SailPoint Connector for Top Secret

Version 4.0.03

Rev 1.2

Administration Guide

Copyright and Trademark Notices

Contents

Integrating SailPoint with Top Secret Source

Operating as a security administration Connector running on the various platforms in the enterprise, Connector processes and transfers security commands and data between managed system and SailPoint.

This document is intended for the above products and IdentityIQ System Administrators and assumes an advance level of technical knowledge.

SailPoint Connector for Top Secret includes the following features:

- Full aggregation
- Provisioning
- Monitoring of Top Secret activities to update SailPoint in real time with User and password changes

The following topics are discussed in this chapter:

Connector Facilities

The Connector facilities enable the Managed System (MS) to be monitored and managed by SailPoint. The Connector facilities include:

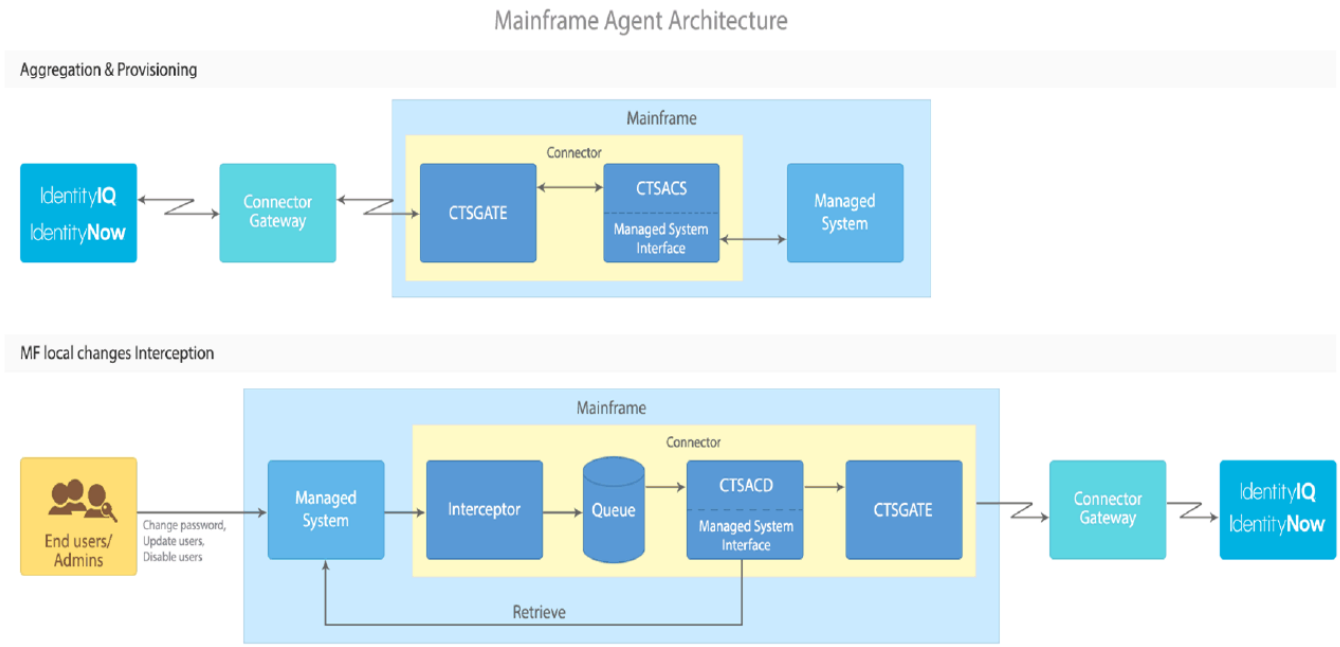
- **Managed System data aggregation to SailPoint** – The data aggregation procedure which is initiated from the SailPoint is controlled by the Connector. The Connector ensures that relevant data is aggregated from the Top Secret Managed System database to the SailPoint database. After all the data of the Top Secret Managed System has been aggregated, a consolidated picture of SailPoint data can be viewed in SailPoint.
- **Translation and execution of SailPoint commands** – Security-related commands (for example, add user, change password or phrase) which are initiated by SailPoint are handled by the Connector. The Connector translates these commands into the format and language recognized by the Managed System and executes them in the Managed System.
- **Managed System activity monitoring** – The Connector intercepts events that occur in the managed system which are initiated from within the platform environment. For example, the MS administrator adds, modifies, or deletes MS users and groups or an MS user changes his/her password or Managed System administrator changes a password for a user. When a significant event occurs, either data defining the event or an up-to-date updated entity is sent by the Connector to IdentityIQ. This functionality is accomplished using the Online

Interceptor component. Note: Online Interceptor requires IdentityIQ setting. For more information, see SailPoint Quick Reference Guide for Gateway Connectors.

- **Secured Communication** – Secured communication can be performed externally using AT-TLS or internally using Transmitted Data Encryption. For more information, see Secured Communication.
- **Stored Data Encryption** – All sensitive data which is stored temporarily in Connector for Top Secret (for example, sensitive security information that is written to the Connector queue file) is encrypted using a stored data encryption key.

Connector Components in Detail

The following diagram illustrates the major components of the Connector, their relationship with one another, and the flow of data between them. This diagram represents the connection between SailPoint and a single Connector installation with a single MS. In practice, multiple MS on different platforms can be administered by multiple Connector installations.



Further information on each of the components above is available below.

- **Connector Gateway** – Resides between SailPoint and Mainframe Connector (CTSGATE) and is responsible for the communication between these two components.
- **Connector** – Enables the interception of managed system events and the translation of SailPoint commands to each specific managed system terminology. The Managed System Interface component of the Connector is a flexible API which is customized for each managed system.

- **CTSGATE** – Mainframe side communicator gateway. Responsible for communication with Connector Gateway and CTSACS /CTSACD. It is also responsible for starting and stopping CTSACS and CTSACD.
- **CTSACS** – Transaction Server – is responsible for SailPoint transactions handling. Note: May be 1 to 3 transaction servers.
- **Managed System Interface** – Responsible on the interface with Top Secret itself. It translates SailPoint transactions into Top Secret commands (provisioning transactions). It uses Top Secret's API to aggregate Top Secret's entities from Top Secret to provisioning module.
- **Managed System** – TSS
- **CTSACD** – The Notification Server, which reads events written to Queue by Interceptor, retrieve relevant entity up-to-date status from Top Secret and pass entity data to CTSGATE.
- **Interceptor** – Responsible for intercepting Mainframe local changes done by Top Secret administrators and end-users and writes them to Queue.

  Two types of interceptors can be used in the Connector:

  - **Online Interceptor** – Detects security administration events as they occur.
  - **Offline Interceptor** – Detects security administration events in batch.
- **Connector Queue** – The Connector queue is a dataset in which all security data is saved before it is sent to SailPoint via the Notification Server. If communication between Connector and SailPoint fails, Managed System events continue to be stored in the Connector queue and are sent to SailPoint when communication is re-established.

Installation

This chapter provides the required information and step-by-step instructions involved in the installation of Connector for Top Secret.

It is recommended that before beginning the installation procedure you review the content of Before Installing the Connector, installation steps in this chapter and, the contents of Top Secret Support Customization.

Before Installing the Connector

Before you begin the connector's installation, ensure the following requirements are met:

Hardware and Software Requirements

The following table outlines the software and hardware requirements to support the Connector for Top Secret .

Component	Description
Hardware Requirements	The Connector operates on any hardware configuration supported by any of the supported operating systems. • Supported operating systems: Connector for Top Secret can be installed on z/OS release 2.4 through 2.5. • Supported Top Secret levels: Connector for Top Secret can be used to manage Top Secret release 14 through 16. Software requirements are as follows: • Job Entry Subsystem: JES2 or JES3 • TSO/E • SMP/E

Component	Description
	• TCP/IP • ISPF or any other text editor that allows submitting jobs and checking their output
Disk Type	The Connector datasets may reside on any disk that is supported by MVS. For example, **3390** IBM disk type is supported.
Disk Space	700 cylinders of a 3390 device are sufficient to store the datasets installed by the Connector installation procedure.

> **Caution**
>
> Any Mainframe tool which handles x37 abends should be avoided or disabled when using the SailPoint Mainframe connector.
>
> If the tool allows it, you should add SailPoint Mainframe connector to the tool's exclude list.

Communication Parameters Coordination

For Top Secret Connector to communicate successfully with a provisioning module, several parameters must be coordinated between provisioning module's Application or Source Definition, Connector Gateway and Top Secret Connector.

The table below summarizes these parameters. Each row in the table describes a set of parameters in all or some of the components which must be coordinated. For more information and description of Connector Gateway parameters, see *SailPoint Integration Guide* or *SailPoint Quick Reference Guide for Gateway Connectors* depending on Connector Gateway release. For full description of IdentityIQ Application definition, see *SailPoint IdentityIQ Administration Guide*. For full description of IdentityNow Source definition, see Top Secret *Source Configuration for IdentityNow*.

Summary of Required Parameter Coordination

For the Connector to communicate successfully with SailPoint, the following Connector installation and environment parameters must be coordinated with parameters specified in the Connector Gateway and in SailPoint:

Parameter name	Top Secret Connector	Connector Gateway	IdentityIQ Application Definition	IdentityNow Source Definition
RSSNAME	MSCS in RSSPARM member in PARM lib-		MSCS Name parameter in Connector Gateway/Connec-	Connector name in Top Secret source

Parameter name		Top Secret Connector	Connector Gateway	IdentityIQ Application Definition	IdentityNow Source Definition
		rary		tor Manager Settings	
RSSTYPE		RSSP_TYPE in RSSPARM member in PARM library. Must be TSS.		MS Type selected for Application Type field should be Top Secret - Full	Top Secret source
MF_PORT		PORT parameter in ECAPARM member in PARM library	**port** parameter in SM section in **init.xml** file.		
SECURED	TRANSMITTED DATA ENCRYPTION	ENCR_EXT_ ACT parameter in CTSPUSR member in PARM library		Encryption parameter in Connector Gateway/ Connector Manager Settings	Not supported
	TLS	AT-TLS implementation	Implementation steps have to be performed	Implementation steps have to be performed	Implementation steps have to be performed

- **RSSNAME** – In Top Secret Connector, the name is set in the **DEFPARMS %RSSNAME%** parameter in the INSTALL library during installation. After installation, this name is automatically set as the MSCS name in RSSPARM member in the PARM library. The MSCS name appears in column 1 of each parameter line (unless **ALL_RSS** is set in the line).

  This name can be up to 32 characters long. For RSSNAME values with more than eight characters, special adjustments must be performed at the end of the Top Secret Connector installation procedure. It is therefore recommended to use RSSNAME values consisting of no more than eight characters. For more information, see11 – Adjust for Longer Managed System Names .

  The same name must be specified for **%RSSNAME%** and for the MSCS Name parameter in Connector Gateway/Connector Manager Settings in Provisioning Module Application Definition (for IdentityIQ) or for the Connector name in Top Secret Source definition (for IdentityNow).

- **RSSTYPE** – During installation RSSTYPE is specified in the **DEFPARMS %RSSGTYPE%** parameter in the INSTALL library. After installation, the RSSTYPE can be found in the RSSPARM RSS_TYPE parameter in the

PARM library. For the Top Secret connector, the value must be **TSS**. The MS Type in Provisioning module Application definition must be TopSecret- Full.

- **MF_PORT** – TCP/IP port number defined for the Top Secret Connector for communication with Provisioning Module. Connector for Top Secret's CTSGATE uses two consecutive TCP/IP ports to communicate with Provisioning Module. By default, the ports used are 2470 and 2471. Verify that these ports are not already in use. If they are in use, locate two other consecutive ports which are available.

  Specify the lower of these two ports when you are instructed to provide a value for parameter PORT during the Connector installation. The same port number must be specified in the port field in the SM section of the Connector Gateway **init.xml** file. for more information, see 9 – Customize Communication Settings.

- **SECURED_COMMUNICATION** – Secured communication can be implemented by using Transmitted Data Encryption or TLS. One of the following options can be selected. When TLS is selected, Transmitted Data Encryption must be set off in all components.

  - **TRANSMITTED DATA ENCRYPTION** – Communication security is gained by encrypting the transmitted data using an encryption key dataset.

    If Encryption parameter is set to Off in Provisioning Module Application Definition, the value set for **ENCR_EXT_ACT** parameter in **CTSPUSR PARM** member must be **N**. Else the value for **ENCR_EXT_ACT** parameter in **CTSPUSR PARM** member must be **Y**.

    For more information, see *9.4 – Set up secured communication* in 9 – Customize Communication Settings.

  - **TLS**: Communication is secured using TLS. Requires implementation of steps in all components.

    In the Mainframe, TLS communication must be configured using AT-TLS. With AT-TLS, the TLS processing is performed by TCP/IP and is transparent to the application (CTSGATE). Hence no settings are required in Connector for Top Secret parameters, except for setting the Transmitted Data Encryption to **Off** as described above.

    For more information, see Secured Communication.

Protecting Temporary Datasets

Temporary data sets are considered protected from any access except by the job or session that created them, and hence are not required to be protected by Top Secret. These files are allocated as new files, held with exclusive SYSDSN ENQ. However in situations like system failure, a temporary data set could be left unprotected. Such file can be accessed by any user, unless protected by Top Secret.

The Connector for Top Secret handles the temporary file allocated by the EXECOUT DD statement in a special way, which causes failures due to security violation when temporary files are protected by Top Secret.

The EXECOUT file is defined in the connector procedure and is created when the connector starts, under the connector User ID. But, when processing requests received from SailPoint, this file is accessed by the Connector for Top Secret under the Managed System Administrator User ID. When the temporary files are protected by Top Secret, the only user that can access temporary files is the user that allocates them. So, when the connector tries to write to the file under the Managed System Administrator User ID. If this User ID does not have permission to access the file, Top Secret fails the request.

During installation, the EXECOUT DD statement allocates the file on VIO. Temporary datasets allocated to VIO are not protected by the Top Secret so no error occurs. But, if VIO is not used in the system, or if VIO is not allowed for large files, the problem occurs.

The above problem can be prevented as follows:

- Allow the files allocated by the Connector for Top Secret to be on VIO.

  Or

- Allocate the EXECOUT DD statement to a permanent file. The Connector for Top Secret utilizes source JCL for started tasks (STCJOB) for starting the procedures to maintain these permanent datasets.

  To use this option, set the name of the STCJOBs library (defined in IEFJOBS DD statement in MSTJCLxx system parameter) to which Connector for Top Secret STCJOBs would be copied, as the value of the **%STCJOBS%** DEFPARMS parameter.

Enhanced Data Integrity Considerations

If Enhanced Data Integrity function is active, ensure that all Connector for Top Secret files are set in its Exclude list. For more information, refer to the following:

- *z/OS DFSMS Using Data Sets*
- *IFGPSEDI (enhanced data integrity)* section in the *z/OS Initialization and Tuning Reference Guide*

Using STCJOBs

STCJOBS can be used to start the Connector for Top Secret started tasks. They are required when temporary datasets are protected (see Protecting Temporary Datasets), but can be used regardless of this protection.

To use STCJOBs, specify the name of the system library for source JCL for started tasks (STCJOB) to which the Connector for Top Secret STCJOBs would be copied. This library must be defined in the IEFJOBS DD statement in the MSTJCLxx system **parmlib** member.

The STCJOBs would be copied to this library, while renamed using the DEFPARMS **%PROCPREFS%** value as the first 3 characters of their names.

When using STCJOBs, the Connector for Top Secret started tasks procedures can be placed in one of the following libraries, according to DEFPARMS **%PROCLIB%** parameter value:

- Copy the Connector for Top Secret started tasks procedures to the system JCL procedures library (PROCLIB). When this option is used, the name of the system JCL procedures library, must be specified in the DEFPARMS **%PROCLIB%** parameter. The started tasks procedures would be copied to this library and renamed using the **%PROCPREFS%** value as the first 3 characters of their name. When the STCJOBs are started, the system would search the standard procedure libraries for the started tasks procedures.

  Or

- Leave the Connector for Top Secret started tasks procedures in the Connector PROCLIB and use the JCLLIB JCL statement to point to this library. When this option is used, LOCALCOPY must be specified in the DEFPARMS **%PROCLIB%** parameter. The procedures are copied to the Connector PROCLIB library and renamed using the **%PROCPREFS%** value as the first 3 characters. When the STCJOBs are started, the system would search for the procedures using the JCLLIB statement in the STCJOB.

> **Note**
>
> When STCJOBs are used, the files allocated by EXECOUT DD statements which are temporary by default, are allocated as permanent files.
>
> When STCJOBs are used, the IGD17054I message could be displayed which is not an error message and can be ignored.
>
> In z/OS V1R13, issuance of the IGD17054I message is controlled by the value specified for the SUPPRESS_DRMSGS parameter, in the IGDSMSxx PARMLIB member. Beginning in z/OS V2R1, issuance of the IGD17054I message is controlled by the new SUPPRESS_SMSMSG parameter, also in the IGDSMSxx PARMLIB member.
>
> For more information, see the *z/OS Initialization and Tuning Reference Guide*of the appropriate z/OS version.

New Installation

This section describes the procedures involved in creating a new installation of Connector for Top Secret.

Installation Summary

The installation of Connector for Top Secret consists of the following procedures:

The following table summarizes the procedures for installing Connector for Top Secret.

Procedure	Step	Job/Member Name	Description
1	**Set the Parameter Values**		
2	**Prepare Installation IMAGE from TRS file**		
	2.1		Transfer the INSTALL.TRS file Using FTP Binary
	2.2		UNCOMPRESS the TRS File
	2.3	$RECEIVE	Tailor the $RECEIVE Job
	2.4	$RECEIVE	RECEIVE the Installation IMAGE
3	**Allocate and Load Connector INSTALL Library**		
	3.1	$LOADINS	Copy, Edit, and Run Member $LOADINS
4	**Allocate and Load Connector for Top Secret Installation Libraries**		
	4.1	LOADCTS	Tailor LOADCTS Member
	4.2	LOADCTS	Submit Job to Allocate and Load Connector for Top Secret Libraries
5	**Tailor Connector for Top Secret Members with Site Parameters**		
	5.1	DEFPARMS	Assign Installation Parameters
	5.2	CHNGEPRS	Modify Connector for Top Secret Members
6	**6 – Copy Connector for Top Secret Procedures, STCJOBs and Other Members**		
	6.1	CPYMTSS	Submit the CPYMTSS Job to Perform the Copy
7	**Customize Connector for Top Secret Installation Parameters**		
	7.1	CTSPARMJ	Create CTSPARM Module
	7.2	RSSPARM	Assign RSSPARM Parameter Values
8	**Format Connector for Top Secret Datasets**		
	8.1	FORMCTS	Edit and Run Member FORMCTS
9	**Customize Communication Settings**		
	9.1		Verify TCP/IP Connectivity
	9.2	ECAPARM	Connector for Top Secret Gateway Communication Parameters
	9.3		Define TCP/IP DATA file
	9.4		Set up Secured Communication

Procedure	Step	Job/Member Name	Description
10	**Define Connector for Top Secret in Top Secret**		
	10.1	CTSTSS	Define a multi-user facility to be used by Connector for Top Secret
	10.2	CTSTSS	Define Connector for Top Secret Started Tasks as valid started tasks in Top Secret
	10.3	CTSTSS	Define the Connector STC ACID to OMVS
	10.4	CTSTSS	Set permissions to Connector Datasets
	10.5	CTSTSS	Protect the Encryption Keys Datasets
	10.6	CTSTSS	Grant CTSGATE with authority to use TCP/IP stack
	10.7	CTSTSS	Grant users permission to facility CTSA
11	**Adjust for Longer Managed System Names**		
12	**Adjusting Managed System Administrator Attributes**		
	12.1	Provide Managed System Administrator Passwords	
	12.2	Verify Managed System Administrator permissions	
13	**Add Connector for Top Secret Libraries to the MVS Authorized Libraries List**		
14	**Review the Installation**		
	14.1		Verify that the Connector for Top Secret Gateway is Installed Properly
	14.2		Verify that the Top Secret Connector Communicates Successfully with SailPoint
	14.3		Verify Top Secret Connector Top Secret Interface
15	**(Optional) Local Changes Migration**		
16	**Configure Automated Startup of Connector for Top Secret**		
17	**Customizing Top Secret Support**		
18	**Post Installation Checks**		

1 – Set the Parameter Values

This section describes about the various tables containing all the parameters for which values must be set.

Set the **Value** column with the selected values, which will be used later to set the installation jobs and parameters.

Datasets Allocation Considerations

The first step of the installation process creates **IMAGE** files from which the Connector files are loaded in later steps.

Ensure that you select different prefix + version combinations to the **IMAGE** files (**%instpref%** in Installation Upload and IMAGE Datasets Parameters) and to the Connector files (**xPREFx** + **xVERx** in Connector for Top Secret Datasets Allocation Parameters). Otherwise, the installation fails due to duplicate datasets.

When selecting high level qualifiers for the files, ensure the product installer has ALTER authority for these files.

Some of the parameters below are the unit name and volume serial number to be used for product datasets allocation. If the datasets must be SMS managed, leave these parameters empty (except for SPCVOL which must be *). If the datasets must not be SMS managed, specify the values for unit and volume serial number to ensure proper handling of the file.

Installation Upload and IMAGE Datasets Parameters

Parameter	Description	Value
Upload dataset name	The name of the dataset into which the product would be transferred using FTP.	
%xmitlib%	Name of the library into which the Connector for Top Secret uploaded file would be uncompressed.	
%instpref%	Prefix selected for Connector for Top Secret installation **IMAGE** datasets that are later used to install Connector for Top Secret.	
%UNIT%	• For non-SMS managed datasets specify UNIT(unitname) where unitname is the name of DASD unit where Connector for Top Secret Installation **IMAGE** datasets would be placed. • For SMS-managed datasets, specify null.	
%VOLUME%	• For non-SMS managed datasets specify VOLUME(volser) where volser is the volume serial number on which Connector for Top Secret Installation **IMAGE** datasets would be placed. • For SMS-managed datasets, specify null.	

Allocate and Load Connector for Top Secret Datasets

This section describes information associated with $LOADINS and LOADCTS jobs.

Considerations before setting Connector for Top Secret file allocation parameters:

- Each Connector for Top Secret file and library used to install and operate the product is assigned a type according to its usage. Each type has separate allocation parameters (prefix (HLQ), version, unit, volser) and is identified by a prefix assigned to its allocation parameters variables.

  File types are:

  - Installation – IL
  - Operation – OL
  - SMP/E CSI – SPC
  - SMP/E files – SPA
  - DLIBs – SPD

  Assign values to the allocation parameters in the following table according to the file types and site standards.

- Some sites use special naming conventions for Load Module libraries. Therefore, the name of the Connector for Top Secret Load library is a user-defined parameter.
- The datasets installed by the Connector for Top Secret installation procedure are described in Appendix C: Connector for Top Secret Datasets and JCL Procedures. The total DASD space they require is listed under Hardware and Software Requirements.

Connector for Top Secret Datasets Allocation Parameters

Parameter	Description	Default Value	Value
DLPREFS	Installation **IMAGE**datasets prefix. The value of this parameter should be the same value specified for %**instpref**% above.	-	
JOBNAME	Job name prefix (1 to 6 characters) to be used for jobs submitted during the Connector for Top Secret installation process.	CTLSA	
JOBCARD	Job card data to be used for all jobs submitted during the Connector for Top Secret installation procedure. Maximum length is 43 characters. The value must not contain blanks and must be enclosed in apostrophes. Example of the use of JOBNAME and JOBCARD parameters:	SA,CLASS=A,MSGCLASS=X	

Parameter	Description	Default Value	Value

	`JOBNAME=CTSINS`
	`JOBCARD=',CTSINST,CLASS=A,MSGCLASS=X'`
	Resulting job card:
	`//CTSINS01 JOB,CTSINST,CLASS=A,MSGCLASS=X`		

STEPLIB	The Connector for Top Secret Load Module library. Note that the last qualifier of this library name must contain a maximum of four characters.	CTLSA.V400.LOAD	
ILPREFS	High level dataset name qualifier (prefix) of the Connector for Top Secret installation libraries.	CTLSA	
ILVERS	Second level dataset name qualifier (version) of the Connector for Top Secret installation libraries.	V400	
ILUNITS	Name of DASD unit where Connector for Top Secret installation libraries will be placed. • For non-SMS managed datasets, specify a generic unit (for example, 3390). • For SMS-managed datasets, it is recommended to specify a null value (that is, specify **ILUNITS=,**).	@@@@	
ILVOLS	• For non-SMS managed datasets specify the volume serial number on which Connector for Top Secret installation libraries will be placed. • For SMS-managed datasets, if a null value was specified in ILUNITS, specify a null value for this parameter as well (that is, specify **ILVOLS=,**).	####	
OLPREFS	High level dataset name qualifier (prefix) of the Connector for Top Secret operation datasets.	CTLSA	
OLVERS	Second level dataset name qualifier of the Connector for Top Secret operation datasets.	V400	
OLUNITS	Name of DASD unit where Connector for Top Secret operation datasets will be placed. • For non-SMS managed datasets, specify a gen-	@@@@	

Parameter	Description	Default Value	Value
	eric unit (for example, 3390). • For SMS-managed datasets, it is recommended to specify a null value (that is, specify **OLUNITS=,**).		
OLVOLS	• For non-SMS managed datasets specify the volume serial number on which Connector for Top Secret operation datasets will be placed. • For SMS-managed datasets, if a null value was specified in OLUNITS, specify a null value for this parameter as well (that is, specify **OLVOLS=,**).	####	
SPCPREF	High level dataset name qualifier (prefix) of the Connector for Top Secret SMP/E CSI dataset.	CTLSA	
SPCVER	Second level dataset name qualifier of the Connector for Top Secret SMP/E CSI dataset.	V400	
SPCVOL	• For non-SMS managed datasets specify the volume serial number on which Connector for Top Secret SMP/E CSI dataset will be placed. • For SMS-managed datasets, you may specify an asterisk (that is, specify **SPCVOL=*,**).	####	
SPAPREF	High level dataset name qualifier (prefix) of the Connector for Top Secret SMP/E datasets.	CTLSA	
SPAVER	Second level dataset name qualifier of the Connector for Top Secret SMP/E datasets.	V400	
SPAUNIT	Name of DASD unit where Connector for Top Secret SMP datasets will be placed, • For non-SMS managed datasets, specify a generic unit (for example, 3390). • For SMS-managed datasets, it is recommended to specify a null value (that is, specify **SPAUNIT=,**).	@@@@	
SPAVOL	• For non-SMS managed datasets specify the volume serial number on which Connector for	####	

Parameter	Description	Default Value	Value
	Top Secret SMP/E datasets will be placed. • For SMS-managed datasets, if a null value was specified in SPAUNIT, specify a null value for this parameter as well (that is, specify **SPAVOL=,**).		
SPDPREF	High level dataset name qualifier (prefix) of the Connector for Top Secret SMP/E distribution libraries.	CTLSA	
SPDVER	Second level dataset name qualifier of the Connector for Top Secret SMP/E distribution libraries.	V400	
SPDUNIT	Name of DASD unit where Connector for Top Secret SMP/E distribution libraries will be placed. • For non-SMS managed datasets, specify a generic unit (for example, 3390). • For SMS-managed datasets, it is recommended to specify a null value (that is, specify **SPDUNIT=,**).	@@@@	
SPDVOL	• For non-SMS managed datasets specify the volume serial number on which Connector for Top Secret SMP/E distribution libraries will be placed. • For SMS-managed datasets, if a null value was specified in SPDUNIT, specify a null value for this parameter as well (that is, specify **SPDVOL=,**).	####	

DEFPARMS Parameters

Parameters	Description	Default value	Value
%HOLDCLASS%	Single-character Held output class for Top Secret Connector procedures and jobs.	X	
%DUMPCLASS%	Single-character Held output class for dumps, diagnostic messages and snaps. This type of output should go to a held output-class which is cleaned frequently (if not printed) in order not to cause spool fill-out.	X	
%WORKUNIT%	Unit name for temporary datasets (for example, SYSDA,	SYSALLDA (exists	

Parameters	Description	Default value	Value
	SORTWORK).	in all z/OS systems)	
%RSSGTYPE%	Type of security product installed. Default value must not be changed.	TSS	
%RSSNAME%	Managed System name. The name must correspond to the Managed system name defined in SailPoint. It is recommended to use a name which is up to 8 characters long. If the name is longer than 8, follow the instructions mentioned in 11 – Adjust for Longer Managed System Names. **Note** For more information, see Communication Parameters Coordination.	MVSTSS	
%BRODCAST%	Name of the system BRODCAST dataset. The file is required when new users are defined.	SYS1.BRODCAST	
%PROCPREFS%	First three characters of the Connector for Top Secret JCL procedure, and optionally STCJOB names, after they are copied to the requested library, as specified in the **%PROCLIB%** and **%STCJOBS%** parameters. This determines the name of started tasks used by Connector for Top Secret. When LOCALCOPY is set for **%PROCLIB%**, the value for this parameter must not be CTS, ACF, TSS or RCF. **Note** When %PROCLIB% is not DONTCOPY or LOCALCOPY, ensure that there are no procedures in the PROCLIB concatenation which start with the value set for this parameter. If STCJOBs are used, verify the same for the system STCJOBs concatenation. Refer the following tables for the inter-relations between	CTS	

Parameters	Description	Default value	Value
	%PROCPREFS%, **%PROCLIB%** and **%STCJOBS%** parameters: • Allowed Combination of Values for %PROCPREFS%, %PROCLIB%, and %STCJOBS% Parameters • Disallowed Combination of Values for %PROCPREFS%, %PROCLIB%, and %STCJOBS% Parameters		
%PROCLIB%	Name of the System JCL Procedure library to which the Connector for Top Secret procedures will be copied. The copy will be done while renaming the procedures using the value specified for **%PROCPREFS%** as the first 3 characters. Other possible values: • **DONTCOPY** – This reserved value can be specified to prevent the Connector for Top Secret procedures from being copied to any procedures library. **Note** DONTCOPY in **%PROCLIB%** forces DONTCOPY in **%STCJOBS%**. • **LOCALCOPY** – This reserved value can be specified to copy the Connector for Top Secret procedures to the Top Secret Connector procedures library while renaming them using the **%PROCPREFS%** value as the first 3 characters. LOCALCOPY can be used when **%STCJOBS%** value is not DONTCOPY. The procedures will be used by the STCJOBs using the JCLLIB statement.	SYS2.PROCLIB	

Parameters	Description	Default value	Value
	Refer the following tables for the inter-relations between **%PROCPREFS%**, **%PROCLIB%** and **%STCJOBS%** parameters: • Allowed Combination of Values for %PROCPREFS%, %PROCLIB%, and %STCJOBS% Parameters • Disallowed Combination of Values for %PROCPREFS%, %PROCLIB%, and %STCJOBS% Parameters		
%STCJOBS%	Name of the System library for source JCL for started tasks (STCJOB) to which the Connector for Top Secret STCJOBs would be copied. This library should be defined in the IEFJOBS DD statement in the MSTJCLxx system PARMLIB member. The reserved value DONTCOPY can be specified to prevent the Connector for Top Secret STCJOBs from being copied. DONTCOPY must not be used when temporary datasets are protected by Top Secret. **Note** For more information see Protecting Temporary Datasets and Using STCJOBs. Refer the following tables for the inter-relations between **%PROCPREFS%**, **%PROCLIB%** and **%STCJOBS%** parameters: • Allowed Combination of Values for %PROCPREFS%, %PROCLIB%, and %STCJOBS% Parameters • Disallowed Combination of Values for %PROCPREFS%, %PROCLIB%, and %STCJOBS% Parameters	DONTCOPY	

Allowed Combination of Values for %PROCPREFS%, %PROCLIB%, and %STCJOBS% Parameters

The following table describes the allowed combination of values for **%PROCPREFS%**, **%PROCLIB%**, and **%STCJOBS%** parameters within DEFPARMS members.

%PROCPREFS%	%PROCLIB%	%STCJOBS%	Results / notes
CTS/xxx	<proclib>	DONTCOPY	Procedures are copied with specified prefix to <proclib> and STCJOBs are not copied.
CTS/xxx	<proclib>	<stcjobs library>	Procedures are copied with specified prefix to <proclib> and STCJOBs are copied with specified prefix to <stcjobs library>.
CTS/xxx	DONTCOPY	Any value	Nothing is copied. If required customer must manually copy the procedures and STCJOBs.
xxx	LOCALCOPY	<stcjobs library>	Procedures are copied with specified prefix to Connector PROCLIB and STCJOBs are copied with specified prefix to <stcjobs library>. The procedures will be used by the STCJOBs using the JCLLIB statement.

Disallowed Combination of Values for %PROCPREFS%, %PROCLIB%, and %STCJOBS% Parameters

The following table describes the combination of values that are not allowed for **%PROCPREFS%**, **%PROCLIB%**, and **%STCJOBS%** parameters within DEFPARMS member.

%PROCPREFS%	%PROCLIB%	%STCJOBS%	Notes
CTS/xxx	LOCALCOPY	DONTCOPY	Prevented with an error message by CHNGEPRS job as the procedures can-

%PROCPREFS%	%PROCLIB%	%STCJOBS%	Notes
			not be started.
CTS/RCF/ACF/TSS	LOCALCOPY		Prevented with an error message by CHNGEPRS job as there are already members with these names in Connector PROCLIB.

RSSPARM Parameters

Parameters	Description	Default value	Value
CTSA_ID	Unique 4-character identifier for Connector for Top Secret to use. If more than one instance of Connector for Top Secret is installed on the platform, each instance should have a unique ID.	`<xxx>R` where `<xxx>` is the value specified for **%PROCPREFS%** in DEFPARMS	
RSS_TYPE	Managed System type. Specify: TSS	TSS	TSS
RSS_WORK_DIR	The prefix used to dynamically allocate working datasets	By default, the prefix used consists of the following: `<prefix>.<version>.<RSS_NAME>` where • `<prefix>` is the value specified for OLPREFS • `<version>` is the value specified for OLVERS • `RSS_NAME` is the value specified for **%RSSNAME%** in DEFPARMS	

Communication Parameters (CTSPUSR and ECAPARM Parameter Members)

Refer to Communication Parameters Coordination before setting values for the parameters mentioned in the table below.

Parameter	In Member	Description	Default value	Value
ENCR_EXT_ACT	CTSPUSR	Select whether Transmitted Data Encryption is required. Possible values are: • **Y** – Enables Transmitted Data Encryption. Default. • **N** – Disables Transmitted Data Encryption. If TLS will be used for secured communication, the value has to be set to N. For more information, see Secured Communication	Y	
PORT	ECAPARM	Lower of the two consecutive port numbers to be used for TCP/IP communication.	2470	
NUMSRV	ECAPARM	The number of Transaction Servers (CSs) defined for Connector for Top Secret. This number determines the number of SailPoint requests that can be processed concurrently. Each Transaction Server handles a single request at a time. The maximum number of CS's that can be specified is 3. **Note** The number of Notification Servers (CDs) is one. This number cannot be changed.	2	
IPLIST	ECAPARM	(*Optional*) Used to validate the incoming IP addresses by CTSGATE. For full description and syntax, see Configuring Incoming IP Address Validation.	-	

2 – Prepare Installation IMAGE from TRS File

Use this procedure to prepare an installation **IMAGE** from the TRS file.

2.1 – Transfer the INSTALL.TRS file using FTP Binary

1. Using FTP, upload file INSTALL.TRS from the TRS file to the target system, using BINARY format.
    - The name of the uploaded dataset must be the name set for **upload dataset name** in the Installation Upload and IMAGE Datasets Parameters table.
    - The uploaded dataset should be pre-allocated with LRECL=1024, BLKSIZE=6144, RECFM=FB
    - The dataset size is approximately 45 cylinders on a 3390 device.
2. At the command prompt, specify the following commands to upload the dataset:

    ```
    FTP <mvssystem>
    ```

    where `<mvssystem>` is the DNS name of your host system.
3. Specify the user ID and password or phrase; specify the following commands:

    ```
    bin
    put <drive>:\TSS\Install\install.trs
    '<upload_dataset_name>'
    ```

2.2 – UNCOMPRESS the TRS file

> **Important**
> Wait for the previous installation job to finish before continuing with this procedure.

1. Tailor the following TRSMAIN job. Replace `upload_dataset_name` and **%xmitlib%** with the values set in the Installation Upload and IMAGE Datasets Parameters table.

    ```
    //UNTERSE  JOB ,'UNTERSE',CLASS=A,MSGCLASS=X
    //*
    //UNTERSE  EXEC PGM=TRSMAIN,PARM=UNPACK
    //SYSPRINT DD   SYSOUT=*
    //INFILE   DD   DISP=SHR,DSN=<upload_dataset_name>     <== Customize
    //OUTFILE  DD   DISP=(NEW,CATLG),UNIT=SYSALLDA,
    //       DSN=%xmitlib%,             <== Customize
    //       SPACE=(CYL,(120,10,10),RLSE)
    ```

2. Run the job.

   The job step should end with a condition code of `0`.

2.3 – Tailor the $RECEIVE Job

> **Important**
> Wait for the previous installation job to finish before continuing with this procedure.

1. Edit the $RECEIVE member in the**%xmitlib%** library. The job performs RECEIVE operations to convert the uncompressed files from XMIT format to the installation **IMAGE** format.
2. Replace **%xmitlib%**, **%instpref%**, **%UNIT%** and **%VOLSER%** with the values set in Installation Upload and IMAGE Datasets Parameters.

   > **Note**
   > Verify that your SMS does not impose attributes on the installation files by SMS DATACLAS or by pre-allocation.

2.4 – RECEIVE the Installation IMAGE

1. Run the $RECEIVE job.

   All job steps should complete with a condition code of `0`.

2. Check the job output and verify that all RECEIVE and COPY instructions ended successfully and all members were received and copied.

3 – Allocate and Load Connector INSTALL Library

Create the INSTALL library and load it from the **IMAGE** INSTALL library.

> **Important**
> Wait for the previous installation job to finish before continuing with this procedure.

3.1 – Copy, Edit, and Run Member $LOADINS

1. Copy $LOADINS member from library **%xmitlib%** to a new member in the **%xmitlib%** library. The job creates the INSTALL library and loads it from the **IMAGE** INSTALL library.
2. Edit the member. Replace **%instpref%** with the value set in the Installation Upload and IMAGE Datasets Parameters table.

3. Set the values for ILPREFS, ILVERS, ILUNITS and ILVOLS procedure parameters with the values set for these parameters in the Connector for Top Secret Datasets Allocation Parameters table.
4. Run the job.

   The job step should end with a condition code of `0`.
5. Check the job output and verify that the INSTALL library members were copied successfully.

4 – Allocate and Load Connector for Top Secret Installation Libraries

> **Important**
> Wait for the previous installation job to finish before continuing with this procedure.

Use the values assigned in the Connector for Top Secret Datasets Allocation Parameters table for setting the required values in this step.

4.1 – Tailor LOADCTS Member

1. Edit the LOADCTS member in INSTALL library.
2. This member contains JCL for the following:
   - Allocating Connector for Top Secret libraries
   - Loading Connector for Top Secret libraries
   - Performing JCL adaptations of a few installation jobs
3. Tailor the job card.

   Specify values for all the procedure parameters using the values set in the Connector for Top Secret Datasets Allocation Parameters table.

4.2 – Submit Job to Allocate and Load Connector for Top Secret Libraries

1. Check the JCL in member LOADCTS.
2. Submit the job.

   All the job steps must end with a condition code of `0`.

> **Note**

> The following error message generated by this job is not an error and can be disregarded:
>
> ```
> "CTS914E - Modifying of cards ended"
> ```

Datasets which are allocated by the job are listed in Appendix C: Connector for Top Secret Datasets and JCL Procedures.

5 – Tailor Connector for Top Secret Members with Site Parameters

> **Important**
> Wait for the previous installation job to finish before continuing with this procedure.

5.1 – Assign Installation Parameters

1. Edit member DEFPARMS in the Connector INSTALL library.

   > **Note**
   > The member contains keywords which must be assigned the required installation values. Keywords are parameters which start and end with percent signs; for example, **%HOLDCLASS%**.

2. Assign values using the values set in DEFPARMS Parameters.

3. Save the member (if it was modified).

> **Note**
> Throughout this book, the samples are based on the assumption that you have defined your Started Task procedures using the default **CTS** prefix. If you are using a prefix other than **CTS**, adapt the started task name in the samples to match the prefix specified in the **%PROCPREFS%** parameter.

5.2 – Modify Connector for Top Secret members

Member CHNGEPRS in the Connector INSTALL library contains JCL cards to modify Connector for Top Secret members so that they conform to the installation naming conventions. The JCL of this job should have already been set for submission by an earlier installation procedure.

> **Note**
> The job modifies members in the Connector for Top Secret libraries according to values set in DEFPARMS during step 5.1 and values set in the previous installation procedures. Any errors in the parameter definitions in DEFPARMS may result in a need for restarting the installation

> procedure from the beginning. Therefore, prior to submitting the job, verify that the parameters defined in member DEFPARMS are correct.

1. Check the JCL.
2. Submit the job.

   The job updates members in various Connector for Top Secret libraries.
3. Save the member (if it was modified).
4. Scan the output of the job for information and error messages issued by the job.

   The job step must end with a condition code of `0`.

6 – Copy Connector for Top Secret Procedures, STCJOBs and Other Members

> **Important**
> Wait for the previous installation job to finish before continuing with this procedure.

This step copies members to the appropriate libraries:

- Procedures are copied to the requested procedures library (when **%PROCLIB%** is not DONTCOPY)
- When **%STCJOBS%** is not DONTCOPY:
  - STCJOBs are copied to the system STCJOBs library.
  - INCLUDE members are copied to the requested procedures library.
- Parameter members are copied to the Connector PARM library.

6.1 – Submit the CPYMTSS job to Perform the Copy

Member CPYMTSS in the Connector INSTALL library copies the required members to the appropriate libraries. When copied, the procedures and if required, the STCJOBs, are renamed using the value assigned to **%PROCPREFS%** as the first 3 characters. The copy is done without replace, therefore, existing procedures or STCJOBs with the same names will not be overwritten.

If you specified the value **DONTCOPY** in **%PROCLIB%** parameter (in DEFPARMS member), job CPYMTSS does not copy the procedures and STCJOBs, if requested, to your system PROCLIB and STCJOBS libraries. Instead you must

copy them manually from the Connector PROCLIB library and set the first three characters to match the value specified in **%PROCPREFS%** parameter in DEFPARMS member.

1. When ready, submit the job.
2. When the job ends, check the whole `sysout`. In each copy step, check IEBCOPY utility messages and verify that all members were copied successfully.

   Expected condition codes:

   - When **%PROCLIB%** is DONTCOPY, only steps CHECCPR and COPYTSS are executed and both must end with a condition code of `0`.
   - When **%PROCLIB%** is not DONTCOPY:
     - When **%STCJOBS%** is DONTCOPY, steps CHECCPR, COPYPROC and COPYTSS are executed. Step CHECCPR must end with a condition code of `4`. The other steps must end with a condition code of `0`.
     - When **%STCJOBS%** is not DONTCOPY, steps CHECCPR, COPYPROC, COPYSTCJ, COPYSTCI and COPYTSS are executed. Step CHECCPR must end with a condition code of `4`. All other steps must end with a condition code of `0`.

See Appendix C: Connector for Top Secret Datasets and JCL Procedures for a list of the JCL procedures and STCJOBs that are copied by this job.

> **Note**
>
> If JES3 is active in your environment, update all SYSOUT DD cards in all the procedures, and optionally STCJOBs, copied by this job. The update is to drop the whole DCB parameter from ALL SYSOUT DD cards in ALL STCs.
>
> For example, instead of:
>
> ```
> //STDMSG DD SYSOUT=&OUT,DCB=(RECFM=FA,LRECL=133,BUFNO=1)
> ```
>
> You should now have:
>
> ```
> //STDMSG DD SYSOUT=&OUT
> ```

7 – Customize Connector for Top Secret Installation Parameters

> **Important**
> Wait for the previous installation job to finish before continuing with this procedure.

7.1 – Create CTSPARM Module

Member CTSPARMJ in the Connector INSTALL library should already be set for submission by earlier installation procedures.

1. Check the JCL.
2. Submit the job.
3. The job creates load module CTSPARM in the Connector LOAD library.

   All job steps must end with a condition code of `0`.

> **Caution**
>
> If the job returns with a condition code of `12`, you may need to make the following change in the `DCB` of the `SYSPUNCH DD` statement. The issue may have been caused by a change in the Binder which occurred during an update. This file is used as the input for the Binder. Under certain conditions, this may fail. This process converts the file to a sequential file, which is handled differently, and it allows the job to process correctly under the conditions which caused the `12` return code.
>
> 1. Locate the following script:
>
> ```
> //SYSPUNCH DD DSN=&&OBJECT,UNIT=&UNIT,SPACE=(80,(200,50)),
> // DCB=(DSORG=PO,RECFM=FB,LRECL=80,BLKSIZE=800),
> // DISP=(,PASS)
> ```
>
> 2. Update it to the following:
>
> ```
> //SYSPUNCH DD  DSN=&&OBJECT,UNIT=&UNIT,SPACE=(80,(200,50)),
> //DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
> //DISP=(,PASS)
> ```

7.2 – Assign RSSPARM Parameter Values

1. Edit member RSSPARM in the Connector PARM library.
2. Assign a value to the parameters using the values set in RSSPARM Parameters.

8 – Format Connector for Top Secret Datasets

> **Important**
> Wait for the previous installation job to finish before continuing with this procedure.

8.1 – Edit and Run Member FORMCTS

Member FORMCTS in the Connector INSTALL library contains a job which allocates and formats the Connector for Top Secret operation datasets described in the following table. The JCL should already be set for submission by earlier installation procedures.

`<prefix>.<version>.QUEUE`
`<prefix>.<version>.DIAGLVL`
`<prefix>.<version>.RSSOFLI`
`<prefix>.<version>.ENCRINT`
`<prefix>.<version>.ENCREXT`
`<prefix>.<version>.RSSKWDS`
`<prefix>.<version>.USER.CLIST`
	`<prefix>.<version>.CAREGRP`
	`<prefix>.<version>.CARECNN`
	`<prefix>.<version>.CAREOE`
	`<prefix>.<version>.CAREUSR`

where:

- `<prefix>` – Value set for the OLPREFS parameter in Connector for Top Secret Datasets Allocation Parameters table.
- `<version>` – Value set for the OLVERS parameter in Connector for Top Secret Datasets Allocation Parameters table.

1. Check the JCL and submit the job.
2. All job steps must end with a condition code of 0 except:
    - CHECRCF – This step ends with a condition code of `4`.
    - CHECACF2 – This step ends with a condition code of `4`.
    - ALLOCRCF – This step is not executed.
    - ALLOCACF – This step is not executed.

- INITACF – This step is not exectuted.
- ALLOCAIT – This step is not executed.

> **Note**
> Message IEC031I, indicating system D37 failure, appears during execution of the JCL step FORMQUE. This message can be ignored since it is expected behavior and does not present a problem.

9 – Customize Communication Settings

Communication customization should be performed at this time to enable the Connector for Top Secret to communicate with SailPoint through the Connector Gateway (CG).

9.1 – Verify TCP/IP Connectivity

The Connector for Top Secret communicates with the Connector Gateway using the TCP/IP protocol.

A functional TCP/IP connection between Connector for Top Secret and the Connector Gateway is required. Any network topology configuration that supports TCP/IP (hardware and software) can be used, as long as TCP/IP connections can be established between Connector for Top Secret and the Connector Gateway. Connectivity should be verified before you start the Connector Gateway (for example, use the ping command, Telnet commands or other TCP/IP applications).

9.2 – Specify Connector for Top Secret Gateway Communication Parameters

Member **ECAPARM** in the Connector PARM library is used to define Connector for Top Secret Gateway communication parameters.

Edit member ECAPARM in the Connector PARM library and set up the parameters using the values set in Communication Parameters (CTSPUSR and ECAPARM Parameter Members).

9.3 – Define the TCP/IP DATA file

> **Note**
> It is likely that no need to update TCP/IP data file to enable CTSGATE to communicate with SailPoint. You may skip this chapter and get back to it only if you encounter an issue in establishing a communication between CTSGATE and SailPoint.

z/OS TCP/IP regards the CTSGATE started task as a client application requiring a client profile dataset. This profile dataset is referred to in MVS documentation as **hlq.TCPIP.DATA** (**hlq** is the high-level qualifier for the dataset).

This dataset is the main resolver configuration dataset as set up in the local TCP by the MVS/TCP systems programmer.

The TCP/IP profile dataset contains information such as the host name, domain origin and the TCPIPJOBNAME parameter. This information identifies the TCP/IP stack to use.

> **Note**
> For more information regarding this dataset, see the IBM document, *z/OS Communications Server IP Configuration Guide*.

When attempting to locate the TCP/IP profile dataset, MVS searches using the following sequence of names:

1. `<jobname>.TCPIP.DATA (for batch jobs and started tasks)`
2. `SYS1.TCPPARMS(TCPDATA)`
3. `TCPIP.TCPIP.DATA`

When located, the dataset is dynamically allocated.

The default value assigned for the high-level qualifier for the TCP/IP profile dataset during TCP/IP setup is **TCPIP**.

If the high-level qualifier for this dataset in your system has been assigned a different value or if this dataset has not been assigned one of the standard names listed above, the dataset name must be specified in parameter TCPDATA in the CTSGATE started task. This parameter is referred to by the //SYSTCPD DD statement.

This issue should be coordinated with the MVS/TCP systems programmer in your organization.

> **Note**
>
> If the high-level qualifier for TCPIP.DATA at your site is TCP01, Modify the TCPDATA parameter in the Connector for Top Secret Gateway JCL procedure (CTSGATE):
>
> ```
> // TCPDATA=TCP01.TCPIP.DATA,
> ```
>
> If the high-level qualifier of this dataset is TCPIP (the default), this Parameter must be left with its default value (NULLFILE).

9.4 – Set Up Secured Communication

Secured communication can be implemented using TLS secured communication or Transmitted Data Encryption.

> **Note**
> For more information, see Communication Parameters Coordinationfor descriptions of each option before selecting the secured communication method.

Install the selected secured communication method using the steps described in Secured Communication.

10 – Define Connector for Top Secret in the Top Secret System

The following Top Secret definitions are required in order for Connector for Top Secret to function properly. Sample definitions can be found in member CTSTSS in the INSTALL library. Read carefully the explanations below and the notes in member CTSTSS and tailor according to site standards before submitting the job or using the commands from this member.

> **Note**
>
> - Before submitting the job or executing the commands verify that the ACID used to submit thejob or execute the commands has the necessary authority for the Top Secret commands. For example, CTSTSS creates an SCA ACID. Therefore, the ACID submitting the job or the one specified in the USER= parameter in the job card should be the MSCA ACID.
> - After submitting the job or executing the commands, check the whole output and verify that all the commands were processed successfully.
> - It is assumed that the user who installed Connector for Top Secret has ACCESS(ALL) authority to all Connector for Top Secret files, ssigned to him at the beginning of the installation process.

10.1 - Define a Multi-user facility to be used by Connector for Top Secret

A sample definition can be found in member CTSTSS in the INSTALL library.

This definition is required only once, on the first time Connector for Top Secret is installed in the system.

Before adding this facility, check in Top Secret if facility CTSA is defined. If not, define it using the commands in member CTSTSS.

To be retained after the next IPL, the definitions must be copied to the Top Secret parameters member.

10.2 – Define Connector for Top Secret Started Tasks as Valid Started Tasks in Top Secret

Started task	Description
CTSGATE	Top Secret Connector Gateway monitor
CTSACS	Connector Transaction Server (CS)
CTSACD	Connector Notification Server (CD)
CTSAONI	Connector Online Interceptor
CTSAOFLI	Connector Offline Interceptor

> **Note**
> In the list of started tasks used in this section, it is assumed that the default value CTS was

> accepted for the DEFPARMS parameter PROCPREFS. If you assigned a different value to this parameter, modify the started task names accordingly.

Connector for Top Secret started tasks must be associated with an ACID of type SCA (Central Security Administrator) and must be granted authority to LIST all data.

The Multi-user facility defined for Connector for Top Secret must be assigned to the Connector for Top Secret started tasks ACID as MASTFAC.

Sample commands for setting these authorities can be found in member CTSTSS in the INSTALL library.

10.3 – Define the Connector STC ACID to OMVS

The Connector STC ACID must have OMVS definitions to allow the CTSGATE to use the TCP/IP services of the z/OS UNIX System Services (USS). When a user attempts to use the USS, Top Secret verifies that the user is a USS user before the system allows access.

To define the Connector STC ACID as a USS user the ACID has to be assigned a UID and has to be connected to a group having a GID.

For more information, see details provided within the CTSTSS member.

10.4 – Set Permissions to Connector Datasets

Permit READ access for the Connector DIAGLVL and CLIST libraries for your MVS system programmers, z/OS staff, or SailPoint Mainframe support team who should be able to see them.

> **Important**
> Do not allow users access to any DIAGLVL or CLIST libraries in the CTSTSS installation job.

Permit all Connector for Top Secret installation and operation files to be accessed by Connector for Top Secret started tasks listed above with read and write authorizations.

10.5 – Protect the Encryption Keys Datasets

Transmitted Data Encryption Keys Dataset

> **Note**
> This permission is only required when Transmitted Data Encryption is implemented.

Set Top Secret to permit only Connector for Top Secret servers (CTSACS and CTSACD) READ access to the encryption key dataset ENCREXT created in Procedure "9.4 – Set up secured communication" in 9 – Customize Communication Settings. No other users, other than the installer User ID, must be authorized to access this dataset (not even READ authorization).

Stored Data Encryption Keys Dataset

Set Top Secret to permit only Connector for Top Secret servers (CTSACS and CTSACD) and Connector Interceptors (CTSAONI and CTSOFLI) READ access to the encryption key dataset ENCRINT created in 8 – Format Connector for Top Secret Datasets. No other users, other than the installer User ID must be authorized to access this dataset (not even READ authorization).

10.6 – Grant CTSGATE with authority to use TCP/IP stack

This permission is required only when Top Secret SERVAUTH resource class is defined to protect TCP/IP resources from unauthorized access. For more information, see details provided within the CTSTSS member.

10.7 - Grant users permission to facility CTSA

All ACIDs whose passwords are managed by Connector for Top Secret and the ACID of the Managed System Administrator (used to perform updates originating in SailPoint) must be permitted to FACILITY(CTSA), defined in "10.1 - Define a Multi-user facility to be used by Connector for Top Secret".

For more information, see details provided within the CTSTSS member.

11 – Adjust for Longer Managed System Names

The name assigned to the Managed System may be up to 32 characters long. However, if the Managed System name length is greater than 8 characters, certain adjustments must be performed.

When the length of the Managed System name is greater than 8 characters, do the following after completing the installation of Connector for Top Secret:

1. Select a short name (up to 8 characters) for the Managed System name (referred to as the *short MS name*). This name must be unique for each Managed System managed by the instance of Connector for Top Secret.
2. Edit the RSSPARM member in the Connector PARM library. The value for the RSS_WORK_DIR parameter contains the prefix for dynamically allocated datasets. The prefix contains the Managed System name as a qualifier. Change the Managed System name to the short Managed System name so that the value conforms to MVS dataset naming conventions.
3. Edit the member CTSOFLI in Connector JCL library. The member contains DELETE and LISTCAT commands regarding Offline Interceptor datasets. The dataset names contain the Managed System name as a qualifier. Change the Managed System name to the short Managed System name so that the dataset names conform to MVS dataset naming conventions.
4. Edit members CTSOFLMI and CTSOFLMR in Connector for Top Secret JCL library. Add the parameter RSSQ= with the value used as the short Managed System name to the EXEC statement in both members. Save the members.

> **Note**
> The short RSSNAME qualifier must be identical in all modifications.

12 – Adjusting Managed System Administrator Attributes

12.1 - Provide Managed System Administrator Passwords

This section is relevant when you define a new Application or when you set a new Managed System Administrator in SailPoint.

The Top Secret Connector doesn't save the Managed System Administrator passwords in a file like many of the other Connectors do.

When a new Manged System Administrator is defined in Application definition in IdentityIQ or in Source definition in IdentityNow, the Managed System Administrator password or phrase is required for verification, unless a protected user is defined as the Managed System Administrator. The Managed System Administrator User ID and password or phrase are sent from SailPoint to the Top Secret Connector where the password or phrase is verified. After verification, the password or phrase is not saved anywhere on the Connector's platform, or in SailPoint.

Before setting the managed System Administrator user and password or phrase in SailPoint, ensure that the password or phrase of the user is not expired (a new password or phrase set using the CREATE or REPLACE commands is usually expired automatically, and must be changed when logging on for the first time). If the password or phrase is expired, password or phrase verification done by Top Secret Connector will fail.

12.2 - Verify Managed System Administrator Permissions

Set operations, activated from SailPoint, are done in Top Secret under the Managed System Administrator User ID. Therefore, this User ID must be a Control ACID. The Control type depends on the scope of the accounts managed by SailPoint. If all accounts are managed by SailPoint, then it must be SCA-Security Control ACID.

13 – Add Connector for Top Secret Libraries to the MVS Authorized Libraries List

Add the Connector LOAD library and Connector CTRANS library to the MVS APF authorized libraries list.

1. Edit member PROG*nn* in the SYS1.PARMLIB library. Add the Connector LOAD and CTRANS libraries and their volumes to the list using the APF statement.
2. Add the libraries to the active APF list by specifying the following operator command:

   ```
   SET PROG=nn
   ```

   where *nn* is the suffix of the PROG*nn* member that is updated.

14 – Review the Installation

You have completed the Connector for Top Secret installation procedure. Before starting Connector for Top Secret, it is recommended that you refer to the Installation checklist and review the installation procedures to make sure none was omitted.

14.1 – Verify that the Connector for Top Secret Gateway is Installed Properly

Verify that the Connector for Top Secret Gateway is installed properly by issuing the following operator command:

```
S CTSGATE
```

The Connector for Top Secret Gateway starts, and then automatically starts the Connector servers: Transaction Server (CS) and Notification Server (CD). The Connector for Top Secret Gateway waits for communication to be established with SailPoint.

> **Note**
> For more information about starting and stopping Connector processes, see Operations.

14.2 – Verify that the Top Secret Connector Communicates Successfully with SailPoint

Perform the required configuration activities in Connector Gateway and SailPoint.

> **Note**
> For more information, see Communication Parameters Coordination and the appropriate Connector Gateway guides for detailed configuration description.

1. Verify that all components communicate successfully.

   For IdentityIQ, perform the **Test Connection** to synchronize the keywords file, perform account or group aggregation, and so on.

14.3 – Verify Top Secret Connector Top Secret Interface

To test Connector for Top Secret without involving SailPoint, see Appendix F: Connector for Top Secret Batch Utility.

It is not recommended to perform LISTUSER to all accounts with all attributes as this produces a very large sysout file.

15 – (*Optional*) Local Changes Migration

Local changes made by customer in previous version of Top Secret Connector environment may be copied manually to new Top Secret Connector environment.

Such local changes may refer to:

- Parameters set in RSSPARM member in PARM library
- RSSAPI member in PARM library
- Scripts in **USER.CLIST** library

> **Note**
> If there are parameters set in CTSPARM member in PARM library of the previous version and there is a need to migrate them to the current version, contact SailPoint Customer

16 – Configure Automated Startup of Connector for Top Secret

If an automatic startup tool is used to start all the required applications after system initialization, add CTSGATE to this automatic startup tool definitions. Otherwise, add the following command to member COMMND*nn* in SYS1.PARMLIB to start Connector for Top Secret:

```
S CTSGATE
```

> **Note**
> If you intend to start the connector for Top Secret during the system initialization process, be sure to start it **after** initializing TCP/IP and the Top Secret subsystem.

17 – Customizing Top Secret Support

Customization of Top Secret support must be performed at this time. For information and step by-step instructions, see Top SecretTop Secret Support Customization. Perform this customization before continuing to the next procedure in the installation.

18 – Post Installation Checks

Performing any operation performs the Test Connection and Keywords synchronization as the first operation and it would update keywords on Mainframe Connector in **RSSKWDS** file.

1. Start the Mainframe Connector.
2. Start Connector Gateway and ensure that in Mainframe **CTSGATE** a connection is established with the Connector Gateway.
3. To synchronize keywords after Mainframe Connector installation, perform the following:
    - Check application schema of configured Top Secret application and perform **Test Connection**.
    - For any existing application

4. Check application schema of configured Top Secret application
5. To use a new schema attribute, add it in to account or group schema attribute accordingly.

> **Note**
> If you install a new Mainframe Connector retaining the same IdentityIQ application then there may be an issue of keywords not synchronizing. In this case, update the description of any schema attribute for any minor changes.

6. Save the application.

Uninstalling the Connector

To uninstall the Connector for Top Secret, perform all of the following procedures.

1. Delete all Connector datasets
2. Delete all Connector procedures from the system PROCLIB
3. Remove references to Connector LOAD and CTRANS libraries from PROGxx member in SYS1.PARMLIB
4. If STCJOBs are installed delete them from the system STCJOBs library.
5. If you configured Connector for automatic start, remove the relevant definitions:
    - Security definition for Connector started tasks
    - Connector dataset permissions
    - Top Secret accounts defined for Connector

Top Secret Support Customization

The following topics are discussed in this chapter:

Connector for Top Secret Interceptors

The Connector for Top Secret detects changes and events in Top Secret via the following components:

- Online Interceptor (CTSAONI)
- Standard Offline Interceptor (CTSOFLI)

These Connector for Top Secret components, described below, process data in cooperation with Top Secret and various components within your z/OS system. This chapter describes the required customization of these components.

> **Note**
>
> - The Online Interceptor is the recommended facility for intercepting Top Secret events.
> - The Online Interceptor and Offline Interceptor are not supported for IdentityNow.

Online Interceptor

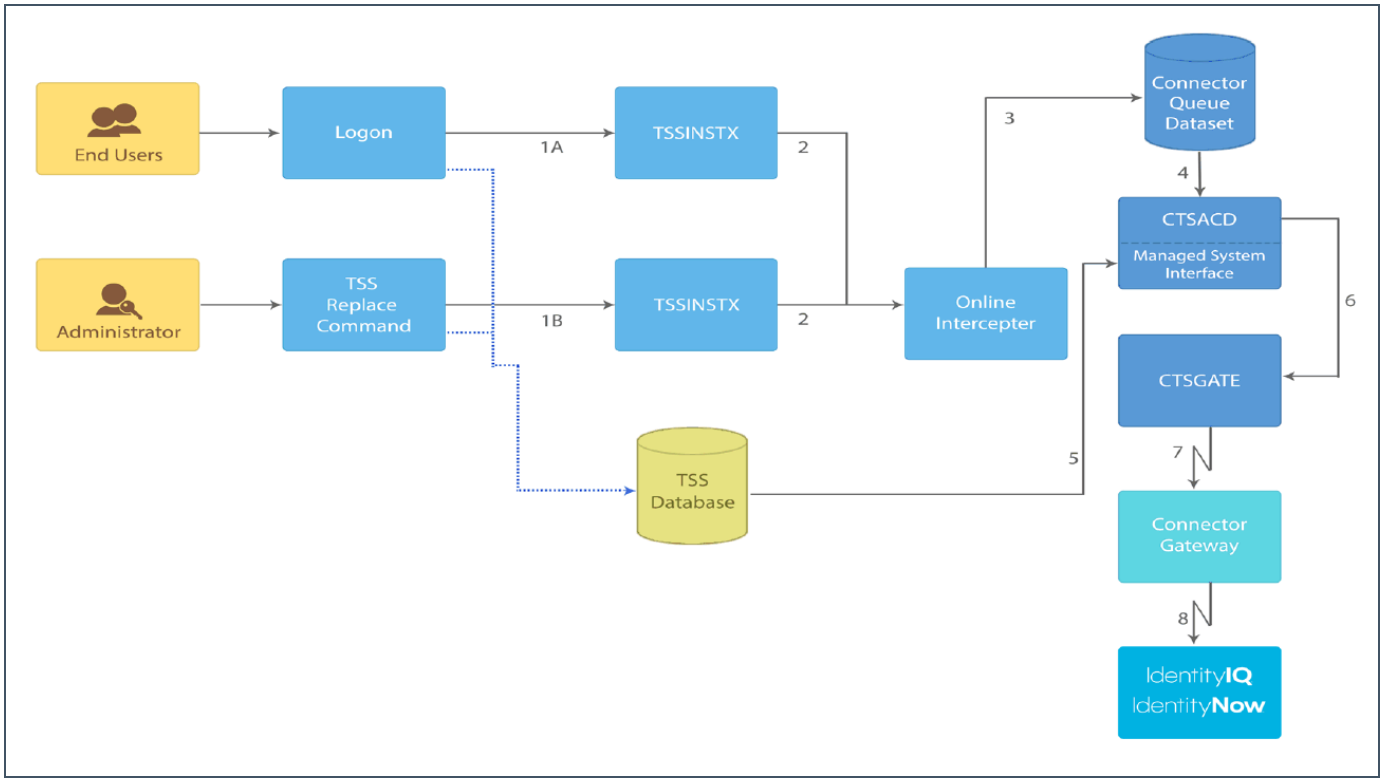
The Connector Online Interceptor detects, in real-time, Top Secret administration events that occur on the platform, and records them so that they can be reported to IdentityIQ. The Online Interceptor also notifies IdentityIQ of password changes made by Top Secret users. This is accomplished using the Top Secret installation exit (TSSINSTX). This exit intercepts the Top Secret commands issued in the system and the password change requests and then transfers information regarding these events to the Online Interceptor.

When password change events are intercepted, the Online Interceptor verifies the new password before notifying IdentityIQ on the password change event. If verification fails, IdentityIQ is not notified on the event. Password verification will also fail when the user whose password is changed is suspended or canceled, because Top Secret returns the same response as when password is incorrect. Therefore, IdentityIQ will not be notified on password change events for suspended or canceled users.

When the Top Secret installation exit intercepts an event, it notifies the Online Interceptor started task via cross-memory services that the event has been intercepted. The Online Interceptor records the event in Connector for Top Secret datasets. The data are then reported to IdentityIQ by the Connector Notification Server (CD), via CTSGATE.

As long as the Online Interceptor is active in the system, Top Secret events and changes are recorded, even if the Connector for Top Secret and the CTSGATE is inactive. When Connector for Top Secret is restarted, the recorded data are transmitted to IdentityIQ.

The processing flow of the Connector Online Interceptor is illustrated in the following figure.



The Connector Online Interceptor detects Top Secret events in one of the following manners:

- When a Top Secret user changes his/her password during the logon process [1A], Top Secret calls TSSINSTX (POST-INIT entry).
- When a user issues a Top Secret command [1B], Top Secret calls TSSINSTX (database change) to log the command.

In any of these situations, the exit that intercepts the event passes the event to the Connector Online Interceptor via cross-memory services [2]. The Online Interceptor then writes the event to the Connector QUEUE dataset [3]. The Connector Notification Server (CTSACD) reads the QUEUE dataset [4], gets the updated entity from Top Secret database, when needed (5) and transfers the event to the Connector for Top Secret Gateway (CTSGATE)

[6] which transfers the event to the Connector Gateway [7] which passes it to IdentityIQ [8].

Installing the Top Secret Installation Exit

> **Note**
> For the exit to intercept the TSS updates, the TSS Recovery file must be active.

To implement the Online Interceptor, the Top Secret Installation Exit must be installed in the system.

This exit intercepts Top Secret commands and password change events and then transfers information to the Online Interceptor regarding each such event that updates a Top Secret database.

For more information regarding this exit, see Online Interceptor.

Installation of the Top Secret Installation exit creates the TSSINSTX load module in the Top Secret load library. If TSSINSTX is already used by the system, it is linked during the install process with Connector supplied exit into a combined module. For every exit call, the Connector exit is invoked first; upon completion, the Connector exit calls the local TSSINSTX module.

The following members in the INSTALL library are involved in this installation procedure:

- CTSTSSX – Connector TSSINSTX exit source code.
- ASMTSSX – Sample job to assemble and link CTSINSTX module.
- CPYTSSX – Sample job to copy CTSINSTX module to the Top Secret load library as TSSINSTX.

1 – Create the CTSINSTX Exit Module

Member ASMTSSX in the Connector INSTALL library is a sample member to assemble and link Connector exit CTSINSTX to Connector load library.

Review the member carefully. Verify that:

- TSSMAC parameter is set to the Top Secret optional materials dataset (TSSOPMAT).
- TSSLOAD parameter is set to the Top Secret load library. In the linkage step, ASMTSSX contains an INCLUDE command for the current copy of TSSINSTX from the Top Secret load library. The TSSLOAD parameter must be specified regardless of whether or not a copy of TSSINSTX exists.
- If TSSINSTX module does not exists in CA-To Secret load library, remove the following lines from the Linkage Editor input statements:

```
ORDER TSSINSTX
INCLUDE TSSLOAD(TSSINSTX)
```

Submit the job to create the CTSINSTX load module in the Connector LOAD library. All job steps must end with a condition code of `0`. The job links the Connector supplied exit with the local TSSINSTX exit, if exists, creating a combined module CTSINSTX in Connector LOAD library.

2 – Copy CTSINSTX to the Top Secret Load Library

Edit member CPYTSSX in the Connector INSTALL library. The job copies load module CTSINSTX (created in 1 – Create the CTSINSTX Exit Module) to your Top Secret load library as TSSINSTX.

Any previous copy of exit TSSINSTX in the Top Secret load library is overwritten by this job. It is recommended that you create a backup copy of the previous exit.

Review the job carefully and submit the job. All job steps must end with a condition code of `0`.

3 – Activate TSSINSTX

Refresh the LINKLIST libraries concatenation via F LLA, REFRESH operator command. Wait for completion of the refresh operation.

Activate TSSINSTX by issuing the following Top Secret command:

```
TSS MODIFY EXIT(ON)
```

or use the operator command:

```
F TSS,EXIT(ON)
```

4 – Verify that the Online Interceptor is installed properly

Verify that the Online Interceptor is installed properly by issuing the following operator command:

```
S CTSAONI
```

The Online Interceptor starts and begins recording Top Secret events to the Top Secret Connector Queue dataset.

Configuring Autostart of Online Interceptor

Make the appropriate changes to the system startup procedures to start Connector for Top Secret Online Interceptor after IPL.

If an automatic startup tool is used to start all the required applications after system initialization, add CTSAONI to this automatic startup tool definitions or add the following command to **COMMND*nn*** member in SYS1.PARMLIB to start the Online Interceptor after IPL:

```
S CTSAONI
```

(Optional) Controlling Online Interceptor Memory Consumption

The memory size used by the Online Interceptor can be controlled by setting the ONLI_MAX_EVENTS optional parameter in the RSSPARM member in the PARM dataset as follows:

```
managedSystemName ONLI_MAX_EVENTS numberOfEvents
```

ONLI_MAX_EVENTS

The **ONLI_MAX_EVENTS** parameter sets the maximum number of events that the Online Interceptor can accumulate in memory. If the ONLI_MAX_EVENTS parameter does not exist in the RSSPARM member, the default value of 20,000 events is used. The minimum value for ONLI_MAX_EVENTS is 2,000. Each entry occupies 2,560 bytes in memory, depending on the event type (user, group, connection, password) and length of userid, group, password.

ONLI_MIN_NOTIFY_EVENT%

The **ONLI_MIN_NOTIFY_EVENT%** optional parameter sets a threshold for issuing a warning message that the memory for accumulating event may soon be entirely filled. When the stated percentage of available memory for accumulating events is filled, message CTS4509W (described below) is displayed.

Use the following syntax for ONLI_MIN_NOTIFY_EVENT% parameter:

```
managedSystemName ONLI_MIN_NOTIFY_EVENT% value%
```

> **Note**
> The value specified must be followed by the % symbol. If the ONLI_MIN_NOTIFY_EVENT% parameter is not specified in the RSSPARM member, the default value of 10% is used.

For example, given the following settings:

```
RSS1 ONLI_MAX_EVENTS 5000
```

```
RSS1 ONLI_MIN_NOTIFY_EVENT% 20%
```

You can expect the following:

- Space for 5000 intercepted events is allocated in memory (approximately 12.5 MB).
- Message CTS4509W will be issued when only 20% of the allocated space remains, by default this is when more than 4,000 events reside in memory.

(Optional) Configuring Event Interception Filtering

Event Interception filtering enables you to decide which entity events intercepted in the Managed System by the Online Interceptor, are transferred to IdentityIQ.

The following RSSPARM parameters enable you to filter the interception of events on the Managed System:

ONLI_EVENT_OE

Specifies whether the Online Interceptor should send container events to IdentityIQ.

Possible values for this parameter are Y/N.

ONLI_EVENT_GROUP

Specifies whether the Online Interceptor should send group events to IdentityIQ. Possible values for this parameter are Y/N.

ONLI_EVENT_USER

Specifies whether the Online Interceptor should send user events to IdentityIQ. Possible values for this parameter are Y/N.

ONLI_EVENT_USER_PWD_ONLY

Specifies which intercepted user events are sent to IdentityIQ.

This parameter is only relevant when ONLI_EVENT_USER is specified as Y Possible values for this parameter are:

- Y – Only user password change events are sent to IdentityIQ.
- N – All intercepted user events are sent to IdentityIQ. This is the default value.

To filter the interception of Managed System events:

1. Stop Connector for Top Secret Online Interceptor.

   For more information, see Starting and Stopping the Online Interceptor.

2. Edit member RSSPARM in the Connector for Top Secret PARM library.

3. Insert or modify one or more of the following parameters as necessary.

```
rss_name ONLI_EVENT_USER_PWD_ONLY  N  <== Modify as required
rss_name ONLI_EVENT_USER           Y  <== Modify as required
rss_name ONLI_EVENT_GROUP          Y  <== Modify as required
rss_name ONLI_EVENT_OE             Y  <== Modify as required
```

4. Save the member and exit.

5. Restart Connector for Top Secret Online Interceptor.

   For more information, see Starting and Stopping the Online Interceptor.

Standard Offline Interceptor

The Standard Offline Interceptor is a process that can be invoked to intercept security administration events occurring outside of Connector for Top Secret.

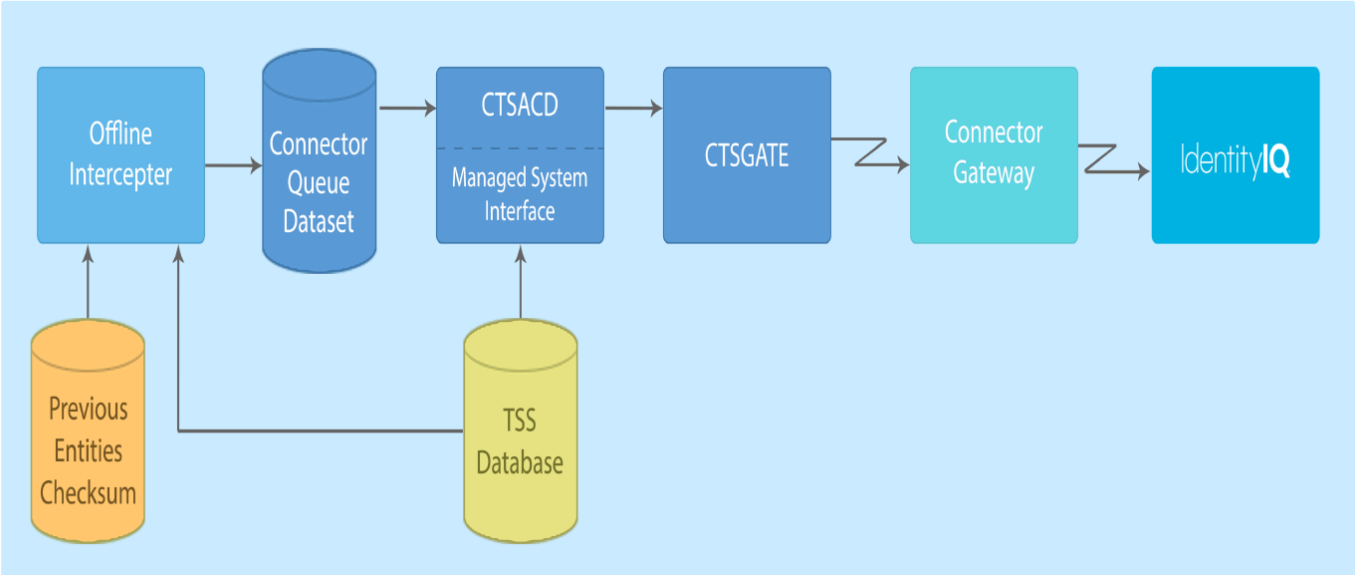
The Offline Interceptor typically scans all managed system objects it monitors in order to determine what events have occurred since the last Offline Interceptor invocation.

While these events are typically intercepted by the Online Interceptor, the Offline Interceptor can be used if, for some reason, the Online Interceptor has not been active for a certain period. It is recommended that you use the Standard Offline Interceptor as an alternative to performing aggregation in these circumstances.

If you wish to run the Standard Offline Interceptor on a regular basis, the Notification Server can be used to schedule the Offline Interceptor using parameters described below.

For more information, see Scheduling the Standard Offline InterceptorThe Standard Offline Interceptor can be scheduled independently of Connector for Top Secret, either manually or automatically via the Notification Server..

The processing flow of the Connector Offline Interceptor is illustrated in the following figure.



The Standard Offline Interceptor reads from Top Secret database all relevant entities (users, groups, profiles) and compares them, using checksum records, to their status on last time Offline Interceptor was run. Only those entities which are different, which means they have been changed, are written to Queue file and then sent by Notification Server (CTSACD) to IdentityIQ via Connector for Top Secret Gateway (CTSGATE) and the Connector Gateway.

> **Note**
> Password change events are not handled by the Offline Interceptor. If needed, the Online Interceptor must be used.

Required Top Secret Definitions

The installation process Top Secret definitions required to run CTSOFLI. For more information, see 10 – Define Connector for Top Secret in the Top Secret System.

CTSOFLI requires UPDATE permission to the CATALOG in which its datasets will be cataloged.

Setting the Volume for Standard Offline Interceptor Working Datasets

During Standard Offline Interceptor operation, working datasets are created and must be retained until the next run of the Offline Interceptor.

> **Note**
> Whenever the CTSOFLI procedure is mentioned, replace CTS with the relevant three-letters prefix used for the procedures in the Connector for Top Secret installation.

The Offline interceptor datasets will be allocated using the values specified for OLUNITS and OLVOLS during installation (see Connector for Top Secret Datasets Allocation Parameters for more information).

To set other values, change the values set for OLUNITS and OLVOLS in the CTSOFLI procedure parameters.

Step **ALLOC1** in member **CTSOFLI** allocates the files listed in the following table.

Member Name	Description
OFLUTMP	Users temporary file
OFLGTMP	Groups temporary file
OFLCTMP	Connections temporary file
OFLOTMP	Containers temporary file
OFLRTMP	Managed System parameters temporary file
OFLUCMP	File for comparing users
OFLGCMP	File for comparing groups
OFLCCMP	File for comparing connections
OFLOCMP	File for comparing containers
OFLRCMP	File for comparing Managed System parameters

Step **ALLOC2** in member **CTSOFLI** allocates the files listed in the following table.

Member Name	Description
OFLUIMG	Users checksum file
OFLGIMG	Groups checksum file
OFLCIMG	Connections checksum file
OFLOIMG	Containers checksum file
OFLRIMG	Managed System parameters checksum file

Determine the size of the files allocated according to the number of users, groups, and connections in Top Secret.

In a 3390 device, one cylinder is approximately 840 KB (56 KB per track, 15 tracks per cylinder: 56 KB * 15 = 840 KB).

For each user or group, allocate 20 bytes should be calculated for allocation of **OFLUTMP**, **OFLGTMP**, **OFLUCMP**, **OFLGCMP**, **OFLUIMG** and **OFLGIMG** files.

For each connection, allocate 30 bytes should be calculated for allocation of **OFLCTMP**, **OFLCCMP**, and **OFLCIMG** files.

For example, for 50,000 users, allocate 2 cylinders:

```
(20 * 50,000) / 840,000 = 1.19 cylinders.
```

This exceeds 1 cylinder, and thus should be considered as 2 cylinders.

> **Important**
> Do not change the allocation size of the **OFLOTMP**, **OFLOCMP**, **OFLOIMG**, **OFLRTMP**, **OFLRCMP** and **OFLRIMG** files in **CTSOFLI**.

Scheduling the Standard Offline InterceptorThe Standard Offline Interceptor can be scheduled independently of Connector for Top Secret, either manually or automatically via the Notification Server.

The following options exist for scheduling activation of the Offline Interceptor:

- The Notification Server can schedule the Offline Interceptor to run at fixed intervals, starting from the time Connector for Top Secret is activated. The interval is specified using parameter OFLI_INTERVAL, described below.
- The Notification Server can schedule the Offline Interceptor to run at specific times during the day. The run times are specified using parameter OFLI_RUN_TIME_LIST, described below.
- You can schedule the Offline Interceptor using an external facility.

When scheduled by the Notification Server, the Notification Server will not begin scheduling the Offline Interceptor until after the initial aggregation of managed system data to IdentityIQ has been performed.

Depending on the method used to schedule the Offline Interceptor, it may be necessary for Notification Server to be aware of the exact time the Offline Interceptor was last activated.

For this purpose, Connector for Top Secret maintains the <prefix>.<version>.RSSOFLI file. This file contains the invocation time of the Offline Interceptor and saves the time stamp of the last event that it processed.

Upon initialization, the Offline Interceptor obtains the time stamp from its previous invocation from function CTSInterceptorInit. When the Offline Interceptor completes its operation, the updated time stamp is written to the RSSOFLI file by calling function CTSInterceptorTerm.

Standard Offline Interceptor Configuration Parameters

Offline Interceptor scheduling is controlled using parameters in file RSSPARM. This file contains both common and Managed System-specific parameters of the different managed system managed by IdentityIQ via Connector for Top Secret. Unless indicated otherwise, the parameters that control Offline Interceptor scheduling are specified for each managed system.

OFLI_INTERCEPT

Offline Interceptor scheduler activation. Possible values are:

- **Y** – The Offline Interceptor is started periodically by the Notification Server based on the value of parameter OFLI_INTERVAL or OFLI_RUN_TIME_LIST (described below). Default.
- **N** – The Offline Interceptor is not started by the Notification server. You must provide another means of scheduling the Offline Interceptor.

  For information on starting the Offline Interceptor manually, see Starting the Offline Interceptor Manually.

OFLI_WAIT_INTERVAL

An interval, starting from the time of the initial communication between Connector for Top Secret and IdentityIQ, during which the Notification Server will not activate the Offline Interceptor, even if it is scheduled to run during that time. This provides a period at Connector for Top Secret startup during which the administrator can perform functions in Connector for Top Secret without interference from the Offline Interceptor. Format: **hhmmss**.

Default: 001000 (10 minutes).

Unlike the other Offline Interceptor scheduling parameters, OFLI_WAIT_INTERVAL applies to all managed systems managed by the Connector for Top Secret installation.

OFLI_INTERVAL

The interval that must pass between consecutive activations of the Offline Interceptor (format: **hhmmss**). The Offline Interceptor is first activated at the earliest opportunity following the activation of Connector for Top Secret, and then after the specified interval. If Connector for Top Secret is stopped and restarted, the Offline Interceptor is not activated until the specified interval has passed from the previous activation.

Default: 010000 (1 hour).

OFLI_RUN_TIME_LIST

A list of times at which the Offline Interceptor should be activated. The times are stated in 24-hour format **hh:mm**, separated by commas. 00:00 represents midnight.

For example: `00:00,03:00,09:30,12:00,15:30,21:25`

The Offline Interceptor will be activated at the times specified in the list. If, for any reason (for example, Connector for Top Secretwas not active), the Offline Interceptor missed a scheduled activation, it will not be activated until the next time specified in the list. See also parameter OFLI_RUN_INTERVAL.

If both parameter OFLI_RUN_TIME_LIST and parameter OFLI_INTERVAL are specified, Connector for Top Secret ignores parameter OFLI_RUN_TIME_LIST.

If neither of the two parameters is specified, Connector for Top Secret assigns the default value **02:30** to parameter OFLI_RUN_TIME_LIST.

OFLI_RUN_INTERVAL

The maximum deviation from scheduled activation time (in parameter OFLI_RUN_TIME_LIST) that the Notification Server will tolerate for activation of the Offline Interceptor. If the Offline Interceptor cannot be activated within this interval from the scheduled time (for example, Connector for Top Secret was not active), the scheduled activation is skipped.

Format: `hhmmss`

Default: 001000 (10 minutes).

> **Note**
> Parameters OFLI_INTERVAL, OFLI_WAIT_INTERVAL and OFLI_RUN_INTERVAL must be expressed as valid time representations in the format **hhmmss**, where **hh** is 00 through 23, **mm** is 00 through 59, and **ss** is 00 through 59. For example: 010000 translates to 01:00:00 (1 hour) and is a valid value; 006000 translates to 00:60:00 which is not a valid time and thus will be ignored.

For example, with the following parameter values:

- OFLI_RUN_TIME_LIST

  00:00,12:00

- OFLI_RUN_INTERVAL

  020000 (2 hours)

- OFLI_WAIT_INTERVAL

  001000 (10 minutes)

Scheduling of the Standard Offline Interceptor by the Notification Server would be performed as follows:

- If Connector for Top Secret was shut down at 23:00 and then restarted at 01:30, the Offline Interceptor would be activated at 01:40 (Connector for Top Secret start time + OFLI_WAIT_INTERVAL), as this is within the 2-

hour deviation permitted by parameter OFLI_RUN_INTERVAL.

- If Connector for Top Secret was shut down at 23:00 and then restarted at 01:55, the Offline Interceptor would not be activated for its scheduled run time of 00:00. This is because the earliest possible activation time for the Offline Interceptor is 2:05 (Connector for Top Secret start time + OFLI_WAIT_INTERVAL). This exceeds the 2-hour deviation permitted by parameter OFLI_RUN_INTERVAL by 5 minutes. The next activation of the Offline Interceptor would occur at 12:00.

Shared Top Secret Database Support

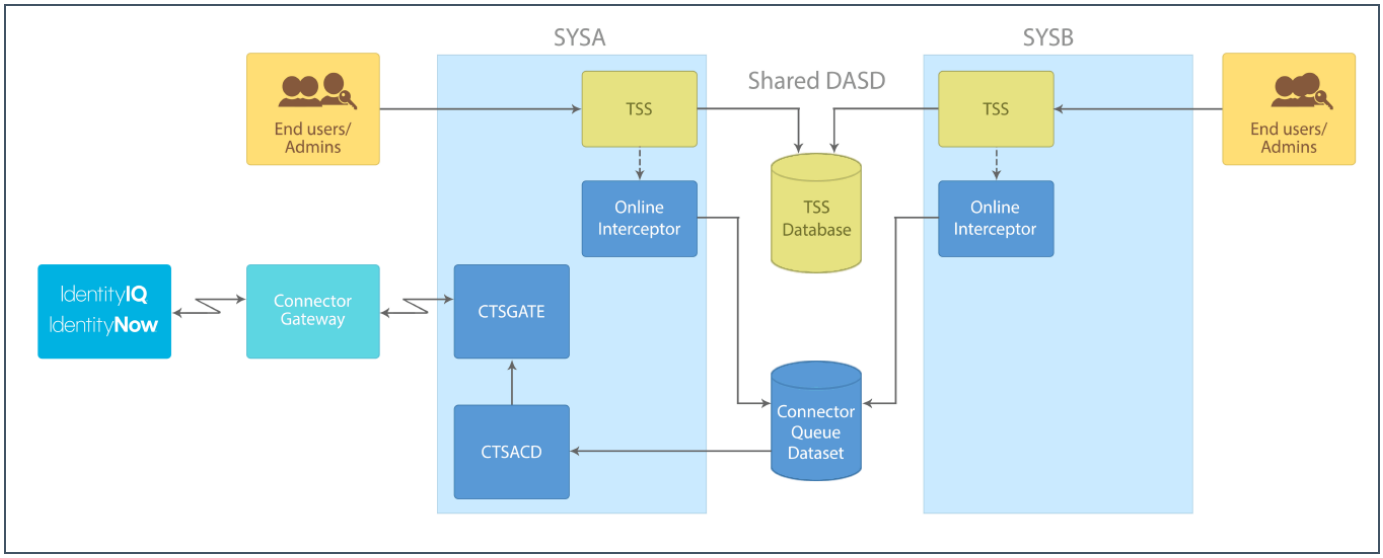
> **Note**
> The Shared Top Secret database support is implemented using the Connector for Top Secret Online Interceptor.

A shared Top Secret database environment consists of two or more Top Secret systems which share the same Top Secret database.

In a shared database configuration, one of the systems runs a full instance of Connector for Top Secret. This system is referred to as the primary system. Each additional Top Secret system sharing the database is referred to as a *secondary system*. Each secondary system only runs the Connector for Top Secret Interceptor.

The primary system communicates with IdentityIQ and executes the transactions generated by IdentityIQ (in the regular manner). The secondary systems runs the Connector Interceptors to ensure that any updates made to the Top Secret database from these systems are propagated to IdentityIQ. In this way, the IdentityIQ and Top Secret databases are kept synchronized.

For example, assume that systems SYSA and SYSB share the Top Secret database. SYSA is the primary system which runs the complete Connector for Top Secret. SYSB is the secondary system which runs only the Online Interceptor. This configuration is shown in the following figure:



When a transaction is issued by IdentityIQ, it is executed by the Connector Transaction Server (CTSACS) running on the primary system, and the appropriate updates are made to the Top Secret database.

When a local update is done in either SYSA or SYSB, it is intercepted by the appropriate Interceptor running in the system and written to the Connector QUEUE dataset.

The Notification Server (CTSACD) running in the primary system reads the local update from the Connector QUEUE dataset and forwards it to IdentityIQ. This ensures that the IdentityIQ and Top Secret databases are kept synchronized.

Install Support for a Shared Top Secret Database

A shared Top Secret database environment consists of two or more Top Secret systems which share the same Top Secret database.

In a shared database configuration, one of the systems runs the complete Connector for Top Secret. This system is referred to as the primary system. The other system(s) is referred to as the secondary system.

The secondary system only runs one component of Connector for Top Secret (that is, the Online Interceptor). After installing Connector for Top Secret on the primary system, you must install the Online Interceptor on each additional secondary system which shares the Top Secret database.

1. **Define Connector QNAME to the ENQ Manager**

   The Connector for Top Secret serializes access to its QUEUE dataset by issuing an ENQ with the SYSTEMS option.

   To allow synchronized update of the QUEUE dataset from multiple systems, you may need to update your external ENQ manager (for example, GRS or MIM) with the Connector QNAME to be propagated to all systems in the ENQ manager complex.

   Another option is to let the Connector for Top Secret use the RNL=NO ENQ/DEQ parameter telling GRS to always propagate the ENQ to all systems in the GRS complex. This can be controlled using the ENQRNL parameter in member CTSPARM in Connector for Top Secret PARM library.

   The default QNAME used by Connector for Top Secret is ***xxx*ASYNC** where ***xxx*** is the value set for

   %PROCPREFS% parameter in the DEFPARMS member in the INSTALL library. The default may be customized by changing the QNAME parameter in member CTSPARM in Connector for Top Secret PARM library.

   For explanations on how to change and activate these parameters, see CTSPARM: Assembler Format Parameters.

2. **Share the Datasets**

   Place the datasets listed below on a shared DASD so that they can be shared by the primary and secondary systems.

Dataset Name	Description
prefix.version.QUEUE	Connector Queue dataset
prefix.version.LOAD	Connector LOAD library
prefix.version.CMSG	Connector Msgs library
prefix.version.PARM	Connector PARM library
prefix.version.DIAGLVL	Connector Diagnostics library
prefix.version.ENCRINT	Connector Stored Data Encryption dataset
prefix.version.CTRANS	Connector SAS/C Runtime Load library

3. **Complete the Installation**

   The following should be done to complete the installation on the secondary system:

   a. If STCJOBS are not used, copy the CTSAONI JCL procedure from the system PROCLIB of the primary system to the system PROLCIB of the secondary system.

      If STCJOBS are used:

      - Copy the CTSAONI STCJOB member to the system STCJOBS library.
      - If the value LOCALCOPY is set for %PROLCIB% in the DEFPARMS member in the INSTALL library, make sure the system has access to the Connector for Top Secret PROCLIB library. Otherwise, copy the CTSAONI JCL procedure from the system PROCLIB of the primary system to the system PROCLIB of the secondary system.

   b. Add Connector LOAD and CTRANS libraries to the APF authorized libraries list. For details, see Installation step 13 – Add Connector for Top Secret Libraries to the MVS Authorized Libraries List.

   c. Define the CTSAONI started task to Top Secret started tasks table. For details, see Installation step 10 – Define Connector for Top Secret in the Top Secret System.

   d. Install Online Interceptor system exits TSSINSTX. For details see Installing the Top Secret Installation Exit.

   e. Start the Online Interceptor.

   f. After verification, configure automated startup of the Online interceptor. For details see Configuring Autostart of Online Interceptor.

Supporting Mixed Case Passwords

Connector for Top Secret supports mixed case passwords. If mixed case is used in your Top Secret, update in PARM library in RSSPARM member the ONLI_PASSWORD_CASE parameter to be ASIS or UPPER instead of the default

lower case value. When mixed case is used in Top Secret and ASIS or UPPER is set in RSSPARM, the passwords are sent to IdentityIQ without any translation made by Top Secret Connector.

> **Note**
> ASIS and UPPER have the same effect - which means no translation is done on password.

User Defined Fields and SailPoint

User Defined fields are fields within the Top Secret database that you customize to store security information about the users at your installation. You can tailor the names and attributes of User Defined fields. Once you define custom fields, use Top Secret commands to add data to custom fields.

Connector for Top Secret can aggregate User Defined fields to SailPoint. To achieve this, the User Defined fields should be added to the Account entity schema of IdentityIQ Application or IdentityNow Source definitions.

To implement support for User Defined fields

1. Add the User Defined fields to the Account entity of the application schema in IdentityIQ or Source schema in IdentityNow.
2. Perform full account aggregation to get the User Defined fields data.

> **Note**
> User Defined fields are supported only in aggregation. They are not supported for provisioning (their values cannot be changed via SailPoint).

Secured Communication

Secured communication has the following aspects:

- Communication security using TLS or using Transmitted Data Encryption
- IP List validation, which allows control of the IP addresses which are allow accessing the Top Secret Connector

The following topics are discussed in this chapter:

TLS Secured Communication

For using TLS secured communication for Top Secret Connector, TLS communication must be defined and configured in the following relevant components:

- TCP/IP in the system where Top Secret Server will be active
- Connector Gateway
- SailPoint IdentityNow or IdentityIQ

System Requirements

- The following respective components for z/OS versions must be installed for TLS communication:

z/OS Version	Cryptographic Services	z/OS Security Level 3
z/OS 2.5	System SSL Base: FMID HCPT450	System SSL Security Level: FMID JCPT451
z/OS 2.4	System SSL Base: FMID HCPT440	System SSL Security Level: FMID JCPT441

- The CSF started task must be active in the system where Top Secret Server would be active.

Implementing Secured Communication for Top Secret Connector

Secured communication to Top Secret Connector must be implemented using AT-TLS policy. The TLS processing is done by TCP/IP and is transparent to the Top Secret Connector.

The secured communication is implemented using server authentication.

> **Note**
> When TLS is used for Top Secret Connector secured communication, Transmitted Data Encryption must be disabled. Edit member CTSPUSR in the Connector PARM library, set the value of ENCR_EXT_ACT to N and save the member.

Implementation Procedure

1. A valid server certificate with its associated server private key must be defined. This certificate must be signed by a trusted Certificate Authority's (CA).

   > **Note**
   > For testing purposes, a local CA can be defined for signing the server certificate.

2. The server certificate and the CA certificate must be connected to a key ring.

3. The CA certificate must be exported to a file, transferred (using FTP with ASCII mode) to the client (with .cer suffix) and installed there to be used for certificate verification by the TLS handshake process.

4. Implementing AT-TLS policy for Top Secret Connector communication.

   For detailed information about implementing AT-TLS policy, see "Application Transparent Transport Layer Security data protection" chapter of z/OS Communications Server IP Configuration Guide.

   The required policy attributes for AT-TLS policy are:

   - Local Port Range – ports defined in ECAPARM for Top Secret Connector
   - Direction = Inbound
   - TLS Enabled = On
   - TLS v1.1 = On
   - TLS v1.2 = On
   - Handshake Role = Server
   - Client Authorization Type = PassThru
   - Application Controlled = Off
   - Secondary Map = Off

- The name of the certificate created for the secured communication and the name of the key ring to which the server certificate and the CA certificate are connected, should be specified.

> **Note**
> TCP/IP must be granted permission to access the key ring to which the Top Secret server certificate and the CA certificate are connected.

Sample File for AT-TLS Policy

```
# RULE for Top Secret Connector CTSGATE
######################################################
TTLSRule CTSGATE
{
LocalAddr ALL RemoteAddr ALL
LocalPortRange 2470-2471 Direction Inbound
Priority 255 # highest priority rule Userid CTSGATE
TTLSGroupActionRef GrpAct_CTSGATE
TTLSEnvironmentActionRef GrpEnv_CTSGATE
TTLSConnectionActionRef GrpCon_CTSGATE
}
TTLSGroupAction GrpAct_CTSGATE
{
TTLSEnabled On
Trace 7
}
TTLSEnvironmentAction GrpEnv_CTSGATE
{
Trace 7
HandshakeRole Server
EnvironmentUserInstance 0
TTLSKeyringParmsRef PrmKeyRing_CTSGATE
TTLSEnvironmentAdvancedParmsRef PrmEnvAdv_CTSGATE
}
TTLSEnvironmentAdvancedParms PrmEnvAdv_CTSGATE
{
TLSv1.1 On
TLSv1.2 On
ClientAuthType PassThru
}
TTLSConnectionAction GrpCon_CTSGATE
{
HandshakeRole Server
TTLSCipherParmsRef PrmCipher_CTSGATE
TTLSConnectionAdvancedParmsRef PrmConAdv_CTSGATE
CtraceClearText Off
Trace 7
}
TTLSConnectionAdvancedParms PrmConAdv_CTSGATE
{
ApplicationControlled Off
CertificateLabel CTSGATE
```

```
SecondaryMap Off
}
TTLSCipherParms PrmCipher_CTSGATE
{
# supported cipher suites - we used a wide list, that should be decreased according
to specific needs
V3CipherSuites TLS_DH_DSS_WITH_DES_CBC_SHA
V3CipherSuites TLS_DH_RSA_WITH_DES_CBC_SHA
V3CipherSuites TLS_NULL_WITH_NULL_NULL
V3CipherSuites TLS_RSA_WITH_NULL_MD5
V3CipherSuites TLS_RSA_WITH_NULL_SHA
V3CipherSuites TLS_RSA_EXPORT_WITH_RC4_40_MD5
V3CipherSuites TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
V3CipherSuites TLS_RSA_WITH_DES_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_DES_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_DES_CBC_SHA
V3CipherSuites TLS_RSA_WITH_AES_256_CBC_SHA256
V3CipherSuites TLS_RSA_WITH_AES_256_CBC_SHA
V3CipherSuites TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_RSA_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
V3CipherSuites TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DHE_RSA_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_DHE_DSS_WITH_AES_128_CBC_SHA
V3CipherSuites TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
V3CipherSuites TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
V3CipherSuites TLS_RSA_WITH_AES_128_GCM_SHA256
V3CipherSuites TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
V3CipherSuites TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
V3CipherSuites TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
V3CipherSuites TLS_DHE_DSS_WITH_AES_128_GCM_SHA256
V3CipherSuites TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA
V3CipherSuites TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA
}
TTLSKeyringParms PrmKeyRing_CTSGATE
{
Keyring CTSRING
}
```

Enabling TLS Between Top Secret Connector and Connector Gateway

For more information on the procedure to be performed on the client side (Connector Gateway), see *SailPoint Integration Guide* or *SailPoint Quick Reference Guide* for Gateway Connectors depending on the Connector Gateway

release.

> **Note**
> Ensure that the Transmitted Data Encryption is not active for communication between the Top Secret connector and the Connector Gateway.

Enabling TLS between SailPoint and Connector Gateway

For more information on the procedure to be performed on SailPoint, refer to the relevant documentation of SailPoint.

> **Note**
> Ensure that the Transmitted Data Encryption is not active for communication between the Connector Gateway and IdentityIQ.

Transmitted Data Encryption

Transmitted Data Encryption must be configured in the CA-Top Secret Connector and in IdentityIQ.

> **Note**
> Transmitted Data Encryption is not supported for communication with IdentityNow.

Implement Transmitted Data Encryption for Top Secret Connector

Implement transmitted data encryption on the Top Secret Connector.

Enable Transmitted Data Encryption

1. Edit member CTSPUSR in the Connector PARM library.
2. Set the value of **ENCR_EXT_ACT** to **Y**.
3. Save the member.

Install Transmitted Data Encryption Key Dataset

The Connector for Top Secret uses keys in the Encryption Key dataset to encrypt data which is transmitted between Connector for Top Secret and IdentityIQ. The Encryption Key file (global or platform-specific) which was generated in IdentityIQ must be copied to the Encryption Key dataset in Connector for Top Secret.

To copy the Encryption Key file, use a file transfer utility available at your site or any other available transfer method. The Encryption Key file is a text dataset. Therefore, the transfer method must convert ASCII to EBCDIC and must support conversion of line breaks.

The Transmitted Data Encryption keys are stored in the following dataset:

```
<prefix>.<version>.ENCREXT
```

where:

- `<prefix>` – Value set for the OLPREFS parameter. For more information, see in Connector for Top Secret Datasets Allocation Parameters.
- `<version>` – Value set for the OLVERS parameter. For more information, see in Connector for Top Secret Datasets Allocation Parameters.

Implement Transmitted Data Encryption for IdentityIQ

To use Transmitted Data Encryption for IdentityIQ, an Encryption Key file must be created and transferred to the Top Secret Connector, and the Transmitted Data Encryption must be enabled.

For detailed implementation, see *SailPoint Quick Reference Guide* for *Gateway Connectors or SailPoint Integration Guide*.

Incoming IP Address Validation

To enable incoming IP address validation, perform the following:

1. In the ECAPARM member of the PARM library, add to the CHANNEL definition: IPLIST=ECAIPLSx statement.
2. A corresponding ECAIPLSx member should be set in the library pointed by DAPARM DD card of the CTSGATE STC.
3. Member ECAIPLSx should contain the relevant Allow and Forbid statements as described in the full feature description that follows.

Configure Incoming IP Address Validation

The following topics describe concepts involved with configuring incoming IP address validation.

Syntax of IP Addresses List

The list of IP addresses is a source table that contains IP addresses of work stations allowed to communicate with CTSGATE that owns this list. An IP address is specified in the standard dotted-decimal notation. From 1 through 4 contiguous from right-to-left sections of an IP address can be specified with an asterisk. Such asterisk in a section of IP address which, actually defines a range of IP address. For example: `80.56.241.*` means in fact, a range of IP addresses from `80.56.241.0` through `80.56.241.255`.

If the above described simplified method of specifying an addresses range turns out too rough to determine a necessary range, then a standard subnet mask can be also used.

Any separate IP address or a range of addresses that should be allowed for communication, must be preceded with the keyword **ALLOW ***. IP address or a range of addresses that should be prevented from communication, must be preceded with the keyword **FORBID ***.

The first statement of the list must be always **[ALLOW *]** or **[FORBID *]**.

An incoming address will be checked against the table of IP addresses (in the internal format) to determine if it is to be confirmed or rejected. A given address may in principle match multiple (and even conflicting) entries in the table.

Only one entry will be used to determine whether the IP address is to be allowed or be forbidden. The entry with the most specific address is the effective entry.

The examples provided below demonstrate how a list of IP addresses can be coded.

- The following example allows any IP address, but excluding (forbidding) one specific address (`81.50.1.241`) and a range of addresses (`80.56.241.0` - `80.56.241.255`):

```
ALLOW * First statement allows any IP
Forbid 81.50.1.241
Forbid 80.56.241.*
```

- The same example may be specified with a subnet mask. For example:

```
ALLOW * First statement allows any IP
Forbid 81.50.1.241
Forbid 80.56.241.0, MASK=255.255.255.0
```

- The following example forbids all IP addresses, allowing only two specific addresses (`172.16.241.128` and `81.50.1.241`) and a range of addresses (`80.56.241.0` - `80.56.241.255`):

```
FORBID * First statement forbids any IP
ALLOW 172.16.241.128
ALLOW 81.50.1.241
ALLOW 80.56.241.*
```

- The following example defines 2 short ranges (`172.16.241.0` - `172.16.241.127`) and (`172.16.241.129` - `172.16.241.255`) of addresses that are allowed:

```
FORBID *
ALLOW 172.16.241.*
FORBID 172.16.241.128
```

- The following example shows usage of a subnet mask to allow a range of addresses (`172.16.240.0 - 172.16.255.255`):

```
FORBID *
ALLOW 172.16.240.0, MASK=255.255.240.000
```

- The following is an example of an ALLOW type IP list:

```
FORBID *
ALLOW 172.16.130.151
ALLOW 172.16.110.*
ALLOW 172.16.241.*
ALLOW 80.56.1.*
```

There is no limit on the number of entries in a list of IP addresses. The order of ALLOW and FORBID statements is not important.

Source format of an IP List is processed at the initialization of CTSGATE (or when the REFRESH modify command is issued as shown below) to detect syntax errors.

Both [ALLOW *] and [FORBID *] statements create the following mask: `0.0.0.0`. The absolute value of a mask as a hexadecimal number defines the degree of specificity.

Location of IP Addresses List

The list of IP addresses must reside in a library allocated by a DAPARM DD statement under ECAIPLS fixed name. A one-character suffix is supported in a member name; for example: ECAIPLSx

List Name of IP Addresses List

The name of the required IP list should be specified by the new Channel parameter of CTSGATE as follows:

```
IPLIST=ECAIPLSx
```

The presence of ECAIPLS source member in a library allocated by a DAPARM DD statement is required as soon as the IPLIST channel parameter is specified in ECAPARM.

Modifying the IP Addresses List

ECAIPLS source member is available to a user for changes. ECAIPLSx source member can be refreshed dynamically by means of the modify command `REFRESH=ECAIPLSx` without need to restart CTSGATE.

Administration of IP Addresses Validation

- **Operations** – To refresh ECAIPLSx source member dynamically, the following modify command should be issued:

```
F <CTSGATE>, REFRESH=ECAIPLSx
```

> **Note**
> The IPLIST only blocks establishing connections. Refreshing the IPLIST does not affect existing connections.

- **Administrative features** – IP address validation becomes available as soon as:

  a. Proper PTF has been applied.

  b. A list of IP addresses resides in the library allocated by a DAPARM statement

  c. The PLIST channel parameter is specified in ECAPARM.

  TRACE=199 should be set on in order to track processing of the IP list.

- **Security requirements** – The feature is not mandatory. To enable the feature, the IPLIST=ECAIPLSx channel parameter should be specified in the ECAPARM parameters member. When the feature is enabled, a list of IP addresses must exist in a library allocated by a DAPARM statement.

- **Internal diagnostics** – TRACE=199 should be set to ON in order to track processing of the IP list. The ECAIPLSx member will be printed in DAPRENV.

  If the feature is enabled, information about specific IP list will be printed in DAIGLOG output.

  If the CTSGATE channel is disabled due to invalid ECAIPLSx, the detected invalid lines in ECAIPLS will be displayed

BIND

IP address that IOAGATE must use to listen for incoming connections. If you want IOAGATE to listen on a specific IP address, such as a DVIPA assigned for IOAGATE, use this parameter to identify that IP address.

Use the following syntax:

```
BIND=INADDR_ANY | IP_address | hostname
```

where

- `INADDR_ANY` instructs IOAGATE to listen for incoming connections from any IP address (adapter) on the system.

- `IP_address` or `hostname` indicates that IOAGATE BINDs to either the given IP_address or the IP_address after hostname resolution.

Default: `INADDR_ANY`

Operations

The Connector for Top Secret does not include a local user interface since almost all the functions it performs are activated via SailPoint. However, certain Connector for Top Secret operations (generally used for maintenance) are available. These operations are described in this chapter.

The following topics are discussed in this chapter:

Starting Connector for Top Secret

The Top Secret Connector Gateway starts and, in turn, automatically starts the Connector servers: Transaction Server (s) and Notification Server. As each of these components of Connector for Top Secret is successfully started, an appropriate message is displayed.

- If the Connector Gateway is active when you start the Connector for Top Secret, communication is automatically established between the gateways.
- If the Connector Gateway is not active when you start the Connector for Top Secret, the Top Secret Connector Gateway waits. When the Connector Gateway is started, it establishes communication with the Top Secret Connector Gateway.

> **Note**
> If Connector for Top Secret will be started using an automatic startup tool during system initialization, the Connector for Top Secret must be started after TCP/IP and the managed Top Secret subsystem have been started.

To start the Connector for Top Secret:

1. Issue the following operator command:

```
S CTSGATE
```

Shutting Down the Connector for Top Secret

The Connector for Top Secret Gateway (CTSGATE) shuts down the Connector Transaction Server and Notification Server, and then shuts itself down.

To shut down the Connector for Top Secret:

1. Issue the following operator command:

```
P CTSGATE
```

Starting and Stopping the Online Interceptor

The Connector Online Interceptor is started independently of the Connector for Top Secret Gateway and the Connector Transaction Server and Notification Server.

> **Note**
> It is recommended that the Online Interceptor be active at all times (even if Connector for Top Secret Gateway is down), in order to ensure that all Top Secret changes are recorded.

Configuring the automated startup of Online Interceptor is described in procedure 9 – Customize Communication Settings.

To start the Online Interceptor:

1. Issue the following operator command:

```
S CTSAONI
```

To shut down the Online Interceptor:

1. Issue the following operator command:

```
P CTSAONI
```

> **Note**
> If two or more instances of Connector for Top Secret on the same computer use the dynamic installation of the new password exit, restrictions apply to the order in which the Online Interceptors for each instance of Connector for Top Secret are shut down. The Online Interceptors must be shut down in the reverse order to which they were started.
>
> For example, if the Online Interceptor for Connector instance A was started, followed by the Online Interceptor for B and then for C, the Online Interceptors should be shut down in the order C, then B, and finally A.If two or more instances of Connector for Top Secret on the same computer use the dynamic installation of the new password exit, restrictions apply to the order in which the Online Interceptors for each instance of Connector for Top Secret are shut down. The Online Interceptors should be shut down in the reverse order to which they were started.

Starting the Offline Interceptor Manually

> **Important**
> Ensure that the Offline Interceptor has been configured before operating for the first time. For more information on configuring the connector, see Standard Offline Interceptor.

To start the Offline Interceptor manually:

1. Issue the following operator command to operate the Offline Interceptor as a started task:

```
S CTSOFLI
```

Stopping the Notification and Transaction Servers

The Notification Server and/or Transaction Server can be stopped without having to shut down the Connector for Top Secret Gateway.

To stop the Notification Server or Transaction Server:

1. Issue the following command:

```
F CTSGATE,STOPASID=<nn>
```

where *<nn>* is the ID of the server to be stopped.

For example:

The Notification Server is typically server 01 and the Transaction Servers are server 02, 03 and so on. Given this situation, you would use the commands that follow.

- To stop the Notification Server:

```
F CTSGATE,STOPASID=01
```

- To stop the Transaction Server:

```
F CTSGATE,STOPASID=02
```

Restarting the Notification and Transaction Servers

If either the Notification Server or a Transaction Server terminates for any reason, it can be restarted without having to shut down the Connector for Top Secret Gateway.

To restart the Notification Server or Transaction Server:

1. Issue the following command:

```
F CTSGATE,STARTASID=<nn>
```

where *<nn>* is the ID of the server to be restarted.

For example:

The Notification Server is typically server 01 and the Transaction Server is server 02. Given this situation, you would use the commands that follow.

1. To restart the Notification Server:

```
F CTSGATE,STARTASID=01
```

2. To restart the Transaction Server:

```
F CTSGATE,STARTASID=02
```

Viewing System Status

This section describes how to display the list of servers and the status of each. If the server is currently handling a specific service, the service is also displayed.

To view the system status:

1. Issue the following command:

```
F CTSGATE,STATUS
```

Scripts

The Connector for Top Secret functions (described in this book) are designed to meet fundamental Managed System requirements.

In addition to the basic requirements, site-specific requirements may exist in established Managed System, and Managed System may have changing needs. Therefore, Connector for Top Secret enables customization of the functions. You can customize any Connector for Top Secret function by writing scripts.

A script is a group of statements that perform one or more actions and that manipulate standard SailPoint and Managed System-specific fields. A script may include conditions that determine when actions should be performed. For example, cleanup activities should be run if the Connector for Top Secret function does not execute successfully.

Scripts can also be used to automate any required actions that are currently performed manually.

The Connector for Top Secret has the ability to call a script before and after executing a Connector for Top Secret function. Parameters may be received from and returned to Connector for Top Secret functions by including script commands in the scripts.

The following topics are discussed in this chapter:

Writing a Script

Scripts are written in REXX language and must adhere to REXX syntax rules. Use a text editor to write the scripts and store them in the following library:

```
<prefix>.<version>.USER.CLIST
```

where:

- `<prefix>` – Value set in parameter OLPREFS in member LOADCTS in the Connector INSTALL library.
- `<version>` – Value set in parameter OLVERS in member LOADCTS in the Connector INSTALL library.

> **Note**
> This library name is defined in parameter SCRIPT_DIR in member RSSPARM stored in the Connector PARM library.

Scripts may be executed before and/or after a Connector for Top Secret function is executed (see Executing a Script for details). All scripts are activated under the Connector for Top Secret address space.

A script should consist of the following sequence of actions:

1. Read the current Connector for Top Secret variables into REXX variables.
2. Examine the REXX variables, modify them (if required) and execute additional actions as required.
3. Update the Connector for Top Secret variables with the resultant REXX variable values.

To enable the script to manipulate Connector for Top Secret variables, Connector for Top Secret provides the script command CTSAVAR which reads the current Connector for Top Secret variables into REXX variables. After the REXX variables have been examined and modified, the script command CTSAVAR is used to update the Connector for Top Secret variables with the resultant REXX variable values. Script commands are described later in this chapter.

Since the script is executed in the TSO-REXX environment, Connector for Top Secret scripts can also issue TSO and REXX commands.

When writing a script, the following must be taken into account:

- Certain variables are unmodifiable (see Script Variables).
- The return code of a script must be set by the script before it terminates (see Setting the Return Code).
- Several TSO commands can only be issued indirectly from within a Connector for Top Secret script (see TSO Considerations).
- Due to REXX limitations, field names for keywords must be all uppercase. This applies both when defining keywords in SailPoint and when specifying the field name of the keyword in a script.

Executing a Script

The Connector for Top Secret determines which scripts to execute and whether or not to run the scripts by examining the values of the following parameters in member RSSAPI in the Connector PARM library.

Pre/Post-Scripts

One script can be called before a Connector for Top Secret function executes and another script (or the same one) can be called after the Connector for Top Secret function executes. The scripts are referred to as:

- Pre-script – This script is run immediately before the Connector for Top Secret function executes.
- Post-script – This script is run immediately after the Connector for Top Secret function executes.

The Connector for Top Secret execution of the functions is as follows:

1. If a pre-script is specified and should be executed, the Connector for Top Secret invokes it. Otherwise, execution of the Connector for Top Secret function begins from step 3.
2. The return code of the pre-script is checked. If the script has returned SKIP or FATAL, the Connector for Top Secret does not invoke the functions and execution skips to step 4.
3. If the Connector for Top Secret function is to be invoked, it is called.
4. If a post-script is specified and should be executed, the Connector for Top Secret invokes it, regardless of the return codes from the pre-script and the actual Connector for Top Secret function.
5. The final return code is the return code of the last item (Pre/Post-script or actual Connector for Top Secret) that was executed.
6. If both of the following conditions are satisfied:
    - The Connector function is an ADD or UPDATE function.
    - A GET post-script is specified for the object and should be executed.

    Connector for Top Secret invokes the GET post-script after invoking any ADD or UPDATE Pre/Post-scripts required by the function. This is done to synchronize the Managed System database with the SailPoint database.

Structure of the RSSAPI Member

Each line in the RSSAPI member (excluding comment lines) represents a Connector for Top Secret call which is followed by various parameters. The parameters influence the behavior of the Connector for Top Secret. Each line conforms to the following format:

```
Managed System-type Managed System-name API-name NUM PRF ACF POF PRN PON
```

The following parameters appear in each line of the RSSAPI members:

- **Managed System-type** – Managed System type or a hyphen (–). Hyphen represents all Managed System types.
- **Managed System-name** – Managed System name or a hyphen (–). Hyphen represents all Managed System names.

- **API-name** – Connector for Top Secret functional name.
- **NUM** – Maximum number of keywords that may be added to a transaction by the Pre-script for use by the Connector for Top Secret function or Post-script.
- **PRF** – Y/N flag indicating whether or not to invoke the Pre-script.
- **ACF** – Y/N flag indicating whether or not to invoke the actual Connector for Top Secret routine.
- **POF** – Y/N flag indicating whether or not to invoke the Post-script.
- **PRN** – Pre-script name.
- **PON** – Post-script name.

For example:

```
- MVS ADDUSER 05 N Y Y TSSADDU1 TSSADDU2
```

This line indicates that the Connector for Top Secret function ADDUSER in Managed System MVS operates as follows:

- Does not call the Pre-script TSSADDU1.
- Performs the actual Connector for Top Secret routine.
- Calls the post-script TSSADDU2.

A script can be called as a pre-script and/or post-script for any Connector for Top Secret function. For example, assume that you wish to perform the same action before every Connector for Top Secret function. First write a script which performs the desired action. Next, specify the script name in parameter *PRN* for all the Connector for Top Secret functions and set the parameter *PRF* to Y.

Script Variables

When executed within the Connector for Top Secret environment, a script has access to certain predefined variables. These variables are used for a variety of purposes, including:

- Receiving and modifying values of Managed System-specific fields
- Receiving information regarding the Connector for Top Secret environment
- Controlling actions performed by the Connector for Top Secret function
- Controlling execution of subsequent scripts

These topics are described in detail in this section and following sections.

Types of Variables

The Connector for Top Secret distinguishes between the different kinds of information that is passed to and from the scripts by using different categories of variables; each category of variable is assigned a different prefix.

The categories of variables are:

- CTSA0.<field_name> (Read-only) – Predefined variables representing SailPoint structure fields (common to all Managed System types) or Connector for Top Secret environment information.

  For a full list of CTSA0 variables, see CTSA0 Variables.

- CTSA1.<field_name> (Read/write) – Predefined Managed System-specific variables containing values sent from SailPoint. If the value of a field was modified in SailPoint, the corresponding CTSA1 variable contains the modified value. The value of the CTSA1 variable can be further modified by the script.

  For more information, see “Appendix E – Managed System-Specific fields.”

  In addition to the variables described above, the following special CTSA1 variables are available:

  - CTSA1.MSG (Write-only) – Used to send a message from the script to SailPoint. This variable is defined by the script; use of the variable is optional. A string value assigned to this variable is sent to SailPoint and is displayed in the Transaction Properties window.

  - CTSA1.RC (Write-only) – Return code of the pre- or post-script. Before Connector for Top Secret executes a function, it checks the pre-script return code to determine whether or not to execute the function. Therefore, the return code of a pre-script must be set by the script before it terminates.

    For more information, see Setting the Return Code.

- CTSA2.<field_name> (Read-only) – Predefined Managed System-specific variables containing the original values of fields. The values are retrieved from the Managed System.

  These variables are similar to CTSA1 variables. The same field can be referenced with both the CTSA1 and CTSA2 prefixes. However, the CTSA1 variable contains the new value of the field as sent from SailPoint while the CTSA2 variable contains the original “old” field value.

  After the Managed System has been updated, each CTSA2 variable passed to the script contains the new, updated value of the corresponding field.

- CTSA9.<rss_parameter> (Read-only) – Variables containing values of parameters in the RSSPARM file.

  The script only receives CTSA9 variable values when the parameter SEND_RSSPRM_TO_SCRIPT in the RSSPARM file is assigned the value Y.

List Fields

Managed System-specific information may include List (table) fields. A List field is passed to a script as a string of entries. This string includes specific characters that are designated to separate entries and sub-fields in the list.

The default entry separator value is a comma (,). To specify a value other than the default, assign any printable character to the Managed System parameter SCRIPT_ SEP_ENTRY in the RSSPARM file.

The default sub-field separator value is a semicolon (;). To specify a value other than the default, assign any printable character to the Managed System parameter in SCRIPT_SEP_FIELD in the RSSPARM file.

For example, given the following:

- SCRIPT_ SEP_ENTRY is set to ^
- SCRIPT_SEP_FIELD is set to $

A List field contains the following data:

A1	A2	A3
B1	B2	B3
C1	C2	C3
D1	D2	D3

The above table of data values will be passed to the variable in a script as:

```
A1$A2$A3^B1$B2$B3^C1$C2$C3^D1$D2$D3
```

CTSA0 Variables

CTSA0 variables and their values are listed on the pages which follow. Note that the variables are listed according to Connector for Top Secret function type.

Connector for Top Secret Environment Variables

The following table lists variables containing information that relates to the environment in which the script is executed. These variables are common to all Connector for Top Secret function types.

Variable	Value	Description
CTSA0.ACTION	SCRIPTPRE	Call as a Pre-script
	SCRIPTPOST	Call as a Post-script

Variable	Value	Description
CTSA0.FUNC_NAME	ADDU2UG	Connect a user to a group
	ADDUG	Create a new group
	ADDUSER	Create a new user
	DELU2UG	Disconnect a user from a group
	DELUG	Delete a group
	DELUSER	Delete a user
	GETUGS	Obtain group details
	GTRSPRM	Obtain Managed System details
	GTUG2UC	Obtain user to group connection details
	GTUSERS	Obtain user details
	REVUSER	Revoke/Restore a user
	UPD_PASS	Update password of a user
	UPDUSER	Update user details. **Note** When a transaction is issued from SailPoint for any of the following actions: • changing the user's password • revoking the user • restoring the user With no added Managed System-specific or user-defined fields, the transaction type can be either UPDUSER, or it can be UPD_PASS / REVUSER, depending on how the action was initiated. However, when the transaction includes added Managed System-specific or user-specific fields, the transaction type is always UPDUSER.
	UPDU2UG	Update user to group connection details
	UPDUG	Update group details
CTSA0.ACT_RC	OK	Return code of the actual Connector for Top Secret. Available in the Post-script only
	ERROR	

Variable	Value	Description
	FATAL	
	<UNDEFINED>	
CTSA0.ADM_G		Group of administrator performing the operation
CTSA0.ADM_VER	4.0.01	Current version of Connector for Top Secret.
CTSA0.ADM_MOD	1	Reserved.
CTSA0.ADM_ID		User ID of administrator performing the operation.
CTSA0.ADM_PASSWD		Managed System Administrator password or phrase. This parameter is passed to the scripts, only when the parameter SEND_PWD_TO_SCRIPT in the file RSSPARM is set to Y.
CTSA0.PRE_RC	OK	Return code of the Pre-script. Available in the Post-script only
	WARN	
	SKIP	
	ERROR	
	FATAL	
	<UNDEFINED>	

Managed System User Connector for Top Secret Function Variables

The following table describes variables available for any Managed System user operation.

Variable	Value	Description
CTSA0.USER_ID		User ID
CTSA0.USER_PWD	<password or phrase>	Managed System user new password or phrase. This parameter is passed only when parameter PASS_PASSWORD in file RSSPARM is set to Y.
CTSA0.UG_DEF		Default group of the user
CTSA0.USER_ADMIN	1	User is a regular user
	2	User is an auditor
	3	User is an administrator
	4	User is an auditor and administrator
	5	Ignore this field
CTSA0.USER_STA	1	User is revoked
	2	User is restored
	3	Ignore this field

Variable	Value	Description
CTSA0.PWD_LIFE	1	Permanent
	2	Temporary
	3	Ignore this field
CTSA0.RSS_NAME		Name of the Managed System
CTSA0.RSS_TYPE		Type of Managed System

Group Connector for Top Secret Functions

The following table describes variables available for any Group Connector for Top Secret operation.

Variables	Value	Description
CTSA0.GROUP_ID		Group ID
CTSA0.GROUP_PR		Parent Group
CTSA0.RSS_NAME		Name of the Managed System
CTSA0.RSS_TYPE		Type of Managed System

Managed System User–Group Connector for Top Secret Functions

The following table describes variables available for any Managed System User—Group Connector for Top Secret operation.

Variable	Value	Description
CTSA0.GROUP_ID		Group ID
CTSA0.USER_ID		User ID
CTSA0.U2UG_ATR	1	Regular connection between a user and a group
	2	Connection is of user to its default group
	3	Ignore this field
CTSA0.U2UG_MSC	1	User is a regular member of the group
	2	User is an administrator of the group
	3	User is an auditor of the group
	4	User is administrator and auditor of the group
	5	Ignore this field
CTSA0.RSS_NAME		Name of the Managed System
CTSA0.RSS_TYPE		Type of Managed System

Setting the Return Code

Before Connector for Top Secret executes a Connector function, it checks the Pre-script's return code to determine whether or not to execute the function. Therefore, the return code of a Pre-script must be set by the script before it terminates.

To set the return code in a Pre-script, set the **CTSA1.RCODE** variable to one of the following values:

- **OK** – Pre-script processing completed successfully.
- **SKIP** – Do not call the Connector for Top Secret function, but do call the Post-script.
- **WARN** – Continue and call both the Connector for Top Secret function and the Post-script.
- **ERROR** – Continue and call both the Connector for Top Secret function and the Post-script.
- **FATAL** – Do not call the Connector for Top Secret function, but do call the Post-script.

To set the return code in a Post-script, set the **CTSA1.RCODE** variable to one of the following values:

- **OK** – Post-script processing completed successfully.
- **WARN** – Post-script processing completed with warning.
- **SKIP** – (GET Post-script only) Do not send the retrieved object (for example, Managed System user, resource) to SailPoint. Using this value enables a Post-script to filter the objects to be aggregated to SailPoint, regardless of the aggregation.
- **ERROR** – Post-script processing completed with error.
- **FATAL** – Post-script processing completed with fatal error.

TSO Considerations

The TSO environment is available to scripts and therefore most TSO commands can be issued by the script. However, the following TSO commands cannot be issued directly from within a Connector for Top Secret script.

- Authorized TSO commands
- SUBMIT command

These commands and the method in which they can be issued are described below.

Authorized TSO Command

Authorized TSO commands cannot be issued directly from a script.

To execute an authorized TSO command, use the CTSAEXC command processor. For example, to activate command IDCAMS DEFINE (which is an authorized command processor) enter the following statement in the script:

```
CTSAEXC DEFINE ....define command arguments...
```

SUBMIT Command

The TSO SUBMIT command cannot be issued directly from a script.

To submit a job, use the CTSASUB command processor. CTSASUB receives the name of a DD statement which contains the job JCL image.

For example

Include the following statement in the script:

```
CTSASUB TEST
```

This activates command CTSASUB which reads DD statement TEST and writes its contents to the JES internal reader. An exclusive JES internal reader is allocated to the Connector for Top Secret started task via DD statement INTRDR in the task's JCL. If no DD statement is specified for command CTSASUB, a default DD statement CTSJOBIN is used and its contents are written to the JES internal reader.

Online TSS Commands

Online TSS commands cannot be issued directly from a script. To issue online TSS commands from a script, use the following command:

```
ADDRESS TSO "CTSAEXC <online_tss_command
```

Script Commands

The following table provides a brief description of the available script commands.

Command	Description
CTSAEXC	Activates authorized TSO command processors.
CTSASUB	Submits a job.
CTSAVAR	Copies Connector for Top Secret variables to REXX variables. *OR* Updates Connector for Top Secret variables based on the REXX variables.
CTSASYNC	Generates a synchronization event.

CTSAEXC

Purpose	Activates authorized TSO command processors in the Connector for Top Secret script
Syntax	`CTSAEXC <TSO_command>` where `<TSO_command>` is any authorized TSO command.
Description	Activates authorized TSO command processors in a Connector for Top Secret script.
Example	The following statement activates the IDCAMS DEFINE command: `CTSAEXC DEFINE ALIAS NAME(USER1) RELATE (CATALOG.USERS)`

CTSASUB

Purpose	Submits jobs from a Connector for Top Secret script
Syntax	`CTSASUB [<ddname>]` where `<ddname>` is the name of the DD statement which contains the JCL cards.
Description	Submits jobs from a Connector for Top Secret script.
Example	The following statement submits the job contained in the dataset allocated to DD statement MYJOB: `CTSASUB MYJOB`

CTSAVAR

Purpose	Copies Connector for Top Secret variables to REXX variables or updates Connector for Top Secret variables based on the REXX variables
Syntax	`CTSAVAR {GET | PUT} <token>` where: • `GET` – Sets the REXX variables to the values in the Connector variables. • `PUT` – Updates the Connector variables to the values in the REXX variables. • `<token>` – Token passed to the script as a parameter which is used to identify the relevant Connector for Top Secret parameters by the CTSAVAR command.
Description	Copies Connector for Top Secret variables to REXX variables or updates Connector for Top Secret variables based on the REXX variables.

Example	The following statement sets the REXX variables to the values in the Connector for Top Secret variables:
	`CTSAVAR GET <token>`
	The following statement updates the Connector for Top Secret variables to the values in the REXX variables:
	`CTSAVAR PUT <token>`

CTSASYNC

Purpose	Triggers synchronization of Managed System details with SailPoint
Syntax	One of the following: • `CTSASYNC <token> USER <user>` • `CTSASYNC <token> GROUP <group>` • `CTSASYNC <token> CONN <conn_user>`

Purpose	Triggers synchronization of Managed System details with SailPoint
	`<conn_group>` where: • `<token>` – Token passed to the script as a parameter which is used to identify the relevant Connector parameters by the CTSASYNC command. • `<user>` – User name of the user to be synchronized. • `<group>` – Group name of the group to be synchronized. • `<conn_user>` – User name of the user in the connection to be synchronized. • `<conn_group>` – Group name of the group in the connection to be synchronized.
Description	Triggers synchronization of the Managed System with SailPoint.
Example	The following statement triggers synchronization of the USER details between SailPoint and the Managed System. `CTSASYNC <token> USER USER1`

Maintenance

The Connector for Top Secret provides general maintenance procedures for the administrator's use. These procedures are run by submitting batch jobs.

The available procedures are listed below:

Formatting the Diagnostic Level Dataset

This procedure formats the diagnostic level dataset and should only be used at the request of Technical Support.

Member CTSDIAG in the Connector JCL library is a sample job which activates this procedure. The M= parameter refers to the relevant diagnostic flags member, which can be CTSACS, CTSACD or CTSAONI. These members are in the PARM library and should be updated with relevant flags based on requests from Technical Support.

Displaying Local Connector for Top Secret Data

This procedure creates a report containing information about the local Connector for Top Secret instance. This job should only be used at the request of Technical Support.

This procedure is activated by member STATUS1 in the Connector JCL library. When STATUS1 job is submitted, the following Connector for Top Secret information is written to the job's sysout:

- SMP's PTFs
- RSSPARM
- CTSPUSR
- RSSAPI
- RSSTABL
- RSSKWDS
- CTSPARM

Setting Transmitted Data Encryption

> **Note**
> Transmitted Data Encryption is not supported for communication with IdentityNow

All data transmitted between Connector for Top Secret and IdentityIQ is (optionally) encrypted using an encryption key from the Encryption Key file.

Since the transmitted data must be understood by IdentityIQ and by Connector for Top Secret, the Encryption Key file in both systems must match.

If encryption was activated during Connector for Top Secret installation, a synchronized Encryption Key file already exists. If you wish to change the Encryption Key file, it must be changed from IdentityIQ and then synchronized with Connector for Top Secret.

Similarly, the enabling or disabling of Transmitted Data Encryption between IdentityIQ and Connector for Top Secret must be synchronized in both systems.

Procedures for changing the encryption key and for enabling or disabling data encryption are described in the *IdentityIQ Administration Guide*. These are two-part procedures in which part of the procedure is performed in IdentityIQ and part in Connector for Top Secret.

Follow the procedures described in the above location. At the appropriate point in the procedure, you are instructed to perform the Connector for Top Secret side of the procedure as described in this section.

The following procedures are described below:

- Changing the Transmitted Data Encryption Key.
- Disabling (or enabling) the encryption of transmitted data.

> **Note**
> Follow the procedures for encryption of security data in the *IdentityIQ Administration Guide*. At the appropriate point in that procedure, perform the steps listed below.

To change the Transmitted Data Encryption key for the Connector for Top Secret platform:

1. Stop the Connector for Top Secret Gateway and servers by specifying the following operator command:

   ```
   P CTSGATE
   ```

2. Transfer a copy of the Encryption Key file from IdentityIQ to Connector for Top Secret as described in 9.4 – Set up secured communication in the 9 – Customize Communication Settings installation procedure.

3. Start the Connector for Top Secret Gateway (which automatically starts the Connector for Top Secret servers) by specifying the following operator command:

   ```
   S CTSGATE
   ```

To disable (or enable) the encryption of transmitted data:

> **Note**
> Ensure that the option Encryption in the Platform Properties Window in IdentityIQ is set to Off (for disabled) or On (for enabled) in accordance with the option selected in this procedure

1. Stop the Connector for Top Secret Gateway and servers by specifying the following operator command:

   ```
   P CTSGATE
   ```

2. Edit member CTSPUSR in the Connector PARM library.

   a. Set parameter **ENCR_EXT_ACT** to **N** to disable Transmitted Data Encryption.

   OR

   Set parameter **ENCR_EXT_ACT** to **Y** to enable Transmitted Data Encryption.

   b. b. Save the member.

3. Start the Connector for Top Secret Gateway (which automatically starts the Connector for Top Secret servers) by specifying the following operator command:

   ```
   S CTSGATE
   ```

Setting Stored Data Encryption

All data which is stored temporarily in Connector for Top Secret is encrypted using a Stored Data Encryption key (which differs from the Transmitted Data Encryption key). For example, sensitive security information that is written by the Interceptors to the Connector queue file is encrypted using the Stored Data Encryption key.

As part of the Connector for Top Secret installation procedure, an encryption key for stored data is created. However, the Stored Data Encryption key can be changed periodically to strengthen security.

> **Note**
> Before changing the encryption key, or before enabling or disabling Stored Data Encryption, verify that the Connector queue file does not contain data. This is because the Notification Server cannot process data in the Connector queue file which was encrypted by a previous key.

The Stored Data Encryption key is used internally by Connector for Top Secret; there is no need to synchronize this key with SailPoint or any other Connector for Top Secret installation.

The following procedures are described below:

- Generating a new Stored Data Encryption key.
- Disabling (or enabling) the encryption of stored data.

To generate a new Stored Data Encryption key

1. Verify that the Connector queue dataset does not contain data (using the procedure Print the Connector Queue).
2. Stop the Connector Online Interceptor by specifying the following operator command:

```
P CTSAONI
```

3. Stop the Connector for Top Secret Gateway and servers by specifying the following operator command:

```
P CTSGATE
```

4. Edit member CTSKGEN in the JCL library. Next, submit the job and a new key is generated. All job steps must end with a condition code of 0.
5. Start the Connector for Top Secret Gateway (which automatically starts the Connector for Top Secret servers) by specifying the following operator command:

```
S CTSGATE
```

6. Start the Connector Online Interceptor by specifying the following operator command:

```
S CTSAONI
```

To disable (or enable) the encryption of stored data

1. Verify that the Connector queue dataset does not contain data (using the procedure Print the Connector Queue).
2. Stop the Connector Online Interceptor by specifying the following operator command:

```
P CTSAONI
```

3. Stop the Connector for Top Secret Gateway and servers by specifying the following operator command:

```
P CTSGATE
```

4. Edit member CTSPRSV in the Connector PARM library.

   a. Set parameter **ENCR_INT_ACT** to **N** to disable Stored Data Encryption.

   OR

   Set parameter **ENCR_INT_ACT** to **Y** to enable Stored Data Encryption.

   b. Save the member.

5. Start the Connector for Top Secret Gateway (which automatically starts the Connector for Top Secret servers) by specifying the following operator command:

```
S CTSGATE
```

6. Start the Connector Online Interceptor by specifying the following operator command:

```
S CTSAONI
```

Formatting the Offline Interceptor Dataset

This procedure formats the Offline Interceptor dataset. This procedure is used during installation and to initialize the Offline Interceptor dataset if it was deleted and re-allocated.

Member FORMOFLI in the Connector JCL library is a sample job which activates the Standard Offline Interceptor procedure.

Recovering the Offline Interceptor After Failure

The Offline Interceptor operates in one of the following modes:

- **Init Mode** – The Offline Interceptor typically runs in Init mode the first time it is activated. In this mode, Offline Interceptor IMG files are created. These files contain an image of the security objects that are currently defined in Top Secret. When operating in Init mode, the Offline Interceptor does not send any events to IdentityIQ.
- **Standard Mode** – The Offline Interceptor typically runs in Standard mode each time it is activated after its initial run. In this mode, the Offline Interceptor compares the current snapshot of Top Secret security objects to that in the IMG files. The differences represent security events and are sent to IdentityIQ.

By default, the Offline Interceptor operates in Init mode if the IMG files do not exist, and operates in Standard mode if the IMG files exist.

In the event an Offline Interceptor failure occurs, (for example due to Top Secret shutdown or manual cancellation), the IMG files may be incomplete or corrupt. To rectify this situation, the Offline Interceptor must be forced to run again in Init mode.

After the Offline Interceptor is forced to run again in Init mode, aggregation must be performed in IdentityIQ. The Aggregation operation should be done as soon as possible after the completion of the Offline Interceptor to ensure that the IMG files and IdentityIQ are synchronized. Until the Aggregation is performed, Top Secret security commands that were issued between the failed Offline Interceptor run and the latest Init run of the Offline Interceptor are not updated in IdentityIQ

To force the Offline Interceptor to run in Init mode:

> **Note**
> SailPoint recommends that you consult with Technical Support before using this procedure.

Method A

a. Delete Offline Interceptor working and IMG files.

   All datasets with the following prefix must be deleted:

   ```
   %OLPREFS%.%OLVERS%.%RSSNAME%.OFL*
   ```

b. Run the Offline Interceptor.

   This may be done manually or automatically. The Offline Interceptor will run in Init mode and creates new IMG files. See Starting the Offline Interceptor Manually for more information.

Method B

Force the Offline Interceptor to run in Init mode by doing one of the following:

- Use sample job CTSOFLMI in the Connector JCL library. This job executes the Standard Offline Interceptor while setting the mode to Init. Make sure the user submitting the job has permissions to all files used by the job.

  OR

- Specify the following command

  ```
  S CTSOFLI,PARM='-I rssName'
  ```

Any future run of the Offline Interceptor will automatically be performed in Standard mode.

Initializing the Connector Queue

This procedure is used to initialize the Connector queue during the installation process. It is also used to initialize the Connector queue dataset if the dataset was deleted and re-allocated.

1. Stop all the Connector for Top Secret and Online Interceptor processes. Ensure that all the Connector for Top Secret and Interceptor processes have shut down.
2. Submit FORMQUE member in the Connector JCL library (a sample job which activates this procedure).
3. Restart the Connector for Top Secret and the Online Interceptor.

Changing the Size of the Connector Queue

Depending on the level of activity in the Managed System managed by Connector for Top Secret, it may become necessary to change the size of the Connector queue. This section describes how to change the queue size.

> **Note**
> You can change the size of the Connector queue dataset regardless of whether or not it contains data. Data contained in the queue is preserved during this procedure.

To change the size of the Connector queue:

1. Stop the Connector Online Interceptor by specifying the following operator command:

   ```
   P CTSAONI
   ```

2. Stop the Connector for Top Secret Gateway and servers by specifying the following operator command:

   ```
   P CTSGATE
   ```

3. Ensure that all the Connector for Top Secret and Interceptor processes have shut down.
4. Allocate a new Queue dataset with increased DASD space, using a dataset name suffix other than ".QUEUE".

   For example if your existing Queue dataset is "CTSA.V400.QUEUE", allocate the new Queue dataset as "CTSA.V400.QUEUEN2".

   For a sample allocation of a Queue dataset, see job FORMCTS in the Connector INSTALL library.
5. Edit sample job CTSQCR in the Connector JCL library, and specify the dataset name suffix for the new Queue (example "QUEUEN2").

   This job consists of two steps:

   a. Step 1 calls the CTSAQCR JCL procedure which calls the CTSQCR utility with appropriate parameters. The utility expects the new Queue dataset to be pre-defined and allocated. The utility formats the new Queue dataset and then copies the contents of the active Queue dataset to the new Queue dataset.

   b. Step 2 calls the IDCAMS utility to change the suffix of the active Queue dataset (for example, to ".OLDQUEUE") so that it is no longer active, and to change the suffix of the new Queue dataset so that it

becomes the active Queue dataset. Renaming of datasets can also be done using other methods available under z/OS, such as TSO or ISPF commands.

6. Submit the job.

7. Upon the successful termination of job CTSQCR and dataset rename commands, restart the Connector for Top Secret and the Online Interceptor. The new Queue dataset is now active.

Print the Connector Queue

This procedure prints the contents of the Connector queue dataset.

Member PRTQUE in the Connector JCL library is a sample job which activates this procedure.

Renaming a Managed System

You can change the name assigned to the Managed System during Connector for Top Secret installation.

Once the Managed System has been defined in SailPoint, renaming the Managed System involves changes both on the SailPoint workstation and in Connector for Top Secret. The procedure described below changes the name of the Managed System only in Connector for Top Secret.

> **Note**
> This procedure should only be performed within the framework of the procedure for changing the name of the Managed System in SailPoint.

The name of an Managed System is changed in Connector for Top Secret using utility CTSAADPT.

Member CTSADAPT in the JCL library contains a sample job to invoke the JCL procedure for the CTSAADPT utility (procedure <prefix>AADPT, where <prefix> is the prefix of your Connector JCL procedures). The utility should be run when Connector for Top Secret and the Interceptors are inactive.

Specify the old and new Managed System names as values of parameters FROMRSS and TORSS respectively when invoking the <prefix>AADPT procedure to execute the CTSAADPT utility.

This will change all occurrences of the old Managed System name in Connector for Top Secret datasets to new Managed System name. Note that the RSSPARM member itself is also modified.

The utility reports which datasets (and how many records in each) were modified. Most datasets considered for modification are allocated via JCL (see the relevant DD statements in the <prefix>AADPT JCL procedure).

One exception to this is the OFLRIMG dataset, used by Offline Interceptor utility. The DSNAME for this dataset is determined during CTSAADPT execution by the value of the RSS_WORK_DIR parameter for TORSS name in the

RSSPARM member. The DSNAME dynamically allocated is the concatenation of the value of RSS_WORK_DIR with the suffix OFLRIMG (for example: CTSA.V400.MYRSS.OFLRIMG). If this DSNAME does not exist, no dynamic allocation (and thus, no modification) is done by CTSAADPT for this dataset.

Before running the CTSAADPT utility, verify that the RSS_WORK_DIR parameter value conforms to MVS dataset naming conventions.

> **Note**
> If the new Managed System name is longer than 8 characters, additional adjustments must be performed. For more information, see 11 – Adjust for Longer Managed System Names.

The CTSAADPT utility does not modify the Connector for Top Secret started task procedures. After running the utility, the following additional changes must be done manually.

Changes to Online Interceptor

Change the name of the Managed System in the CTSAONI (Online Interceptor) started task procedure as follows:

1. Stop all the Connector for Top Secret and Online Interceptor processes. Ensure that all the Connector for Top Secret and Interceptor processes have shut down.
2. Edit the CTSAONI member in the PROCLIB library containing Connector for Top Secret procedures. The name of the PROCLIB library is defined in variable %PROCLIB% of member DEFPARMS in the INSTALL library.
3. Locate the following line:

   ```
   RSS=<rss_name>
   ```

4. Modify the value for *<rss_name>* to the new Managed System name.
5. Save the member.
6. Restart the Connector for Top Secret and the Online Interceptor.

Changes to Offline Interceptor

Change the name of the Managed System in the CTSOFLI (Offline Interceptor) procedure as follows:

1. Edit member CTSOFLI in the PROCLIB library containing Connector for Top Secret procedures.

   The name of the PROCLIB library is defined in variable %PROCLIB% of member DEFPARMS in the INSTALL library.

2. Locate the following line:

   ```
   RSS=<rssName> RSSQ=<rssName>
   ```

3. Modify the value for <rssName> to the new Managed System name.
4. Save the member.

5. Edit member CTSOFLI in the Connector for Top Secret JCL library. The 3rd qualifier of the Offline Interceptor datasets contains the managed system name. Change the qualifier to the new managed system name.

Filtering Interception Messages

Events affecting the Managed System database that are intercepted by Connector for Top Secret are recorded in Connector for Top Secret log files. However, not all intercepted events are relevant for Connector for Top Secret. You can set RSSPARM parameters in the member RSSPARM to filter the interception messages that are recorded in the log files.

The following Managed System parameters from the member RSSPARM are used to control filtering of interception messages.

LOG_INTERCEPT_MSG

Specifies the types of interception messages to be recorded in the CD log file.

Possible values for this parameter are described in the following list:

- **ALL** – All messages are recorded, indicating in each case whether the intercepted event was sent to SailPoint. Default.
- **ACCEPTED** – Only Managed System data included in aggregation (and therefore sent to SailPoint) is recorded.
- **IGNORED** – Only Managed System data not included in aggregation (and therefore not sent to SailPoint) is recorded.
- **NONE** – No messages are recorded.

LOG_GET_MSG

Specifies the filtering option for intercepted messages generated by the synchronization action (Managed System Retrieval Transaction). The messages are recorded in the CS log file.

Possible values for this parameter are described in the following list:

- **ALL** – All Sync messages are recorded. Default.
- **NONE** – No Sync messages are recorded.

To filter interception messages:

1. Stop the Connector for Top Secret (described on “Shutting down the Connector for Top Secret” on page 57).
2. Edit the RSSPARM member in the Connector PARM library.
3. Insert or modify either or both of the following parameters as necessary.

```
rss_name    LOG_INTERCEPT_MSG      ALL    <== Modify as required
rss_name    LOG_GET_MSG            ALL    <== Modify as required
```

4. Save the member and exit.
5. Restart the Connector for Top Secret.

Interception Acknowledgment

The Connector for Top Secret sends intercepted Managed System events to IdentityIQ. When the Interception Acknowledgment function is active, IdentityIQ sends acknowledgment for events received to the Connector for Top Secret. This process is managed by the **INTERCEPT_SEND_MAX** parameter in the RSSPARM file. This parameter determines whether or not Connector for Top Secret waits for acknowledgment from IdentityIQ for each intercepted Managed System event sent before sending the next event.

- When the **INTERCEPT_SEND_MAX** parameter is not present in the RSSPARM file, or is set to 0, the interception acknowledgment mechanism is disabled. Connector for Top Secret sends events without waiting for acknowledgment.
- When **INTERCEPT_SEND_MAX** is set to **1**, Connector for Top Secret waits for acknowledgment after each event is sent.

To set Interception Acknowledgment

1. Stop the Connector for Top Secret.

   See Starting and Stopping the Online Interceptor.
2. Edit member RSSPARM in the Connector PARM library.

```
ALL_RSS        INTERCEPT_SEND_MAX        1
```

3. Save the member and exit.
4. Restart the Connector for Top Secret.

Maintain User-Defined Field Related Keywords

When changes are done to User Defined fields definitions in Top Secret (new User Defined Fields are added or User Defined Fields are altered or deleted), the User Defined Fields related attributes in the Account entity should be re-defined according to the new User Defined Fields definitions.

To re-define the User Defined Fields related attributes

1. Update Application schema in IdentityIQ or Source schema in IdentityNow with the new User Defined fields.
2. Aggregate all Accounts in order to retrieve the new User Defined Fields data.

Remove Support for User Defined Fields

If there is a need to stop handling User Defined Fields data via Connector for Top Secret and SailPoint, the User Defined Fields support can be removed. The removal process includes removing the User Defined Fields related attributes defined for Account entity in IdentityIQ Application schema or IdentityNow Source schema and removing all the User Defined Fields data from accounts.

To remove support for User Defined Fields:

1. Delete all User Defined fields from application schema in IdentityIQ or Source schema in IdentityNow.
2. Aggregate all Accounts in order to remove the Custom Fields data.

Appendix A: Maintaining Connector for Top Secret using SMP/E

The design of the Connector for Top Secret SMP/E implementation accomplishes the following:

- Having as few modifications to the existing installation process as possible.

  This is achieved by enhancing the installation process to load the SMP/E dataset from tape into datasets with customizable names, and establish a complete SMP/E environment.

- Providing the user with an SMP/E environment that represents, as closely as possible, the target system, and that requires very few local modifications and tailoring during the Connector for Top Secret installation.

  This is achieved by shipping a complete set of SMP/E datasets, representing the environment being installed by the Connector for Top Secret installation process. These datasets are pre-loaded with information representing the environment installed. There is no need to perform any of SMP/E's traditional installation steps (that is, RECEIVE/APPLY/ACCEPT) to perform the Connector for Top Secret installation. Upon completion of the installation process, the SMP/E datasets loaded represent the installed Connector for Top Secret product environment.

This appendix describes the following information.

Packaging of Connector for Top Secret using SMP/E

The Connector for Top Secret product is shipped using a pre-installed SMP/E environment. The environment shipped includes SMP/E's CSI, service datasets, and distribution and target libraries. All these datasets are unloaded during the Connector for Top Secret installation jobs.

SMP/E parameters that are environment dependent are modified by the Connector for Top Secret installation process to contain the values specified by the user. When the installation process is complete, the SMP/E CSI and related datasets are customized to reflect the site environment.

Following the customization step, you will find a complete SMP/E environment, containing the Global zone, a target zone and a distribution zone.

The DDDEF names for the target and distribution libraries may be seen in Appendix C: Connector for Top Secret Datasets and JCL Procedures.

Zone Structure

The Connector for Top Secret product environment supplied contains three zones in a single CSI. These zones are:

- **GLOBAL** – The SMP/E global zone
- **CTSATZN** – Connector for Top Secret product target zone
- **CTSADZN** – Connector for Top Secret product distribution zone

All three zones are contained in a single VSAM KSDS cluster, in a single CSI structure. After the CSI has been loaded and customized by the Connector for Top Secret installation process, you will be able to move the loaded zones to other CSIs or rename these CSIs to suit their local standards.

Functions Installed

In the SMP/E environment supplied, the functions installed are:

Function	Description
CACF400 CRCF400 CTSA400 CTSS400	Connector for Top Secret elements
ECA7000 IOA700C IOA700E	Connector for Top Secret Gateway elements (Level 7.0.0)

Maintenance

Fixes supplied for Connector for Top Secret are module or other element replacement fixes and are shipped in PTF format.

Possible Changes in Top Secret Commands Usage Introduced by Maintenance

The Connector for Top Secret performs changes in a Top Secret by issuing Top Secret commands (for example: ADDTO, REPLACE, CREATE).

Often a patch (fix or enhancement) to Connector for Top Secret adds usage of new Top Secret commands, or usage of new operands to the Top Secret commands currently used.

Since Top Secret allows a Top Secret administrator to control who is permitted to issue various Top Secret commands and their operands, it is important for z/OS sites using the Connector for Top Secret to be aware and prepare/adapt

the capabilities of the Top Secret users under which Connector for Top Secret issues the above Top Secret commands.

The Connector for Top Secret issues commands to Top Secret using Managed System administrator. This is the Top Secret user defined to handle operations performed from SailPoint. This administrator is typically SCA ACID.

Running SMP/E Jobs

The Connector for Top Secret installation supplies a JCL procedure (CTSASMP) designed to run SMP/E, and allocate its CSI, auxiliary datasets, and target and distribution libraries. This procedure should be used for all SMP/E operations involving Connector for Top Secret.

> **Note**
> The samples below are only intended to demonstrate the most basic use of each SMP/E command; they are not intended to provide the complete format of each command.

For a full description of SMP/E commands, refer to the appropriate SMP/E documentation, especially the SMP/E User's Guide and SMP/E Reference. Full publication names and IBM Form Numbers are provided in the References section at the end of this appendix.

SMP/E requires the user to define, for each operation, the zone for which the operation will take effect. This definition is done via the SET BOUNDARY SMP/E command. This command must be the first in SMP/E's command stream, and is effective for all subsequent commands up to the next SET command.

Receiving Maintenance

The SMP/E RECEIVE command loads SYSMODs into SMP/E's Global zone. The RECEIVE command does not update any element, and its major purpose is to store the SYSMOD in the Global zone for subsequent processing.

The following is a sample RECEIVE job:

```
//jobname JOB jobparms
//RECEIVE EXEC CTSASMP SET BOUNDARY (GLOBAL).
RECEIVE [SYSMODS] [SELECT(sysmod1,sysmod2,...)].
/*
//SMPPTFIN DD ...
```

The SMPPTFIN DD statement should point to a sequential dataset (or PDS member) holding the SYSMODs to be processed.

The SELECT parameter of the RECEIVE command can be used to limit its scope to the SYSMODs specified in it. Omitting this parameter will make SMP/E receive all the SYSMODs contained in the dataset pointed to by the SMPPTFIN DD statement. Omitting the SYSMODS parameter will cause SMP/E to attempt and receive HOLD information contained in the SMPHOLD dataset.

Member SMPRECIV in the Connector JCL library, contains JCL tailored to RECEIVE PTFs for Connector for Top Secret.

Applying Maintenance

After the maintenance is received, it must be installed into the software product to take effect. The installation is done via the APPLY command. The following is a sample APPLY job:

```
//jobname JOB jobparms
//APPLY EXEC CTSASMP
  SET BOUNDARY(CTSATZN).
  APPLY [SELECT(sysmod1,sysmod2,    )].
/*
//
```

Member SMPAPPLY in Connector JCL library, contains JCL tailored to APPLY PTFs for Connector for Top Secret.

Accepting Maintenance

Following the SYSMODs application into the target zone, and after sufficient time has passed, the SYSMOD should be ACCEPTed into the distribution zone. The following is a sample ACCEPT job:

```
//jobname JOB jobparms
//ACCEPT EXEC CTSASMP
  SET BOUNDARY(CTSADZN).
  ACCEPT [SELECT(sysmod1,sysmod2,    )].
/*
//
```

Member SMPACCPT in Connector JCL library, contains JCL tailored to ACCEPT PTFs for Connector for Top Secret.

Producing SMP/E Reports

It is recommended that you ACCEPT the maintenance after a certain period of time has elapsed since the maintenance was applied. To verify which SYSMODs have been applied and are not ACCEPTed yet, run the following sample job:

```
//jobname JOB jobparms
//REPORT EXEC CTSASMP
  SET BOUNDARY(GLOBAL).
  REPORT SYSMODS INZONE(CTSATZN) COMPAREDTO(CTSADZN).
/*
//
```

Appendix B: Connector for Top Secret Configuration Parameters

This appendix describes configuration parameters used by Connector for Top Secret. Many of these parameters can be modified to suit user requirements.

The Connector for Top Secret configuration parameters are stored in the following members in the Connector PARM library:

- **CTSPUSR** – Parameters in this member are common to all Connector for Top Secret platforms. For example, this member includes the parameter that determines whether or not Transmitted Data Encryption is enabled.
- **RSSPARM** – Parameters in this member are specific to Top Secret. For example, this member includes the parameter that determines the interval between runs of the Offline Interceptor.
- **RSSAPI** – This member contains all Connector for Top Secret calls and the corresponding script-related parameters.
- **CTSPARM** – Parameters in assembler format which require compile in case they are updated.

The parameters in these members are set during the Connector for Top Secret installation and customization process and should not be modified.

This appendix describes the following information:

CTSPUSR – Connector for Top Secret Parameters

The following table describes the CTSPUSR parameters.

Parameter	Description
ENCR_EXT_ACT	Whether Transmitted Data Encryption is enabled (Y/N).
WRITE_TO_QUEUE	Whether the messages of account and group aggregations are written to Queue file. Default: N.

Parameter	Description
	Note MAIN_CS MUST be specified in column 1 of CTSPUSR with WRITE_TO_QUEUE parameter.

RSSPARM – Managed System Parameters

This section contains a description of parameters in the RSSPARM member, followed by a listing of the member as it appears after installing the Connector for Top Secret.

Description of Parameters

Each parameter in the RSSPARM member is applicable either for all MSCSs managed by the Connector for Top Secret installation or for a specific Managed System.

Each parameter in the RSSPARM member has the following syntax:

```
mscs parameterName value
```

where:

- *mscs* – Name of the Managed System to which the parameter applies. If the parameter applies to all MSCSs, contains ALL_RSS.
- *parameterName* – Name of the RSSPARM parameter.
- *value* – Value assigned to the parameter.

> **Note**
> Many of the parameters in the tables are not automatically present in the RSSPARM file after Connector for Top Secret installation. If you wish to assign a value to a specific parameter, it may be necessary to add the parameter to the file. The value labeled as Default appearing in the Values column of the tables that follow indicates the value assigned if the parameter is not present in the member or if the parameter is assigned an invalid value.

Each table contains the following columns:

Column Name	Description
Parameter	Name of the RSSPARM parameter.
	The presence of the symbol * in this column indicates that if the parameter is assigned an invalid value, Connector for Top Secret automatically assigns the parameter the “default” value specified (see the description of the **Values** column below).
	The presence of “(ALL_RSS)” in this column indicates that the parameter is applicable to all MSCSs managed by the Connector for Top Secret installation. If the parameter is specific to a certain type of Managed System, the parameter is applicable to all MSCSs of that type.
Description	Description of the parameter.
Values	Possible parameter values, or limitations. Where specified, the **Default** value in this column indicates the value assigned if the parameter is not present in the RSSPARM member or if the parameter is assigned an invalid value.

General Parameters

The following table lists descriptions of RSSPARM parameters which are specified once for all MSCSs managed by the Connector for Top Secret installation. Each parameter name is preceded by "ALL_RSS".

Parameter	Description	Value
CHECK_SYNC_OBJS	During aggregation – Number of entity/connection operations handled by Connector for Top Secret, after which an “active” confirmation message is sent to SailPoint.	Default – 100
INTERCEPT_SEND_MAX	When INTERCEPT_SEND_MAX is set to a positive numeric value, Connector for Top Secret waits for acknowledgment after the amount of events specified are sent. For example, when INTERCEPT_SEND_MAX is set to 10, Connector for Top Secret sends 10 event messages to IdentityIQ before waiting for an acknowledgment message.	**0** – Do not wait (Default) **1 or above** – Number of events which are sent to IdentityIQ before waiting for an acknowledgm ent from IdentityIQ.
OFLI_VERBOSE	Whether the Log Message of the Online and Offline Interceptor is sent to Managed System console.	Y, N Default – N
OCCUPIED_QUEREU_DATA	For future use. Do not change the default value.	Y, N Default – N
STAT_CHKSUM_INTRVL	During aggregation – Number of entity/connection check-	Default – 5000

Parameter	Description	Value
	sums received by Connector for Top Secret from SailPoint, after which a message is sent to the Connector for Top Secret log and an event is sent to SailPoint.	
STATIST_INTRVL	During aggregation – Number of entities or connections received by Connector for Top Secret from SailPoint, after which a message is sent to the Connector for Top Secret log and an event is sent to SailPoint.	Default – 20
STATUS_INTERVAL	When a Managed System is not active, interval at which Connector for Top Secret checks the status of the Managed System. When the Managed System is active, an event is sent to SailPoint.	Format – **hhmmss** Default – **000500** (5 minutes)
STOP_REQ_MSGS	During aggregation – Number of entity/connection operations handled by Connector for Top Secret, after which an "active" confirmation message is sent to SailPoint.	Default – 10
WAIT_LOCK	Wait Lock (seconds)	Default – 60
WAIT_QUEUE	Wait Queue (seconds)	Default – 60

Managed System-Specific Parameters

The following table lists descriptions of parameters which are specified separately for each individual Managed System managed by the Connector for Top Secret installation. The name of the Managed System must appear before the parameter name in the record.

Parameters	Description	Values
ADMIN_CASE_SENS*	Whether the Administrator name is case-sensitive.	Y, N Default – Y
ADMIN_USER_REQ*	Whether a default administrator is used.	Y, N Default – N
DEFAULT_ADMIN	Name of Connector for Top Secret default administrator account, which is used for GET operations. Only applicable when ADMIN_USER_REQ=Y.	
DEFAULT_CD_ADMIN	Name of default administrator for the CD process. If special administrator for CD process is not required, then DEFAULT _ADMIN is used. Only applicable when ADMIN_USER_REQ=Y.	

Parameters	Description	Values
DEFAULT_CS_ADMIN	Name of default administrator for the CS process. If special administrator for CS process is not required, then DEFAULT _ADMIN is used. Only applicable when ADMIN_USER_REQ=Y.	
DEFAULT_OFLI_ADMIN	Default administrator for the Offline Interceptor process. If special administrator for Offline Interceptor process is not required, then DEFAULT _ADMIN is used. Only applicable when ADMIN_USER_REQ=Y.	
DELETE_INTERCEPT_CHECK	When a delete event is detected, whether the Notification server should call a **get** function to check that an object was actually deleted on the Connector for Top Secret platform. If the call determines that the object exists, an update event is sent to IdentityIQ instead of a delete event.	Y, N Default – N
LOG_GET_MSG	Filtering option for messages generated by the synchronization action-Managed System Retrieval Transaction). The messages are recorded in the Transaction Server (CS) log file.	ALL – All Sync messages are written to the CS log file. NONE – No Sync messages are written to the CS log file. Default – ALL
LOG_INTERCEPT_MSG	Types of interception messages to be recorded in the Notification Server (CD) log file.	ALL – All messages are recorded, indicating in each case whether the event was sent to IdentityIQ ACCEPTED – Only Managed System data included in the aggregation (and therefore sent to IdentityIQ) is recorded. IGNORED – Only Managed System data not included in

Parameters	Description	Values
		the aggregation (and therefore not sent to IdentityIQ) is recorded. NONE – No messages are recorded.
MAX_Q_TRY	When the Queue file is full, the Offline Interceptor may retry to write to it according to the number set in MAX_Q_TRY in RSSPARM.	Values – Any number Coded – Forever (retries continue indefinitely.)
OFLI_INTERCEPT	Whether the Offline Interceptor is started automatically by the Notification Server.	Y – The Offline Interceptor is started periodically by the Notification server. N – The Offline Interceptor is not started by the Notification server. You must provide another means of scheduling the Offline Interceptor. Default – Y
OFLI_INTERVAL	Minimum interval between consecutive activations of the Offline Interceptor.	Value in the format **hhmmss** Default – 010000
OFLI_MAX_DELETE_PERCENT	The maximum number (expressed as a percent) of existing objects that you realistically expect would be deleted between consecutive executions of the Offline Interceptor. If the Offline Interceptor determines that the number of deleted objects exceeds this threshold, the Offline Interceptor stops operation and places an error message in its message file. The purpose of this parameter is to detect situations where a large number of objects are temporarily unreachable and therefore appear to have been deleted. If the number of such	Default – 100 (no limit on objects deleted)

Parameters	Description	Values
	"deleted" objects exceeds the threshold, the Offline Interceptor will not send delete events to SailPoint.	
ONLI_EVENT_USER_PWD_ONLY	This parameter controls whether user and group events are intercepted and sent by Online Interceptor. By default it is set to N, meaning that all users, groups, connections and password events are sent to SailPoint. When set to Y, only the password events are sent to SailPoint. In this case the password violation events controlled by SEND_PASS_VIOLATION parameter are not sent to SailPoint.	Y, N
ONLI_MAX_EVENTS	Number of events which may be active in memory, when Queue file is full. Each entry holds 2,560 bytes in memory, depending on the event type (user, group, connection, password) and length of userid, group, password (above 16M line). A queue full situation might occur when Online Interceptor is active and CD component of the connector is down or reads events from the Queue too slowly comparing to the written events by the Online Interceptor. When queue full situation occurs, Online Interceptor continues accepting events from the SMF exit and from the password exits and these events are accumulated in memory, until queue full situation is relieved. The number of events which can be accumulated in memory is determined by this parameter. ONLI_MIN_NOTIFY_EVENT% Percent number of residual place for new events left in memory that below it, CTS4509W message will be sent by Online Interceptor, each time it handles a new event. The 100% is ONLI_MAX_EVENTS which its default is 20,000. When CTS4509W message is sent, Online Interceptor is	Coded – 20000 Coded – 10% Which is 2,000 by default, as ONLI_MAX_EVENTS is 20,000 by default.

Parameters	Description	Values
	able to handle only ONLI_MIN_NOTIFY_EVENT% more events from SMF exit and password exits. If this situation is not relieved fast enough, new events may get lost and not be handled by the Online Interceptor.	
ONLI_MIN_NOTIFY_EVENT%	Percent number of residual place for new events left in memory that below it, CTS4509W message will be sent by Online Interceptor, each time it handles a new event. 100 percent equals the value of ONLI_MAX_EVENTS, which is 20,000 by default. When CTS4509W message is sent, Online Interceptor is able to handle only ONLI_MIN_NOTIFY_EVENT% more events from SMF exit and password exits. If this situation is not relieved fast enough, new events may get lost and not be handled by the Online Interceptor.	Values – 10% of default The default is 2,000 (which is 10% of the default value of ONLI_MAX_EVE NTS).
ONLI_SEMAPHORE	The name of lock obtained during Online Interceptor operation. Valid only for RSSs that support the Online Interceptor.	Y, N Default – N
PASS_PASSWORD	Whether the password is passed to pre/post-scripts in update password transactions.	Y, N Default – N
PASSWORD_EVENT_FILTER	This parameter makes it possible to filter all password_change events and user_update events. Filtering is done based on jobname or prefix of jobname which we want to filter its commands.	To filter specific jobname specify full jobname. To filter multiple jobnames with same prefix, specify jobname prefix with '*' adjacent to it. For example, if you want to filter all commands issued by jobnames starting with ABC, specify: ABC* .
RSS_TYPE	Type of Managed System.	TSS

Parameters	Description	Values
RSS_WORK_DIR	The prefix used to dynamically allocate working datasets.	Must conform to MVS dataset naming conventions.
SCRIPT_DIR	The name of a dataset containing customer scripts.	
SCRIPT_SEP_ENTRY	Separator entry value for list fields passed to scripts.	Default – comma (,)
SCRIPT_SEP_FIELD	Separator field value for list fields passed to the scripts.	Default – semicolon (;)
SEND_PASS_VIOLATION	This parameter controls whether password violation events are intercepted and sent by Online Interceptor. By default, it is set to Y (send the interception). By setting it to N, all these events are filtered out and not sent to IdentityIQ. Events are sent only if the ONLI_EVENT_USER_PWD_ ONLY parameter is not set to Y.	Y, N
SEND_PWD_TO_SCRIPT	Whether the Managed System Administrator password or phrase is sent to scripts.	Y, N Default – N
SEND_RSSPRM_TO_ SCRIPT	Whether to send RSSPARM parameters to scripts.	Y, N Default – Y
SYNC_SEMAPHORE	Name of lock obtained while the Offline Interceptor or aggregation is running (in order to avoid concurrent execution).	
ONLI_EVENT_ CONTAINER	Whether intercepted container events are sent to SailPoint.	Y, N Default – Y
ONLI_EVENT_GROUP	Whether intercepted group events are sent to SailPoint	Y, N Default – Y
ONLI_EVENT_USER	Whether intercepted user events are sent to SailPoint.	Y, N Default – Y
ONLI_EVENT_USER_ PWD_ONLY	(Only relevant when ONLI_EVENT_USER = Y) Type of intercepted user events sent to SailPoint.	Y – Only user password change events are sent. N – All user events are sent. Default – N
USAAPI_LIB_NAME	The name of an optional DD statement in the CS and CD STC procedure. This parameter is used for customized Connector for Top Secret that are called from a non-APF load lib-	

Parameters	Description	Values
	rary.	
VERIFY_PASSWORD_ NUMB ER	The number of trials Online Interceptor performs when the login with a new password fails.	Minimum value – 1 Maximum value – 9 Default value – 4
WAIT_WHEN_Q_EOF	When the Queue file is full, the Offline or the Online Interceptor may retry to write to the Queue file according to the value set in the MAX_Q_TRY parameter. The WAIT_WHEN_Q_EOF parameter sets the number of seconds to wait between retries.	Number of seconds to wait.

The following table contains descriptions of RSSPARM parameters which only appear in Connector for Top Secret.

Parameter	Description	Values
CTSA_ID (ALL_RSS)	An ID that uniquely identifies the Connector for Top Secret installation.	Default – CTSA
HANDLE_ABENDS (ALL_ RSS)	Determines the behavior of Connector for Top Secret function CTSAPITerm if Connector for Top Secret abends. Relevant for custom Connector for Top Secret developed using the SDK for OS/390. The ability to call the CTSAPITerm function if Connector for Top Secret abends is useful when the Connector for Top Secret includes databases and files that should be properly disconnected.	Y – The recovery routines from the function **CTSAPITerm** are invoked. N – The standard SAS-C abend handler is invoked.The SAS-C abend handler only provides DUMP and diagnostic information and does not perform application-specific recovery. Default – N
LOCKED_TYPES	RU_LOCKED is set to 'Y' depending on values of LOCKED_TYPES parameter.	The valid values are A, P, V, X which represent the following suspension events in Top Secret: • **A** – An account is suspended due to administrator action. • **P** – An account is suspended due to password violation when it exceeded the maximum passwords attempts allowed in its

Parameter	Description	Values
		site (PTHRESH value). • **V** – An account is suspended due to access violation when it exceeds the maximum attempts to access a resource which is not allowed in its site (VTHRESH value). • **X** – An account is suspended due to Top Secret installation exit. The default value types are – P, V - Which means by default RU_ LOCKED is set to Y when ACID's LOCKED_TYPE is either P or V. Values are mutually exclusive with SUSPENDED_TYPES's values.
MAX_SCRIPT_NOTIFY (ALL_RSS)	Number of entries in the Script Notify Buffer.	Default – 250
OFLI_STCNAME	Offline Interceptor started task name.	Default – %PROCPREFS%AOFI
ONLI_ACSJBN	Name of the Transaction Server reported to the Online Interceptor.	Default – %PROCPREFS%ACS
ONLI_DETAIL_MSGS	Whether to retrieve detailed messages from the Online Interceptor.	Y, N Default – N
ONLI_PASSWORD_CASE	Case in which the Online Interceptor sends intercepted passwords to SailPoint.	**LOWER** – Send in lowercase. **ASIS** – Send as received (with no translation). **UPPER** – Same as ASIS. Default – LOWER
ONLI_PASSWORD_ FILTER	Whether to send password updates to SailPoint.	SUPPRESS – password updates are not sent to IdentityIQ. FORWARD – password updates are

Parameter	Description	Values
		sent to IdentityIQ. Default – FORWARD
ONLI_MAX_TSS_EVENTS	The CTSACS notifies the Online Interceptor on any command it issued so that the Online Interceptor can ignore them. This parameter defines how many such commands can be accumulated.	Default – 900 **Important** Do not change this value unless instructed by SailPoint Customer Support.
ONLI_EXPIRE_INTERVAL	Time in seconds, a command notification received from the CTSACS would be kept by the Online interceptor, when not matched by an intercepted event.	Default:120 **Important** Do not change this value unless instructed by SailPoint Customer Support.
ONLI_DELAY_INTERVAL	Time interval between CTSACS command notification and actual event occurrence for matching the event and the notification by Online Interceptor.	Default:10 **Important** Do not change this value unless instructed by SailPoint Customer Support.
SEND_PASS_VIOLATION	Whether an interception is sent to SailPoint when a user specifies an incorrect password while logging in to the managed system.	Y, N Default – Y
SUSPEND_TYPES	RU_SUSPENDED is set to 'Y' depending on values of SUSPENDED_TYPES parameter.	The valid values are A, P, V, X which represent the following suspension events in TSS: • A – An account is suspended due to administrator action. • P – An account is suspended due to password violation when

Parameter	Description	Values
		it exceeded the maximum pass-words attempts allowed in its site (PTHRESH value). • V – An account is suspended due to access violation when it exceeds the maximum attempts to access a resource which is not allowed in its site (VTHRESH value). • X – An account is suspended due to TSS installation exit. The default value types are: 'A','X' - Which means by default RU_ SUSPENDED is set to Y when ACID's suspend_type is either A or X. Values are mutually exclusive with LOCKED_TYPES's values.
USERS_TO_FILTER	For Online Interceptor, userid filtering is enabled based on first letter of the userid. Set this parameter to filter users with first let-ters. For example, set as follows to filter all events of all users which their first letter P, Q, or R: `<rssname> USERS_TO_FILTER PQR`	

CTSPARM: Assembler Format Parameters

This CTSPARM member in PARM library includes few parameters in an assembler source format. This means that, once a parameter is updated, the member must be saved and then it must be compiled and linked. This is done with the CTSPARMJ member located in the INSTALL library. So once CTSPARM is updated, CTSPARMJ must be sub-mitted.

Parameter	Description
ENQRNL	When global resource serialization (GRS), (for example, ENQ or DEQ) encounters a

Parameter	Description
	request for a resource with a scope of SYSTEMS, it scans the SYSTEMS exclusion resource name list (RNL) to determine the scope of the requested resource. However, if the request specifies RNL=NO, GRS would not scan the SYSTEMS exclusion RNL and bypass the RNL search. By default, the CTSPARM ENQRNL parameter is set to **Y**. If GRS environment requires that resource requests bypass the RNL search, set ENQRNL to **N**.
QNAME	Unique name used to protect access to the Queue file by Interceptors and Notification Server. The default QNAME is *xxx*ASYNC where *xxx* is the value set for %PROCPREFS% in the DEFPARMS member in the INSTALL library.

RSSAPI: Connector Parameters

Each line in this member (excluding the comment lines) represents a Connector call which is followed by various parameters. The parameters influence the behavior of the Connector.

For a description of the parameters in each line, see "Table 20—RSSAPI Member Parameters".

Pre-scripts and Post-scripts must be stored in the PDS library which was set in parameter SCRIPT_DIR of member RSSPARM. The default value for this parameter is:

```
prefix.version.USER.CLIST
```

where:

- *prefix* – Value set for parameter OLPREFS in member LOADCTS in the INSTALL library
- *version* – Value set for parameter OLVERS in member LOADCTS in the INSTALL library.

Appendix C: Connector for Top Secret Datasets and JCL Procedures

This appendix describes the following information.

Connector for Top Secret Dataset List

This section lists the different datasets used by the Connector for Top Secret.

Connector for Top Secret Installation Datasets

The following datasets are allocated during the Connector for Top Secret installation procedure. In the names of these datasets, **<prefix>** is the value specified for parameter **ILPREFS**, and **<version>** is the value specified for parameter **ILVERS**. Both parameters are located in LOADCTS member in the Connector INSTALL library. These parameters are set in procedure "4.1 – Tailor member LOADCTS" in 4 – Allocate and Load Connector for Top Secret Installation Libraries using values set in Connector for Top Secret Datasets Allocation Parameters.

Dataset	Description
<prefix>.<version>.CLIST	REXX library
<prefix>.<version>.CTRANS	SAS/C Runtime Load library
<prefix>.<version>.CMSG	Message text library
<prefix>.<version>.INSTALL	Installation library
<prefix>.<version>.ISMSGENG	ISPF English messages of CTSGATE ISMSGENG
<prefix>.<version>.JCL	Sample jobs library
<prefix>.<version>.LOAD	Load library
<prefix>.<version>.LOADE	SSL load modules LOADE
<prefix>.<version>.MAC	Macros library
<prefix>.<version>.MSG	Message text library
<prefix>.<version>.MSGENG	English messages of CTSGATE MSGENG
<prefix>.<version>.PARM	Parameters library
<prefix>.<version>.PANELENG	English panels of CTSGATE PANELENG
<prefix>.<version>.PROCLIB	JCL procedures library
<prefix>.<version>.SAMPLE	Sample REXX and REXX library

Dataset	Description
<prefix>.<version>.SECSRC	Sample security exits for CTSGATE (not used)
<prefix>.<version>.UPGRADE	UPGRADE jobs

Operation Datasets

The following datasets are allocated during the Connector for Top Secret installation procedure. In the names of these datasets, **<prefix>** is the value specified for parameter **OLPREFS**, and **<version>** is the value specified for parameter **OLVERS**. Both parameters are located in member LOADCTS in the Connector INSTALL library. These parameters are set in procedure "4.1 – Tailor member LOADCTS" in 4 – Allocate and Load Connector for Top Secret Installation Libraries using values set in Connector for Top Secret Datasets Allocation Parameters.

Dataset	Description
<prefix>.<version>.CARECNN	Managed user to group connections
<prefix>.<version>.CAREGRP	Managed groups
<prefix>.<version>.CAREOE	Managed organization elements
<prefix>.<version>.CAREUSR	Managed users
<prefix>.<version>.DIAGLVL	Diagnostics setup
<prefix>.<version>.ENCREXT	Transmitted Data Encryption
<prefix>.<version>.ENCRINT	Stored Data Encryption
<prefix>.<version>.QUEUE	Interception Queue dataset
<prefix>.<version>.RSSKWDS	Managed System specific keywords table
<prefix>.<version>.RSSOFLI	Offline interceptor table
<prefix>.<version>.USER.CLIST	Scripts dataset
<prefix>.<version>.TSSCACHE	Cache file for connections

SMP/E Distribution Datasets

The following datasets are allocated during the Connector for Top Secret installation procedure. In the names of these datasets, **<prefix>** is the value specified for parameter **SPDPREF**, and **<version>** is the value specified for parameter **SPDVER**. Both parameters are located in member LOADCTS in the Connector INSTALL library. These parameters are set in procedure "4.1 – Tailor member LOADCTS" in 4 – Allocate and Load Connector for Top Secret Installation Libraries using values set in Connector for Top Secret Datasets Allocation Parameters.

Dataset	Description
<prefix>.<version>.ACMSG	CMSG Dlib
<prefix>.<version>.ACLIST	REXX Dlib ACLIST

Dataset	Description
<prefix>.<version>.AINSTALL	INSTALL Dlib
<prefix>.<version>.AISMSGEN	ISMSGENG Dlib AISMSGEN
<prefix>.<version>.AIOALOAD	GATEWAY LOAD Dlib
<prefix>.<version>.AJCL	JCL Dlib
<prefix>.<version>.ALOADE	SSL Load modules Dlib ALOADE
<prefix>.<version>.AMAC	Macro Dlib
<prefix>.<version>.AMSG	MSG Dlib
<prefix>.<version>.AMSGENG	English messages of CTSGATE Dlib AMSGENG
<prefix>.<version>.APARM	PARM Dlib
<prefix>.<version>.APROCLIB	PROCLIB Dlib
<prefix>.<version>.APANELEN	PANELENG Dlib APANELEN
<prefix>.<version>.ASAMPLE	SAMPLE Dlib
<prefix>.<version>.ASECSRC	SECSRC Dlib ASECSRC
<prefix>.<version>.AUPGRADE	UPGRADE Dlib AUPGRADE

SMP/E Datasets

The following datasets are allocated during the Connector for Top Secret installation procedure. In the names of these datasets, **<prefix>** is the value specified for parameter **SPAPREF**, and **<version>** is the value specified for parameter **SPAVER**. Both parameters are located in member LOADCTS in the Connector INSTALL library. These parameters are set in procedure "4.1 – Tailor member LOADCTS" in 4 – Allocate and Load Connector for Top Secret Installation Libraries using values set in Connector for Top Secret Datasets Allocation Parameters.

Dataset	Description
<prefix>.<version>.SMPLOG	SMP/E Work dataset
<prefix>.<version>.SMPLOGA	SMP/E Work dataset
<prefix>.<version>.SMPLTS	SMP/E Work dataset
<prefix>.<version>.SMPMTS	SMP/E Work dataset
<prefix>.<version>.SMPPTS	SMP/E Work dataset
<prefix>.<version>.SMPSCDS	SMP/E Work dataset
<prefix>.<version>.SMPSTS	SMP/E Work dataset

SMP/E CSI

The following dataset is allocated during the Connector for Top Secret installation procedure. In the names of this dataset, **<prefix>** is the value specified for parameter **SPCPREF**, and **<version>** is the value specified for parameter

SPCVER. Both parameters are located in member LOADCTS in the Connector INSTALL library. These parameters are set in procedure "4.1 – Tailor member LOADCTS" in 4 – Allocate and Load Connector for Top Secret Installation Libraries using values set in Connector for Top Secret Datasets Allocation Parameters.

Dataset	Description
<prefix>.<version>.CSI	SMP/E CSI

Connector for Top Secret JCL Procedures

The following JCL procedures are copied to your system PROCLIB library during the Connector for Top Secret installation procedure.

JCL Procedure	Description
CTSAADPT	Connector for Top Secret Managed System rename utility
CTSACD	Connector Notification Server (CD)
CTSACS	Connector Transaction Server (CS)
CTSADFR	Connector Offline Interceptor file formatting utility
CTSALERT	Connector for Top Secret alert data utility
CTSAONI	Top Secret Online Interceptor
CTSAQCR	Connector for Top Secret copy and format Queue dataset utility
CTSAQFR	Connector for Top Secret Queue formatting utility
CTSAQPR	Connector for Top Secret Queue printing utility
CTSASMP	Connector for Top Secret SMP/E procedure
CTSCDIAG	Connector for Top Secret diagnostics file formatting utility
CTSDIAGI	Connector for Top Secret diagnostic initialization utility
CTSGATE	Connector for Top Secret Gateway
CTSKGEN	Connector for Top Secret Stored Data Encryption utility.
CTSUPMSG	Connector for Top Secret message update table
CTSOFLI	Standard Offline Interceptor
CTSACDJ	STCJOB of Connector Notification Server (CD)
CTSACSJ	STCJOB of Connector Transaction Server (CS)
CTSAONIJ	STCJOB of Top Secret Online Interceptor
CTSGATEJ	STCJOB of Top Secret Gateway Connector Monitor
CTSOFLIJ	STCJOB of Standard Offline Interceptor

Appendix D: Copying a Connector for Top Secret Installation

This appendix describes the procedure to copy an existing installation of Connector for Top Secret from one MVS system to another.

The following terms are used throughout this appendix:

- **source system** – MVS system where Connector for Top Secret was originally installed.
- **target system** – MVS system to which Connector for Top Secret installation is copied.

This appendix describes the following information.

Installation Copy Procedure

Use the procedure that follows to copy an existing installation of Connector for Top Secret from one MVS system to another.

> **Note**
> If LOCALCOPY value is set for the %PROCLIB% parameter in the DEFPARMS member in the Connector INSTALL library, the same procedures will be used by the source and new Connector for Top Secret environments. If the Managed System name has to be changed (see 3 – Adjust Connector for Top Secret Parameters below), contact SailPoint Customer Support for instructions.

1 – Copy Connector for Top Secret JCL Procedures

During installation of Connector for Top Secret, various JCL procedures were copied to the system procedures library. The JCL procedures are used by Connector for Top Secret started tasks and by jobs activating the Connector for Top Secret utilities.

These JCL procedures must be copied to the target system.

The procedures may be copied manually or using the instructions specified in Procedure 5 – Tailor Connector for Top Secret Members with Site Parameters.

1A – Connector for Top Secret Started Tasks

The following JCL procedures are Connector for Top Secret started tasks and must be copied to the target system JCL procedures library:

- CTSACD
- CTSACS
- CTSGATE
- CTSAONI
- CTSOFLI

When STCJOBs are used:

1. The corresponding STCJOB members must be copied to the target system STCJOBS library.
2. If LOCALCOPY value is set for the %PROCLIB% parameter in DEFPARMS member in the Connector INSTALL library, make sure the target system has access to the Connector for Top Secret PROCLIB library.

1B – Connector for Top Secret Utilities

The following JCL procedures are used by Connector for Top Secret utilities and maintenance jobs. They are not required for the daily operation of Connector for Top Secret. However, they are required for customization and installation operations performed in subsequent steps described in this procedure.

- CTSADFR
- CTSDIAG
- CTSAQFR
- CTSDIAGI
- CTSAQCR
- CTSAQPR
- CTSKGEN
- CTSALERT
- CTSUPMSG
- CTSAADPT

2 – Copy Connector for Top Secret Datasets

The following parameters are used in this step:

- `<i_prefix>` – Value set for parameter ILPREFS in member LOADCTS in the Connector INSTALL library.

  Default: CTSA

- `<i_version>` – Value set for parameter ILVERS in member LOADCTS in the Connector INSTALL library.

  Default: V400

- `<o_prefix>` – Value set for parameter OLPREFS in member LOADCTS in the Connector INSTALL library.

  Default: CTSA

- `<o_version>` – Value set for parameter OLVERS in member LOADCTS in the Connector INSTALL library.

  Default: V400

For more information regarding parameters in member LOADCTS, see step “4.1 – Tailor Member LOADCTS” in 4 – Allocate and Load Connector for Top Secret Installation Libraries.

2A – Operations Datasets

The following datasets are used during the operation of Connector for Top Secret and must be copied from the source system to the target system:

```
<i_prefix>.<i_version>.CLIST
<i_prefix>.<i_version>.CMSG
<i_prefix>.<i_version>.LOAD
<i_prefix>.<i_version>.MSG
<i_prefix>.<i_version>.PARM
<i-prefix>.<i-version>.CTRANS
```

2B – Installation Datasets

The following datasets are not used by Connector for Top Secret started tasks and utilities but are required for customization and installation operations performed later in this procedure:

```
<i_prefix>.<i_version>.INSTALL
<i_prefix>.<i_version>.MAC
<i_prefix>.<i_version>.JCL
<i_prefix>.<i_version>.SAMPLE
<o_prefix>.<o_version>.USER.CLIST
<i_prefix>.<i_version>.PROCLIB
```

3 – Adjust Connector for Top Secret Parameters

Perform the customization steps described below on the target system.

> **Note**
>
> As an alternative to steps 3A and 3B below, you can use utility CTSAADPT to rename the Managed System on the target system.
>
> For more information, see Renaming a Managed System.

3A – RSSPARM Parameters

Edit member RSSPARM member in the Connector PARM library.

The RSSPARM member contains the Managed System parameters for the installed Managed System. Each line is in the format:

```
<Managed System-name> <parameter_name> <parameter_value>
```

where `<Managed System-name>` is either ALL_RSS or the Managed System name specified during installation.

If the Managed System name defined in SailPoint for the target system is different from the one specified in the member RSSPARM, you must update member RSSPARM to reflect that difference.

To perform the required update in member RSSPARM, change the <Managed System-name> value on all lines that specify the source system Managed System name to the new Managed System name that is correct for the target system.

In addition, the Managed System name qualifier in the value specified for parameter RSS_WORK_DIR should be modified.

> **Note**
> The Managed System name specified in the RSSPARM parameters file must match the name defined in SailPoint for the target system. If the names do not match, SailPoint will not be able to connect to Connector for Top Secret.

3B – RSSAPI Parameters

Edit member RSSAPI member in Connector PARM library.

RSSAPI member contains the script activation definitions for the Managed System. Each line is in the format:

```
RSS_type RSS_name additional_parameters
```

where:

- `RSS_type` – hyphen (–)
- `RSS_name` – Managed System name specified during installation. Default: MVSTSS

The complete syntax of this member is described under “Structure of the RSSAPI member” on page 73.

If the Managed System name defined in SailPoint for the target system is different from the one specified in member RSSAPI, you must update member RSSAPI to reflect that difference.

To perform the required update in member RSSAPI, change the **RSS_name** parameter on all the lines containing the source system Managed System name to the new Managed System name that is correct for the target system.

3C – Adjust Procedures

If the Managed System name was changed, modify the “Managed System=” parameter in each of the following procedures:

- CTSAADPT
- CTSADFR
- CTSOFLI
- CTSAONI

3D – Allocate Connector for Top Secret Work Datasets

The datasets listed below are created during Connector for Top Secret installation. These datasets should *not* be copied from the source system to the target system. Instead, they should be allocated and formatted directly on the target system.

Originally, allocation of these datasets was performed by job FORMCTS, which is run in Procedure 8 – Format Connector for Top Secret Datasets.

You may run job FORMCTS in the Connector INSTALL Library to allocate the datasets on the target system.

```
<o_prefix>.<o_version>.CARECNN
<o_prefix>.<o_version>.CAREGRP
<o_prefix>.<o_version>.CAREUSR
<o_prefix>.<o_version>.CAREOE
<o_prefix>.<o_version>.DIAGLVL
<o_prefix>.<o_version>.QUEUE
<o_prefix>.<o_version>.RSSKWDS
<o_prefix>.<o_version>.RSSOFLI
```

3E – Encryption Datasets

The following datasets are used during the operation of Connector for Top Secret and must be copied from the source system to the target system:

```
<o_prefix>.<o_version>.ENCREXT
```

```
<o_prefix>.<o_version>.ENCRINT
```

These datasets are allocated in step 3D – Allocate Connector for Top Secret Work Datasets and should be over-written in this step.

3F – Adjust Offline Interceptor Files

Edit member CTSOFLI in the Connector for Top Secret JCL library. The 3rd qualifier in the datasets referred to by the member should be modified to match the new managed system name.

4 – Adjust MVS System Parameters

APF Authorized Library

The Connector LOAD and CTRANS libraries must be defined as APF authorized libraries in the target system. (for more information, see Procedure 9 – Customize Communication Settings).

5 – Customize Top Secret Support

The customization process required for completing Connector for Top Secret installation is described in the Installation chapter.

The process includes the installation of the SMF IEFU83 exit that is required for interception of Top Secret database security administration events.

To install support for this capability for the target system, perform the necessary instructions as described in the above chapter.

Appendix E: Managed System-Specific fields

This appendix provides reference tables for Managed System-specific fields.

This appendix describes the following information:

Description of Table Column Titles

Due to the many columns of information contained in the tables in this appendix, abbreviated column names are used. This section describes the meaning of the column titles for the Managed System-specific field tables later in the appendix.

The columns described in the following table appear in all the Managed System-specific Field tables in this appendix.

Column Title	Description
Field	Field name (as it appears in the Details window of SailPoint). For list fields, the sub-fields are indented. By default, field labels are displayed in a Details window. To view field names, click the right mouse button anywhere in the Details window in the SailPoint (except on a field) and choose the option **Show Field Names** from the pop-up menu. The field names are displayed instead of the field labels.
L	Whether or not the field accepts a list of values. A list consists of values separated by commas. Possible values in this column are: • **L** – Identifies a list field. • **S** – Identifies a subfield of a list field (names of subfields are indented in the **Field** column).
T	Type of input accepted in the field. Possible values in this column are: • **C** – Character. All input is treated as characters even if all are digits. • **F** – Flag. Input must be Y or N. • **N** – Integer. Input must be numeric.

Column Title	Description
	• **T** – Time. Input must be in the time format specified in the column Restrictions • **D** – Date/Time. Input must be in the format specified in the column Restrictions. This format generally requires that the value be specified as a string consisting of the date or date/time. • **S** – Selection from a list of predefined values.
Len	Maximum number of characters in a character field. This field length only applies if the type (column **T**) is **C**. (Field length limitations for other data types are determined by information in columns **L** and **Restrictions**.)
Restrictions	Validation restrictions such as numeric ranges or list of possible values. Underlined values (if any) are the default values.

The column titles for each type of entity differ slightly. The following table describes the meaning of the single-letter column titles used to indicate the type of function for which each Managed System-specific field is relevant.

An **X** appearing in a column for a given field indicates that the field is relevant to that function.

Function Type	Column Title	Description
Account	A	Add user
	U	Update user
	G	Get user
	D	Delete user
	R	Revoke/restore user
	P	Update password or phrase
Group	A	Add group
	U	Update group
	G	Get group
	D	Delete group
User–Group Connection	C	Connect user to group
	U	Update user to group connection
	G	Get user to group connection
	D	Disconnect user from group

Managed System User Fields

The following table lists managed system user fields.

Field	L	T	Len	Restrictions	M	A	U	G	R	D
USER_TYPE		S	8	MSCA | ZCA | VCA |DCA | SCA | LSCA | USER	X	X	X	X		
NAME		C	32		X	X	X	X		
EXPIRES		D		YYYYMMDD		X	X	X		
groups	L	C	16K					X		
RESUMES		D		YYYYMMDD		X	X	X		
INFO.ACID_SIZE		N	4					X		
INFO.CREATED_DATE		D		YYYYMMDD				X		
INFO.SUSPEND_TYPE		S		ASUPSEND | SUSPEND				X		
FACILITY	L					X	X	X		
– NAME	S		8			X	X	X		
– ACTION	S	C	25	AUDIT, NOTIFY, DENY		X	X	X		
– EXPIRES	S	D		YYYY/MM/DD		X	X			
– DAYS	S	S	30	MON, TUE, WED, THU, FRI, SAT, SUN, WEEKDAYS, WEEKENDS, ALL		X	X	X		
– TIMES	S			HH,HH | ANY		X	X	X		
MASTFAC		C	8			X	X	X		
USER	L					X	X	X		
– CLASS	S	C	8			X	X	X		
– RESOURCE	S	C	256			X	X	X		
INTERVAL		N	3			X	X	X		
INFO.PASSWOR-D_EXP_DATE		D		YYYYMMDD				X		
DEFNODES	L	C	2303			X	X	X		
INSTDATA		C	255			X	X	X		
AUDIT		F	1			X	X	X		

Field	L	T	Len	Restrictions	M	A	U	G	R	D
CONSOLE		F	1			X	X	X		
NOSUSPEND		F	1			X	X	X		
NOADSP		F	1			X	X	X		
NOATS		F	1			X	X	X		
NODSNCHK		F	1			X	X	X		
NOLCFCHK		F	1			X	X	X		
NOPWCHG		F	1			X	X	X		
NORESCHK		F	1			X	X	X		
NOSUBCHK		F	1			X	X	X		
NOVMDCHK		F	1			X	X	X		
NOVOLCHK		F	1			X	X	X		
DUFUPD		F	1			X	X	X		
DUFXTR		F	1			X	X	X		
MULTIPW		F	1			X	X	X		
TRACE		F	1			X	X	X		
PHYSKEY		C	256			X	X	X		
TZONE		N	3			X	X	X		
SOURCE	L	C	45			X	X	X		
LTIME	L					X	X	X		
– TIME	S	N	3			X	X	X		
– FACILITY	S	C	8			X	X	X		
COMMAND	L					X	X	X		
– FACILITY	S	C	8			X	X	X		
– NAME	S	C	11			X	X	X		
XCOMMAND	L					X	X	X		
– FACILITY	S	C	8			X	X	X		
– NAME	S	C	11			X	X	X		
OIDCARD		F								
OPCLASS	L	N	256	1-24		X	X	X		
OPIDENT		C	3			X	X	X		
OPPRTY		N	3			X	X	X		
SCTYKEY		N	256	1-255		X	X	X		
SITRAN	L					X	X	X		

Field	L	T	Len	Restrictions	M	A	U	G	R	D
TRANSACTION		C	8			X	X	X		
FACILITY		C	8			X	X	X		
SMSAPPL		C	8			X	X	X		
SMSDATA		C	8			X	X	X		
SMSMGMT		C	8			X	X	X		
SMSSTOR		C	8			X	X	X		
TSOCOMMAND		C	80			X	X	X		
TSODEFPRFG		N	3			X	X	X		
TSODEST		C	8			X	X	X		
TSOHCLASS		C	1			X	X	X		
TSOJCLASS		C	1			X	X	X		
TSOLACCT		C	40			X	X	X		
TSOLPROC		C	8			X	X	X		
TSOLSIZE		N	7			X	X	X		
TSOMCLASS		C	1			X	X	X		
TSOMPW		F	1			X	X	X		
TSOMSIZE		N	7			X	X	X		
TSOOPT		S	50	MAIL | NOMAIL | NOTICES | NONOTICES | OIDCARD | NOOIDCARD		X	X	X		
TSOSCLASS		C	1			X	X	X		
TSOUDATA		C	4			X	X	X		
TSOUNIT		C	8			X	X	X		
UID		N	6			X	X	X		
DFLTGRP		C	8			X	X	X		
HOME		C	1024			X	X	X		
OMVSPGM		C	1024			X	X	X		
MCSALTG		C	8			X	X	X		
MCSAUTH		S		INFO | MASTER | SYS | IO | CONS | ALL		X	X	X		

Field	L	T	Len	Restrictions	M	A	U	G	R	D
MCSAUTO		S		YES | NO		X	X	X		
MCSCMDS		C	8			X	X	X		
MCSDOM		S		NORMAL |ALL |NONE		X	X	X		
MCSKEY		C	8			X	X	X		
MCSLEVL		S		ALL | NB | R | I | CE | E | IN		X	X	X		
MCSLOGC		S		YES | NO		X	X	X		
MCSMFRM		S		M | J | S | T | X		X	X	X		
MCSMGID		S		YES | NO		X	X	X		
MCSMON		S		JOBNAMES | JOBNAMES-T | SESS | SESS-T | STATUS		X	X	X		
MCSROUT		S		NONE | ALL | list of rout codes (1-128)		X	X	X		
MCSSTOR		N	4			X	X	X		
MCSUD		S		YES | NO		X	X	X		
PCADMIN		C	8			X	X	X		
PCDSDAYS		N	2			X	X	X		
PCIDLE		N	2			X	X	X		
PCLGTYPE		S	2	1 | 2 | 3 | 4		X	X	X		
PCMINPWD		N	1			X	X	X		
PCOPTS	L	S	10	CONNECT | NOCONNECT | SGNOFFMSG | NOSGNOFFMS-G | PRIVPLUS		X	X	X		
WAACCNT		C	255			X	X	X		
WABLDG		C	60			X	X	X		
WADEPT		C	60			X	X	X		
WAADDR1		C	60			X	X	X		
WAADDR2		C	60			X	X	X		

Field	L	T	Len	Restrictions	M	A	U	G	R	D
WAADDR3		C	60			X	X	X		
WAADDR4		C	60			X	X	X		
WANAME		C	60			X	X	X		
WAROOM		C	60			X	X	X		
IMSMSC		C	255			X	X	X		
LANGUAGE		C	1			X	X	X		
ADMIN.ACID	L					X	X	X		
– AUTH	S	S	256	XAUTH | AUDIT | CREATE | INFO | DEFNODES | REPORT | MAINTAIN | ALL		X	X	X		
ADMIN.DATA	L					X	X	X		
– AUTH	S	S	256	BASIC | RESOURCE | XAUTH | LCF | SOURCE | INSTDATA | CICS | PROFILE | ADMIN | NAMES | ACID | WORKATTR | SESSKEY | PASSWORD | PWVIEW | ALL		X	X	X		
ADMIN.MISC1	L					X	X	X		
– AUTH	S	S	256	LCF | INSTDATA | USER | LTIME | RDT | SUSPEND | NOATS | TSSSIM | ALL		X	X	X		
ADMIN.MISC2	L					X	X	X		

Field	L	T	Len	Restrictions	M	A	U	G	R	D
– AUTH	S	S	256	SMS | TSO | NDT | DLF | TARGET | PC* | WORKATTR* |APPCLU* | ALL		X	X	X		
ADMIN.MISC8	L					X	X	X		
– AUTH	S	S	256	LISTRDT | LISTSTC | MCS | REMASUSP | ALL		X	X	X		
ADMIN.MISC9	L					X	X	X		
– AUTH	S	S	256	BYPASS | TRACE | CONSOLE | STC | MASTFAC | MODE | GLOBAL | GENERIC | ALL		X	X	X		
ADMIN.RESOUR-CE	L					X	X	X		
– AUTH	S	S	256	OWN | XAUTH | AUDIT | INFO | REPORT | ALL		X	X	X		
– ACCESS	S	S	256	ALL | BLP | BROWSE | CONTROL | CREATE | DELETE | FETCH | FEOV | FIND | LOAD | MULTI | MREAD | MWRITE | NONE | PURGE | READ | REPLACE | SCRATCH |		X	X	X		

Field	L	T	Len	Restrictions	M	A	U	G	R	D
				UPDATE | WRITE						
ADMIN.RESCLAS-S	L					X	X	X		
– RESCLASS	S	C	8			X	X	X		
– AUTH	S	S	256	OWN | XAUTH | AUDIT | INFO | REPORT | ALL		X	X	X		
– ACCESS[1]	S	S	256	ALL | BLP | BROWSE | CONTROL | CREATE | DELETE | FETCH | FEOV | FIND | LOAD | MULTI | MREAD | MWRITE | NONE | PURGE | READ | REPLACE | SCRATCH | UPDATE | WRITE		X	X	X		
ADMIN.FACILITY	L	C	256			X	X	X		
ADMIN.SCOPE	L					X	X	X		
– ACID	S	C				X	X	X		
UUID		C	36			X	X	X		
DCENAME		C	1024			X	X	X		
HOMECELL		C	60			X	X	X		
HOMEUUID		C	36			X	X	X		
DCEFLAGS		C	10	AUTOLOGIN		X	X	X		
DCEKEY		C	20			X	X	X		
NETVCONS		C	8			X	X	X		
NETVCTL	L	S	255	GENERAL | GLOBAL |		X	X	X		

Field	L	T	Len	Restrictions	M	A	U	G	R	D
				SPECIFIC						
NETVIC		C	255			X	X	X		
NETVDMNS	L	C	255			X	X	X		
– Domain Name	S	C	8			X	X	X		
NETVMSGR		C	3	YES,NO		X	X	X		
NETVNGMF		C	3	YES,NO		X	X	X		
NETVOPCL	L	S	255			X	X	X		
– Option	S	N	4	1-2040		X	X	X		
targetPermissions[2]		C	16K					X		

1. Not all ACCESS subfields are applicable for any RESCLASS value. Consult the *Top Secret Command Functions Guide* for the available ACCESS keywords for each specific RESCLASS.
2. All permissions are included within targetPermissions attribute.

   Full description of targetPermissions content appears in Resource ACL Data Translation Tables.

Group Fields

The following table lists group fields.

Field	L	T	Len	Restrictions	M	A	U	G	D
NAME		C	32		X	X	X	X	
UG_TYPE	L	C	8	PROFILE | GROUP	X	X		X	
FACILITY	L					X	X	X	
– NAME	S	C	8			X	X	X	
– ACTION	S	C	50			X	X	X	
– EXPIRES	S	D		YYYYMMDD		X	X		
– DAYS	S	S	8	MON | TUE | WED | THU | FRI | SAT | SUN | WEEKDAYS | WEEKENDS | ALL		X	X	X	

Field	L	T	Len	Restrictions	M	A	U	G	D
– TIMES	S			HH | HH | ANY		X	X	X	
GAP		F	1			X	X	X	
AUDIT		F	1			X	X	X	
NOADSP		F	1			X	X	X	
NOATS		F	1			X	X	X	
NODSNCHK		F	1			X	X	X	
NOLCFCHK		F	1			X	X	X	
NOPWCHG		F	1			X	X	X	
NORESCHK		F	1			X	X	X	
NOSUBCHK		F	1			X	X	X	
NOVMDCHK		F	1			X	X	X	
NOVOLCHK		F	1			X	X	X	
DUFUPD		F	1			X	X	X	
DUFXTR		F	1			X	X	X	
TRACE		F	1			X	X	X	
TZONE		N	3			X	X	X	
SOURCE	L	C	45			X	X	X	
LTIME	L					X	X	X	
– TIME	S	N	3			X	X	X	
– FACILITY	S	C	8			X	X	X	
OIDCARD		F	1						
OPCLASS	L	N	256	1-24		X	X	X	
OPIDENT		C	3			X	X	X	
OPPRTY		N	3			X	X	X	
SCTYKEY	L	N	256	1-24		X	X	X	
SITRAN	L					X	X	X	
TRANSACTION		C	8			X	X	X	
FACILITY		C	8			X	X	X	
GID		N	6			X	X	X	
LANGUAGE		C	1			X	X	X	
COMMAND	L	C				X	X	X	
– FACILITY	S	C	8			X	X	X	
– NAME	S		11			X	X	X	

Field	L	T	Len	Restrictions	M	A	U	G	D
XCOMMAND	L	C				X	X	X	
– FACILITY	S	C	8			X	X	X	
– NAME	S	C	11			X	X	X	
targetPermissions[1]		C	16K					X	

1. All permissions are included within targetPermissions attribute.

   Full description of targetPermissions content appears in Resource ACL Data Translation Tables.

Account–Group Fields

The following table lists account-group fields.

Field	L	T	Len	Restrictions	M	C	U	G
EXPIRES		D		YYYYMMDD		X	X	X
RELATIVE_POS	L	S	8	AFTER | BEFORE | TOP | BOTTOM		X	X	X
RELATIVE_ PROFILE		C	8			X	X	X
POS		N	3					X

Appendix F: Connector for Top Secret Batch Utility

The batch utility enables a user to execute a batch job containing provisioning or list requests directly on the Top Secret Connector without the requirement of a partner (such as SailPoint). The provisioning or list requests are processed by the requested Managed System Interface together with the requested security product, Top Secret.

The batch utility would be helpful in the following scenarios:

- When a new installation is performed and a connection has not yet been established with SailPoint.

  Provisioning or list transactions can be issued locally within the Mainframe using the batch utility to ensure that the Top Secret Connector is installed and working properly.

- For testing or debugging purposes as instructed by SailPoint Support.

- When multiple provisioning transactions are required to be performed quickly and easily from the Mainframe than from SailPoint.

Whenever provisioning transactions are issued by the batch utility and if the Online Interceptor is not active, it is required to issue full aggregation in order to update SailPoint with the changes done by these provisioning transactions.

This appendix describes the syntax rules, provisioning and list requests, and invocation JCL required to execute the utility and an example of utility control statements with the sample job output.

This appendix describes the following information.

Security Requirements

Before running the utility, ensure that the following security requirements are met:

- When invoking the batch utility with provisioning requests, the user specified as ADMIN_UNAME must have sufficient authority in order to execute the requests.

- If **STCJOBS** are used for the product started tasks, the file allocated for EXECOUT DD statement is a permanent file.

- The user submitting the job must have ALTER authority to this file.
- The user specified as ADMIN_UNAME must have UPDATE authority to this file. The file name is:

  ```
  <olprefs>.<olvers>.EXECOUT.<procprefs>BATCH
  ```

  where *olprefs*, *olvers* and procprefs are the values set in 1 – Set the Parameter Values.

Input Control Statements Syntax Rules

The following rules must be followed for the utility control statements:

- Must contain environment and provisioning / list request lines
- Data must be written in columns 1 to 72.
- The file must not have sequence numbers, that is, number mode off or unnum.
- A comment line must begin with an asterisk in column 1. The Input Control Statements can have many comment lines. Comments must not be inserted in a non-comment line.
- The input for the utility must begin with environment lines as defined in Environment Definition Syntax.
- The input for the utility may contain multiple provisioning / list requests. Each request begins with a request line which defines the request and is followed by one or more parameter lines which provides request details.
- The input control statements – keywords and values – would not be converted to upper or lowercase. Hence all data must be specified in the correct case.
- If a value is longer than 1 line, it must be surrounded by parentheses and continued in column 1 of the next line. For example,

  ```
  Keyword=(value1,value2,value3,value4,value5,value6,
  value7,value8,value9,value 10,value11,value12,value13)
  ```

Environment Definition Syntax

The utility input must begin with environment definition lines.

```
:ENV
```

Parameter	Mandatory/Optional	Description
RSS_TYPE	Mandatory	TSS
RSS_NAME	Mandatory	The name defined for RSSNAME in the DEFPARMS member

Parameter	Mandatory/Optional	Description
ADMIN_UNAME	Mandatory	The administrator User ID
ADMIN_GROUP	Optional	The group to be used for the administrator. **Note** Used to override the default group used during logon processing.

Batch Provisioning and List Requests

This section describes the following provisioning and list requests:

- **ADDUSER** – Add a User
- **UPDUSER** – Update a User
- **DELUSER** – Delete a User
- **DISABLEUSER** – Disable a User
- **ENABLEUSER** – Enable a User
- **LISTUSER** – List User Information
- **CHGPWD** – Change a User password
- **ADDGROUP** – Add a Group
- **UPDGROUP** – Update a Group
- **DELGROUP** – Delete a Group
- **LISTGROUP** – List Group Information
- **ADDCONN** – Add a User-Group Connection
- **UPDCONN** – Update a User-Group Connection
- **DELCONN** – Delete a User-Group Connection

- **LISTCONN** – List User-Group Connection information
- **LISTACL** – List ACL permission information

ADDUSER – Add a User

Use the following request and parameter lines to add a user.

```
:ADDUSER=userid
```

Parameter	Mandatory/Optional	Description/Value
USER.DFLTGRP	Optional	Default group name. `USER.DFLTGRP=default-group-name`
USER.PASSWORD	Optional	User's password `USER.PASSWORD=[()]password[,PERM / TEMP)]`
USER.DISABLE	Optional	`USER.DISABLE=YES`
USER.AUTH	Optional	`USER.AUTH=REG / ADMIN / AUDIT / ADMINAUDIT`
USER.PARENT_ CONTAINER	Mandatory	Parent container name. For example, `USER.PARENT_CONTAINER=parent-container-na me`
USER_TYPE	Mandatory	USER_TYPE=USER/DCA/SCA/VCA/ZCA/LSCA/MSCA

kwd	Optional	The kwd lines represent attributes which are specified in the SailPoint schema.
		`kwd=value`
		or
		`kwd=(value1,value2,value3,value4...)`

UPDUSER – Update a User

Use the following request and parameter lines to update a user.

```
:UPDUSER=userid
```

Parameter	Mandatory/Optional	Description
USER.PARENT_ CONTAINER	Mandatory	Parent Container name. For example: `USER.PARENT_CONTAINER=parent-container-name`
USER.AUTH	Optional	`USER.AUTH=REG / ADMIN / AUDIT / ADMINAUDIT`
USER.DFLTGRP	Optional	`USER.DFLTGRP= (default-group-name, DROPOLD /`

Parameter	Mandatory/Optional	Description
		KEEPOLD)
USER.PASSWORD	Optional	USER.PASSWORD=[()]password[,PERM / TEMP)]
USER.DISABLE	Optional	USER.DISABLE=YES
USER.ENABLE	Optional	USER.ENABLE=YES
kwd	Optional	The kwd lines represent attributes which are specified in the SailPoint schema. kwd=value or kwd=(value1,value2,value3,value4...) To nullify a keyword, the keyword must be specified with a null value. To delete one value from a multi-value keyword field, specify the keyword with all remaining values.

DELUSER – Delete a User

Use the following request line to delete a user.

```
:DELUSER=userid
```

DISABLEUSER – Delete a User

Use the following request line to disable a user.

```
:DISABLEUSER=userid
```

ENABLEUSER – Enable a User

Use the following request line to enable a user.

```
:ENABLEUSER=userid
```

LISTUSER – List User Information

Use the following request and parameter lines to list user information.

```
:LISTUSER
```

Default parameter setting:

```
FILTER.USERID=*
```

Parameter	Mandatory/Optional	Description
FILTER.USERID	Optional	`FILTER.USERID={* | [(]userid[,userid…][)] | userid-mask}`
USER.GETCONN	Optional	Indicates that the groups keyword would contain all the groups associated with the user. `USER.GETCONN=NO/YES`
kwd	Optional	The specified `kwd` lines are attributes which are retrieved and displayed. If no `kwd` is specified all user related attributes are retrieved and displayed. `kwd`

CHGPWD – Change a User Password

Use the following request and parameter lines to change a user's password.

```
:CHGPWD=userid
```

Parameter	Mandatory/Optional	Description
USER.PASSWORD	Optional	User password `USER.PASSWORD =[(]password [,PERM / TEMP)]`
VERIFY_PWD	Optional	Indicates that the password is to be verified, not changed. `VERIFY_PWD=N / Y`

ADDGROUP – Add a Group

Use the following request and parameter lines to add a group.

```
:ADDGROUP=groupid
```

Parameter	Mandatory/Optional	Description
GROUP.PARENT	Optional	`GROUP.PARENT =parent-group-name`
GROUP.PARENT_ CONTAINER	Mandatory	`GROUP.PARENT_CONTAINER=parent-container-name`
kwd	Optional	The kwd lines represent attributes which are specified in the SailPoint schema. `kwd=value` or

Parameter	Mandatory/Optional	Description
		`kwd=(value1,value2,value3,value4...)`

UPDGROUP – Update a Group

Use the following request and parameter lines to update a group.

```
:UPDGROUP=groupid
```

Parameter	Mandatory/Optional	Description
GROUP.PARENT	Optional	`GROUP.PARENT =parent-group-name`
GROUP.PARENT_ CONTAINER	Mandatory	`GROUP.PARENT_CONTAINER=parent-container-name`

kwd	Optional	The kwd lines represent attributes which are specified in the SailPoint schema.
		`kwd=value`
		or
		`kwd=(value1,value2,value3,value4...)`
		To nullify a keyword, the keyword must be specified with a null value.
		To delete one value from a multi-value keyword field, specify the keyword with all remaining values.

DELGROUP – Delete a Group

Use the following request to delete a group.

```
:DELGROUP=groupid
```

LISTGROUP – List Group Information

Use the following request and parameter lines to list group of information.

```
:LISTGROUP
```

Default parameter setting:

```
FILTER.GROUPID=*
```

Parameter	Mandatory/Optional	Description
FILTER.GROUPID	Optional	`FILTER.GROUPID={*|[(]groupid[,groupid…][)]}`
kwd	Optional	The specified `kwd` lines are attributes which are retrieved and

Parameter	Mandatory/Optional	Description
		displayed. If no `kwd` is specified all group-related attributes are retrieved and displayed. `kwd`

ADDCONN – Add a User-Group Connection

Use the following request and parameter lines to add a user-group connection.

```
:ADDCONN=(userid, groupid)
```

Parameter	Mandatory/Optional	Description
CONN.AUTH	Optional	`CONN.AUTH=REG/ADMIN/AUDIT/ADMINAUDIT`

kwd	Optional	The `kwd` lines represent attributes which are specified in the SailPoint schema.
		`kwd=value`
		or
		`kwd=(value1,value2,value3,value4...)`

UPDCONN – Update a User-Group Connection

Use the following request and parameter lines to update a user-group connection.

```
:UPDCONN=(userid, groupid)
```

Parameter	Mandatory/Optional	Description
CONN.AUTH	Optional	`CONN.AUTH=REG/ADMIN/AUDIT/ADMINAUDIT`

kwd	Optional	The kwd lines represent attributes which are specified in the SailPoint schema.
		`kwd=value`
		or
		`kwd=(value1,value2,value3,value4...)`
		To nullify a keyword, the keyword must be specified with a null value.
		To delete one value from a multi-value keyword field, specify the keyword with all remaining values.

DELCONN – Delete User-Group Conn

Use the following request and parameter lines to delete a user-group connection.

```
:DELCONN=(userid, groupid)
```

Parameter	Mandatory/Optional	Description
CONN.AUTH	Optional	`CONN.AUTH=REG/ADMIN/AUDIT/ADMINAUDIT`
OWNER	Optional	`OWNER=connection-owner`

LISTCONN – List User-Group Connection Information

Use the following request and parameter lines to list user-group connection information.

```
:LISTCONN
```

Default parameter setting:

```
FILTER.CONN=*
```

All the parameters listed below are mutually exclusive.

Parameter	Mandatory/Optional	Description
FILTER.CONN	Optional	All user-group connections or a specific user-group connection `FILTER.CONN={* | (userid,groupid)}`
FILTER.GROUP	Optional	All user connections with the specific group(s) `FILTER.GROUP=[(]groupid[,groupid…)]`
FILTER.USER	Optional	All group connections with the specific user(s) `FILTER.USER=[(](userid[,userid…)]`

LISTACL – List ACL (Permission) Information

Use the following request and parameter lines to list ACL (permission) information.

```
:LISTACL
```

Parameter	Mandatory/Optional	Description
FILTER.RESTYPE	Mandatory	`FILTER.RESTYPE=resource-type`
FILTER.RESNAME	Mandatory	`FILTER.RESNAME=resource-name`
kwd	Optional	The specified kwd lines are attributes which are retrieved and displayed. If no kwd is specified, all permission-related attributes are retrieved and displayed. `kwd=`

Invocation JCL

> **Note**
> Before running the utility, refer to System Requirements.

Use the CTSBATCH JCL member to invoke the batch utility. This job calls the batch utility from the CS Server procedure but uses a different program name (EXEC CTSACS,PROG=CTSCBAT). The input control statements for the utility are provided in DD statement SYSIN.

To use a different input file name, specify the TOKEN parameter and add the DD statement for the input file as follows:

```
//CALLCBAT        EXEC CTSACS,PROG=CTSCBAT,TOKEN=(DDIN(BATINPUT))
//CTSACS.BATINPUT DD        DSN=SPIIQ.V4000.INPUT(BATCHFAC),DISP=SHR
```

SYSIN File Example

The following is an example of a SYSIN file specifying several provisioning /list requests:

```
:ENV
RSS_TYPE=TSS
RSS_NAME=T4PTSS
ADMIN_UNAME=SECAL
*****************
:ADDUSER=SECSTT
USER.AUTH=ADMINAUDIT
USER.PASSWORD=PSW@STT
USER_TYPE=USER
USER.PARENT_CONTAINER=YON1DEP
NAME=SELIG
*******************
:LISTUSER
FILTER.USERID=SECSTT
*******************
:UPDUSER=SECSTT
USER.PARENT_CONTAINER=YON1DEP
NAME=ZIGGY
*******************
:LISTUSER
FILTER.USERID=SECSTT
*******************
:DELUSER=SECSTT
*****************
:ADDGROUP=SELIG3
UG_TYPE=GROUP
GROUP.PARENT_CONTAINER=YON1DEP
NAME=TEST
****************
:LISTGROUP FILTER.GROUPID=SELIG3
```

```
****************
:UPDGROUP=SELIG3
GROUP.PARENT_CONTAINER=YON1DEP
NAME=SELIG
****************
:LISTGROUP
FILTER.GROUPID=SELIG3
*****************
:DELGROUP=SELIG3
```

Sample Batch Job Output

When executing the batch utility with the provisioning/list requests as displayed in the SYSIN input file above, the following output is printed in the SYSPRINT file:

```
================================
*** Control-SA Batch Program ***
Control-SA 4.0.00 - CTSA400
================================
:ADDUSER:
User    : SECSTT
Group    :
Password : PSW@STT
Status    : Ignored
Authority: Administrator & Auditor
Password
    life    : Ignored
    ADDINFO: TYPE    KEYWORD/VALUE
             1A:    USER_TYPE = USER
             1A:    NAME = SELIG
API call :ADDUSER OK
API msg(s):
2017/09/12 2:18:24 CTS1380I R SA-Agent Batch Utility version 4.0.00 ID
SECSTBAT/JOB08979 started
2017/09/12 2:18:25 CTS3600I R TSS CREATE(SECSTT) NAME(UNKNOWN) TYPE(USER) PASSWORD
(*******) DEPARTMENT(YON1DEP)
2017/09/12 2:18:25 CTS3600I R TSS0300I CREATE    FUNCTION SUCCESSFUL
2017/09/12 2:18:26 CTS3600I R TSS REPLACE(SECSTT) NAME(SELIG)
2017/09/12 2:18:26 CTS3600I R TSS0300I REPLACE FUNCTION SUCCESSFUL
:LISTUSER:
User List
=========
User: SECSTT
Group     :
Password :
Status     : Normal
Authority: Regular
Password
    status: Ignored
ADDINFO: TYPE    KEYWORD/VALUE
         1B:    ADMIN.SCOPE =
```

```
         1A:    DCEKEY =
         1A:    WAADDR4 =
         1A:    WAADDR3 =
         1A:    DFLTGRP =
         1A:    WAADDR2 =
         1B:    XCOMMAND =
         1A:    TSOLACCT =
         1A:    WAADDR1 =
         1B:    DEFNODES =
         1A:    DIV_NAME = * UNKNOWN DIVISION *
         1A:    NOVOLCHK = N
         1A:    INFO.CREATED_DATE = 20170912
         1A:    USER_TYPE = USER
         1A:    LAST_USED_CPU =
         1B:    SOURCE =
         1A:    DUFXTR = N
         1A:    OPIDENT =
         1A:    WABLDG =
         1A:    INFO.LAST_MOD = 09/12/17 02:18
         1A:    LAST_USED_COUNT =
         1A:    MULTIPW = N
         1A:    ParentContainerACID = YON1DEP
         1A:    NAME = SELIG
         1A:    CONSOLE = N
         1A:    MCSDOM =
         1A:    ZONE_ACID =
         1B:    FACILITY_51 =
         1A:    SMSMGMT =
         1A:    IMSMSC =
         1A:    INFO.PASSWORD_EXP_DATE = 20171012
         1A:    OIDCARD = N
         1A:    MCSKEY =
         1A:    DEPT_NAME = * UNKNOWN DEPARTMENT *
         1A:    DEPT_ACID = YON1DEP
         1A:    HOME =
         1B:    ADMIN.DATA =
*** Total number of users found: 1 ***
:UPDUSER:
    User    : SECSTT
    Group    : \
    Password: \
    Status    : Ignored
    Authority: Ignored
    Password
        life    : Ignored
Old Def UG Action: Ignored
    ADDINFO: TYPE    KEYWORD/VALUE
             1A:    NAME = ZIGGY
API call :UPDUSER OK
API msg(s):
2017/09/12 2:18:26 CTS3600I R TSS MOVE(SECSTT) DEPARTMENT(YON1DEP) TYPE(USER)
2017/09/12 2:18:26 CTS3600I R TSS0300I MOVE    FUNCTION SUCCESSFUL
2017/09/12 2:18:26 CTS3600I R TSS REPLACE(SECSTT) NAME(ZIGGY)
2017/09/12 2:18:27 CTS3600I R TSS0300I REPLACE FUNCTION SUCCESSFUL
```

```
:LISTUSER:
User List
=========
User: SECSTT
    Group     :
    Password :
    Status    : Normal
    Authority: Regular
    Password
        status: Ignored
    ADDINFO: TYPE    KEYWORD/VALUE
             1B:    ADMIN.SCOPE =
             1A:    DCEKEY =
             1A:    WAADDR4 =
             1A:    WAADDR3 =
             1A:    DFLTGRP =
             1A:    WAADDR2 =
             1B:    XCOMMAND =
             1A:    TSOLACCT =
             1A:    WAADDR1 =
             1B:    DEFNODES =
             1A:    DIV_NAME = * UNKNOWN DIVISION *
             1A:    NOVOLCHK = N
             1A:    INFO.CREATED_DATE = 20170912
             1A:    USER_TYPE = USER
             1A:    LAST_USED_CPU =
             1B:    SOURCE =
             1A:    DUFXTR = N
             1A:    OPIDENT =
             1A:    WABLDG =
             1A:    INFO.LAST_MOD = 09/12/17 02:18
             1A:    LAST_USED_COUNT =
             1A:    MULTIPW = N
             1A:    ParentContainerACID = YON1DEP
             1A:    NAME = ZIGGY
             1A:    CONSOLE = N
             1A:    MCSDOM =
             1A:    ZONE_ACID =
             1B:    FACILITY_51 =
             1A:    SMSMGMT =
             1A:    IMSMSC =
             1A:    INFO.PASSWORD_EXP_DATE = 20171012
             1A:    OIDCARD = N
             1A:    MCSKEY =
             1A:    DEPT_NAME = * UNKNOWN DEPARTMENT *
             1A:    DEPT_ACID = YON1DEP
             1A:    HOME =
             1B:    ADMIN.DATA =
*** Total number of users found: 1 ***
:DELUSER:
    User    : SECSTT
        ADDINFO: TYPE    KEYWORD/VALUE
API call :DELUSER OK
API msg(s):
```

```
2017/09/12 2:18:27 CTS3600I R TSS DELETE(SECSTT)
2017/09/12 2:18:27 CTS3600I R TSS0300I DELETE    FUNCTION SUCCESSFUL
:ADDGROUP:
    Group    : SELIG3
    Parent Group:
    Parent
        Container: YON1DEP
    ADDINFO: TYPE    KEYWORD/VALUE
             1A:    UG_TYPE = GROUP
             1A:    NAME = TEST
API call :ADDGROUP OK
API msg(s):
2017/09/12 2:18:27 CTS3600I R TSS CREATE(SELIG3) NAME(UNKNOWN) TYPE(GROUP)
DEPARTMENT(YON1DEP)
2017/09/12 2:18:27 CTS3600I R TSS0300I CREATE    FUNCTION SUCCESSFUL
2017/09/12 2:18:27 CTS3600I R TSS REPLACE(SELIG3) NAME(TEST)
2017/09/12 2:18:27 CTS3600I R TSS0300I REPLACE FUNCTION SUCCESSFUL
:LISTGROUP:
Group List
==========
Group: SELIG3
    Parent group:
        ADDINFO: TYPE    KEYWORD/VALUE
                 1A:    AUDIT = N
                 1A:    INFO.LAST_MOD = 09/12/17 02:18
                 1A:    OIDCARD = N
                 1A:    DEPT_NAME = * UNKNOWN DEPARTMENT *
                 1A:    ParentContainerACID = YON1DEP
                 1A:    INFO.CREATED_DATE = 20170912
                 1B:    COMMAND =
                 1A:    DIV_ACID = DIV=YON
                 1A:    TRACE = N
                 1A:    LANGUAGE =
                 1A:    GID =
                 1A:    ParentContainerType = DEPT
                 1A:    NAME = TEST
                 1A:    NOATS = N
                 1A:    NODSNCHK = N
                 1A:    NOSUBCHK = N
                 1A:    ZONE_NAME =
                 1A:    OPPRTY =
                 1B:    OPCLASS =
                 1A:    DEPT_ACID = YON1DEP
                 1A:    UG_TYPE = GROUP
                 1A:    OPIDENT =
                 1A:    NOPWCHG = N
:UPDGROUP:
    Group    : SELIG3
    Parent Group:
    Parent
        Container: YON1DEP
    ADDINFO: TYPE    KEYWORD/VALUE
             1A:    NAME = SELIG
API call :UPDGROUP OK
```

```
API msg(s):
2017/09/12 2:18:28 CTS3600I R TSS MOVE(SELIG3) DEPARTMENT(YON1DEP) TYPE(GROUP)
2017/09/12 2:18:28 CTS3600I R TSS0300I MOVE    FUNCTION SUCCESSFUL
2017/09/12 2:18:28 CTS3600I R TSS REPLACE(SELIG3) NAME(SELIG)
2017/09/12 2:18:28 CTS3600I R TSS0300I REPLACE FUNCTION SUCCESSFUL
:LISTGROUP:
Group List
==========
Group: SELIG3
Parent group:
ADDINFO: TYPE    KEYWORD/VALUE
          1A:    AUDIT = N
          1A:    INFO.LAST_MOD = 09/12/17 02:18
          1A:    OIDCARD = N
          1A:    DEPT_NAME = * UNKNOWN DEPARTMENT *
          1A:    ParentContainerACID = YON1DEP
          1A:    INFO.CREATED_DATE = 20170912
          1B:    COMMAND =
          1A:    DIV_ACID = DIV=YON
          1A:    TRACE = N
          1A:    LANGUAGE =
          1A:    GID =
          1A:    ParentContainerType = DEPT
          1A:    NAME = SELIG
          1A:    NOATS = N
          1A:    NODSNCHK = N
          1A:    NOSUBCHK = N
          1A:    ZONE_NAME =
          1A:    OPPRTY =
          1B:    OPCLASS =
          1A:    DEPT_ACID = YON1DEP
          1A:    UG_TYPE = GROUP
          1A:    OPIDENT =
          1A:    NOPWCHG = N
:DELGROUP:
    Group    : SELIG3
    Parent Group:
    ADDINFO: TYPE    KEYWORD/VALUE
API call :DELGROUP OK
API msg(s):
2017/09/12 2:18:28 CTS3600I R TSS DELETE(SELIG3)
2017/09/12 2:18:28 CTS3600I R TSS0300I DELETE    FUNCTION SUCCESSFUL
2017/09/12 2:18:28 CTS1381I SA-Agent Batch Utility version 4.0.00 ID
SECSTBAT/JOB08979 ended OK
```

Appendix G: Connector for Top Secret Profile Ordering

This appendix describes the procedure for Connector for Top Secret Profile Ordering from IdentityIQ while adding entitlement from IdentityIQ.

Perform the following steps:

1. Edit the application in IdentityIQ debug page.
2. Add the following to the Attributes Map:

```
<entry key="splConnectionAttributes">
     <value>
        <Map>
          <entry key="RELATIVE_POS" value="false"/>
          <entry key="RELATIVE_PROFILE" value="false"/>
        </Map>
     </value>
</entry>
```

3. Add the following provisioning policy to the application:

```
<Form name="Update account" objectType="account" type="Update"><Attributes>
<Map>
<entry key="pageTitle" value="Update account"/>
</Map>
</Attributes>
<Section name="Section">
<Field displayName="Relative Position" helpKey="Applicable for adding enti-
tlement." name="RELATIVE_POS" reviewRequired="true" type="string">
<AllowedValuesDefinition>
<Value>
<List>
<String>TOP</String>
<String>BEFORE</String>
<String>AFTER</String>
<String>BOTTOM</String>
</List>
</Value>
</AllowedValuesDefinition>
</Field>
<Field displayName="Relative Profile" helpKey="Name of the relative pro-
file/group. Applicable if relative position is BEFORE or AFTER." name-
e="RELATIVE_PROFILE" reviewRequired="true" type="string"/>
</Section>
</Form>
```

Note the following:

- This policy is applicable only for entitlement requests. But since the policy is common for all update requests, this will be prompted for non-entitlement update requests too. User must skip these to proceed with the requests.
- The provisioning policy introduced in this solution is applicable for all update user requests. When single/multiple entitlements are added, user will be prompted for the ordering attributes only once. Then the same ordering attribute values are applicable for all the entitlements which are getting added. Hence to avoid this, if one entitlement is added in one request, the ordering attribute values can be provided for each profile individually.
- In case, the order for the entitlements is already identified and always remains same, the input needs not to be asked each time. In such scenario, the above provisioning policy must be added to application. A plan initializer script can be introduced which will populate the above attributes depending on the fix entitlements order.

Connection Data Translation Tables

Field name	Top Secret parameter
EXPIRES	Translates to the Top Secret parameter UNTIL.
RELATIVE_POS	Determines position of a Profile within user ACID. Can be ‘TOP’, ‘AFTER’, ‘BEFORE’, ‘BOTTOM’.
RELATIVE_PROFILE	Determines relative Profile name (in case RELATIVES_POS=AFTER/BEFORE).
POS	(Read-only). Shows Serial number of Profile with the user ACID.

These attributes are relevant and may be used only within the provisioning transaction.