interface usage and multi-level inheritance

Program Plugin (ZSAPLINK_PROGRAM)

- Full support of programs with multiple text languages
- Full support of programs with dynpros

Contributors

```
*/---------------------------------------------------------------------\
*| /  __ \           | |      (_) |         | |                        |
*| | /  \/ ___  _ __ | |_ _ __ _| |__  _   _| |_ ___  _ __ ___         |
*| | |    / _ \| '_ \| __| '__| | '_ \| | | | __/ _ \| '__/ __|        |
*| | \__/\ (_) | | | | |_| |  | | |_) | |_| | || (_) | |  \__ \        |
*|  \____/\___/|_| |_|\__|_|  |_|_.__/ \__,_|\__\___/|_|  |___/        |
*|---------------------------------------------------------------------|
*| Lead Developers : ed herrmann                                       |
*|                        ewherrmann+saplinkcred@gmail.com             |
*|                   dan mcweeney                                      |
*|                        daniel.mcweeney+saplinkcred@gmail.com        |
*|                                                                     |
*| Plugin Developers: Enhanced programs with dynpros - Rich Heilman    |
*|                    Tables - Thomas Jung                             |
*|                    Data Elements - Thomas Jung                      |
*|                    Domains - Thomas Jung                            |
*|                    Lock Objects - Thomas Jung                       |
*|                    Index - Thomas Jung                              |
*|                    Search Helps - Thomas Jung                       |
*|                    Structures - Thomas Jung                         |
*|                    Table Technical Settings - Thomas Jung           |
*|                    Table Types - Thomas Jung                        |
*|                    View Technical Settings - Thomas Jung            |
*|                    Views - Thomas Jung                              |
*|                    BSP Extensions - Phillip T Young                 |
*|                    Infocubes - Arjen Gideonse                       |
*|                    Smartforms - Onno Bagijn                         |
*|                                                                     |
*| Patch submissions: Michael Diehl                                    |
*|                                                                     |
*| Fearless Alpha Testers: Matt Ammons                                 |
*|                         Phillip T Young                             |
*|                         Ryan Quackenboss                            |
*|                         Sascha Kiefer                               |
*|                         Todd Rowland                                |
*\---------------------------------------------------------------------/
```

- This page was last modified 14:00, 11 Oct 2006.