**SST Generator User Guide**

**User Guide**

**Eli Lilly and Company**

**Introduction**

This document provides guidance for using the SST Generator.

This guide will be helpful for anyone needing to generate a SST for their study needs. It is assumed that anyone doing this work will have knowledge of using SAS macros, and the ability to run SAS in AWE or CLUWE.

The SST Generation Tool is a SAS macro that generates an SST based on input provided in the SST Input Spreadsheet. More details about the SST Input Spreadsheet are provided later in the user guide.

This user guide describes what is necessary to:

* Configure the SST Input Spreadsheet.
* Steps to execute the macro to generate an SST using the above input spreadsheet.

This guide will not cover the behind the scene steps performed to allow the SST to pick up latest specs for SST generation.

**Location of SST Generator**

The SST Generator has been developed, tested and validated for AWE and CLUWE. The macro is available in the below locations:

|  |  |
| --- | --- |
| **Environment** | **Macro Path** |
| AWE | \\MANGO\sddext.grp\SDDGENERAL\macro\_tools\sst |
| CLUWE | /lillyce/prd/general/rums/macro\_library/sst |

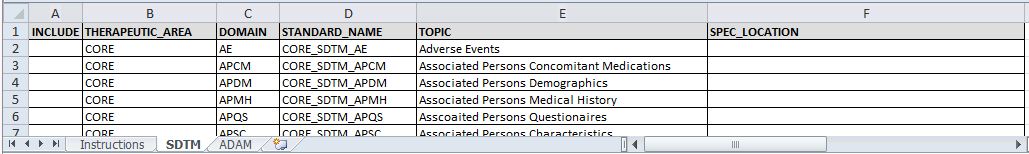
**Configuration**

The SST Generator gathers information regarding what domains to include in the SST from the SST Input Spreadsheet (SIS). The SIS is a simple spreadsheet that can be downloaded from the Business Document Repository:

<http://lillynetcollaboration.global.lilly.com/sites/CDFTProcess/Business%20Document%20Repository/Forms/AllItems.aspx>

It is stored under Topic: Data Analysis and Delivery => Sub-Topic: 1.1 Specification Development.

The SST Input Spreadsheet has a simple template as shown below:



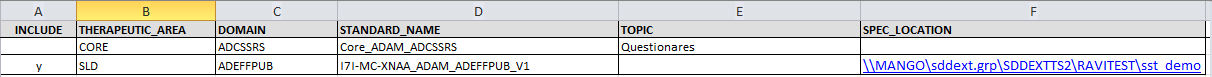
As can be seen from above screenshot, the SST Input Spreadsheet has three Excel tabs.

* Instructions : List of instructions in filling the spreadsheet.
* SDTM : List of SDTM domains to select from.
* ADAM : List of ADAM datasets to select from.

The SDTM and ADAM tabs have the below columns:

|  |  |
| --- | --- |
| **COLUMN NAME** | **DESCRIPTION** |
| INCLUDE | To include the domain in the SST, type ‘Y’ or ‘Yes’ besides the domain |
| THERAPEUTIC\_AREA | Specifies the therapeutic area of the domain |
| DOMAIN | Specifies the name of the domain |
| STANDARD\_NAME | Specifies the name of the standard. Note that this does not include the version allowing the SST Generator to always pick the latest version available. |
| TOPIC | Brief description of the domain |
| SPEC\_LOCATION | Location of the standard. If missing, then the SST Generator assumes that it is a core domain and look for the spec in a pre-defined location. |

While selections can be made from the list of domain the spreadsheet has been pre-populated with, there are times when the user may want to generate a SST with a SLD. In such cases, the user can go ahead and add another row to the end of the spreadsheet in the respective Excel tab (SDTM or ADAM) and provide the information as described in the above table. For example:



In the above screenshot, a SLD was added and a SPEC\_LOCATION was populated allowing the tool to use the SLD in the generation of the SST.

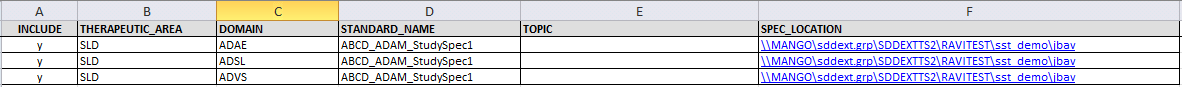
NOTE 1: The SLD should be placed in a server location that the SST Generator has access to.

NOTE 2: When specifying STANDARD\_NAME for the SLD, the SST Generator will expect the full name of the standard including the version (if any). The SST Generator will \*NOT\* select the latest version if multiple versions exist as it does for the Core specs.

The SST Generator also has the capability to use a configured study specification from another study as a SLD. In this case, the SST Generator requires an individual row to be entered in the SST Input Spreadsheet for each domain or dataset to be used from the study specification. For example, if a study team wanted to use the ADAE, ADSL, and ADVS datasets from the ABCD study specification as shown below:



In the SST Input Spreadsheet, the following entries would be required (one for each domain/dataset to be included):



Note that the STANDARD\_NAME references the same Study Specification document for all three domains / datasets – ADAE, ADSL and ADVS.

**SST Generator Usage**

Using the SST Generator requires setting up the macro library, and then calling the macro.

Syntax:

options nocenter sasautos=(sasautos "<Macro Path for environment>");

%***sst\_create*** (

specmodel = <REQUIRED - SDTM or ADAM> ,

spectype = <REQUIRED ONLY FOR SDTM – CLRM or GLS> ,

metafile = <REQUIRED - location/name of the SST Input Spreadsheet> ,

output = <REQUIRED - location/name of the SST file>,

);

Full list of Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| **PARAMETER** | **REQUIRED / OPTIONAL** | **DEFAULT** | **DESCRIPTION** |
| SPECMODEL | Required | N/A | Specify ADAM or SDTM |
| SPECTYPE | Required for SDTM | N/A | Specify CLRM or GLS |
| METAFILE | Required | N/A | Full path of SST Input Spreadsheet |
| OUTPUT | Required | N/A | Full path of output file |
| OVERWRITE | Optional | N | Overwrites the SST if it already exists |
| DEBUGMODE | Optional | N | Capability to add debug parameters |
| PRODMODE | Optional | Y | Y / N – Runs the Prod or QA version of the tool |

**The OPTIONS Statement:**

The user needs to point SASAUTOS to the location of the SST\_CREATE macro.

Example for AWE:

%let maclib = \\MANGO\sddext.grp\SDDGENERAL\macro\_tools\sst;

options sasautos=(sasautos "&maclib") mautosource nomprint;

Example for CLUWE:

%let maclib = /lillyce/prd/general/macro\_library/sst;

options sasautos=(sasautos "&maclib") mautosource nomprint;

**SPECMODEL Parameter:**

The SPECMODEL parameter expects the user to specify whether the SST generated is SDTM or ADAM. Hence, allowed values are – SDTM or ADAM. The parameter can either be in small case, upper case or mixed case. The macro will throw an error if an invalid parameter is passed.

Example: specmodel=SDTM

**SPECTYPE Parameter:**

The SPECTYPE parameter expects the user to specify whether the SST generated is CLRM or GLS. Hence, allowed values are – CLRM or GLS. The parameter can either be in small case, upper case or mixed case. The macro will throw an error if an invalid parameter is passed. The SPECTYPE parameter is required for an SDTM SST but can be deleted for an ADaM SST.

Example: spectype=CLRM

**METAFILE Parameter:**

The METAFILE parameter takes the full path of the completed SST Input Spreadsheet (SIS). Please make sure you download the latest SIS from the Business Document Repository. Also, ensure that the configuration has been done as per instructions provided in this document.

Example: metafile = \\MANGO\sddext.grp\SDDEXTTS2\demo\a1b\_cd\_efgh\_sdtm\_input1.xlsx

Please save the configured input spreadsheet in a location that is accessible by the macro when executed.

**OUTPUT Parameter:**

The OUTPUT parameter takes the full path of the SST to be generated. Note that the SST will either be generated as XLS or XML file. Please save the generated SST as XLSX if needed.

Example: output = \\MANGO\sddext.grp\SDDEXTTS2\demo\a1b\_cd\_efgh\_sdtm\_sst1.xls

**OVERWRITE Parameter:**

By default, if the same name SST exists as the OUTPUT parameter, then the macro will stop execution and alert the user that a SST with same name already exists. If user wants to overwrite the SST, parameter OVERWRITE=Y should be passed.

Example: overwrite = Y

**DEBUGMODE Parameter:**

The DEBUGMODE parameter is a provision to pass debugging parameters to the macro like mprint symbolgen, etc. By default, all debugging parameters are disabled.

Example: debugmode=mprint symbolgen

**PRODMODE Parameter:**

The PRODMODE parameter chooses the prod / qa version of the macro to execute. By default, the prod version of the macro is used. If prodmode=N, then QA version of the macro is executed. This mode is not meant for the end-user. This is used by the development team. End users should always use the production version of the macro.

Example: prodmode=N

**Example Macro Call**

**In AWE:**

dm 'clear log; clear out';

%let maclib = \\MANGO\sddext.grp\SDDGENERAL\macro\_tools\sst;

options noquotelenmax nocenter sasautos=(sasautos "&maclib") mautosource nomprint;

%let basepath = \\MANGO\sddext.grp\SDDEXTTS2\demo;

%***sst\_create*** (

specmodel=sdtm,

spectype=clrm,

metafile=&basepath\SST\_INPUT\_SPREADSHEET.xlsx,

output=&basepath\XNAA\_SDTM\_SST1.xls

);

**In CLUWE:**

%let maclib = /lillyce/prd/general/rums/macro\_library/sst;

options noquotelenmax nocenter sasautos=(sasautos "&maclib") mautosource nomprint;

%let basepath = /lillyce/qa/general/other/demo;

%***sst\_create*** (

specmodel=sdtm,

spectype=clrm,

metafile=&basepath/SST\_INPUT\_SPREADSHEET.xlsx,

output=&basepath/XNAA\_SDTM\_SST1.xls

);

**Support and Troubleshooting**

In troubleshooting the tool, please follow these steps:

Step One

Examine any messages for some clues as to what might be wrong, to see if you can fix the issue yourself.

Step Two

Talk with your colleagues to see if they have experienced the same issue.

Step Three

Check the FAQ section of the Virtual Coding Team Collaboration Site:

<http://lillynetcollaboration.global.lilly.com/sites/vct/SitePages/Home.aspx>

Step Four

Submit a ticket at the above collaboration site.

Tickets submitted should see a response within 2-4 business hours. Requests for training and/or coaching can also be submitted using the ticket system.

**Enhancement Requests**

Enhancement request are always welcome from the users. If you have any request / suggestion, please submit a service request ticket on the Reusable Code site and it will be actively monitored.

<http://lillynetcollaboration.global.lilly.com/sites/vct/Lists/Code%20RequestHelp/Public%20View.aspx>

**Revision History**

|  |  |  |
| --- | --- | --- |
| **Version Number** | **Effective Date** | **List of Major Changes** |
| 1.0 | 01-Sep-2016 | Initial Release |