

SAS Life Science Analytics Framework Macro API 1.5

Installation Qualification / Operational Qualification Checklist

Install LSAF SAS Macro API 1.5

**Version 1.00**

**Date February 6, 2017**

Checklist A: “Install LSAF SAS Macros”

***Note:*** This checklist assumes you have completed an installation of LSAF Java API 1.8.1.

***Prerequisites:*** You will need to set the following property in the **lsaf.properties** configuration file on any tcServer instance that is running:

**sassession.enable.trusted.connections=true**

| Checklist A: “Install LSAF SAS Macros” | | **Corresponding Document:** *None* | |
| --- | --- | --- | --- |
| ***#*** | ***Activity*** | ***Expected Results*** | ***Actual Results*** |
|  | Login to an admin PC. | Windows display appears. |  |
|  | Using the putty tool, create an **ssh** connection from the admin PC to the **SAS server** and login as **sasadmin**. | You’re prompted for the password. |  |
|  | Enter the *password* for sasadmin. | You’re logged into SAS server. |  |
|  | Use the **script** command to capture commands that are typed in the Unix window. The syntax is **script /var/staging\_logs/<***servername***>\_lsafmacro1.5\_<***date***>.txt** | System should indicate script has started. |  |
|  | From the admin PC, download the **LSAF Java API 1.8.1 Client** distribution from the SAS Technical Support site. After the distribution has been downloaded, use the WinSCP2 tool to copy the file to **/sso/sfw/installers** on the SAS server. Transfer it as a **binary** file using the **sasadmin** userid.  **lsaf-java-api-client-1.8.1.zip** | The file is transferred to the SAS server. |  |
|  | From the admin PC, download the **LSAF Macro 1.5** distribution from the SAS Technical Support site. After the distribution has been downloaded, use the WinSCP2 tool to copy the file to **/sso/sfw/installers** on the SAS server. Transfer it as a **binary** file using the **sasadmin** userid.  **lsaf-sas-macro-1.5.zip** | The file is transferred to the SAS server. |  |
|  | Stop the Object Spawner by typing **/sso/biconfig/940/Lev1/ObjectSpawner/ObjectSpawner.sh stop** | The Object Spawner is stopped. |  |
|  | Type **ps –ef | grep sasadmin** | The following process should **not** be running:  /bin/sh -p /sso/biconfig/940/Lev1/ObjectSpawner/ObjectSpawner.sh start2\_tag |  |
|  | Type **cd /sso/sfw/sas/940/SASFoundation/9.4** | Navigate to that directory. |  |
|  | Type **mkdir lsafapi** | Directory is created. |  |
|  | Type **cd lsafapi** | Navigate to that directory. |  |
|  | Type **unzip –q /sso/sfw/installers/lsaf-java-api-client-1.8.1.zip** | Contents of zip file are extracted. |  |
|  | Type **unzip –q /sso/sfw/installers/lsaf-sas-macro-1.5.zip** | Contents of zip file are extracted. |  |
|  | Type **touch lsaf\_macros.log** | The file is created. |  |
|  | Type **chmod 777 lsaf\_macros.log** | Permissions are updated. |  |
|  | Type **ls –al** | You see the following directories created:  lsaf-java-api-client-1.8.1  lsaf\_macros.log  lsaf-sas-macro-1.5 |  |
|  | Type **cd lsaf-sas-macro-1.5/conf/linux** | Navigate to that directory. |  |
|  | Type **cp log4j.properties log4j.properties.original** | A backup of the file is made. |  |
|  | Type **vi log4j.properties** | File opens for edit. |  |
|  | Change the **log4j.category.com.sas.hls.macro** property from **DEBUG** to **INFO** so that the line looks like the following:  **log4j.category.com.sas.hls.macro=INFO** | Property is set to INFO. |  |
|  | Hit **Esc** and type **:x** to save the file. | File is saved with changes made. |  |
|  | Type **cd /sso/biconfig/940/Lev1/SDDApp/WorkspaceServer**  **Note:** For single SAS server installs, this application name will be SASApp. | Navigate to that directory. |  |
|  | Type **vi sasv9\_usermods.cfg** | File is opened for edit. |  |
|  | After the comment block, add the following lines:  **-insert sasautos "/sso/sfw/sas/940/SASFoundation/9.4/lsafapi/lsaf-sas-macro-1.5/sasmacros"**  **-JREOPTIONS (**  **-Dsas.app.class.dirs=/sso/sfw/sas/940/SASFoundation/9.4/lsafapi/lsaf-java-api-client-1.8.1/lib:/sso/sfw/sas/940/SASFoundation/9.4/lsafapi/lsaf-sas-macro-1.5/lib**  **-Dsession.strategy.classname=com.sas.hls.client.impl.trusted.TrustedContextSessionStrategyImpl**  **-Dsas.lsaf.macros.trusted=true**  **-Dlog4j.configuration=file:/sso/sfw/sas/940/SASFoundation/9.4/lsafapi/lsaf-sas-macro-1.5/conf/linux/log4j.properties**  **)**  **/\* this prevents a classpath not set warning from javaobj \*/**  **-SET CLASSPATH !CLASSPATH**  ***Note:*** Copy and paste this text from the sasv9\_usermods.txt file in our staging area. | File is modified. |  |
|  | Hit **Esc** and type **:x** to save the file.  ***Note:*** The **log4j.properties** file is configurable and should be modified to point to the correct path to the **lsaf\_macros.log** file. | File is saved with changes made. |  |
|  | Type **more sasv9\_usermods.cfg** | Verify that changes have been saved. |  |
|  | Start the Object Spawner by typing **/sso/biconfig/940/Lev1/ObjectSpawner/ObjectSpawner.sh start** | Object Spawner is started and system displays:  Spawner is started (pid *nnnnn*)... |  |
|  | Type **ps –ef | grep sasadmin** | The following process should be running:  /bin/sh -p /sso/biconfig/940/Lev1/ObjectSpawner/ObjectSpawner.sh start2\_tag |  |
|  | Hit **Ctrl-D** to stop the script command. | System indicates script stopped running. |  |
|  | Type **exit** | The sasadmin userid is logged off the SAS server. |  |
|  | Repeat **steps A1 – A30** to deploy the LSAF SAS Macros to any additional SAS processing servers. | Steps repeated as necessary. |  |

***Signature below indicates completion of Checklist A, items A1 – A31, above.***

**Name (print or type): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Sign-off : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**