MHS Information Platform (MIP)
SAS Computing Environment User’s Guide

Version 1.5

Revision History

Version	Date	Description of Changes
1.01	December 2022	● Major updates to restructure document for MIP SCE environment considerations will be published as a working draft for use
1.1	January 2023	● Includes updates identified during the MIP MDR Limited User Testing event. Removed the "DRAFT" watermark to finalize the version
1.2	January 2023	● Made document 508 compliant ● Enhance modifiedDate example
1.3	February 10, 2023	● Edited the filesGet syntax for Command Line Interface ● Edited smart quotes
1.4	May 2, 2023	● Added contact information for MDR Operations and Processing ● Added clarification on filesGet,Remove functions ● Updated user training and resource links
1.5	September 13, 2023	● Updated saswork workspace and job submission information

CTRL + Click to go to the locations in the Table of Contents below:

Contents

I. Introduction and MDR Overview 4

II. System Overview and Technical Architecture 5

Connections to Internal and External Resources 5

Data Refresh and Availability 5

System Training 5

III. Space Allocation and Related Issues 6

IV. Storing and Securing Privacy Act and HIPAA Data 7

Keeping Logs 7

V. Getting Started 9

Connecting to the SCE 9

MIP AWS- WinSCP Application: See Appendix C 10

MIP AWS- PuTTY Application Guide: See Appendix D 10

MIP AWS-SAS EG (Enterprise Guide) Application: 11

Working with AWS S3 Files 11

Modified Dates 15

Submitting SAS Jobs in the SCE -- SAS Job Queue 15

Scheduling SAS Jobs 20

Moving Files to the SCE from External Systems: Data Imports and Extractions 21

Commonly Used UNIX Commands 22

Linux File Naming Conventions 26

The Linux Directory Tree 27

Getting Around Directories 28

Appendix A: The NANO Editor (For editing in Linux, a clone of PICO) and VI editor 29

Basic VI Editor 32

Appendix B: MIP SCE Quick Reference Guide 35

Appendix C: AVHE MIP AWS -- WinSCP Login to MIPSAPSASUSRP01 37

Appendix D: SAS Launch PuTTY SSH Session to MIPSAPSASUSRP01 43

Appendix E: SAS Create Enterprise Guide Profile to MIPSAPSASUSRP01 48

I. Introduction and MDR Overview

The SAS Computing Environment (SCE) provides authorized individuals with access and query capability to specified Military Health System (MHS) Data Repository (MDR) data sets within the MHS Information Platform (MIP). This User Guide addresses the processes for accessing and utilizing the SCE. The SCE is for experienced healthcare analysts, who are enterprise-level power users possessing a thorough knowledge of SAS and existing MHS data sets.

Data sets contained within the MDR include workload, cost, population, enrollment, demographic, and reference data for the MHS. The data sets are arrayed in a structured manner. They are available post processing and are represented as a subset of the MDR file catalog:

- According to a defined periodicity or update cycle;
- Processed using known, tested software - with identified business rules;
- Quality controlled, documented, and reviewed; and
- Released for specified use.

MDR Information Resources

Additional information regarding the MDR Alerts, MDR Data Dictionary, MDR Functional Guide, and data import/export transmittal can be found at the following link below.

https://health.mil/Military-Health-Topics/Technology/Support-Areas/MDR-M2-ICD-Functional-References-and-Specification-Documents

The MDR Users Group

This group meets every second Wednesday of each month at 1300 ET. Topics covered are relevant to all MDR users and MDR data partners. Historical documentation is available at the link below.

https://info.health.mil/staff/analytics/MDRUsersGroup/Forms/AllItems.aspx

MDR Blasters

Periodically, all MDR users will receive a “blaster” message which is sent BCC to all MDR users to communicate significant milestones and/or events (e.g. outages, system wide changes…).

For questions about user profiles, MDR blasters, MDR data feeds and import/export transmittals use the following email.

dha.jbcharleston.j-6.list.eids-mip-mdr-ops@health.mil

For questions about data quality or currency (Process control worksheet) users should use the processing email.

dha.jbcharleston.j-6.list.eids-mip-mdr-processing@health.mil

II. System Overview and Technical Architecture

The SCE resides in the MHS Information Platform (MIP) which is in an AWS GovCloud environment specifically managed by the EIDS program office. The data in the SCE is a combination of SAS datasets and ASCII flat files. Most processed data exists in SAS data set form and takes advantage of SAS procedures and library functions.

Available MIP Tools and Software

The MIP offers a variety of tools for analysts to utilize (e.g. Python, WinSCP). See the link below to find the current versions in AVHE.

https://carepoint.health.mil/sites/MIP/mugteam/SitePages/Tools%20Main%20Page.aspx

Connections to Internal and External Resources

Authorized users have the ability to access the following external databases using their credentials. TED-Bench, TED-Query, PCDW-Q, PDW, DMHRSi, PEPR

TED_Bench, TED_Query, PCDW_Q, PEPR, PDW please see the lib name statements to access.

```
- LIBNAME db2TEDSB ODBC DSN="TEDSTGB2" USER='sasUser' PASSWORD='sasPass' ; *TED-Bench Stage;
- LIBNAME db2TEDOB ODBC DSN="TEDODSB2" USER='sasUser' PASSWORD='sasPass' ; *TED-Bench Prod;
- LIBNAME db2TEDSP ODBC DSN="TEDSTGP2" USER='sasUser' PASSWORD='sasPass' ; *TED-Query Stage;
- LIBNAME db2TEDOP ODBC DSN="TEDODSP2" USER='sasUser' PASSWORD='sasPass' ; *TED-Query Prod;
- LIBNAME db2PCDW6 ODBC DSN="PCDW6BP1" USER='sasUser' PASSWORD='sasPass' ; *PCDW;
- LIBNAME db2PPRRP ODBC DSN="PPRRPT1P" USER='sasUser' PASSWORD='sasPass' ; *PEPR;
- LIBNAME sqlPDW SQLSVR DSN="PDW" SCHEMA=dbo USER=sasUser PASSWORD=sasPass ;
```

Authorized users can access MIP resources such as Redshift, MIP FileShare and SQL Farm. Users will need to submit an update to their SAARs access.

Please see the following link for assistance.

https://carepoint.health.mil/sites/MIP/mugteam/SitePages/SAS%20EG%20Training.aspx

Data Refresh and Availability

The MDR data sets in this environment will be refreshed with current week, month, quarter, or yearly data immediately after MDR processing is completed. For MDR data sets that are processed, the update will be loaded upon receipt and completion of Quality Control (QC). For more information on data sets you can consult the MDR data dictionary located at

https://health.mil/Military-Health-Topics/Technology/Support-Areas/MDR-M2-ICD-Functional-References-and-Specification-Documents

System Training

The MIP Users Group (MUG) offers a multitude of training content to assist in learning how to leverage MIP capabilities. See the site navigation panel on the left side of the page for more information on Training.

https://carepoint.health.mil/sites/MIP/mugteam/SitePages/Home.aspx

SAS specific training can be found at the link below:

https://carepoint.health.mil/sites/MIP/mugteam/SitePages/SAS%20EG%20Training.aspx

Additional training materials are below:

Crontab Manual: https://linux.die.net/man/5/crontab

Versioning

Backup of files that exist in the MDR/SCE are handled by Amazon Web Services (AWS) S3 versioning which is enabled for the SCE. Any output/work product removed from AWS S3 requires MIP system administration staff to restore. Versioning in Amazon S3 is a means of keeping multiple variants of an object in the same bucket. S3 Versioning is active in the SCE and allows users to preserve, retrieve, and restore every version of every object stored in S3 buckets. Versioning allows for recovery more easily from both unintended user actions and application failures.

All requests for file restoration are to be submitted via GSC ticket.
https://carepoint.health.mil/sites/MIP/mugteam/SitePages/PMO%20EIDS%20Report%20Issues.aspx

III. Space Allocation and Related Issues

A directory is a method of storing and organizing data files, such as SAS programs, output files, SAS datasets, and text files. It also allows for manipulation and retrieval of data within and across directories. Every directory on the SCE has rules that set permissions for user access directories will be determined based on sponsoring organization and project designation.

All requests for additional space must be submitted by email to the SCE Functional Proponent, Mr. James Huber (james.t.huber4.civ@health.mil).

As these directories contain private rather than MDR files, the government and/or functional owner of the directory is responsible for:

- Ensuring the directory contains only data authorized by law and regulation;
- Ensuring Privacy Act and Health Insurance Portability and Accountability Act (HIPAA) laws and regulations are followed; and
- Notifying EIDS of both who is empowered to provide access authorizations for that directory, and through that agent, the list of users allowed into those project or sponsor- designated files.

IV. Storing and Securing Privacy Act and HIPAA Data

Health Insurance Portability and Accountability Act (HIPAA) requires healthcare systems to record any use or disclosure of an individuals' healthcare data.

Users are responsible for full compliance with HIPAA and the Privacy Act for data created or placed into all files systems on the SCE. EIDS does not track data stored in user workspace within these file systems. Private Health Information (PHI) data created as a result of programs extracting data from the MDR datasets for download or those originating outside of the SCE and imported/uploaded/keyed into these directories must be reported via e-mail to the SCE Functional Proponent (james.t.huber4.civ@health.mil) with the subject line HIPAA download Request. This applies to the movement of data outside of the SCE architecture.

The email should be sent **before the file is downloaded or uploaded**. Multiple files can be listed in the same email. The e-mail needs to include a brief description of the file, approximate size, source, and destination directory, along with your identity and organization. This information may be used to determine if an Import or Export Transmittal form is required (e.g., large amounts of data).

Downloads of PHI data that are 10,000 rows or greater requires an Export Transmittal request. Users should not download this data themselves via WinSCP or any other FTP software.

EXAMPLE OF DOWNLOADING EMAIL:
25-March 2022: Downloaded Direct Care SIDR and SADR records from /xyz/abc to hard drive; ~2,500 records total, with PHI and PII data. Delivered to OTSG Decision Support Center via secure FTP (HIPAA compliant). Data deleted from hard drive after delivery. John Doe, ABC Company (Decision Support).

EXAMPLE OF UPLOADING EMAIL:
25-March- 2022: Uploaded DMDC-provided file containing 2,500 activated guard/reserve records with SSNs and minimal activation information, saved into the "ABC Company" file system. John Doe, ABC Company (Decision Support).

Restricted Files

Some data in the MDR is considered sensitive and requires special justification. To obtain access to files in /mdr/restricted, users must complete a special justification form signed by their Government Sponsor/supervisor. The form can be found at the following link and must be submitted to Mr. James Huber.

https://www.health.mil/Military-Health-Topics/Technology/Support-Areas/MDR-M2-ICD-Functional-References-and-Specification-Documents

Keeping Logs

In healthcare systems such as the SCE, EIDS can identify data access or disclosure to the point of knowing which SCE user extracted data from an MDR data set on a specific date. In order to fulfill the remainder of the requirement for records of access, **each SCE user must retain logs of program executions on any files containing Privacy Act data**. These logs can be the normal electronic logs produced by SAS and stored by the user on the SCE. Upon request, these logs must be made available to EIDS, DHA Privacy Office, and/or Decision Support.

One example for automatically saving the SAS logs to a dedicated log directory includes the following SAS macro which is coded at the beginning of every SAS program:

```
%let dir1=%str(/main/secondary/task/log/);
%let pgmname=%str(xxxpgm_name);
%let date=&sysdate;
%let time=&systime;
%let ext1=%str(log);
%let break=%str(-);
%let final1=&dir1&pgmname&date&break&time..&ext1;
%put &final1;
proc printto log="&final1";
run; quit;
```

/main = the main directory of your dedicated workspace

/secondary = secondary directory, if applicable

/task = specific task name

/log = log directory set up for each task

xxx = users initials

pgm_name = program name

Before running this, make sure you have created the log directory in the file path. This is where all the logs, every time the program is run, will be stored for future reference, if necessary.

Notes:
1) create task areas for easier reference
2) use program names that make sense for the task
3) user initials allows for identification of who ran the program

If you use the above code, you MUST use the following to submit the program:

```
/main/secondary/task/log > sas pgm_name.sas --altlog pgm_name.log
```

The --altlog <program_name>.log portion of the command also puts a *.log file in the directory you submitted the program from and is where the programmer can review the results of the program. So, essentially two duplicate logs are created, one for the programmer to review, the other to be HIPAA compliant. The HIPAA compliant log never gets overwritten. The reviewer log gets overwritten every time the program is rerun.

V. Getting Started

Connecting to the SCE

Users are required to use the Application Virtualization Hosting Environment (AVHE) to authenticate and open virtual applications. Users requiring access to the MDR/SCE need to follow the process listed at the following link under “MDR Access Forms”.

> https://health.mil/Military-Health-Topics/Technology/Support-Areas/MDR-M2-ICD-Functional-References-and-Specification-Documents

AVHE Login:

Use this link https://avhe.health.mil/Citrix/AVHEWeb/ to access AVHE.

Submitting GSC tickets

In order to submit a Global Service Center (GSC) Ticket for issues, please send an email in the following format. The Subject MUST contain “MHS Information Platform (MIP)” in the SUBJECT in order to get the ticket to the correct team.

TO: **dha.jbsa.j-6.mbx.MHS-Service-Desk@health.mil**

CC: **DHA JB Charleston J-6 List PS dha.jbcharleston.j-6.list.ps@health.mil**

SUBJECT: **GSC Ticket: MHS Information Platform (MIP)** Ticket **for MIP MDR**

BODY:

Route To: MHS Information Platform (MIP)

Name: [REQUIRED]

Phone Number: [REQUIRED]

E-mail Address: [REQUIRED]

Site: (optional)

Organization: [REQUIRED]

Department: (opt)

Building Number: (opt)

Room Number: (opt)

Full Computer Name (opt)

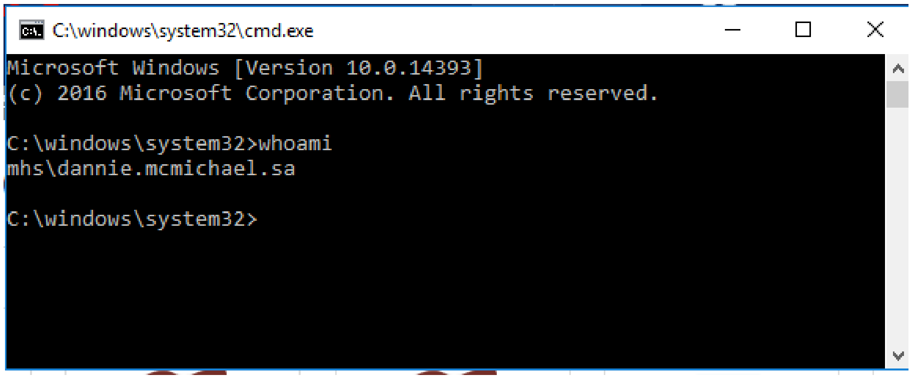
A detailed description of the issue/request: (include as much details as possible such as URLs, screenshots, Tableau bitab folders, application names, etc.) [REQUIRED]

SIGNATURE BLOCK

Obtain "Username":

Background: Your "Username" is required when accessing the SCE through Putty or WinSCP. SAS Enterprise Guide does not require a user to know their username. Usernames do not follow a standard naming convention. To determine what your "Username" is:

1. Open AVHE 'MIP AWS -- Command Prompt'
2. Enter 'whoami' command
3. Username does not include the domain 'mhs\' (e.g. dannie.mcmichael.sa)



```
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\windows\system32>whoami
mhs\dannie.mcmichael.sa

C:\windows\system32>
```

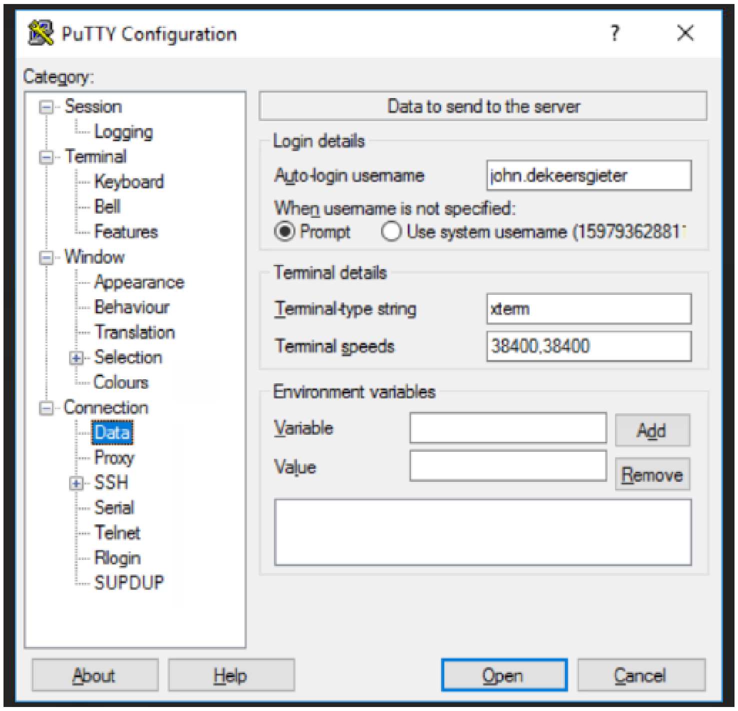
Once the account is approved, permissions are established, and the user obtains their username, the user will be able to access the SCE.

MIP AWS- WinSCP Application: See Appendix C

https://carepoint.health.mil/sites/MIP/mugteam/Qlik/SAS%20Setup%20WinSCP%20Login%20to%20MIPSAPSASUSRP01.docx

MIP AWS- PuTTY Application Guide: See Appendix D

https://carepoint.health.mil/sites/MIP/mugteam/Qlik/SAS%20Launch%20PuTTY%20SSH%20Session%20to%20MIPSAPSASUSRP01.docx



Login details

Auto-login username: john.dekeersgieter

When username is not specified:
● Prompt ○ Use system username (15979362881

Terminal details

Terminal-type string: xterm

Terminal speeds: 38400,38400

Environment variables

Variable:

Value:

**Note: The purpose of this is to auto login to PuTTY without entering username

MIP AWS-SAS EG (Enterprise Guide) Application:

The MIP Users Group (MUG) site for SAS EG is at the following link for training, access, and issue reporting.

https://carepoint.health.mil/sites/MIP/mugteam/Qlik/SAS%20Create%20Enterprise%20Guide%20Profile%20to%20MIPSAPSASUSRP01.docx

The following document contains useful information about SAS Enterprise Guide. See Appendix E.

https://carepoint.health.mil/sites/MIP/mugteam/Qlik/SAS%20Create%20Enterprise%20Guide%20Profile%20to%20MIPSAPSASUSRP01.docx

Working with AWS S3 Files

SCE user's data is stored in AWS S3 which acts as a data clearinghouse and must be retrieved for use. This data can be accessed through several commands. filesGet, filesPut, filesRemove, these commands are specific to SCE users and provide a method of interacting with AWS S3. Each command is detailed below. The commands are specific to the user's interface, the Command Line Interface (CLI) and SAS Enterprise Guide commands are slightly different.

AWS S3 is the main data store for all MDR/SCE data. Any data that is on the SAS server will be synced to AWS S3 on a daily basis. Data on the SAS server will periodically be cleaned to provide more disk space for process execution. The process can obtain the data from AWS S3 upon re-execution.

S3 permissions are dependent upon the user's access. If a user has read access, they will be able to read/obtain files from S3. If a user has write access, they will be able to overwrite a file within S3.

This is the current retention policy for MIP SCE:

1. Pub files- most recent 3 years will be retained on the SAS server. Once the file is retrieved, it will persist.

2. These directories persist on the SAS server: /apps   /mdr/aprod/util   /mdr/ref

3. These file extensions persist on the SAS server: .sas7bcat, .fmt, .lst, .sas, .log

4. All other files will be retained on the SAS server for 5 days unless a filesGet is used to retrieve the file, then the 5-day rule restarts the clock

Every 2 hours - metadata file (.s3 files) creation and removal occurs.

This command provides a full list of current disk space usage: Disk Usage: df -h

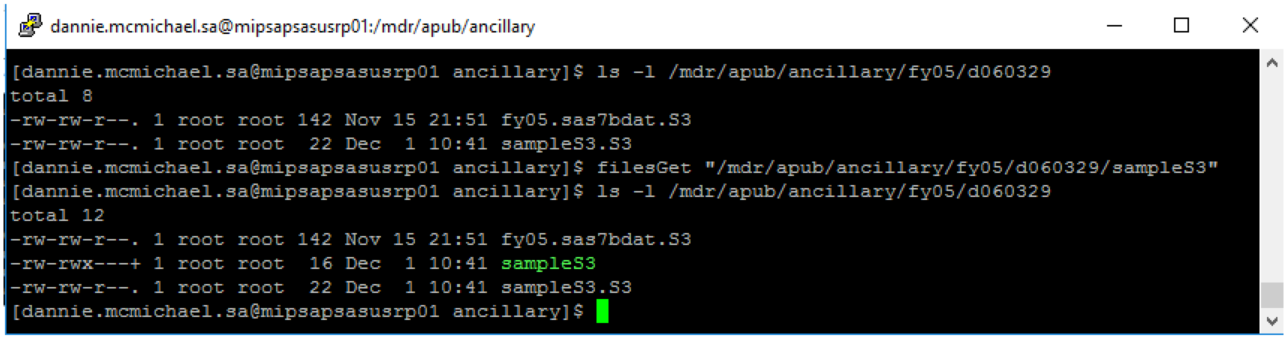
As of September 2023, as files in /mdr/pub and /mdr/ref are updated, those updates will be pushed from S3 into the SAS server. This will ensure that the files present on the SAS server are always the latest available version.

filesGet:
Background:
The filesGet command provides a user the ability to obtain files that reside within S3. This step is required for users to interact with files on the SAS server. It is best practice to include this within your program for **ALL inputs**, as it will obtain files from S3 if S3 has an updated version. The syntax requires the full path (directory and file name) to retrieve the file from S3.

****NOTE: Do not include the .s3 extension when you create your filesGet statement**

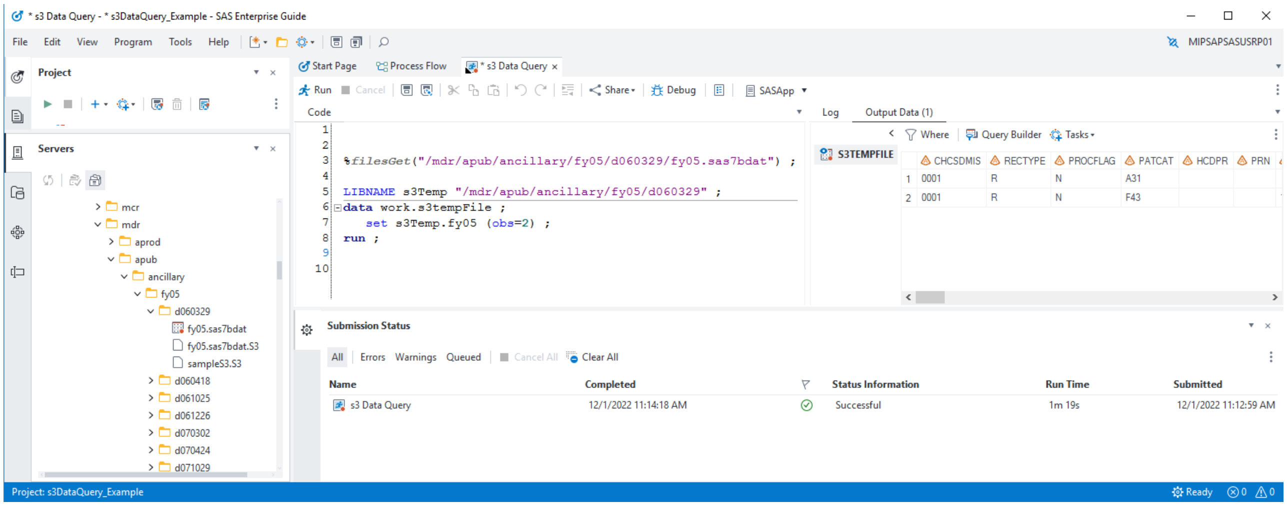
*****NOTE 2: Do not include /mdrfile at the beginning of the file path in the* filesGet *statement***

Command Line Interface (CLI):
Single file syntax: filesGet "/dir/file.ext"
Multiple files syntax: filesGet "/dir/file1.ext, /dir/file2.ext"



```
[dannie.mcmichael.sa@mipsapsasusrp01 ancillary]$ ls -l /mdr/apub/ancillary/fy05/d060329
total 8
-rw-rw-r--. 1 root root 142 Nov 15 21:51 fy05.sas7bdat.S3
-rw-rw-r--. 1 root root  22 Dec  1 10:41 sampleS3.S3
[dannie.mcmichael.sa@mipsapsasusrp01 ancillary]$ filesGet "/mdr/apub/ancillary/fy05/d060329/sampleS3"
[dannie.mcmichael.sa@mipsapsasusrp01 ancillary]$ ls -l /mdr/apub/ancillary/fy05/d060329
total 12
-rw-rw-r--. 1 root root 142 Nov 15 21:51 fy05.sas7bdat.S3
-rw-rwx---+ 1 root root  16 Dec  1 10:41 sampleS3
-rw-rw-r--. 1 root root  22 Dec  1 10:41 sampleS3.S3
[dannie.mcmichael.sa@mipsapsasusrp01 ancillary]$
```

Within a SAS program:
Single file syntax: %filesGet("/dir/file.ext") ;
Multiple files syntax: %filesGet("/dir/file1.ext, /dir/file2.ext") ;



```

%filesGet("/mdr/apub/ancillary/fy05/d060329/fy05.sas7bdat") ;

LIBNAME s3Temp "/mdr/apub/ancillary/fy05/d060329" ;
data work.s3tempFile ;
   set s3Temp.fy05 (obs=2) ;
run ;

```

	CHCSDMIS	RECTYPE	PROCFLAG	PATCAT	HCDPR	PRN
1	0001	R	N	A31		
2	0001	R	N	F43		

Name	Completed	Status Information	Run Time	Submitted
s3 Data Query	12/1/2022 11:14:18 AM	Successful	1m 19s	12/1/2022 11:12:59 AM

filesPut:

Background:

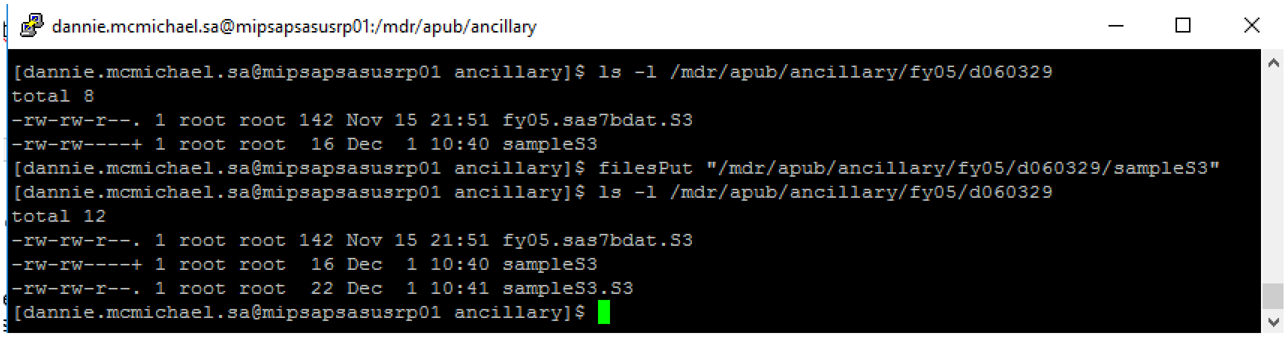
The filesPut command provides a user the ability to place files onto S3. The daily sync process will execute and place any new/modified files onto S3. This command provides the user to explicitly place a file within S3. This may be required depending on the user's use case.

**NOTE: Do not include /mdrfile at the beginning of the file path in the filesPut statement

Command Line Interface (CLI):

  Single file syntax: filesPut "/dir/file.ext"

  Multiple files syntax: filesPut "/dir/file1.ext, /dir/file2.ext"

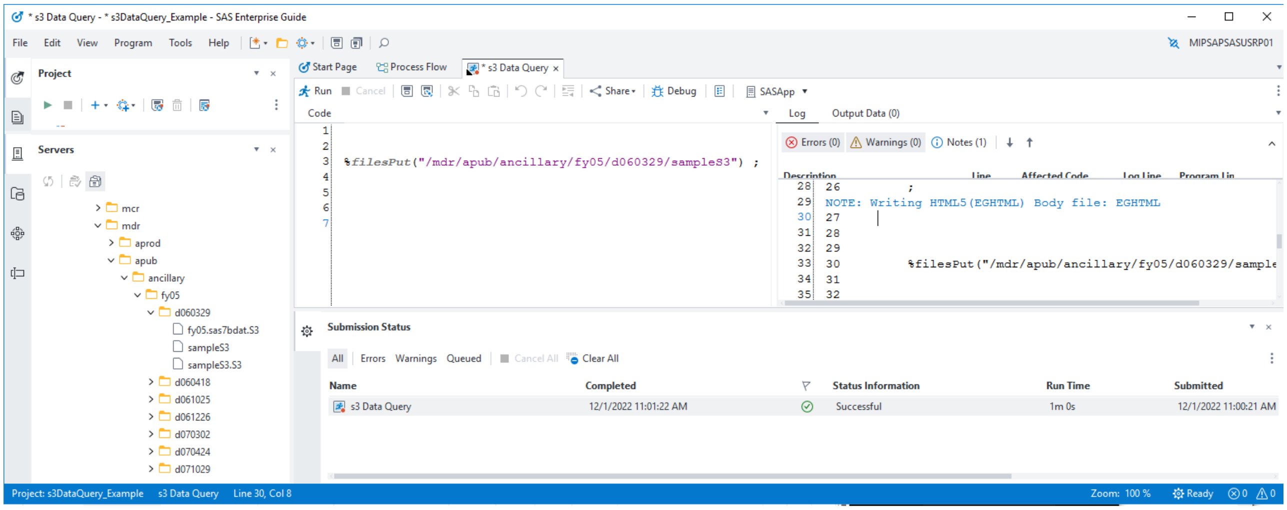


```
[dannie.mcmichael.sa@mipsapsasusrp01 ancillary]$ ls -l /mdr/apub/ancillary/fy05/d060329
total 8
-rw-rw-r--. 1 root root 142 Nov 15 21:51 fy05.sas7bdat.S3
-rw-rw----+ 1 root root  16 Dec  1 10:40 sampleS3
[dannie.mcmichael.sa@mipsapsasusrp01 ancillary]$ filesPut "/mdr/apub/ancillary/fy05/d060329/sampleS3"
[dannie.mcmichael.sa@mipsapsasusrp01 ancillary]$ ls -l /mdr/apub/ancillary/fy05/d060329
total 12
-rw-rw-r--. 1 root root 142 Nov 15 21:51 fy05.sas7bdat.S3
-rw-rw----+ 1 root root  16 Dec  1 10:40 sampleS3
-rw-rw-r--. 1 root root  22 Dec  1 10:41 sampleS3.S3
[dannie.mcmichael.sa@mipsapsasusrp01 ancillary]$
```

Within a SAS program

  Single file syntax: %filesPut("/dir/file.ext") ;

  Multiple file syntax: %filesPut("/dir/file1.ext, /dir/file2.ext") ;



```
%filesPut("/mdr/apub/ancillary/fy05/d060329/sampleS3") ;
```

```
26          ;
NOTE: Writing HTML5(EGHTML) Body file: EGHTML
27
28
29
30          %filesPut("/mdr/apub/ancillary/fy05/d060329/sample
31
32
```

Name	Completed		Status Information	Run Time	Submitted
s3 Data Query	12/1/2022 11:01:22 AM	✓	Successful	1m 0s	12/1/2022 11:00:21 AM

filesRemove:

Background:

If a user exceeds the size of the assigned directory, they will need to clean up the directory. The filesRemove command provides a user the ability to remove files within S3. This requires that a user has write permissions to the file. A user can use this command to remove files from S3 that are no longer needed, releasing storage.

Note: ‘filesRemove’ only removes the file from AWS S3. This can be verified by the removal of the ‘/dir/file.ext.S3’ metadata file. If desired, the user must also use ‘rm’ to remove the file from the SAS server.

Note 1. PuTTY: rm <FILENAME>will remove the file from the SAS server; WinSCP: Right click to delete

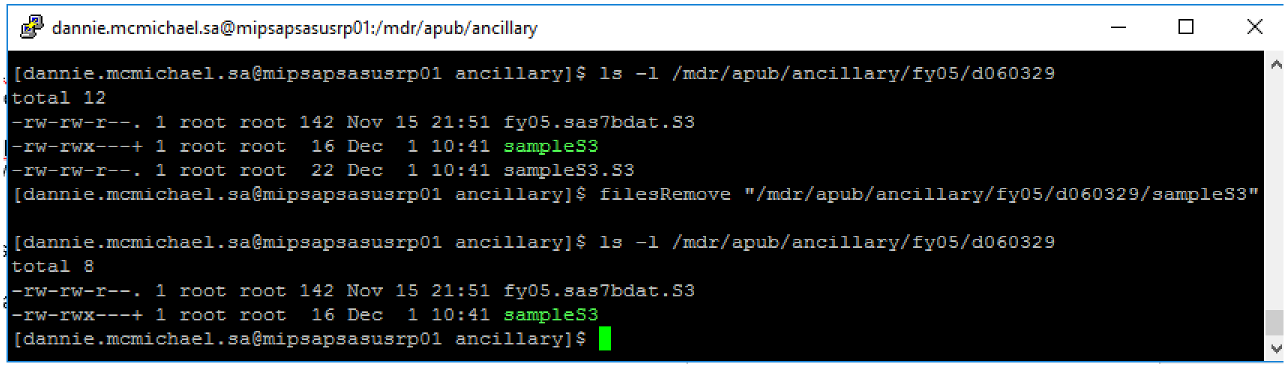
2. filesRemove <FILENAME>will remove the file in S3

**NOTE: Do not include /mdrfile at the beginning of the file path in the filesRemove statement

Command Line Interface (CLI):

Single file syntax: filesRemove "/dir/file.ext"

Multiple files syntax: filesRemove "/dir/file1.ext, /dir/file2.ext"

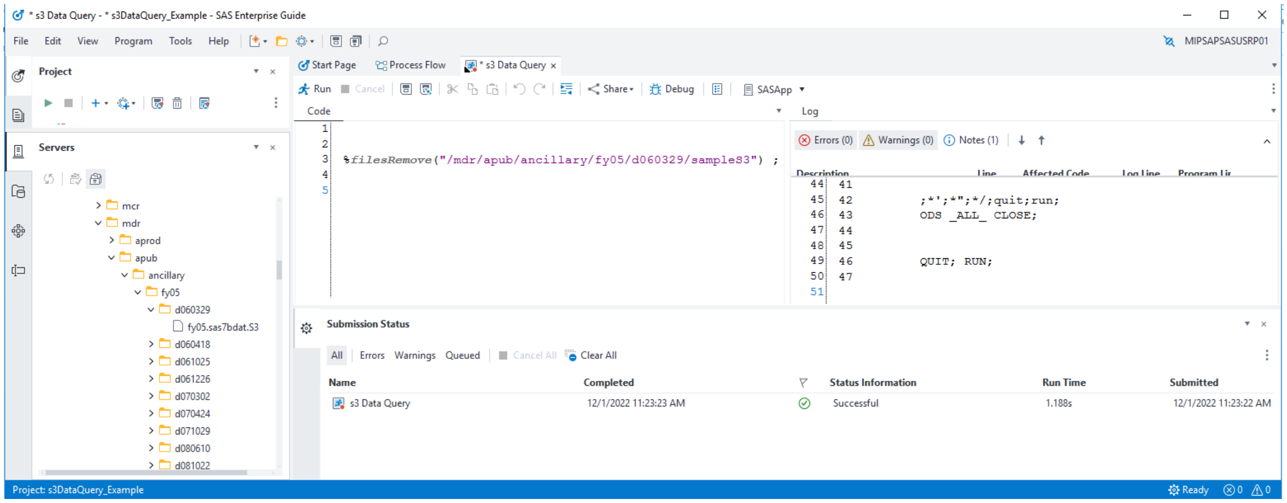


```
[dannie.mcmichael.sa@mipsapsasusrp01 ancillary]$ ls -l /mdr/apub/ancillary/fy05/d060329
total 12
-rw-rw-r--. 1 root root 142 Nov 15 21:51 fy05.sas7bdat.S3
-rw-rwx---+ 1 root root  16 Dec  1 10:41 sampleS3
-rw-rw-r--. 1 root root  22 Dec  1 10:41 sampleS3.S3
[dannie.mcmichael.sa@mipsapsasusrp01 ancillary]$ filesRemove "/mdr/apub/ancillary/fy05/d060329/sampleS3"

[dannie.mcmichael.sa@mipsapsasusrp01 ancillary]$ ls -l /mdr/apub/ancillary/fy05/d060329
total 8
-rw-rw-r--. 1 root root 142 Nov 15 21:51 fy05.sas7bdat.S3
-rw-rwx---+ 1 root root  16 Dec  1 10:41 sampleS3
[dannie.mcmichael.sa@mipsapsasusrp01 ancillary]$ 
```

Within a SAS program:

Single file syntax: %filesRemove("/dir/file.ext") ;

Multiple file syntax: %filesRemove("/dir/file1.ext, /dir/file2.ext") ;



```
%filesRemove("/mdr/apub/ancillary/fy05/d060329/sampleS3") ;
```

```
41
42         ;*';*";*/;quit;run;
43         ODS _ALL_ CLOSE;
44
45
46         QUIT; RUN;
47
```

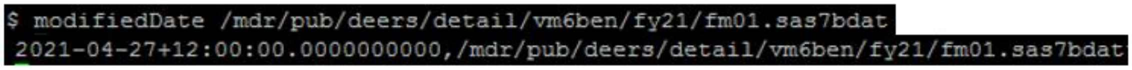
Submission Status

Name	Completed	Status Information	Run Time	Submitted
s3 Data Query	12/1/2022 11:23:23 AM	Successful	1.188s	12/1/2022 11:23:22 AM

Modified Dates

Tracking when a file was most recently modified can be a helpful tool in data forensics. With the transition to the new MIP AWS GovCloud environment, understanding modified dates is more complicated. The transition generated new modified dates for all files, even if the files haven't really been changed.

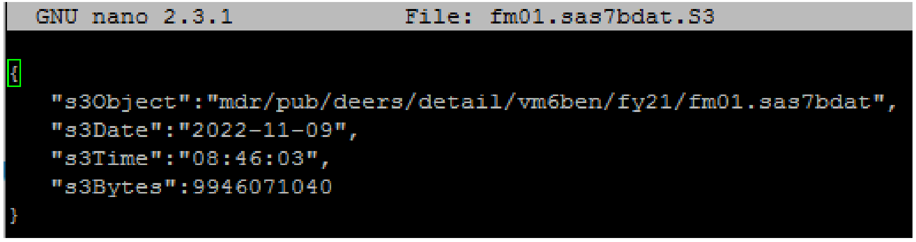
To determine when the file was most recently modified *before* the transition, use the "modifiedDate" function. This will return the date the file was most recently modified in the old DISA Capacity Services environment. The syntax is similar to the filesGet and other commands above. You must put the full path of the file:



```
$ modifiedDate /mdr/pub/deers/detail/vm6ben/fy21/fm01.sas7bdat
 2021-04-27+12:00:00.0000000000,/mdr/pub/deers/detail/vm6ben/fy21/fm01.sas7bdat
```

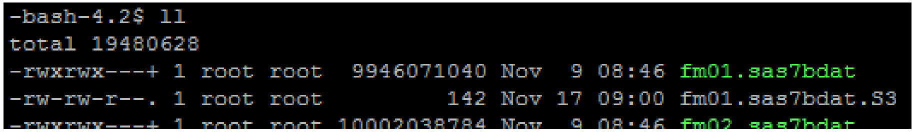
To determine when the file was most recently modified *after* the transition, you can do either of the following:

1) Open the metadata file (with the .S3 extension) and view the "s3Date". This is when the file was most recently modified in S3.



```
  GNU nano 2.3.1              File: fm01.sas7bdat.S3

{
   "s3Object":"mdr/pub/deers/detail/vm6ben/fy21/fm01.sas7bdat",
   "s3Date":"2022-11-09",
   "s3Time":"08:46:03",
   "s3Bytes":9946071040
}
```

2) If the file is on the SAS Server, you can simply observe the date using the ll command.



```
-bash-4.2$ ll
total 19480628
-rwxrwx---+ 1 root root  9946071040 Nov  9 08:46 fm01.sas7bdat
-rw-rw-r--. 1 root root         142 Nov 17 09:00 fm01.sas7bdat.S3
-rwxrwx---+ 1 root root 10002038784 Nov  9 08:46 fm02.sas7bdat
```

Notice that both options show a date of November 9. In this case, this represents when the file was initially loaded to S3 as part of the transition, but if the file is updated in the new environment, these dates will show when the update occurred.

Submitting SAS Jobs in the SCE -- SAS Job Queue

SAS jobs run in the background (by default) on the SCE to prevent running jobs from failing if the SSH session times out. You may create or edit SAS programs on the SCE using the NANO (PICO) or vi editor. If you prefer to work on your PC, you may edit your programs in Windows and then use a secure file transfer to copy them to the server. Please note that Windows may add hidden characters that cannot be included within the program.

Due to the high usage of the SCE, there is a limit of 3 programs per user running at a time. If you already have three programs running, you'll need to wait until one job finishes before setting off another SAS program. If more than 3 are submitted, the most recent submission will be 'in queue' until one of the prior programs finish and release resources.

Important: In order for programs to run, they MUST be named with a .sas extension. To submit a SAS job:

1. Log on to the server using a SSH client and change to the directory where the SAS program is located in your directory space.

   Example:

   ```
   [mipsapsasusrp01}:/home/med.ds.osd.mil/$USER > cd /hpae2/
   ```

   Note: Do not store code or data in your home directory, specifically /home/med.ds.osd.mil/$USER. Files in this area may be deleted without warning.

2. Remember, if the macro described in Keeping Logs is used (page 8), you must use the syntax provided in that section to submit your programs.

   ```
   sas <program_name.sas> -altlog <program_name.log>
   ```

Your program will be submitted to the jobQueue. The SCE can have up to 26 jobs running at a time. Any jobs after that enter a queue to be executed sequentially. Keep this in mind on busy days and try to only run jobs you need when you see jobs waiting.
You can see the queue by typing jobQueue --list.

Using Environment Variables

Any environment variable defined in the shell when the jobQueue is invoked will be passed to the SAS program. The jobQueue has key arguments, which can be found by executing jobQueue --help. These key arguments are specifically jobQueue options, all other arguments will be passed the SAS program. Passing a SAS argument can be done using the following:

```
sas <program_name> -<variable_name> <value>
```

SAS Temporary Workspace

There is one SAS temporary workspace area on the SCE, located here: /sas_data/saswork. When submitting a SAS job with no workspace argument, it will default to this location.

Commands for Submitting SAS Jobs

- To start a job running in the default workspace, the command is **sas *jobname.sas***
- To determine what jobs are running in the SAS Temporary Workspaces simply type **'jobQueue --list'** at the command prompt and look under "Queue" to determine the jobs running or queued to run in the

SAS Temporary saswork area.

Previously, there was a “sasbig” command and a --workspace option, but those are no longer recommended.

When you submit a SAS job, a temporary work folder will be created like this:
/sas_data/saswork/$USER/<jobPID>/
(see below for where you can find jobPID)

For example: /sas_data/saswork/khofmann/68971/

SAS Arguments Configuration

Memory

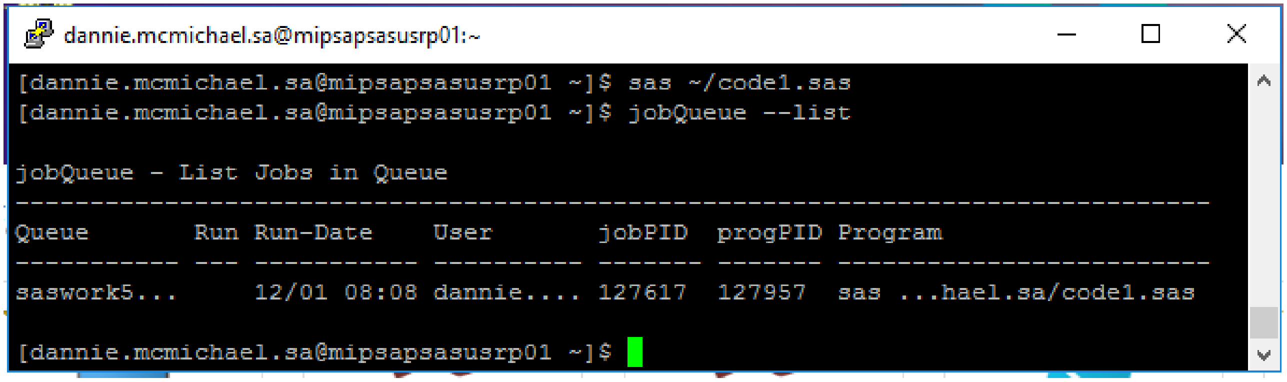
Users can control the amount of memory available to their SAS program. The default is 2GB and should be sufficient for most users. Using more than 2GB will impact performance for other users.

If necessary, use the -MEMSIZE option to make adjustments for a single use program run:

sas *jobname.sas -MEMSIZE 4G*

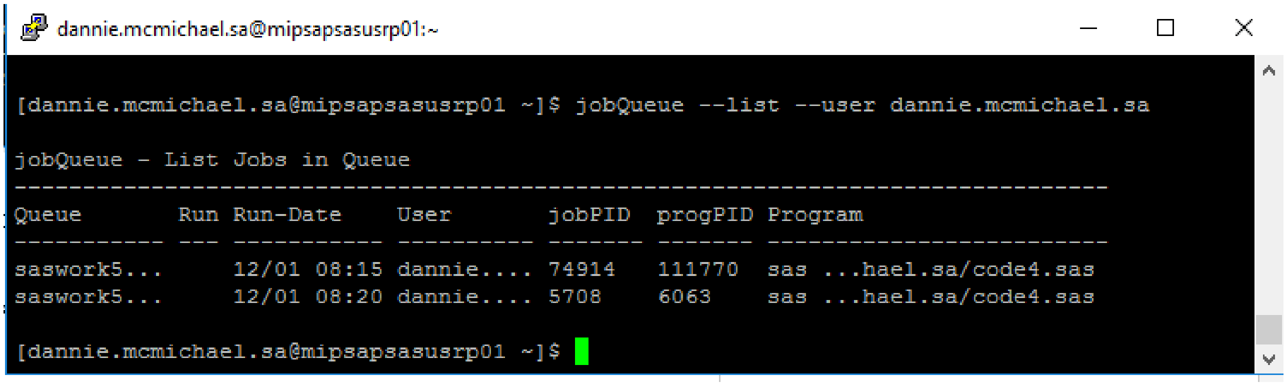
Checking Job Status

To find the current status of running jobs, type **'jobQueue --list'** at the command prompt.



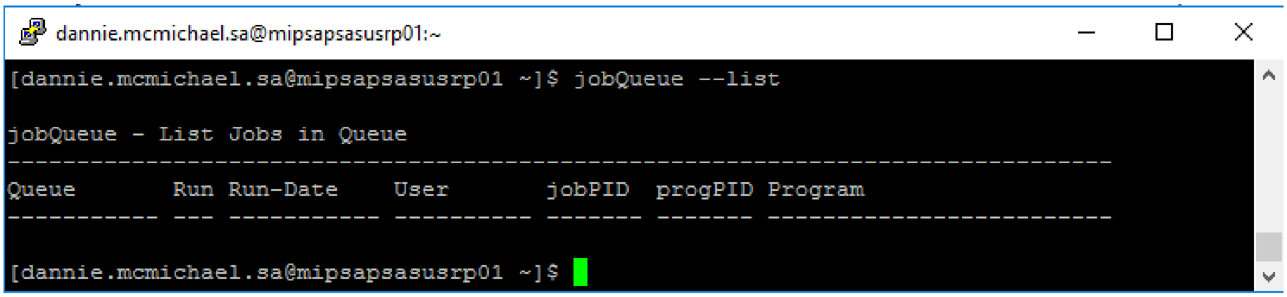
```
[dannie.mcmichael.sa@mipsapsasusrp01 ~]$ sas ~/code1.sas
[dannie.mcmichael.sa@mipsapsasusrp01 ~]$ jobQueue --list

jobQueue - List Jobs in Queue
--------------------------------------------------------------------------------
Queue       Run Run-Date    User       jobPID  progPID Program
----------- --- ----------- ---------- ------- ------- -------------------------
saswork5...     12/01 08:08 dannie.... 127617  127957  sas ...hael.sa/code1.sas

[dannie.mcmichael.sa@mipsapsasusrp01 ~]$
```

To find the current status of running jobs only for the user, type **'jobQueue --list --user $USER'** at the command prompt.



```
[dannie.mcmichael.sa@mipsapsasusrp01 ~]$ jobQueue --list --user dannie.mcmichael.sa

jobQueue - List Jobs in Queue
--------------------------------------------------------------------------------
Queue       Run Run-Date    User       jobPID  progPID Program
----------- --- ----------- ---------- ------- ------- -------------------------
saswork5...     12/01 08:15 dannie.... 74914   111770  sas ...hael.sa/code4.sas
saswork5...     12/01 08:20 dannie.... 5708    6063    sas ...hael.sa/code4.sas

[dannie.mcmichael.sa@mipsapsasusrp01 ~]$
```

All submitted jobs have completed when the output is empty:



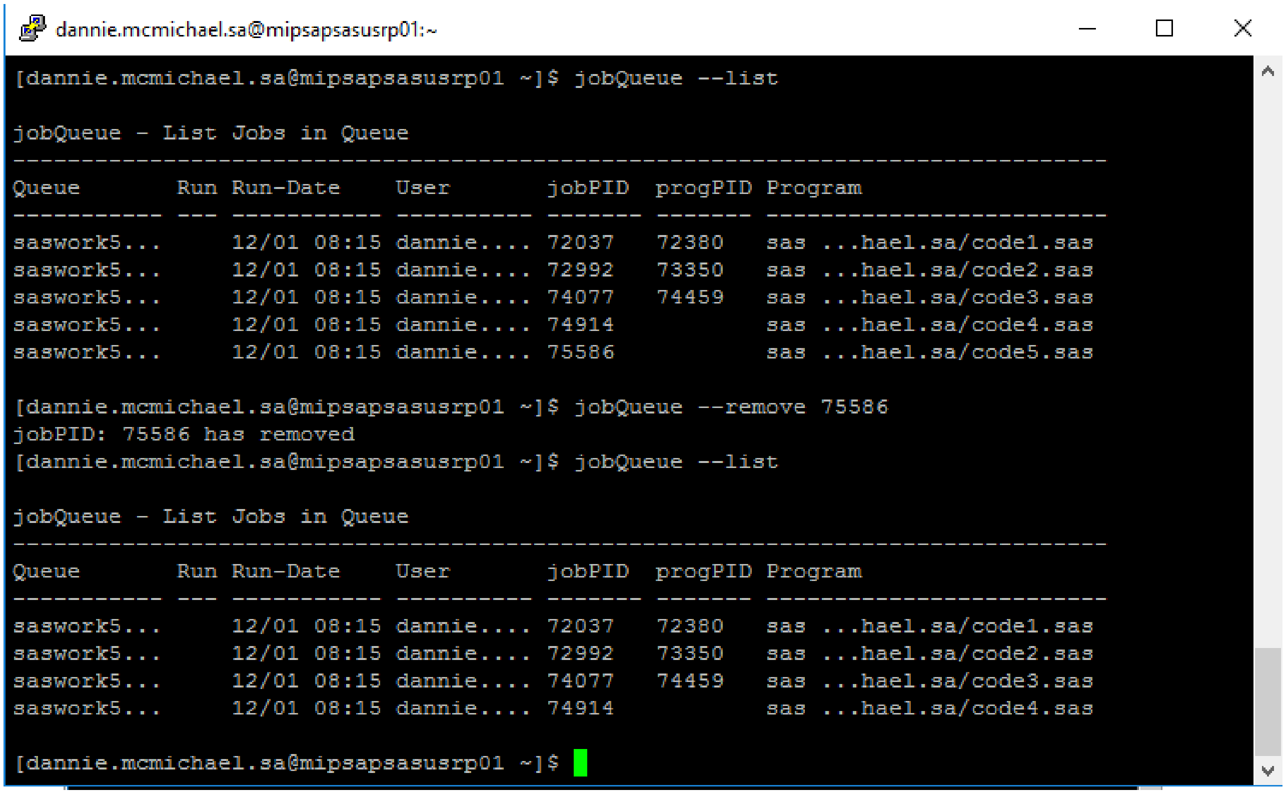
```
[dannie.mcmichael.sa@mipsapsasusrp01 ~]$ jobQueue --list

jobQueue - List Jobs in Queue
--------------------------------------------------------------------------------
Queue       Run Run-Date    User       jobPID  progPID Program
----------- --- ----------- ---------- ------- ------- -------------------------

[dannie.mcmichael.sa@mipsapsasusrp01 ~]$
```

Check the .log files for results.

Canceling a submitted Job -- When job is in queue or scheduled status, but before job is 'Running'

To cancel a submitted job, use the **jobQueue --remove <jobPID>** command. It will remove the job from the jobQueue.

1. List your jobs within the jobQueue: **jobQueue --list --user $USER**
2. Take the 'jobPID' of the job you wish to remove from the Queue: **jobQueue --remove <jobPID>**



```
[dannie.mcmichael.sa@mipsapsasusrp01 ~]$ jobQueue --list

jobQueue - List Jobs in Queue
--------------------------------------------------------------------------------
Queue       Run Run-Date    User       jobPID  progPID Program
----------- --- ----------- ---------- ------- ------- -------------------------
saswork5...     12/01 08:15 dannie.... 72037   72380   sas ...hael.sa/code1.sas
saswork5...     12/01 08:15 dannie.... 72992   73350   sas ...hael.sa/code2.sas
saswork5...     12/01 08:15 dannie.... 74077   74459   sas ...hael.sa/code3.sas
saswork5...     12/01 08:15 dannie.... 74914           sas ...hael.sa/code4.sas
saswork5...     12/01 08:15 dannie.... 75586           sas ...hael.sa/code5.sas

[dannie.mcmichael.sa@mipsapsasusrp01 ~]$ jobQueue --remove 75586
jobPID: 75586 has removed
[dannie.mcmichael.sa@mipsapsasusrp01 ~]$ jobQueue --list

jobQueue - List Jobs in Queue
--------------------------------------------------------------------------------
Queue       Run Run-Date    User       jobPID  progPID Program
----------- --- ----------- ---------- ------- ------- -------------------------
saswork5...     12/01 08:15 dannie.... 72037   72380   sas ...hael.sa/code1.sas
saswork5...     12/01 08:15 dannie.... 72992   73350   sas ...hael.sa/code2.sas
saswork5...     12/01 08:15 dannie.... 74077   74459   sas ...hael.sa/code3.sas
saswork5...     12/01 08:15 dannie.... 74914           sas ...hael.sa/code4.sas

[dannie.mcmichael.sa@mipsapsasusrp01 ~]$
```

Note: jobQueue --remove <jobPID> will only allow a user to remove their own submitted jobs.

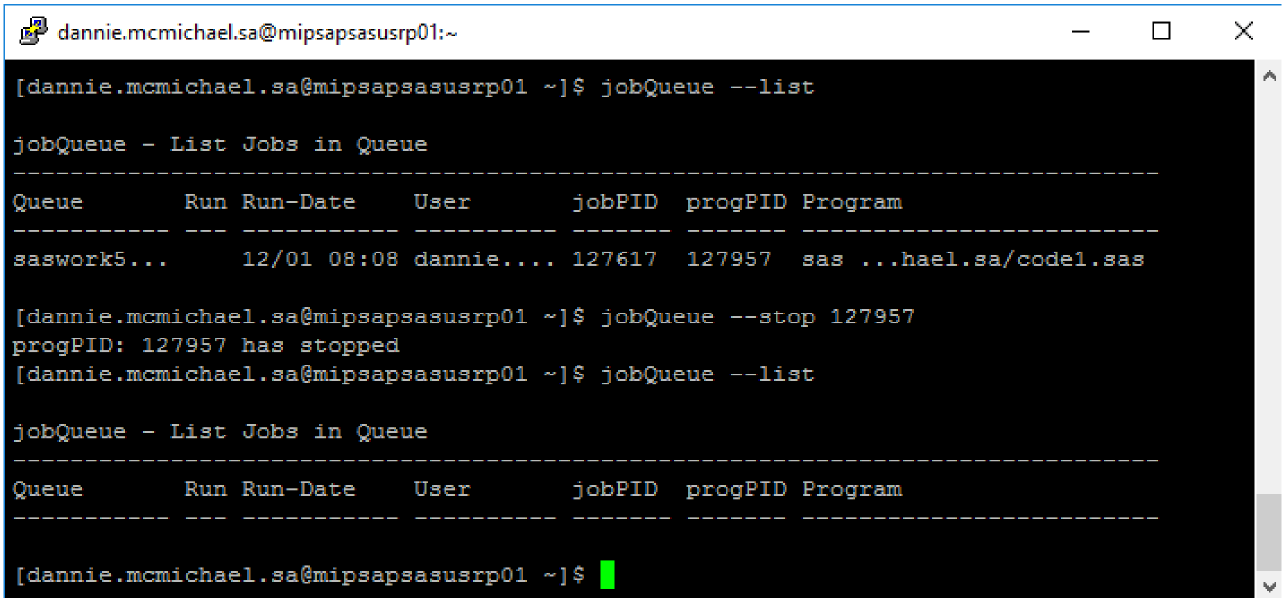
Note: Sometimes SAS jobs which terminate abnormally leave residual data within the temporary SAS workspace. This data will be removed by the system administrator on a routine basis.

Remember, do not store programs, scripts, data, formats, etc. under your user home directory: /home/med.ds.osd.mil/$USER/. There is very limited space available for the home directories. If this space becomes full, all users will be locked out of the SCE. If the system administration staff sees files in the home directories other than those placed there by the staff, they will be removed after due diligence.

Canceling a submitted Job -- When job is 'Running'

To cancel a submitted job, use the **jobQueue --stop <progPID>** command. It will remove the job from the jobQueue.

3. List your jobs within the jobQueue: **jobQueue --list**
4. Take the 'progPID' of the job you wish to remove from the Queue: **jobQueue --stop <progPID>**



```
[dannie.mcmichael.sa@mipsapsasusrp01 ~]$ jobQueue --list

jobQueue - List Jobs in Queue
--------------------------------------------------------------------------------
Queue       Run Run-Date    User       jobPID  progPID Program
----------- --- ----------- ---------- ------- ------- -------------------------
saswork5...     12/01 08:08 dannie.... 127617  127957  sas ...hael.sa/code1.sas

[dannie.mcmichael.sa@mipsapsasusrp01 ~]$ jobQueue --stop 127957
progPID: 127957 has stopped
[dannie.mcmichael.sa@mipsapsasusrp01 ~]$ jobQueue --list

jobQueue - List Jobs in Queue
--------------------------------------------------------------------------------
Queue       Run Run-Date    User       jobPID  progPID Program
----------- --- ----------- ---------- ------- ------- -------------------------

[dannie.mcmichael.sa@mipsapsasusrp01 ~]$
```

Note: jobQueue --stop <progPID> will only allow a user to stop their own submitted jobs

Scheduling SAS Jobs

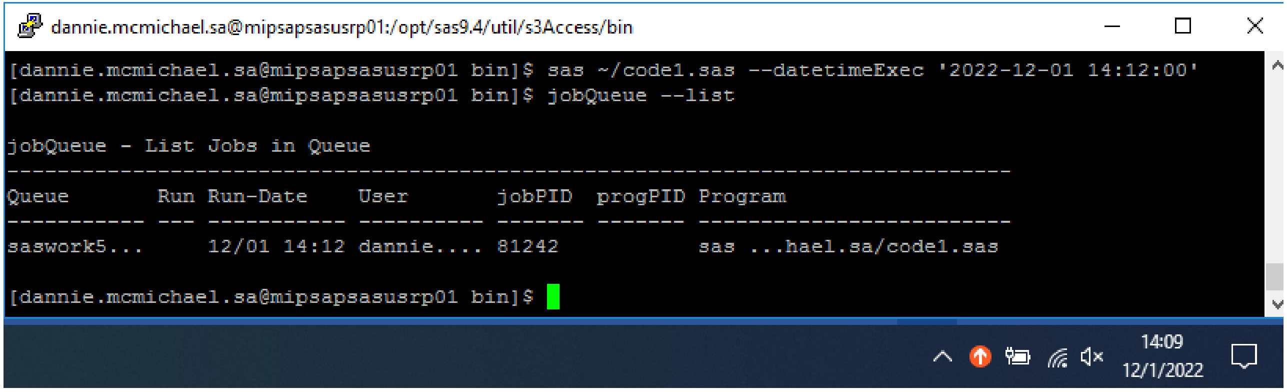
Background: There are two different types of scheduled jobs, one time execution and recurring. Depending on execution requirements, refer to the appropriate scheduling method.

One time execution:

A onetime execution utilized the jobQueue to schedule the job at the set time. The jobQueue will hold the job until execution time has been met, then releases the job into the queue. If there are other jobs exceeding the queue limits, the job will execute once resources are available.

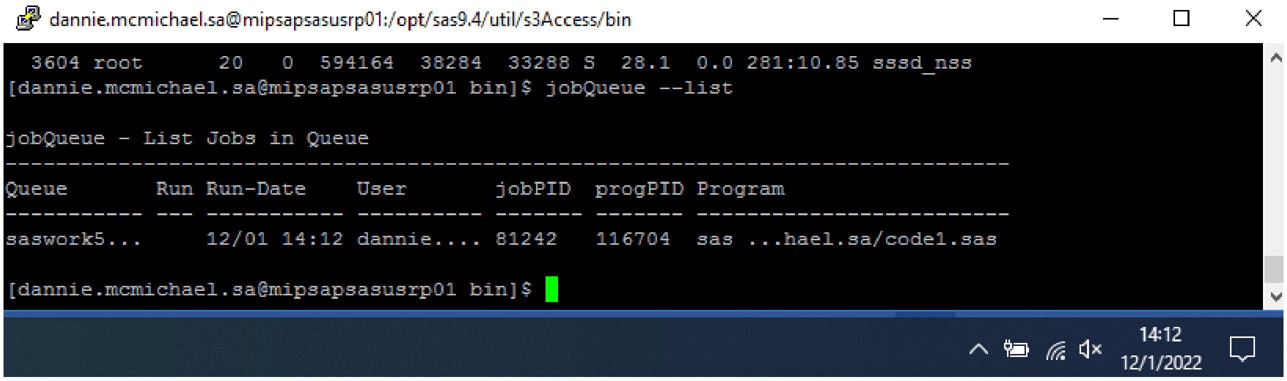
Syntax: sas /dir/file.sas --datetimeExec '2023-01-01 23:59:59'

jobQueue holding the job until datetimeExec:


dannie.mcmichael.sa@mipsapsasusrp01:/opt/sas9.4/util/s3Access/bin
14:09
12/1/2022

```
[dannie.mcmichael.sa@mipsapsasusrp01 bin]$ sas ~/code1.sas --datetimeExec '2022-12-01 14:12:00'
[dannie.mcmichael.sa@mipsapsasusrp01 bin]$ jobQueue --list

jobQueue - List Jobs in Queue
--------------------------------------------------------------------------------
Queue       Run Run-Date    User       jobPID  progPID Program
----------- --- ----------- ---------- ------- ------- -------------------------
saswork5...     12/01 14:12 dannie.... 81242           sas ...hael.sa/code1.sas

[dannie.mcmichael.sa@mipsapsasusrp01 bin]$
```

JobQueue release job at datetimeExec:


dannie.mcmichael.sa@mipsapsasusrp01:/opt/sas9.4/util/s3Access/bin
14:12
12/1/2022

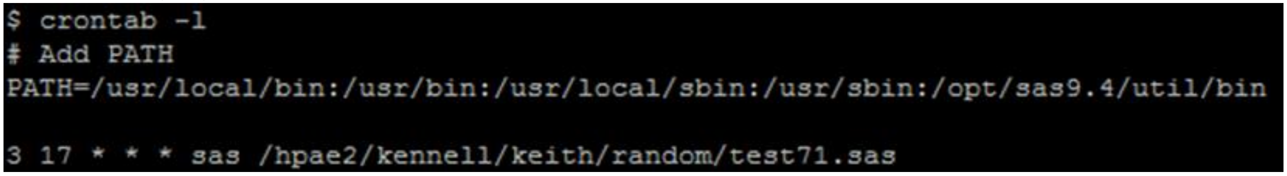
```
 3604 root      20   0  594164  38284  33288 S  28.1  0.0 281:10.85 sssd_nss
[dannie.mcmichael.sa@mipsapsasusrp01 bin]$ jobQueue --list

jobQueue - List Jobs in Queue
-------------------------------------------------------------------------------
Queue       Run Run-Date    User       jobPID  progPID Program
----------- --- ----------- ---------- ------- ------- -------------------------
sasworkS...     12/01 14:12 dannie.... 81242   116704  sas ...hael.sa/code1.sas

[dannie.mcmichael.sa@mipsapsasusrp01 bin]$
```

Recurring execution:
Crontab is a Linux service that allows users to execute commands at specific times.
Crontab Man: https://linux.die.net/man/5/crontab

A recurring execution requires the use of cron. A user can edit their cron by using the following command: **crontab --e.** Users of cron tab need to consider that most MDR data is refreshed weekly or monthly. Please do not run daily cron jobs unnecessarily.

PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/opt/sas9.4/util/bin

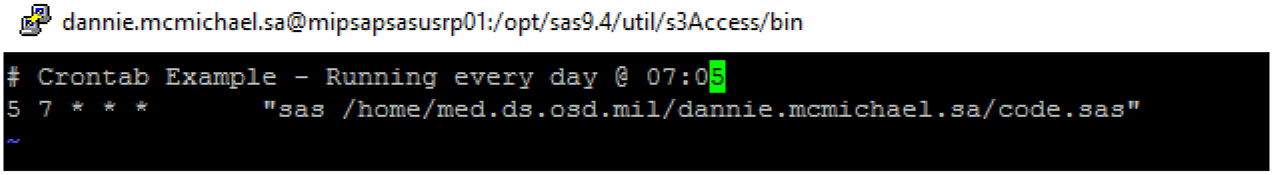
SAS Scheduled entry:
e.g. Cron scheduled task to run every hour at *42* minutes past the hour.
42 * * * * sas /dir/file.sas



```
$ crontab -l
# Add PATH
PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/opt/sas9.4/util/bin

3 17 * * * sas /hpae2/kennell/keith/random/test71.sas
```

Reference the crontab manual for help creating a scheduled execution. When done creating the entry, **:w** will write the entry to crontab and **:q** will quit crontab.

Note: Your processes are dependent on Kerberos authentication. Kerberos operates by validating users and issuing tickets. These tickets expire and requires to re-authenticate. Periodically logging into the SCE server will refresh your Kerberos ticket.


dannie.mcmichael.sa@mipsapsasusrp01:/opt/sas9.4/util/s3Access/bin

```
# Crontab Example - Running every day @ 07:05
5 7 * * *        "sas /home/med.ds.osd.mil/dannie.mcmichael.sa/code.sas"
~
```

Moving Files to the SCE from External Systems: Data Imports and Extractions

SCE users may periodically require files be made available on the server that only exist on external systems or export data from the MIP. Please use the following link to find the data import and export transmittal forms.
https://health.mil/Military-Health-Topics/Technology/Support-Areas/MDR-M2-ICD-Functional-References-and-Specification-Documents

Commonly Used UNIX Commands

Following are some of the basic commands that you will need in the UNIX environment.

Command Reference Table

ls

The “ls” command lists files in a directory. There are many options you can use with the "ls" command. For example "ls -a" will list ALL of the files in a directory (files and directories whose name begin with a period are considered hidden files). "ls -al" will list ALL files in LONG format. Here is an example directory listing:

```
www:~> ls -laF
total 34
drwxr-xr-x 4 root  other   2048 Sep 24 11:25 ./
drwxr-xr-x 102 root  root   2048 Sep 24 11:24 ../
-rw-r--r-- 1 root  other   384 Jan 6 1998 .bashrc
-rw-r--r-- 1 root  other   383 Jan 6 1998 .cshrc
-rw-r--r-- 1 root  other   706 Jan 6 1998 .defaults
-rw-r--r-- 1 root  other   1858 Jan 6 1998 .emacs
-rw------- 1 root  other   2797 Jan 6 1998 .history
-rw-r--r-- 1 root  other   137 Jan 6 1998 .login
-rw-r--r-- 1 root  other   138 Jan 6 1998 .logout
-rw-r--r--        1 root        other        232  Jan  6  1998  .mailrc  drwx------
                  2 root        other        512  Jan  6  1998  Mail/  drwxr-xr-x
                  2 root        other        512 Jan 6 1998 bin/
```

In this example, the first block of each row (-rw-r--r--) shows the type and permissions for an individual file within your current directory. The first column tells you the type of file: '-' means it is a normal file, 'd' means it is a directory, and an 'l' means it is a link to another file (much like a shortcut in Windows). The next nine positions show the file's Read, Write, and Execute permissions in blocks of three. The first three characters (often 'rw-') are the permissions for the owner of the file, the second three are group permissions, and the final three are permissions for anyone else besides the owner and anyone in the group.

The next column is a number representing how many physical links to the directory or file exist on the file system. Files generally have one link to them due to the fact that directories cannot link to files specifically. Notice that the directory up one level, '..', contains a large number of links (102) due to the number of home-directories that reference it as '..'.

The next four columns contain the name of the owner of the file (equivalent to the first three 'rwx' permissions), the group owner (second three 'rwx'), the file size, file date, and file name. Note that by using the '-F' flag (must be a capital F), directories are marked with the / symbol, and the / is not actually part of the directory name. Also, files marked as executable are shown with the ‘*’ symbol following the name (only when the -F flag is set).

cd

The “cd” command is used to change directories. The "cd /" will take you to the root directory of the directory tree; "cd /usr" will take you to the directory named 'usr' that is located off of the root directory; "cd mail" will take you into a directory called 'mail' off from your current directory (note the lack of ‘/’ before the directory name); and "cd .." will take you to the parent directory in the directory tree. Some command-line shells, such as "bash" or "tcsh" expand the ~ symbol to mean a person's home directory. You could then either type "cd ~" to change back to your home directory, or "cd ~another user" to go into another user's home directory.

pwd

The “pwd” command prints/outputs your present working directory to standard output.

mkdir, rmdir

These commands are used to create and remove directories. The "mkdir class" will create a directory called class in your current directory. Whereas "rmdir class" will delete the directory named class. Note that "rmdir" only works on directories that do not contain any files.

cp

To copy a file, use the “cp” command. The syntax is "cp *source destination*", where *source* is a source file and *destination* is either a file name or directory in which to place the file. For example, "cp /usr/local/seminar.c ." will copy a file located in '/usr/local' named 'seminar.c' to '.' which is the symbol that stands for your current directory. "cp /usr/local/seminar.c my.c" will copy the ‘seminar.c’ file to your current directory under the name 'my.c'.

mv

The “mv” command is used to move *and/or* rename a file. "mv my.c first.c" will rename a file called 'my.c' to 'first.c'. Assuming that 'class' is a directory, "mv my.c class" would move the file 'my.c' into the ‘class’ directory. If 'class' is a file, this operation will instead delete 'class' and then rename 'my.c' as 'class', which is probably something you didn't want. You can also rename and move at the same time. For example, "mv my.c class/first.c" would move the file 'my.c' into the 'class' directory and rename it to 'first.c' all in one step.

rm

The “rm” command deletes, or removes, files from the files ystem. Typing "rm my.c" would delete the file 'my.c'. You can also use wildcards. For example, "rm my*" will delete every file whose name started with the word 'my', including 'my.c'. Unlike Windows, these wildcards can appear anywhere in the file name, and do not need to be confined to the standard 8.3 notation used by MS-DOS. Directories and their contents can be deleted by using the '-r' flag which stands for recursiveness. For example, "rm -r class" would delete ALL files and directories under 'class' and 'class' itself regardless if the directory is empty or full.

cat, less, more, head, tail

These commands are used to view the contents of files. "cat" displays the entire file on the screen, regardless if your terminal can view everything at once (the top portion scrolls off, but can be retrieved by scrolling back in a window or sometimes by holding down the 'Shift' key and pressing 'Page Up'). "less" and "more" both display the contents of a file one full screen at a time. "head" is used to display the first few lines of a file, and "tail" is used to display the last few lines of a file. For example, "tail -25 my.c" displays the last 25 lines of the file 'my.c'. "cat first.c" would display all of 'first.c', and "more first.c" will display the file 'first.c' one screen at a time.

grep

Search a file for a particular pattern by using the “grep” command. "grep" stands for Get Regular Expression Pattern. This command searches a file for all lines matching a particular pattern and prints those matching lines on standard output. For example, "grep fun my.c" prints every line of the file ‘my.c’ that contains the word 'fun'.

man

The “man” command brings up the UNIX Manual Pages for a particular command. For example, to learn about the rest of the options for the "ls" command, type: "man ls"

w, who, whoami, id

These commands are used to find out the list of users currently logged into the system and what they are doing. "w" provides slightly different output than "who". "whoami" returns the username of the person currently logged in. "id" prints your User ID, and Group ID, and a list of all the groups you have access to.

chmod

The “chmod” command is used to change the permission bits, called mode, on a file or directory. Only the owner of a file or directory and the superuser can change the permission bits/mode. See the “ls” command for information on viewing a file or directory mode.

Every file and directory is assigned nine permission bits that comprise the mode. The mode is used to determine what operations may be performed on a file and by whom. The mode consists of Read, Write, and Execute bits in blocks of three (Execute is also called “search” when referring to a directory): one block for the owner of the file or directory, one block for the group that the file or directory belongs to, and one block for others. Each user will fit into only one of the three blocks/categories.

For files, the read bit allows the file to be opened and read. The write bit allows the contents of the file to be modified or truncated. The execute bit allows an executable file to be executed.

For directories, the combination of read and execute (aka “search”) bits allows the contents of the directory to be listed. The combination of write and execute bits allows files within that directory to be created, deleted, and renamed. Having only the execute bit will allow the directory to be entered but not have any of its contents listed.

File or directory modes may be changed using octal notation or symbolic format. The octal notation format is described below. The basic syntax for the “chmod” command is:

```
chmod <octal or symbolic notation of read, write, execute permissions> <file or directory name>
```

The octal notation format sets file or directory permissions based on three octal numbers. The first number defines access for the user, the second number for the group, and the third number defines access for others. Each of the three numbers is a combination of permission bits. The octal notation is between 0 and 7.

Use the UNIX command CHMOD to change access permissions by summarizing the numerical values for Owner, Group and Other.

Digit	Permissions
0	None
1	Execute only
2	Write only
3	Write and execute
4	Read only
5	Read and execute
6	Read and write
7	Read, write, and execute

Examples:
#1) chmod 770 tedn.sas - allows owner and group to have read/write/execute privileges, other no privileges
#2) chmod 764 tedn.sas - allows owner read/write/execute, allows group read/write only, allows other read only

Using the “ls --l” command, example #1 and #2 above would appear as such:

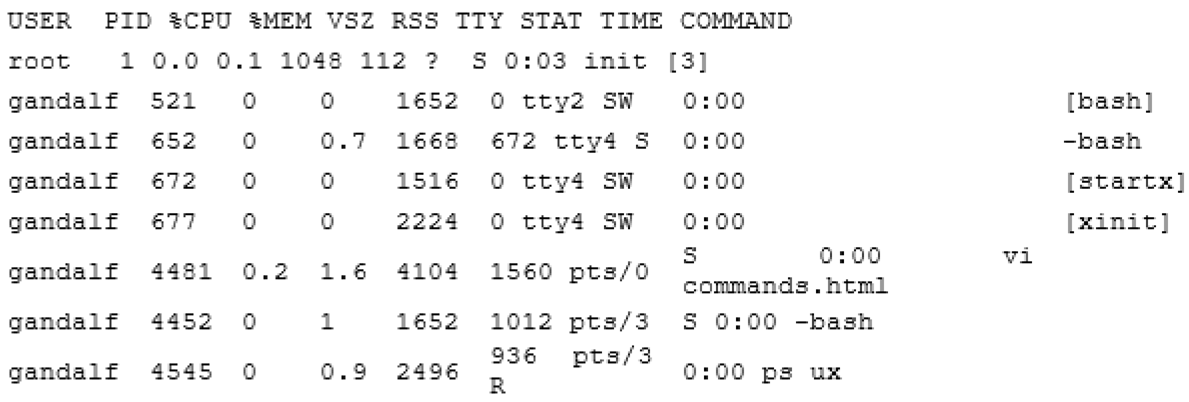
```
#1) rwxrwx---      example1 groupname   2567  Sep 10 12:15 tedn.sas
#2) rwxrw-r--      example1 groupname   2567  Sep 10 12:15 tedn.sas
```

ps

The “ps” command is used to find the Process Status of all processes currently running on the machine. Typing "ps" without any options lists all programs currently running in your process group. “ps ux” or "ps -ef" lists all of the processes on the machine (Some UNIX versions prefer one method over the other. If one way does not work, try the other). Combining the "ps" and "grep" commands through a pipe can give you a list of all the processes you have running, and can be done like this:

```
ps aux | grep <yourusername>
```

The '|' symbol, called a pipe, takes the results of the first command (in this case, "ps -aux") and uses it as input to the second command ("grep yourusername"). Let’s take a look at an example:



```
USER   PID %CPU %MEM VSZ RSS TTY STAT TIME COMMAND
root    1 0.0 0.1 1048 112 ?  S 0:03 init [3]
gandalf  521   0    0    1652  0 tty2 SW   0:00                      [bash]
gandalf  652   0    0.7  1668  672 tty4 S  0:00                      -bash
gandalf  672   0    0    1516  0 tty4 SW   0:00                      [startx]
gandalf  677   0    0    2224  0 tty4 SW   0:00                      [xinit]
gandalf  4481  0.2  1.6  4104  1560 pts/0  S       0:00        vi
                                           commands.html
gandalf  4452  0    1    1652  1012 pts/3  S 0:00 -bash
                               936  pts/3
gandalf  4545  0    0.9  2496  R           0:00 ps ux
```

The first column contains the user ID of the owner of that process. The second column contains the PID, or Process Identification number. The next two columns show the percentage of CPU Time and Memory the process has consumed during its entire running life. The next two columns show the Virtual Size of the program (total size of all memory in use) and how much of that memory is currently resident in the computer at the moment. Thus, the RSS number is usually much less than VSZ, because data are either shared, or is marked unused and are swapped to disk to free up room for other processes to use resident memory.

TTY shows which terminal the process is currently running on. Usually, tty1 through tty6 are processes controlled by a user at the main console (ALT-F1 through F6 to switch), and pts/# are processes from either inside X-Windows or through telnet in the network. STAT corresponds to the type of process status running at the time: 'R' means the process is currently active (as is the case for our “ps” command gathering the process data), 'S' means the process is sleeping, 'W' means the process is waiting for a signal to trigger swapping from disk and back into memory (note the 0 Resident Set Size), and 'Z' stands for a zombie process (see below). TIME shows the total amount of time the process has taken up CPU. In this case, each process except 'init' has taken up trace amounts of CPU, while 'init' has taken 3 seconds since the computer was started.

The last column of the "ps" command output contains the name of the process. The name can be represented in several different ways depending on the state of the process. If the name begins with a '-' symbol, such as '-bash', then it is recognized as a login shell. If the name is surrounded by [ ]'s, then the process is currently swapped out of memory to disk (with an RSS of zero).

Zombie Processes. Processes marked 'Z' occur during a coding error on behalf of the programmer to not properly *wait* for one of its children to exit after a *fork* call in the C language ("man" pages are available for both of these functions). Such processes are marked <defunct> and will only terminate when their parent process (PPID) exits.

kill, killall

Force a process to exit or terminate by using the “kill” command. This command is used to send signals to processes, most commonly the signal to kill a process (hence its name). Looking back at the "ps" example, to kill the "vi" process, you can type "kill 4481". Only programs that do not block the Terminate signal can be killed in this fashion. You instead must specify the “kill”
command using a '-9' as the first argument to the prompt. For example, to kill the unused,
swapped out 'bash' process, type: kill -
9 521

The "killall" command behaves in the same way as "kill", only each process matching the name you give it will be sent the signal. To kill all of your "bash" programs (and consequently kick yourself back to the login prompt), type:

killall -9 bash

Note: If you own a process that is using close to 99% of the CPU, it is generally a good idea to "kill -9" it, as it most likely won't respond to normal signal handling. The system PID 1, "init", can never be killed.

Note: Do not use kill to kill a SAS job, use the jobQueue --stop or
jobQueue --remove commands.

compress

The “compress” command compresses data using adaptive Lempel-Zev coding to reduce the size of files. When possible, a compressed file replaces the original file, specified by the file parameter, with a .Z appended to its name. The compressed file retains the same ownership, modes, and modification time of the original file. If no files are specified, the standard input is compressed to the standard output. If compression does not reduce the size of a file, a message
is written to standard error and the original file is not replaced. Files being compressed must have
correct permissions to be replaced. For example, to compress the ‘foo’ file and write the percentage of compression to standard error, type:

```
compress -v foo
```

The foo file is compressed and renamed foo.Z.

For more information about different options and techniques on the formation of the ‘compress’ command, reference the man pages.

uncompress

The “uncompress” command restores original files that were compressed by the “compress”
command. Each compressed file specified by the file parameter is removed and replaced by an uncompressed copy. The uncompressed file has the same name as the compressed version but without the .Z extension. If the user has root authority, the expanded file retains the same owner, group, modes, and modification time as the original file. If the user does not have root authority, the file retains the same modes and modification time, but acquires a new owner and group. If no files are specified, standard input is expanded to standard output. For example, to uncompress the foo.Z file, enter:

```
uncompress foo.Z
```

The foo.Z file is uncompressed and renamed foo.

Linux File Naming Conventions

Absolute Naming

Within the UNIX directory structure there are two ways to name any file: relative naming and absolute naming. An absolute name, or absolute path as it is often called, specifies exactly where in the directory the particular file is. It tells the whole name of the file, starting at the root of the directory. An absolute name starts with /, the root, and names each directory along the path to the file, separating the directories with /. This is very similar to DOS, except for the direction of the slash and the fact that there is no disk drive designation. As an example, the absolute name for your mailbox might be /home/iris2/class9/mbox. The ‘pwd’ command always reports an absolute pathname.

Relative Naming

The second way to name a file in UNIX is with a relative name. Whereas an absolute name specifies exactly where in the directory a particular file exists, a relative name specifies how to get to it from your current directory. The look of a relative name may vary a lot. Depending on your starting directory there are a number of paths to a particular file.

In the simplest case, just naming a file in your current directory is a relative name. You are specifying how to get to this file from your current directory, and the path is to just open the file in the current directory.

When using relative paths, the special directories ‘.’ and ‘..’ that are contained in every directory are used frequently. Recall that ‘.’ specifies the directory itself, and ‘..’ specifies its parent. So, if the file mbox is contained in your current directory, naming the file with ./mbox and mbox are equivalent. The special directory '..' is used to name a directory at the same level in the tree as your current directory, that is, a sibling of your current directory. The following example illustrates using '..' to look at a sibling directory.

```
% pwd
/home/iris2/class9
% ls
NewBibligraphy.refer          bibs                mbox
ShortFile               bin                 src
baby.1                 ig.discography             unix.refer
% cd bin
% ls
pwgen
% ls ../src
helloworld.c        pwgen.c
```

Short-cuts for File Naming

Since certain file naming patterns are used over and over, UNIX provides some short-cuts for file naming. Actually it is the shell that provides these, but the distinction is not critical at this point. In particular, many file accesses are either to your own home directory or to the home directory of another user. To make it easier to point to these places, the ~ character is used. Alone, ~ refers to your home directory. So, the file ~/mbox refers to the file mbox in your home directory. Likewise, ~ username refers to the home directory of that particular user. So,
~dickson/mbox refers to the file mbox in the user dickson's home directory.

File Naming Limitations

Linux allows you great flexibility in naming your files. Many systems allow the name of individual files to be up to 256 characters, and the maximum absolute pathname to be about 1023 characters. This means that files can be given meaningful names quite easily. Also, since Linux is sensitive to the case of the file names, you can use mixed-case names to add clarity. While long, meaningful names can be quite nice, it is also possible to go overboard with file naming. So, try to give your files meaningful names, but try not to overburden yourself or others that have to access the files with overly long names. Since this environment works with AWS S3, file/directory naming should exclude special characters including spaces. Linux may be case sensitive, but to avoid file naming contention with AWS S3, it is recommended not to rely on case sensitivity to distinguish between two different files.

File Name Extensions

While Linux does not actually have the same notion of a file extension as is found in some other systems, notably DOS, many user and applications programs behave as it did. Linux does not consider the '.' character any differently than any other character in a file name. However, applications programs often use it to add an extension to a file name that specifies something about the contents of the file. These extensions sometimes tell what programs were used to create a file. If more than one was used, and is needed to decode the file, multiple extensions are used. Typically, the extension will tell what language the data in the file is in. A good example of this is the C compiler. It assumes that its input files will have an extension of '.c'. Its executable output generally has no extension, unlike DOS which uses an .EXE extension for executables. If we have a program called hello.c, its executable version is likely to be called just hello.

The Linux Directory Tree

Files in Linux are stored in directories, which are organized into a tree type configuration. At the top of the tree is the root, called /.

Locations in the directory tree are called paths and are specified as a list of directories separated by a '/'. Barney’s home directory, for instance, is in the home directory, which is in the root directory, and therefore the path is "/home/med.ds.osd.mil/barney".

If you leave off the leading "/", the path is assumed to start in the directory you are currently in, for instance if you are in "home", "barney/songs" would be the directory songs contained in Barney's home directory. You can also use "..", which specifies the parent directory of the one you are in "../etc" from home would be the "/etc" directory.

There are a few standard directories you will find in pretty much all UNIX systems.

/bin This contains the basic system commands.

/etc This contains system configuration files and programs used for administrating the system.

/lib This contains the system libraries.

/tmp This is used to store temporary files.

/usr/bin This contains all the commands not in /bin.

/usr/man This contains manual pages for programs.

/usr/local This contains local programs, that were installed by your sysadmin and not included with the original system. In particular, /usr/local/bin is local binaries and /usr/local/man is local man pages.

/home The location of this varies greatly from system to system, but somewhere on the system will be a place where all the home directories are, including yours.

Getting Around Directories

You have been given a directory in which you can store your personal files, called your *home directory*. When you log in to the system, this is where you start off. You probably don't have any directories to start off with. If you want to create a directory underneath your home directory, use the *mkdir* command. For instance, to create a directory to hold your programs, type ‘*mkdir programs*’. Please remember that ‘~/’, your home directory, is not meant to store data, any files within your home directory may be removed without notice. This does not affect linked directories like ‘~/MIPFILE’ or ‘~/mdrfile’.

Now you have a directory you can get into with the *cd* command. For instance, to get into your *programs* directory, type ‘*cd programs*’. To get back to your home directory from anywhere, simply type ‘*cd*’. To go up a directory, type ‘*cd ..*’.

If you want to copy or move a file into a directory, use the *cp* or *mv* command as normal, but instead of specifying a destination file, specify a directory. For instance, if you type ‘*cp helloworld.c programs*’ from your home directory, it will copy the file *helloworld.c* into your *programs* directory.

To remove a directory, use the *rmdir* command. Note that the directory must be fully empty to do this. To remove your *programs* directory, change to that directory using the ‘*cd*’ command and type ‘*rm **’. This deletes all files in that directory so make sure you are in the right place! You can find out where you are by typing ‘*pwd*’. Now that all files have been deleted, you can delete the directory. Go back to your home directory and type ‘*rmdir programs*’.

Appendix A: The NANO Editor (For editing in Linux, a clone of PICO) and VI editor

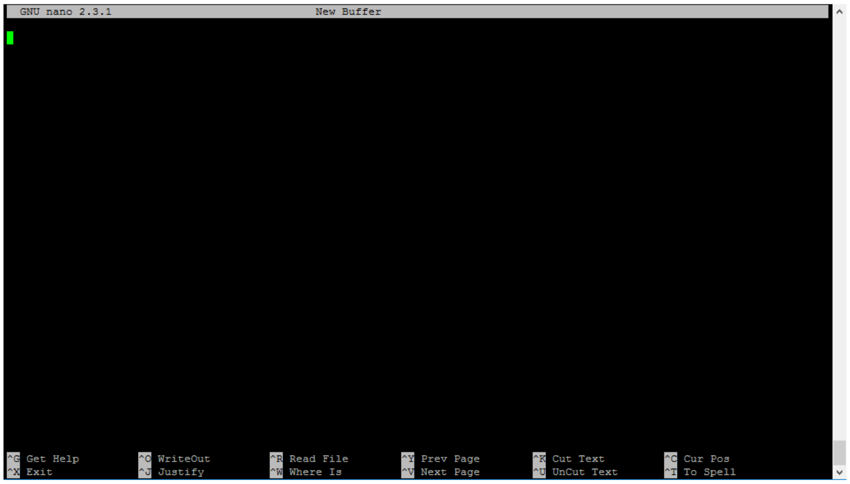
NANO Editor

(https://www.nano-editor.org/dist/v2.2/nano.html)

NANO is a line-mode text editor used to create files, reports and letters. You will need a line- mode editor if you use your account on a non-X Windows computer.

To start using NANO, enter ‘nano filename‘ or ‘pico filename’ at the system prompt where filename is either the name of the file you want to edit (one that already exists) or the name of a new file you want to create. If you do not provide a file name, NANO will open a new file, but it will be unnamed until you save it.

The screen for a new file will look something like this:



```
  GNU nano 2.3.1                       New Buffer

^G Get Help     ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit         ^J Justify      ^W Where Is     ^V Next Page    ^U UnCut Text   ^T To Spell
```

In the list of commands at the bottom of the NANO screen, caret (^) stands for the control (ctrl) key. ‘^G’ means to hold down the control key and press the letter ‘G’. NANO commands are always a combination of the control key and another key. After you enter a NANO command you do not need to press the return key.

Quitting NANO

Press ‘^x’ to quit or exit the NANO editor. NANO will prompt you with a message asking if you want to save your changes. ‘Y’ saves the changes and ‘N’ quits NANO without saving what you've done since the last time you saved.

Entering Text

At the system prompt, type ‘nano’ to create a new file. You should see NANO in your window. The top line of your screen is called the status line. It shows what version of NANO you are using, the file name (it will say New Buffer if it hasn’t been named yet), and whether or not you have made changes (Modified will be in the upper right corner of the NANO screen if something has been entered but not saved). At the bottom of the screen are two lines showing the commands available for your use.

Now that you have an empty file, you can start typing. NANO automatically starts a new line when you run out of room (called "wrapping text," "text wrap," or "autowrap") so you do not need to watch the screen while you type. Press [return] when you want to start a new line. If you want a blank line between lines of text or data, press [return] twice.

Saving the file

Now that you have a file with something in it, this is a good time to save your work. You should save your files often, if for no other reason than strange things can happen and sometimes do. The commonsense rule is to save often enough so that if you lose your work, you will not have much to re-do. Some people recommend saving every 15 minutes.

To save a file, use '^o', which stands for write out because you are writing what you have done to a file as output. When you press '^o', NANO will prompt you for the file name. If the file does not have a name yet, you will need to type one in. If the file does have a name, it will follow the file name prompt, and all you need to do is press [return].

You may also save the file during the process of quitting NANO. '^x' quits NANO after prompting you to save the file. Press 'Y' to save the file. Pressing 'N' will return the file to the state it was in the last time you saved it (if you had not saved the file during this editing session, the file will return to the state it was in when you opened it). If you alter a file and want to save the original version as well as the modified version, enter 'Y' to save the file. When you are prompted for a file name enter a new file name.

Editing in NANO

You can easily edit short files using only the following keys:

- [ret]: Inserts a new line in the file.
- cursor (arrow) keys: The four arrow keys move the cursor up, down, left, and right.
- backspace/delete key: Deletes the character preceding the cursor.

Cutting text

Cutting or deleting text is a major component of any kind of editing. NANO gives you two ways to cut text: (1) cut a character at a time, or (2) cut a line at a time (unfortunately, you cannot cut a word at a time).

To delete a character, place the cursor on the character you want to delete and press ^d. Pressing your keyboard's delete key (some keyboards label it backspace) will delete the character preceding the cursor.

To delete an entire line, place the cursor anywhere within the line and press '^k' (for kill or cut text). The entire line will be cut. If you change your mind about deleting the line before you delete another one, you may press '^u' to bring back the line. The restored line will be placed wherever the cursor is.

Pasting text

Since '^u' lets you restore the cut line wherever the cursor is and not just to wherever it came from, you may use '^k' and '^u' to "cut and paste" text. In NANO, you move blocks of text similar to if you were using a word processing package. You first need to mark (highlight) the text you want to move. Cut the text, move the cursor to where you want the text inserted, then paste it. Position your cursor at the beginning of the first line of the text block you want to move. Press
'^^' (ctrl-shift-6 key). Move the cursor to the end of the block you want to cut. The text should then be highlighted. Once the block is highlighted, press '^k' to cut the text. Place the cursor to
where you want the text placed, and press '^u' to paste. The text should then appear.

Reformatting text

To justify or reformat a paragraph (fill in gaps left by starting a new line) press ‘^j’. NANO defines paragraphs as text surrounded by blank lines or indentation. However, if you had a blank line between each line of text pressing ‘^j’ would not affect the text.

If you do not like the result of the justification, press ‘^u’ to "unjustify" the paragraph before you move the cursor outside the paragraph. Once you have justified text, the ‘^u’ command function changes from undelete to unjustify. However, when you move the cursor from the justified paragraph, the ‘^u’ command function returns to undelete. Therefore, if you need to unjustify the paragraph, you must do so before you move the cursor outside the paragraph.

To join lines that are separated by blank lines use ‘^k’ to delete the blank line and ‘^j’ to fill the paragraph.

Searching a NANO file

If you are proofreading a file by looking at a printout, it may be tricky to find the specific line you want to revise. An easy way to get to the right point in your file is to use NANO’s search, or ‘where is’ command, which is ‘^w’. This command lets you enter a character(s) or word(s) to search for, and then moves the cursor to the first occurrence. NANO begins the search from the point in the file where the cursor is located. So, if the cursor is at the beginning of the second paragraph, the search begins at the beginning of the second paragraph.

To find a specific character or word, press ‘^w’ (for where is). A black strip with the word ‘search:’ will replace the two lines of commands at the bottom of the screen. You should then type the character(s) or word(s) you want to find. This search command is not case sensitive. This search command is also not limited to finding only whole words. For example, if you were to tell NANO to find “th”, it would find “that”, “the”, “they”, and “thick”, etc.

Using the spellchecker

NANO has a spellchecker that will check spelling in your text files.

To start the spell checker, press ‘^t’. The NANO spellchecker compares the words in your file to words that are in its dictionary. When it finds a word that is not in its dictionary, it sends the prompt ‘Edit a replacement:’ followed by the misspelled word. If the word is correct, just press return. If the word is misspelled, you may change it by backspacing over it (to delete it) and then typing the correct word. You may also use the cursor and delete keys to edit parts of the word.

Inserting a file into NANO

To insert an existing file into the NANO file you are currently editing:

- Locate the cursor where you want to place the inserted file
- Press ‘^r’. After you press ‘^r’, you will receive the prompt ‘Insert file:’
- Type in the name of the file you want to insert and press return. The inserted file's contents will be inserted at the cursor.

This command does not delete or in any way alter the file that is to be inserted (a copy of the file is inserted).

If you change your mind about inserting the file after you have pressed ‘^r’, when prompted for the file name enter any character string that does not correspond with a file. The system will attempt to process the command, but when it does not find a corresponding file it will cancel the command and your text will not be altered. You will, however, receive the message ‘No such file:’ followed by the character string you entered.

Control keys for moving around in a file

If for some reason you cannot use cursor (arrow) keys, you may use the following commands to move around in your file.

- ‘^p’ moves to the Previous line.
- ‘^n’ moves to the Next line.
- ‘^f’ moves Forward a character.
- ‘^b’ moves Backward a character.

Clearing the display

If your display becomes cluttered (most likely because of interactive messages), you may clear or refresh it by entering ‘^l’. The messages are erased from your screen and the file you are working with remains.

Command list for NANO

The commands you can use in NANO are listed across the bottom of the screen. Remember, ‘^’ symbolizes the control key. ‘^G’ means to hold down the control key and press G.

^G Help. More information on the commands.
^X Exit NANO
^O Save a NANO file
^J Justify text within a paragraph. This command will fill in paragraphs that are missing, etc.
^R Insert an existing file into NANO.
^W Find a word or character string in the NANO file.
^Y Go back to the previous page.
^V Go to the next page.
^K Cuts the line of text where the cursor is located.
^U Uncuts the last text cut.
^C Gives you the exact location of your cursor within the file.
^T Run the spell checker.

When you have selected ^O to save a file or ^R to insert a file, you will be prompted for a file name. You will also be given the following new options:

^C Cancels the command.
^T Gives you a list of files in your home directory. If you select a file that already exists, NANO will ask you if you want to overwrite the file.

Basic VI Editor

Beginning Your Editing Session

To edit a file	vi [filename]
To recover an editing session	vi -r [filename]

Notes on vi commands and modal editing

All vi commands are entered in command mode. To enter command mode, press the ESC key. Some vi commands cause vi to enter another mode. For example, the i (insert command) causes vi to enter insert mode after which all keystrokes are inserted as text. To return to command mode from insert mode, press the ESC key. The :set showmode command will cause vi to display the current editing mode in the lower right corner of the editing screen.

Controlling the Screen Display of Your Session

Repaint the current screen	{^l}
Display line #, # of lines	{^g}

Moving the Cursor

Beginning of current line	0 or ^
Beginning of first screen line	H
Beginning of last screen line	L
Beginning of middle screen line	M
Down one line	j, {return}, +
End of current line	$
Left one character	h, {ctrl-h}
Left to beginning of word	b, B
Right one character	l, {space}
Right to end of word	e, E
Right to beginning of word	w, W
Up one line	k, -
Beginning of next sentence	)
Beginning of previous sentence	(

Paging Through Text

Back one screen	{^b}
Down half a screen	{^d}\
Down one screen	{^f}
Forward to end of file	G
Move cursor to specified line	line no.G
Up half a screen	{^j}

Special Pattern Characters

Beginning of line	^
End of line	$
Any character except newline	.
Any number of the preceding character	*
Any set of characters (except newline)	*

Creating Text

Append text after cursor	a
Append text after end of line	A
Insert text before cursor	i Insert
text at beginning of line	I Open
new line after current line	o Open

new line before current line	O
Take next character literally (i.e. control characters...) and display it	{^v}

Ending Your Editing Sessions

Quit (no changes made)	:q
Quit and save changes	ZZ, :wq
Quit and discard changes	:q!

Using ex Commands From Within vi

Copy specified lines	:co, t
Display line numbers	:set nu
Disable display of line numbers	:set nonu
Move lines after specified line	:m
Read file in after specified line	:r filename
Review current editor options	:set Review
editor options	:set all
Set new editor option	:set option
Write changes to original file	:w
Write to specified file	:w filename
Force write to a file	:w! filename

Some Useful ex commands for use in vi

Some useful set options for your ~/.exrc file:

:set all	Display all Set options
:set autoindent	Auto magically indent following lines to the indentation of previous line.
:set ignorecase	Ignore case during pattern matching.
:set list	Show special characters in the file.
:set number	Display line numbers.
:set shiftwidth=n	Width for shifting operators << and >>
:set showmode	Display mode when in Insert, Append, or Replace mode.
:set wrapmargin=n	Set right margin 80-n for autowrapping lines (inserting newlines). 0 turns it off.

Searching Through Text

Backward for pattern	?pattern
Forward for pattern	/pattern
Repeat previous search	n
Reverse direction of previous search	N
Show *all* lines containing pattern	beg,endg/pattern/p

Example

:1,$g/compiler/p Will print all lines with the pattern compiler.

Substitute patt2 for all patt1 found.	:beg,ends/patt1/patt2/g

Example

:%s/notfound/found/g Will change all occurrences of ‘notfound’ to ‘found’.

Appendix B: MIP SCE Quick Reference Guide

*****Note: There is no password requirement for the MIP SCE, the new hostname is MIPSAPSASUSRP01**

*****Reminder:** Do not store programs, scripts, data, formats, etc. under your user home directory /home/med.ds.osd.mil/$USER/. If this space becomes full, all users will be locked out of the SCE.
Users can set up their profile in MIP SCE? (e.g. For aliases and environmental variables)
Use PICO or VI to edit: '~/.bash_profile'

filesGet Retrieve files from S3

*****NOTE 1: Do not include the .s3 extension when you create your filesGet statement***

*****NOTE 2: Do not include /mdrfile at the beginning of the file path in the filesGet statement***

Command Line Interface (CLI):
Single file: filesGet "/dir/file.ext"
Multiple files: filesGet "/dir/file1.ext, /dir/file2.ext"

Within a SAS program:
Single file: %filesGet("/dir/file.ext") ;
Multiple files: %filesGet("/dir/file1.ext, /dir/file2.ext");

filesPut The automatic daily S3 sync process will execute and place any new/modified files onto S3. This command provides the user to explicitly place a file within S3.

Command Line Interface (CLI):
Single file: filesPut "/dir/file.ext"
Multiple files: filesPut "/dir/file1.ext, /dir/file2.ext"

Within a SAS program:
Single file: %filesPut("/dir/file.ext") ;
Multiple files: %filesPut("/dir/file1.ext, /dir/file2.ext");

filesRemove removes the file from S3

Command Line Interface (CLI):
Single file: filesRemove "/dir/file.ext"
Multiple files: filesRemove "/dir/file1.ext, /dir/file2.ext"

Within a SAS program:
Single file: %filesRemove("/dir/file.ext") ;
Multiple files: %filesRemove("/dir/file1.ext, /dir/file2.ext");

To submit a SAS job:

- Log on to the server using a SSH client and change to the directory where the SAS program is located in your directory space.

Example:

```
[mipsapsasusrp01}:/home/med.ds.osd.mil/$USER > cd /hpae2/
```

Note: Do not store code or data in your home directory, specifically /home/med.ds.osd.mil/$USER. Files in this area may be deleted without warning.

- Remember, if the macro described in Keeping Logs is used, you must use the syntax provided in that section to submit your programs.

```
sas <program_name.sas> -altlog <program_name.log>
```

Passing a SAS argument sas <program_name> -<variable_name> <value>

Commands for submitting SAS jobs

To start a job running in the default workspace: **sas *jobname.sas***

Current status of running jobs: **jobQueue --list**

Job Queue Help**:** **jobQueue --help**

Current status of running jobs only for the user: **jobQueue --list --user $USER**

Cancel a submitted job, before job starts running: **jobQueue --remove <jobPID>**

Cancel a submitted job, when job is running: **jobQueue --stop <progPID>**

Monitoring job queue alternative: **ps -ef | grep -i sasexe | grep -i *username* | grep -v grep**

Scheduling SAS Jobs

One time execution: sas /dir/file.sas --datetimeExec '2023-01-01 23:59:59'

Recurring execution: Edit cron: **crontab --e**

Write the entry to crontab **:w**

Quit crontab **:q**

Crontab Man: https://linux.die.net/man/5/crontab

Appendix C: AVHE MIP AWS -- WinSCP Login to MIPSAPSASUSRP01

Requirements:

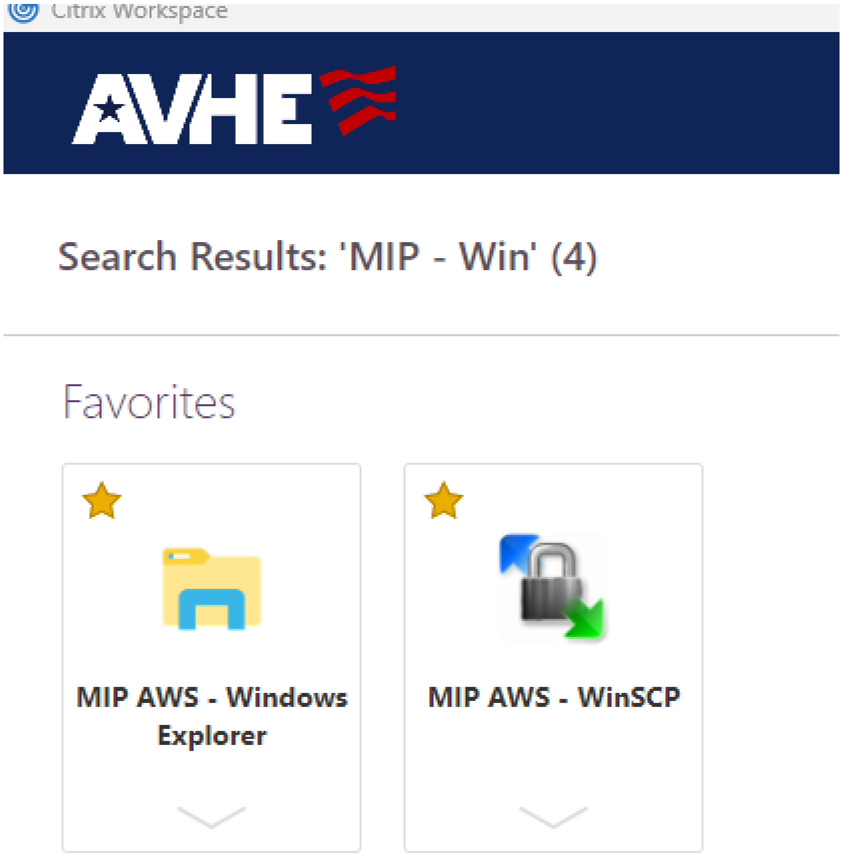
Requirements:

1. User must have access to AVHE with MIP
    a. 'MIP AWS – WinSCP'
    b. This requires a CAC

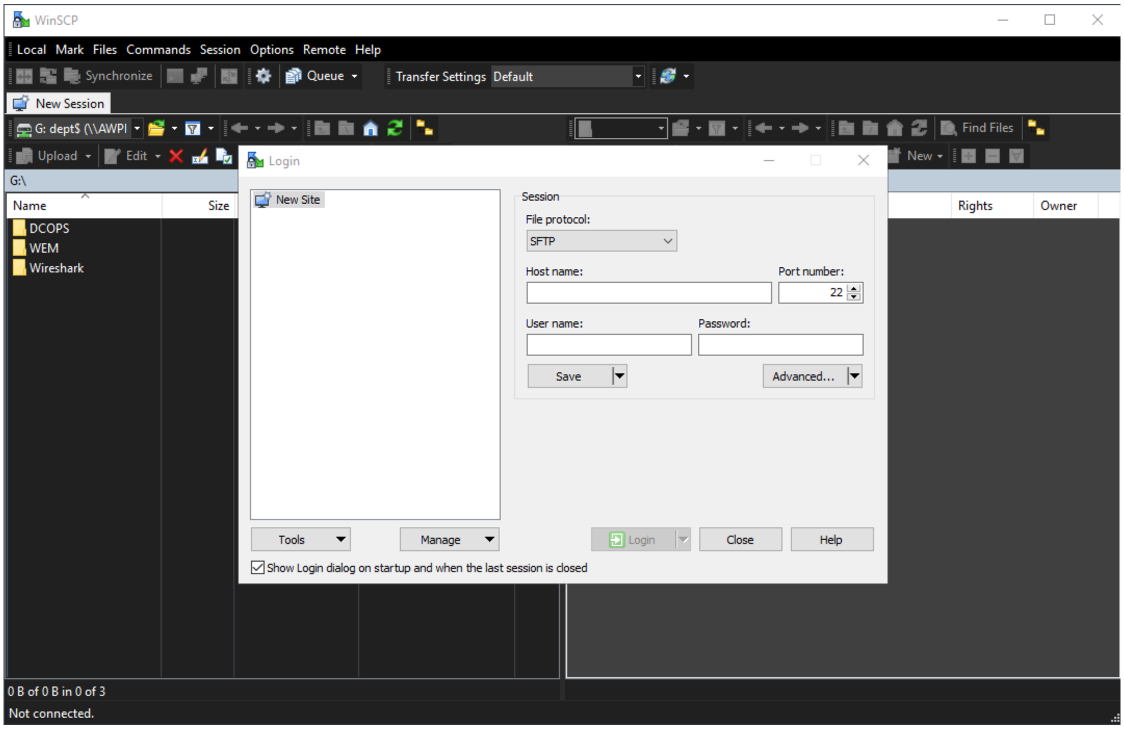
2. User must be added to SAS RBAC group 'MIPS_AWS_AVHE_SASEG'

Directions to access MIPSAPSASUSRP01:

1. Navigate to https://avhe.health.mil/
2. Search for 'MIP AWS – WinSCP'



3. Open 'MIP AWS – WinSCP'



File protocol: SFTP

Host name:

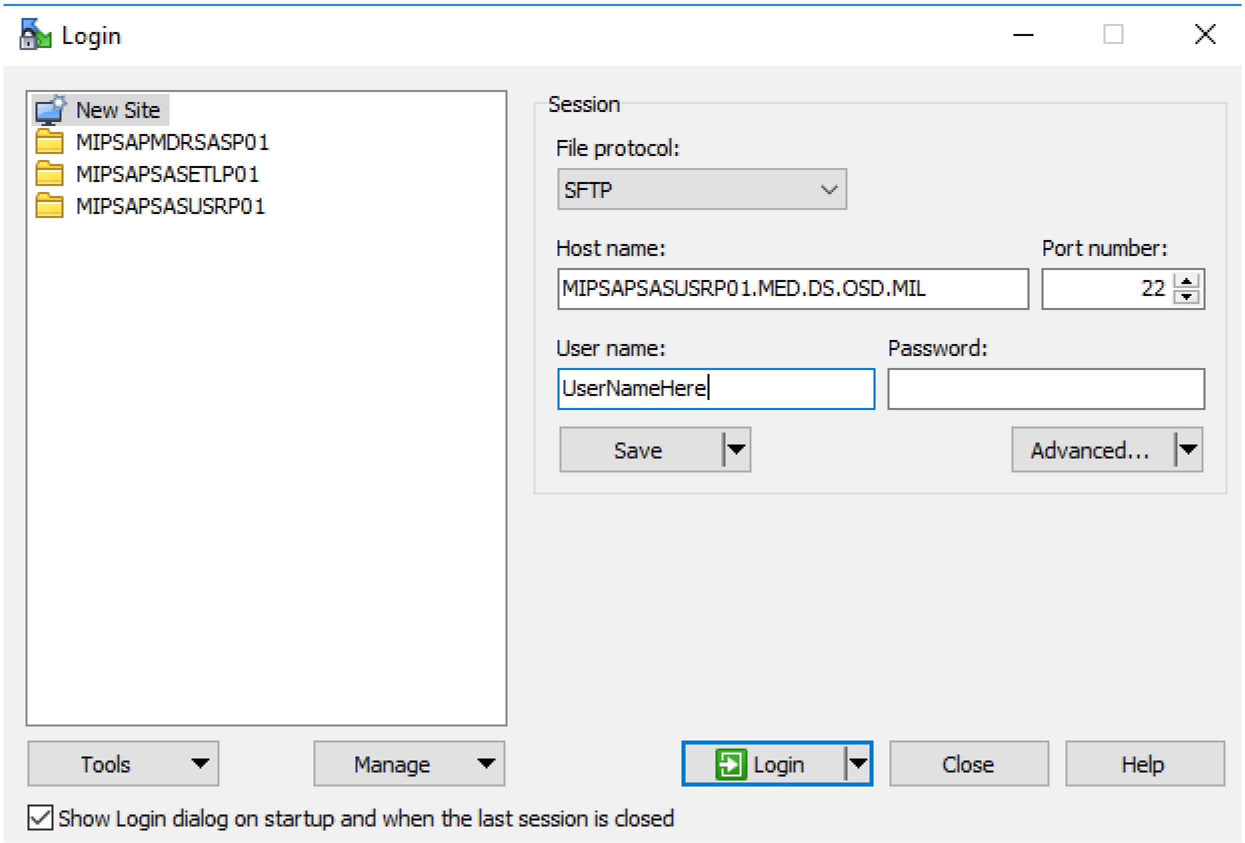
Port number: 22

User name:

Password:

☑ Show Login dialog on startup and when the last session is closed

4. Fill out the Session:
5. in the connection information
   a. File protocol: SFTP
   b. Host name: MIPSAPSASUSRP01.MED.DS.OSD.MIL
   c. User name: Your AVHE user name. If you don't know your username please reference "Who Am I" document for how to obtain.
   d. Password: leave blank



File protocol: SFTP

Host name: MIPSAPSASUSRP01.MED.DS.OSD.MIL

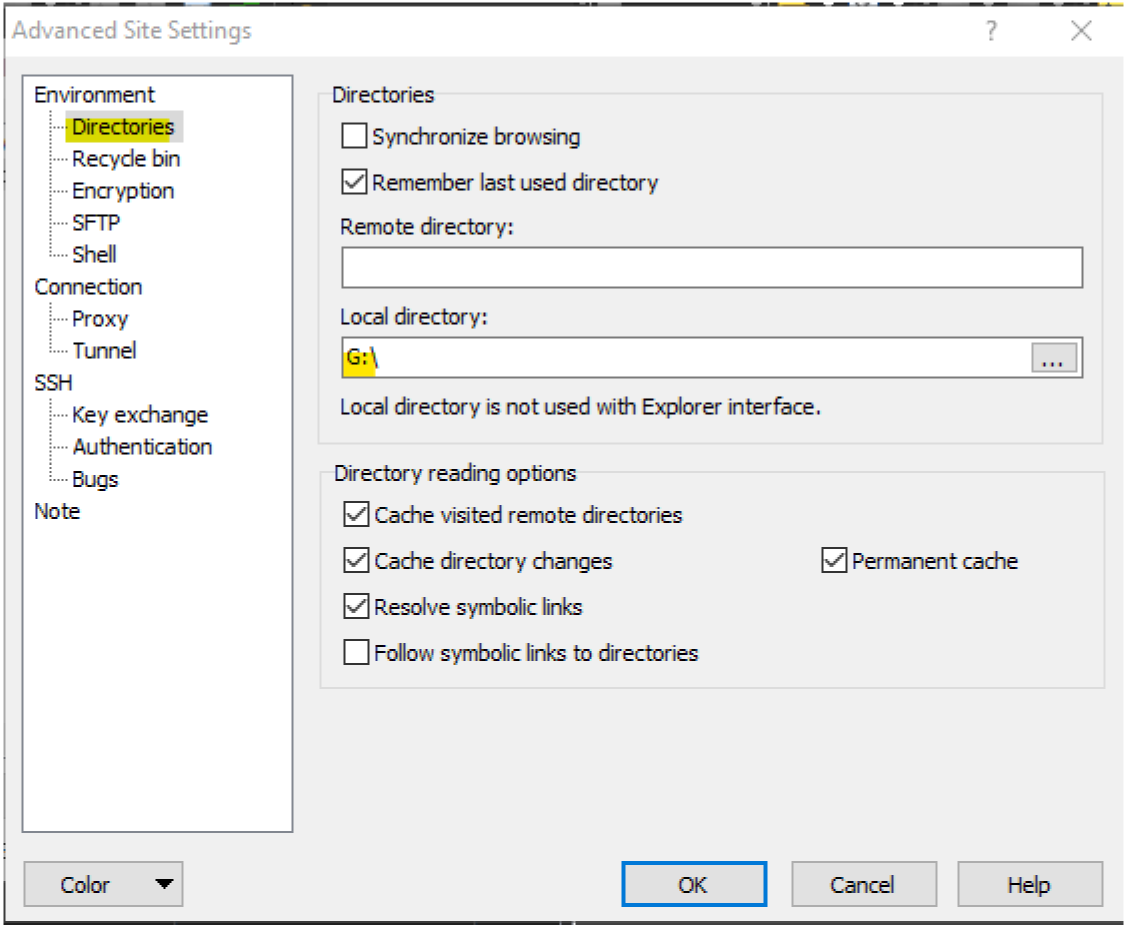
Port number: 22

User name: UserNameHere

Password:

☑ Show Login dialog on startup and when the last session is closed

6. Advanced
   a. Add Directories:
   b. Local directory 'G:\'



Advanced Site Settings

Directories

☐ Synchronize browsing

☑ Remember last used directory

Remote directory:

Local directory: G:\

Local directory is not used with Explorer interface.

Directory reading options

☑ Cache visited remote directories

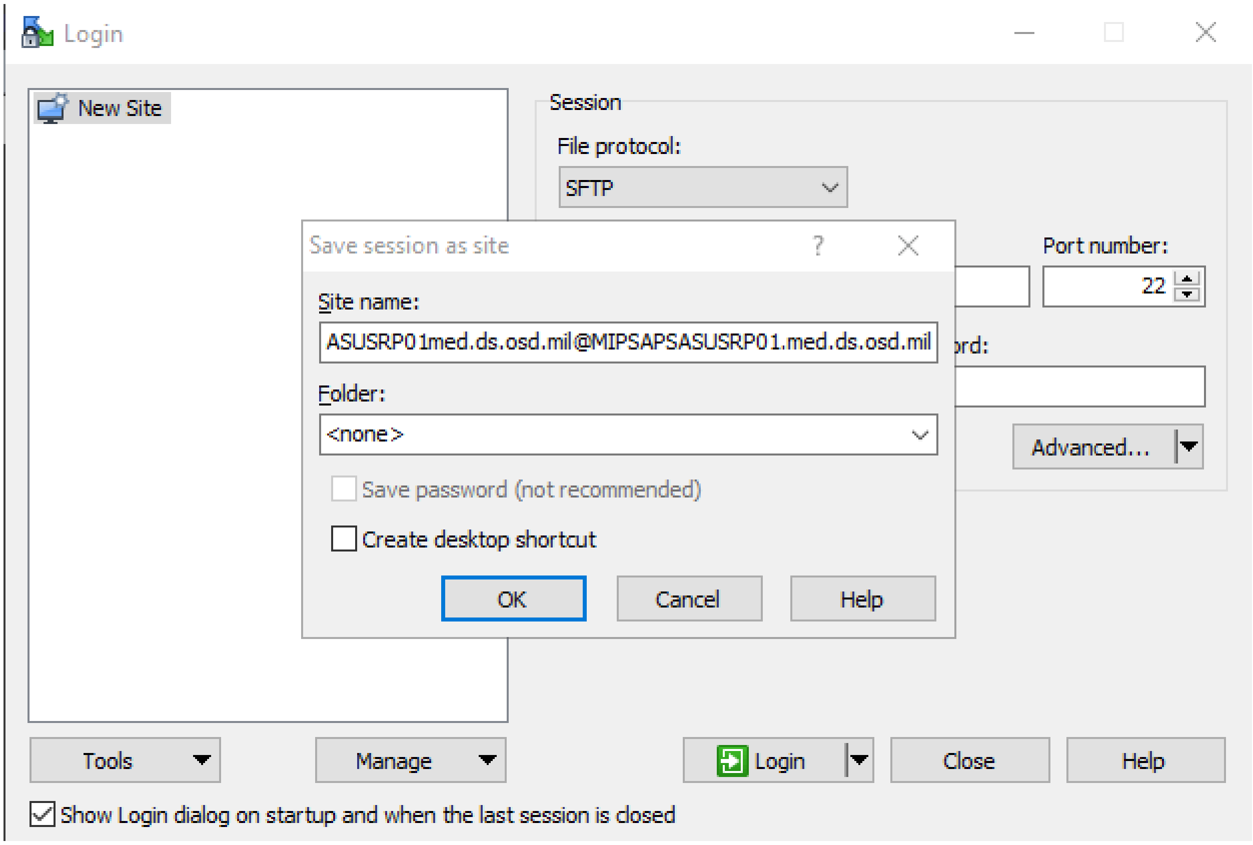
☑ Cache directory changes

☑ Permanent cache

☑ Resolve symbolic links

☐ Follow symbolic links to directories

   c. OK
   d. Save



File protocol: SFTP

Port number: 22

Site name: ASUSRP01med.ds.osd.mil@MIPSAPSASUSRP01.med.ds.osd.mil

Folder: <none>

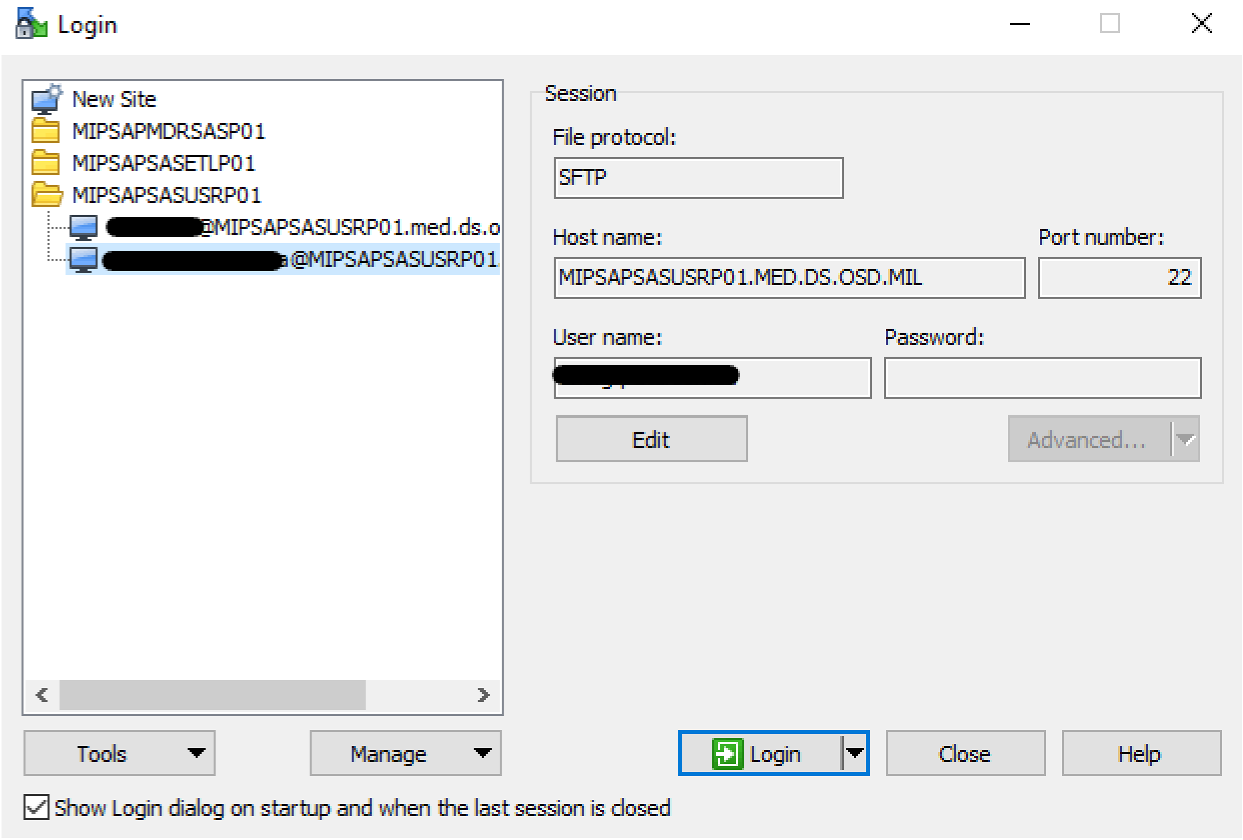
□ Save password (not recommended)

□ Create desktop shortcut

OK Cancel Help

☑ Show Login dialog on startup and when the last session is closed

e. OK



- New Site
- MIPSAPMDRSASP01
- MIPSAPSASETLP01
- MIPSAPSASUSRP01
  - [redacted]@MIPSAPSASUSRP01.med.ds.o
  - [redacted]@MIPSAPSASUSRP01

File protocol: SFTP

Host name: MIPSAPSASUSRP01.MED.DS.OSD.MIL

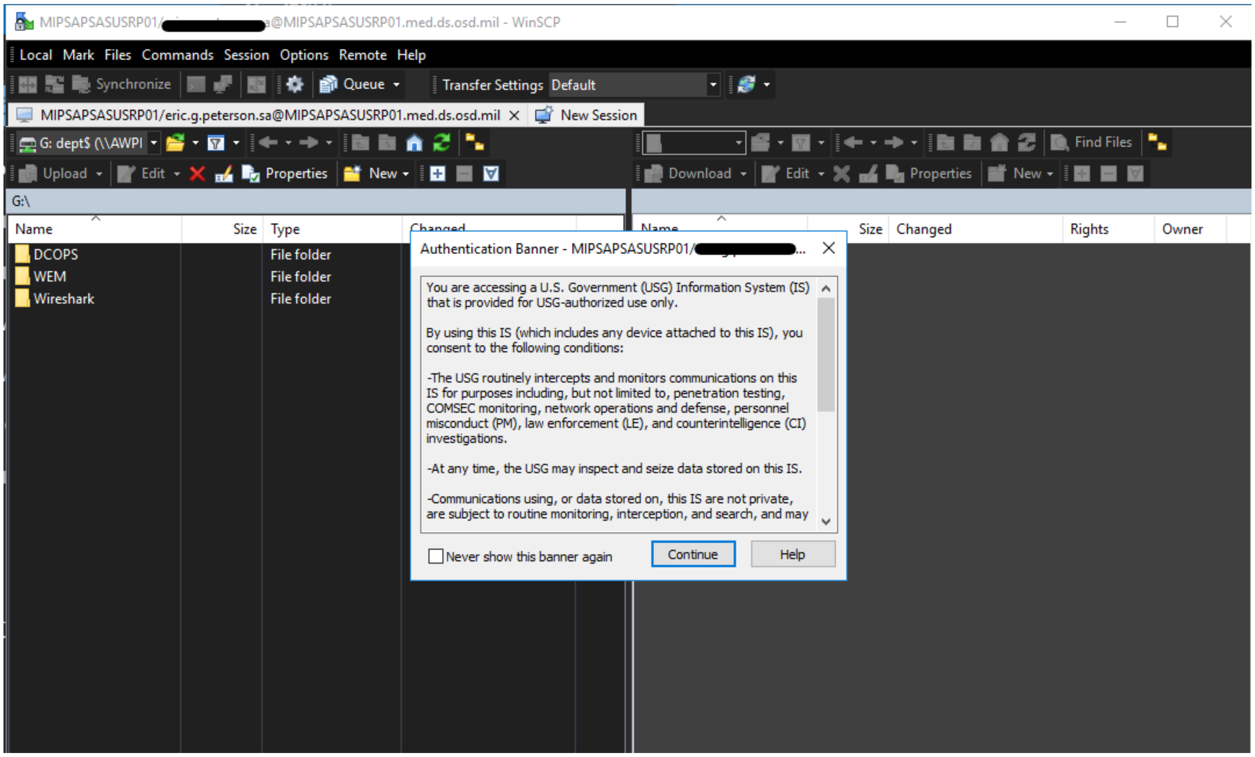
Port number: 22

User name: [redacted]

Password:

☑ Show Login dialog on startup and when the last session is closed

7. Login


MIPSAPSASUSRP01/[redacted]@MIPSAPSASUSRP01.med.ds.osd.mil - WinSCP
Local Mark Files Commands Session Options Remote Help
Synchronize
Queue
Transfer Settings Default
MIPSAPSASUSRP01/eric.g.peterson.sa@MIPSAPSASUSRP01.med.ds.osd.mil
New Session
G: dept$ (\\AWPI
Find Files
Upload
Edit
Properties
New
Download
Edit
Properties
New
G:\
Name
Size
Type
Changed
DCOPS
WEM
Wireshark
File folder
File folder
File folder
Name
Size
Changed
Rights
Owner
Authentication Banner - MIPSAPSASUSRP01/[redacted]...

You are accessing a U.S. Government (USG) Information System (IS) that is provided for USG-authorized use only.

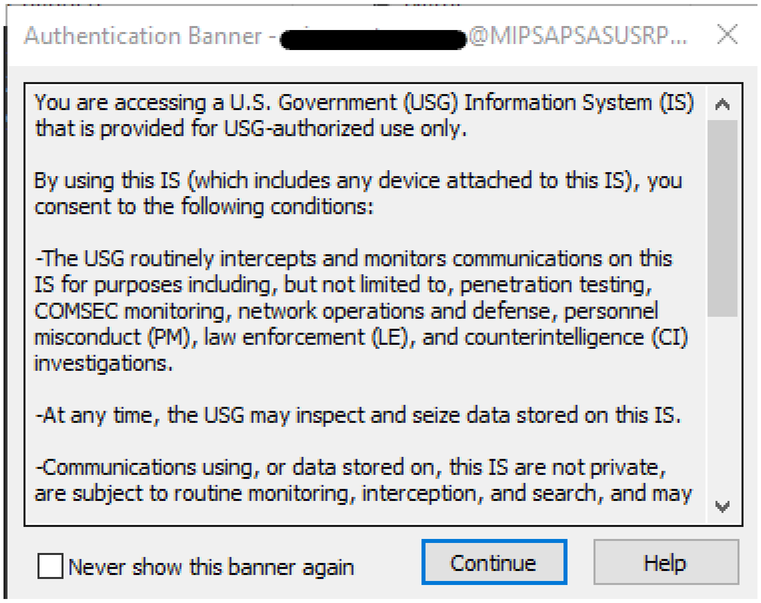
By using this IS (which includes any device attached to this IS), you consent to the following conditions:

-The USG routinely intercepts and monitors communications on this IS for purposes including, but not limited to, penetration testing, COMSEC monitoring, network operations and defense, personnel misconduct (PM), law enforcement (LE), and counterintelligence (CI) investigations.

-At any time, the USG may inspect and seize data stored on this IS.

-Communications using, or data stored on, this IS are not private, are subject to routine monitoring, interception, and search, and may

8. Yes



You are accessing a U.S. Government (USG) Information System (IS) that is provided for USG-authorized use only.

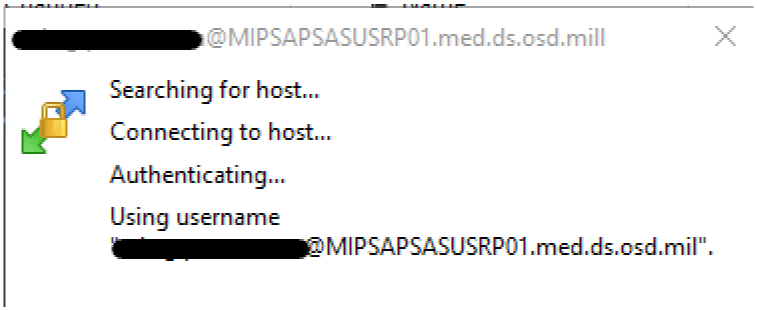
By using this IS (which includes any device attached to this IS), you consent to the following conditions:

-The USG routinely intercepts and monitors communications on this IS for purposes including, but not limited to, penetration testing, COMSEC monitoring, network operations and defense, personnel misconduct (PM), law enforcement (LE), and counterintelligence (CI) investigations.

-At any time, the USG may inspect and seize data stored on this IS.

-Communications using, or data stored on, this IS are not private, are subject to routine monitoring, interception, and search, and may

9. Continue

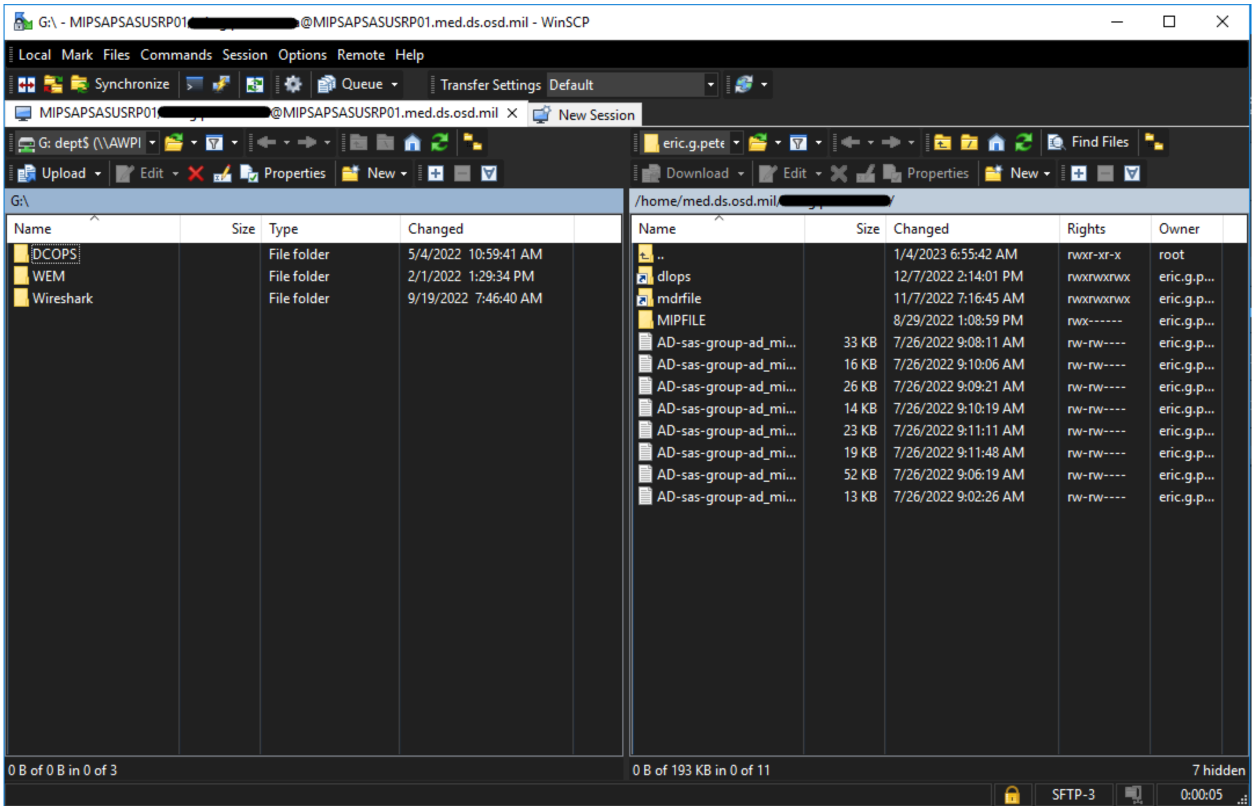


Searching for host...

Connecting to host...

Authenticating...

Using username "[redacted]@MIPSAPSASUSRP01.med.ds.osd.mil".



G:\ - MIPSAPSASUSRP01[redacted]@MIPSAPSASUSRP01.med.ds.osd.mil - WinSCP

MIPSAPSASUSRP01[redacted]@MIPSAPSASUSRP01.med.ds.osd.mil

Transfer Settings Default

G:\

Name	Size	Type	Changed
DCOPS		File folder	5/4/2022 10:59:41 AM
WEM		File folder	2/1/2022 1:29:34 PM
Wireshark		File folder	9/19/2022 7:46:40 AM

0 B of 0 B in 0 of 3

/home/med.ds.osd.mil/[redacted]/

Name	Size	Changed	Rights	Owner
..		1/4/2023 6:55:42 AM	rwxr-xr-x	root
dlops		12/7/2022 2:14:01 PM	rwxrwxrwx	eric.g.p...
mdrfile		11/7/2022 7:16:45 AM	rwxrwxrwx	eric.g.p...
MIPFILE		8/29/2022 1:08:59 PM	rwx------	eric.g.p...
AD-sas-group-ad_mi...	33 KB	7/26/2022 9:08:11 AM	rw-rw----	eric.g.p...
AD-sas-group-ad_mi...	16 KB	7/26/2022 9:10:06 AM	rw-rw----	eric.g.p...
AD-sas-group-ad_mi...	26 KB	7/26/2022 9:09:21 AM	rw-rw----	eric.g.p...
AD-sas-group-ad_mi...	14 KB	7/26/2022 9:10:19 AM	rw-rw----	eric.g.p...
AD-sas-group-ad_mi...	23 KB	7/26/2022 9:11:11 AM	rw-rw----	eric.g.p...
AD-sas-group-ad_mi...	19 KB	7/26/2022 9:11:48 AM	rw-rw----	eric.g.p...
AD-sas-group-ad_mi...	52 KB	7/26/2022 9:06:19 AM	rw-rw----	eric.g.p...
AD-sas-group-ad_mi...	13 KB	7/26/2022 9:02:26 AM	rw-rw----	eric.g.p...

0 B of 193 KB in 0 of 11

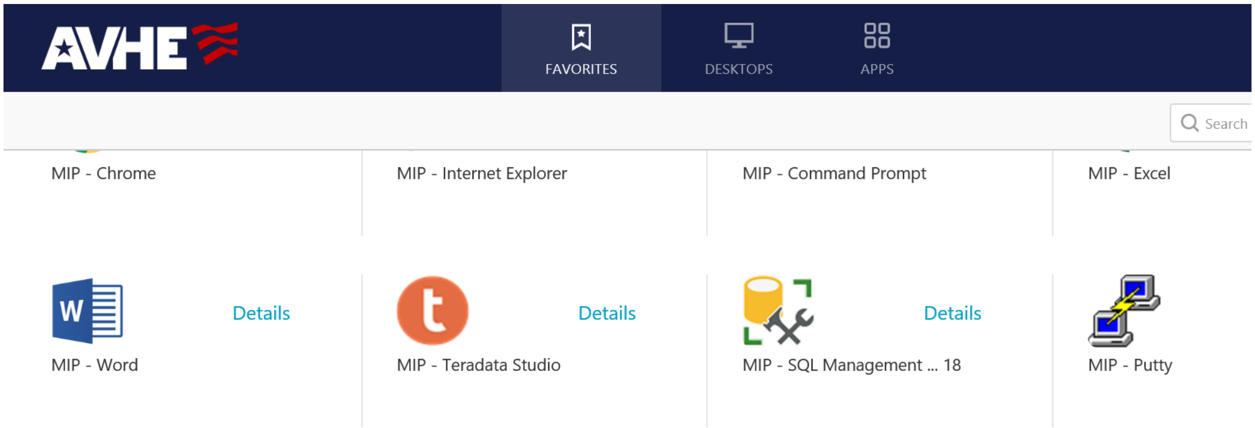
7 hidden

You are now connected to the folder structure on server. The folders are RBAC based so now all users will see or be able to access the same folders.

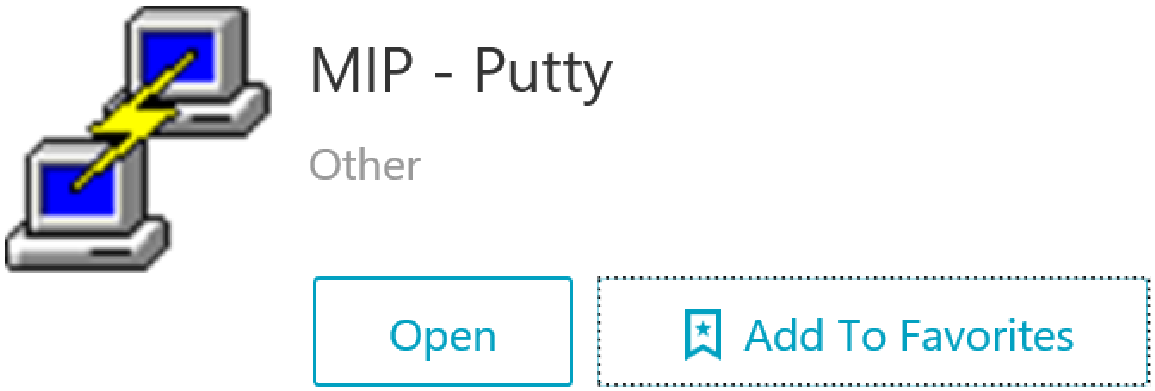
Repeat the process for other desired connections as needed.

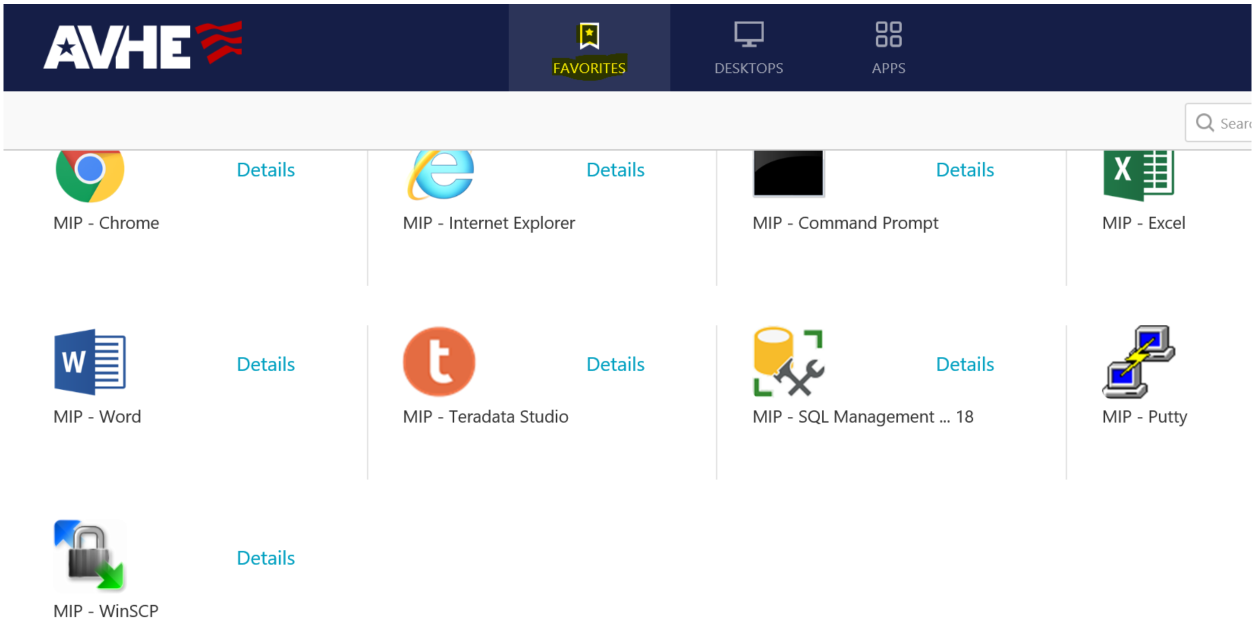
Appendix D: SAS Launch PuTTY SSH Session to MIPSAPSASUSRP01

Navigate to "MIP -- Putty"

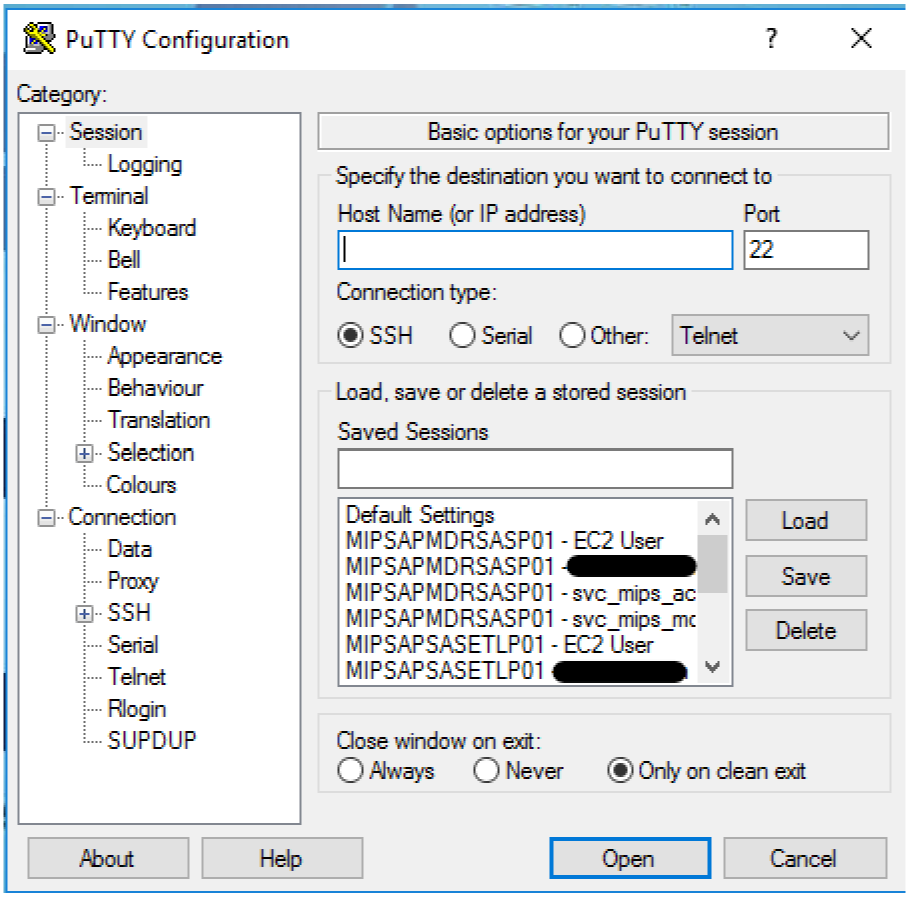


Click Details and Add To Favorites





Launch MIP -- Putty



Specify the destination you want to connect to

Host Name (or IP address):

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

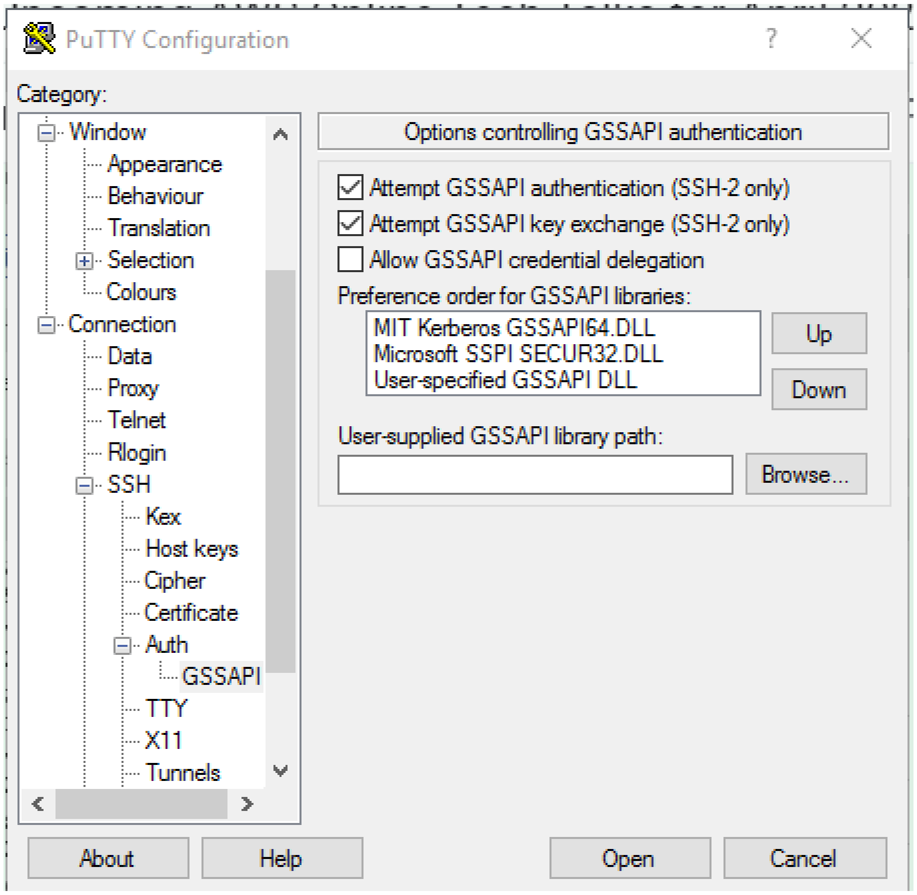
Load, save or delete a stored session

Saved Sessions

Default Settings
MIPSAPMDRSASP01 - EC2 User
MIPSAPMDRSASP01 - [redacted]
MIPSAPMDRSASP01 - svc_mips_ac
MIPSAPMDRSASP01 - svc_mips_mc
MIPSAPSASETLP01 - EC2 User
MIPSAPSASETLP01 [redacted]

Close window on exit: ○ Always ○ Never ● Only on clean exit

Navigate Category to Connection > SSH > Auth > GSSAPI



☑ Attempt GSSAPI authentication (SSH-2 only)
☑ Attempt GSSAPI key exchange (SSH-2 only)
☐ Allow GSSAPI credential delegation

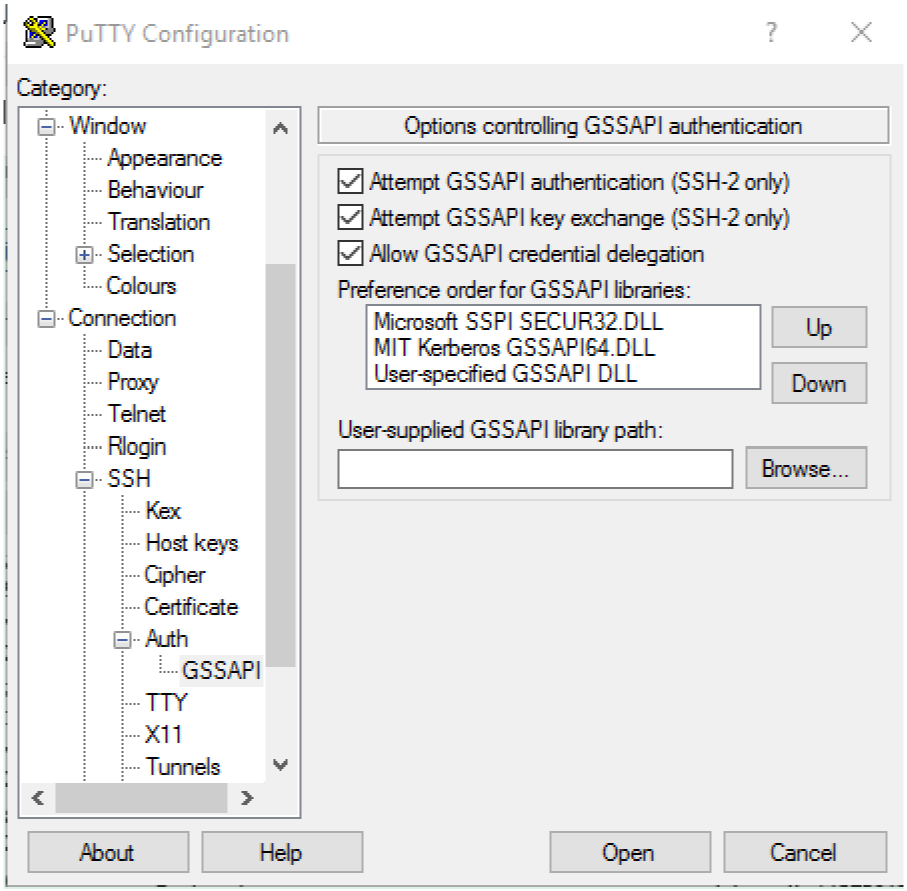
Preference order for GSSAPI libraries:

MIT Kerberos GSSAPI64.DLL
Microsoft SSPI SECUR32.DLL
User-specified GSSAPI DLL

User-supplied GSSAPI library path:

Check Allow GSSAPI credential delegation

Move “Microsoft SSPI SECUR32.DLL” Up



Options controlling GSSAPI authentication

☑ Attempt GSSAPI authentication (SSH-2 only)

☑ Attempt GSSAPI key exchange (SSH-2 only)

☑ Allow GSSAPI credential delegation

Preference order for GSSAPI libraries:

Microsoft SSPI SECUR32.DLL
MIT Kerberos GSSAPI64.DLL
User-specified GSSAPI DLL

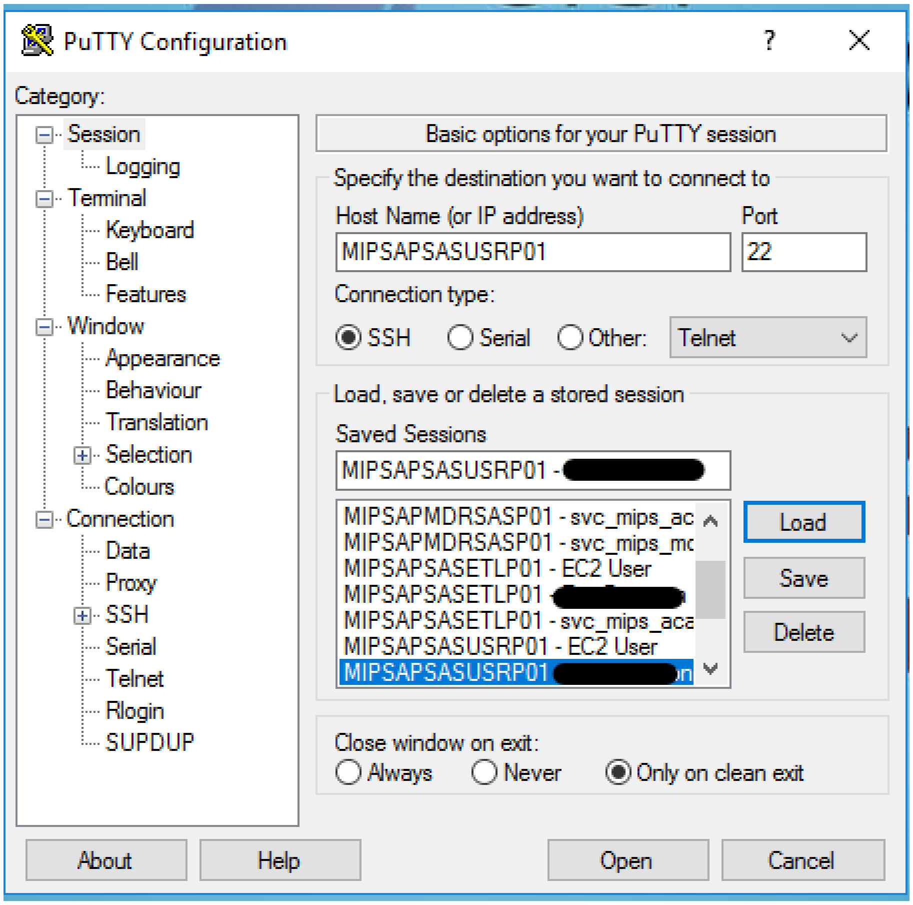
User-supplied GSSAPI library path:

Add Host Name

MIPSAPSASUSRP01.med.ds.osd.mil

Add the Saved Sessions. I called mine “MIPSAPSASUSRP01 – Mynamehere”

Save



Specify the destination you want to connect to

Host Name (or IP address): MIPSAPSASUSRP01

Port: 22

Connection type: ● SSH ○ Serial ○ Other: Telnet

Load, save or delete a stored session

Saved Sessions: MIPSAPSASUSRP01 - [redacted]

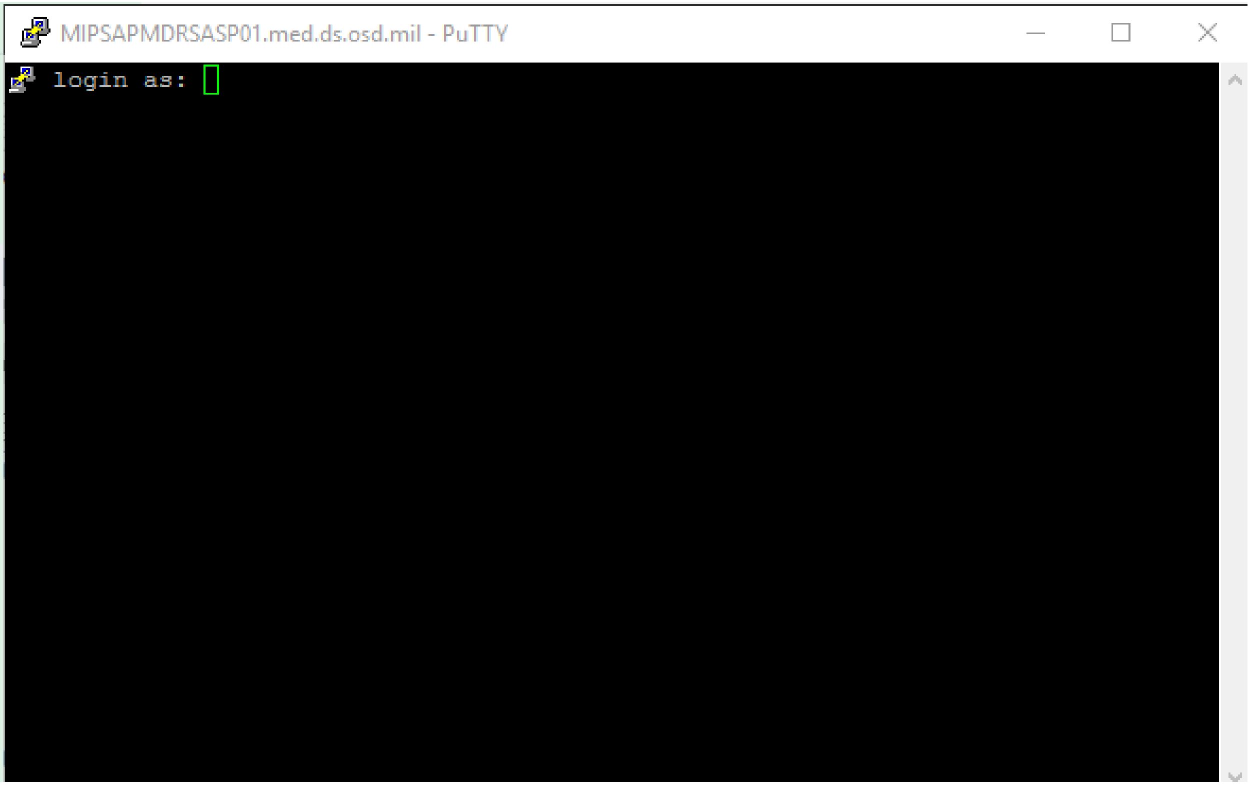
MIPSAPMDRSASP01 - svc_mips_ac
MIPSAPMDRSASP01 - svc_mips_mc
MIPSAPSASETLP01 - EC2 User
MIPSAPSASETLP01 [redacted]
MIPSAPSASETLP01 - svc_mips_aca
MIPSAPSASUSRP01 - EC2 User
MIPSAPSASUSRP01 [redacted]

Close window on exit: ○ Always ○ Never ● Only on clean exit

Save

Load

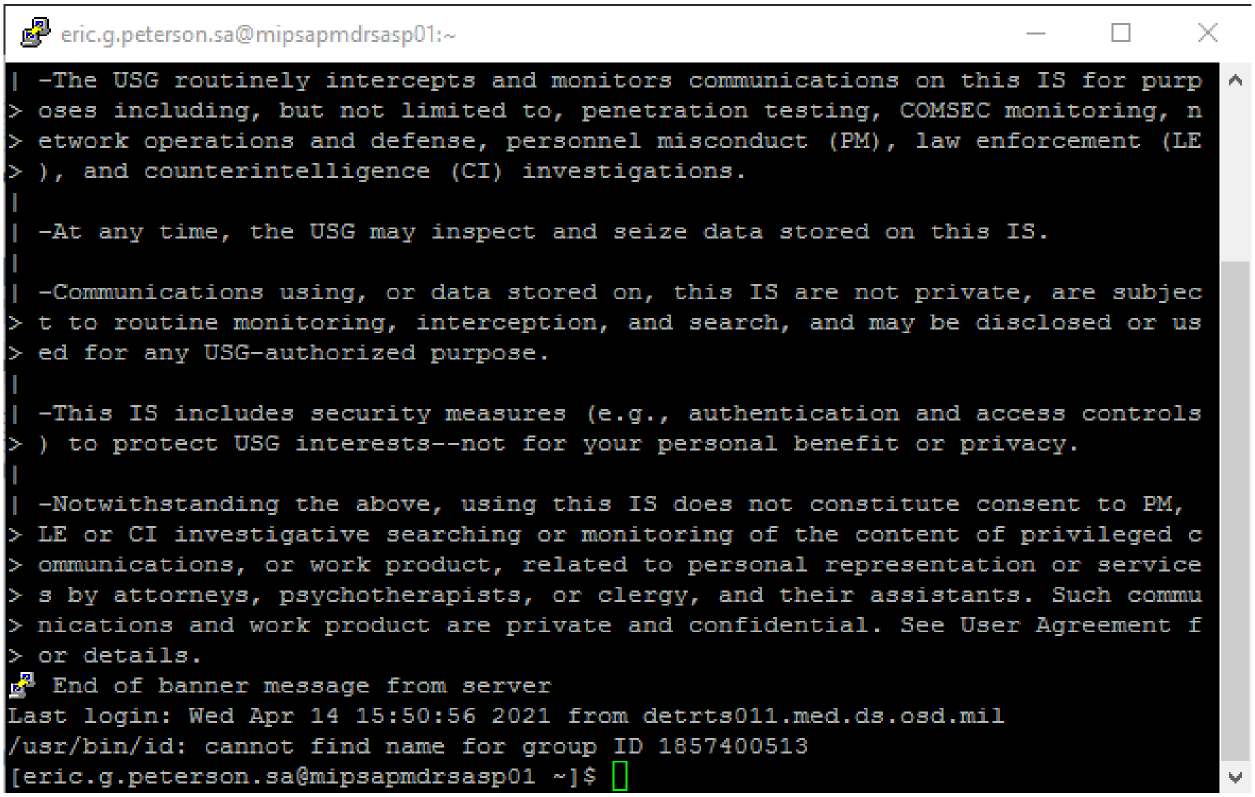
Open



MIPSAPMDRSASP01.med.ds.osd.mil - PuTTY

```
login as: 
```

Enter your MHS domain name



```
| -The USG routinely intercepts and monitors communications on this IS for purp
> oses including, but not limited to, penetration testing, COMSEC monitoring, n
> etwork operations and defense, personnel misconduct (PM), law enforcement (LE
> ), and counterintelligence (CI) investigations.
|
| -At any time, the USG may inspect and seize data stored on this IS.
|
| -Communications using, or data stored on, this IS are not private, are subjec
> t to routine monitoring, interception, and search, and may be disclosed or us
> ed for any USG-authorized purpose.
|
| -This IS includes security measures (e.g., authentication and access controls
> ) to protect USG interests--not for your personal benefit or privacy.
|
| -Notwithstanding the above, using this IS does not constitute consent to PM,
> LE or CI investigative searching or monitoring of the content of privileged c
> ommunications, or work product, related to personal representation or service
> s by attorneys, psychotherapists, or clergy, and their assistants. Such commu
> nications and work product are private and confidential. See User Agreement f
> or details.
  End of banner message from server
Last login: Wed Apr 14 15:50:56 2021 from detrts011.med.ds.osd.mil
/usr/bin/id: cannot find name for group ID 1857400513
[eric.g.peterson.sa@mipsapmdrsasp01 ~]$
```

Connection established

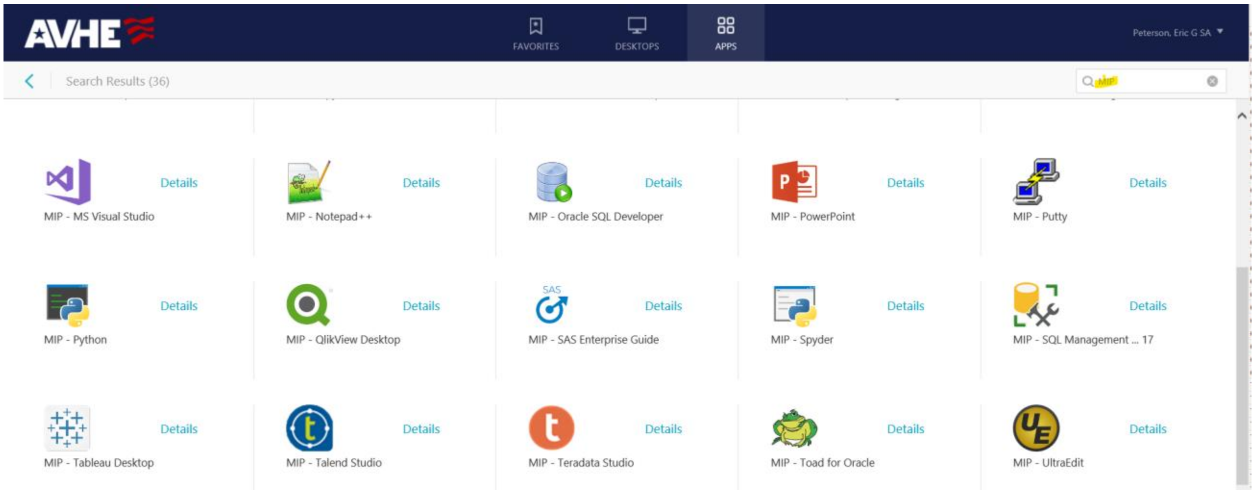
Appendix E: SAS Create Enterprise Guide Profile to MIPSAPSASUSRP01

MIP Enclave -- SAS Enterprise Guide

These requirements must be met to use SAS Enterprise Guide:

1. **Must have access to AVHE**
2. **Your user account must be in the "mip_avhe_sas" Active Directory group**
3. **You must have a SAS Metadata account created within the SAS server**
4. **You must have access to the various database platforms**
5. **MDR SAS and MIP SAS both use the same AVHE app, "MIP -- SAS Enterprise Guide" with different Connection Profiles.**

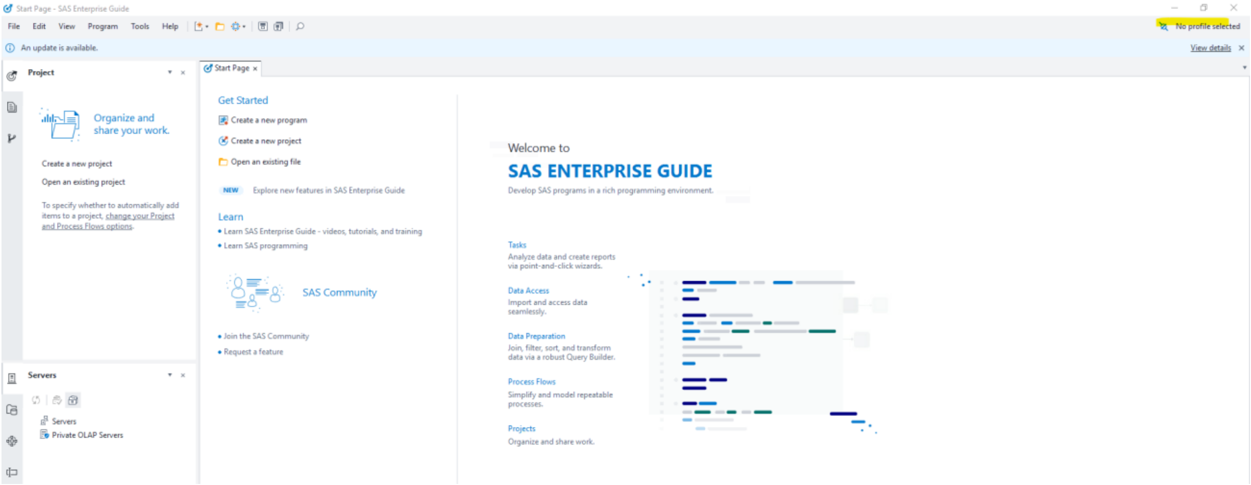
Search for MIP in the "Search Favorites" section of https://avhe.health.mil/Citrix/AVHEWeb/



Click Details and Add to Favorites

Launch

Create SAS Enterprise Guide Profile



Organize and share your work.

Create a new project

Open an existing project

To specify whether to automatically add items to a project, change your Project and Process Flows options.

Explore new features in SAS Enterprise Guide

Learn SAS Enterprise Guide - videos, tutorials, and training

Learn SAS programming

Welcome to

SAS ENTERPRISE GUIDE

Develop SAS programs in a rich programming environment.

Analyze data and create reports via point-and-click wizards.

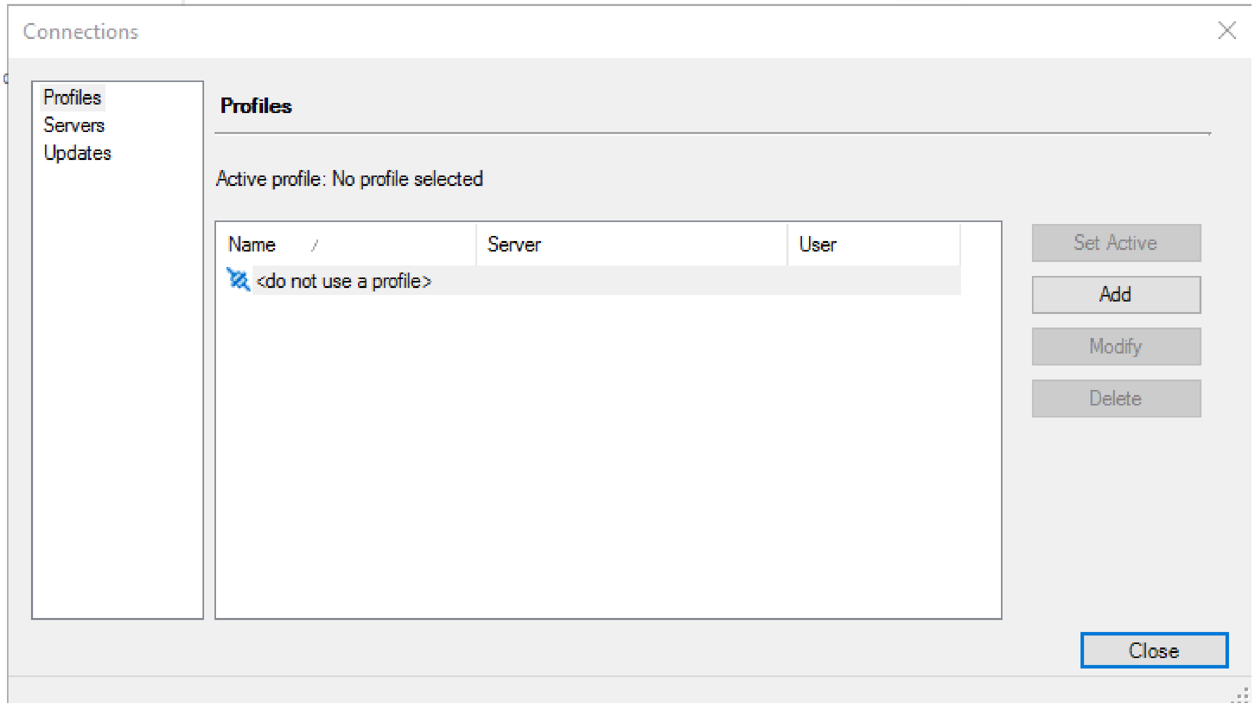
Import and access data seamlessly.

Join, filter, sort, and transform data via a robust Query Builder.

Simplify and model repeatable processes.

Organize and share work.

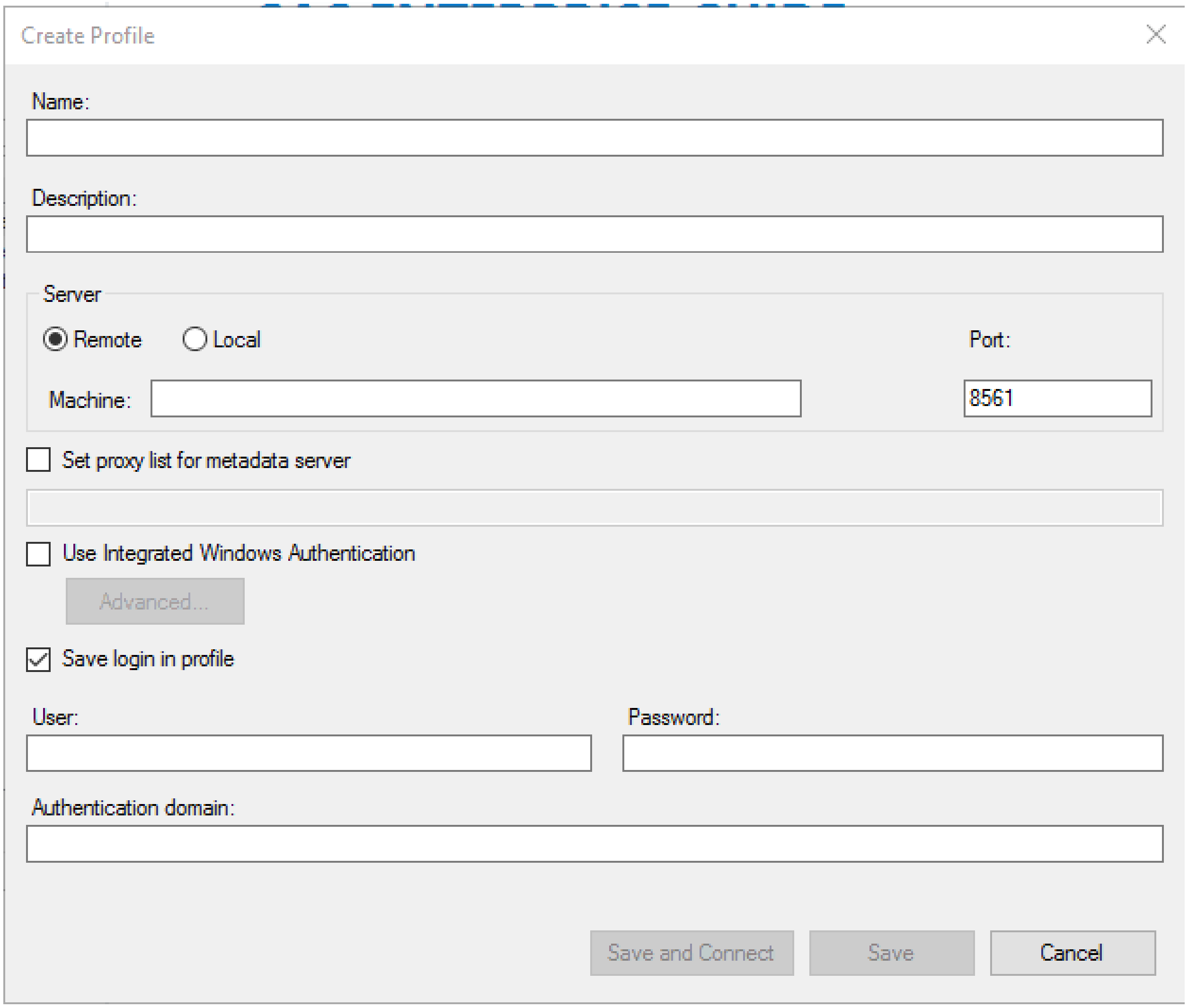
Add



Profiles

Active profile: No profile selected

Name	Server	User
<do not use a profile>		



● Remote ○ Local

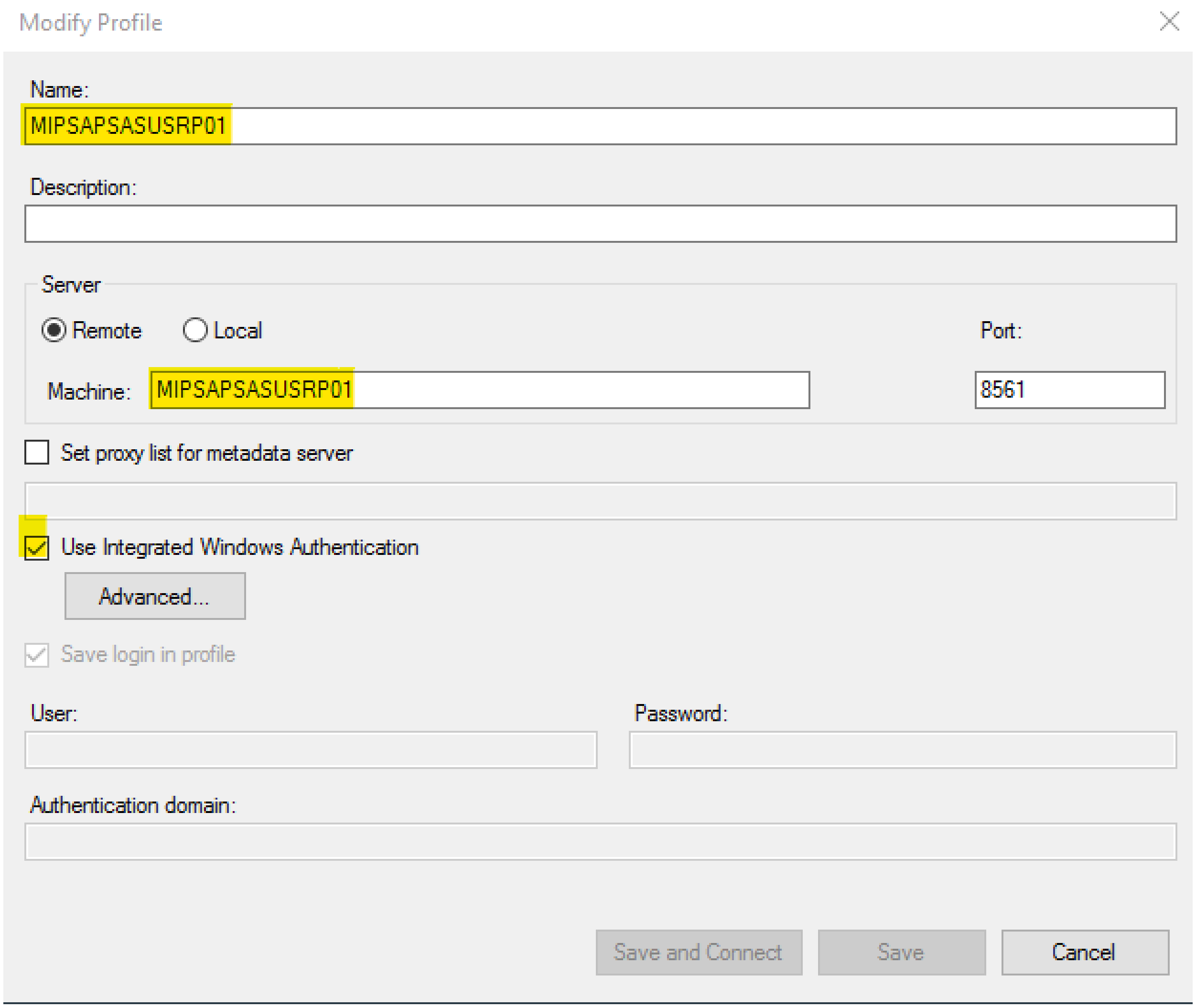
Port: 8561

☑ Save login in profile

Name: “MIPSAPSASUSRP01”

Machine: “MIPSAPSASUSRP01”

Check “Use Integrated Windows Authentication”



Modify Profile

Name: MIPSAPSASUSRP01

Description:

Server

● Remote ○ Local

Machine: MIPSAPSASUSRP01

Port: 8561

□ Set proxy list for metadata server

☑ Use Integrated Windows Authentication

☑ Save login in profile

User:

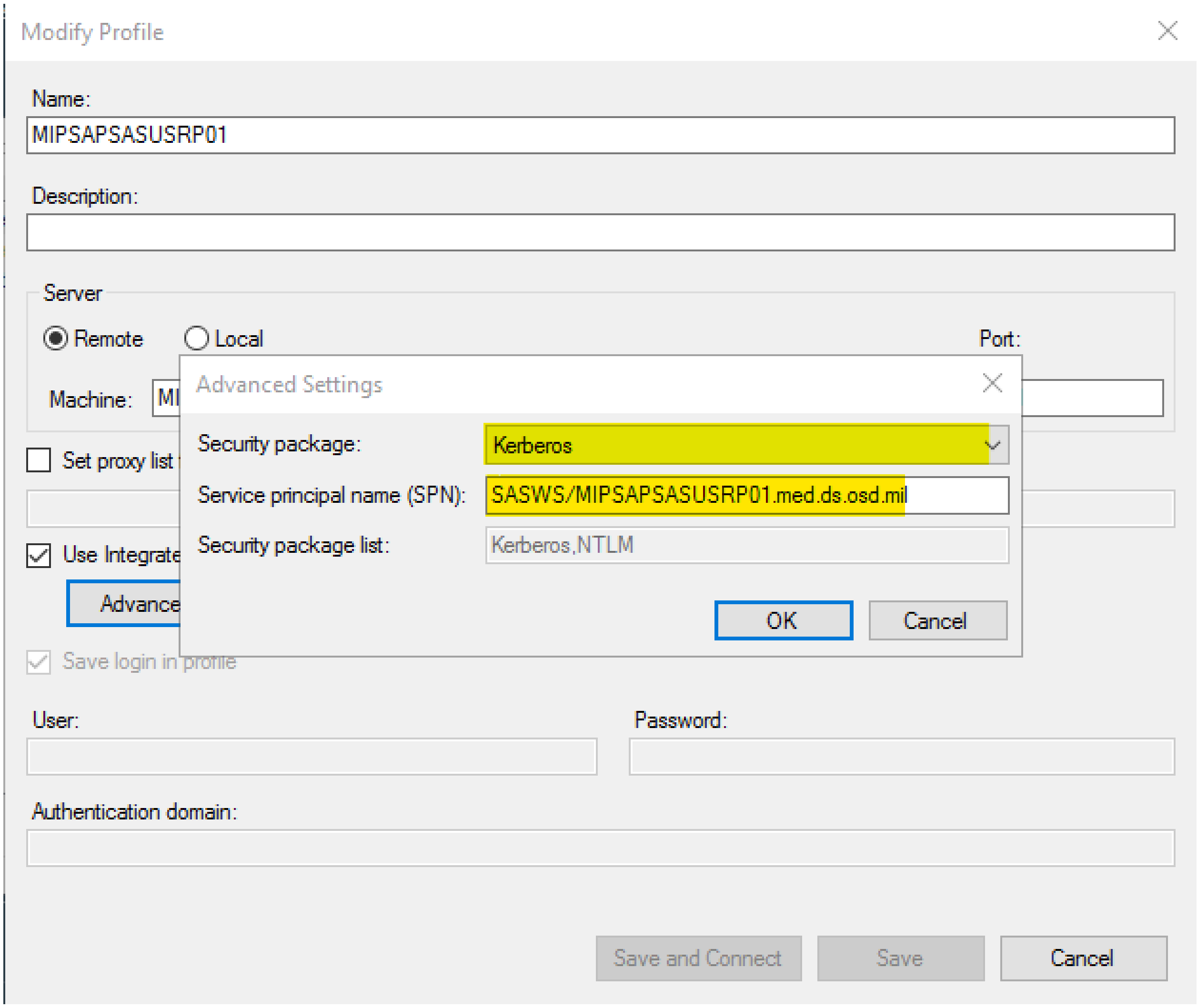
Password:

Authentication domain:

Advanced

Modify Security package: Kerberos

Add the service principal name (SPN): SASWS/MIPSAPSASUSRP01.med.ds.osd.mil



Modify Profile

Name: MIPSAPSASUSRP01

Description:

Server

● Remote ○ Local

Machine: MI

Port:

□ Set proxy list

☑ Use Integrate

Advance

☑ Save login in profile

User:

Password:

Authentication domain:

Advanced Settings

Security package: Kerberos

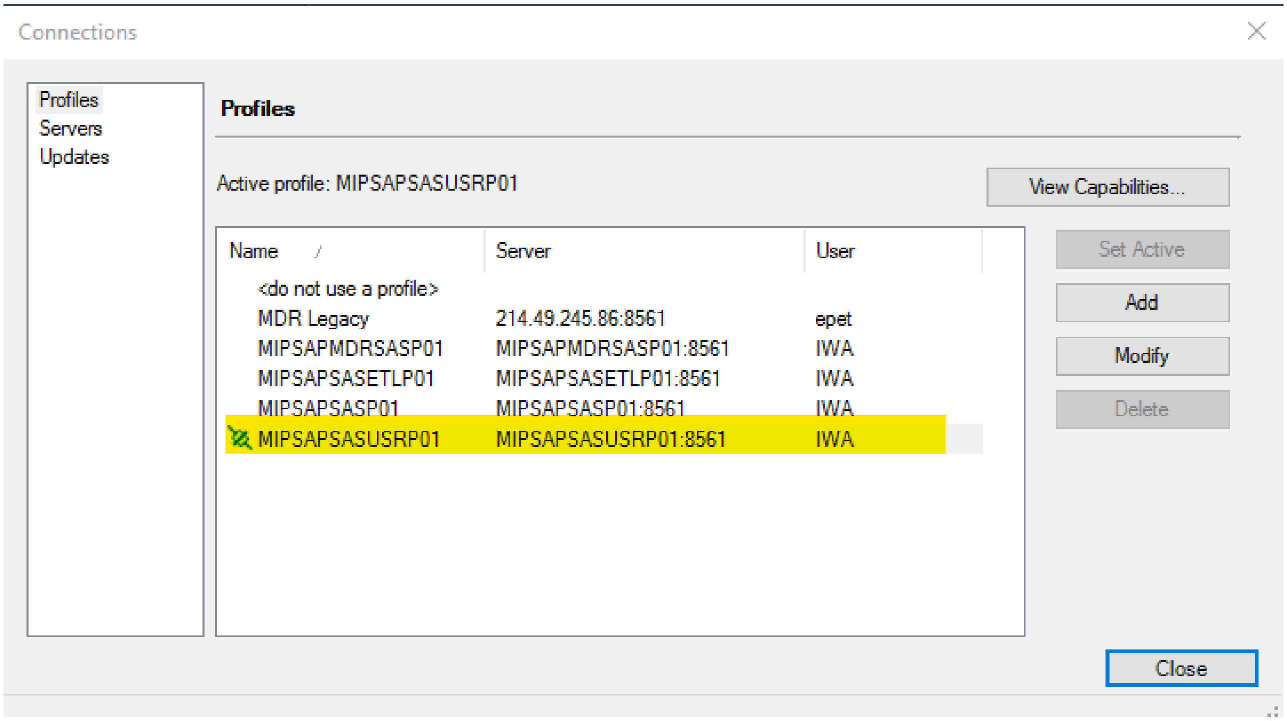
Service principal name (SPN): SASWS/MIPSAPSASUSRP01.med.ds.osd.mil

Security package list: Kerberos,NTLM

OK

Save and Connect

Set Active



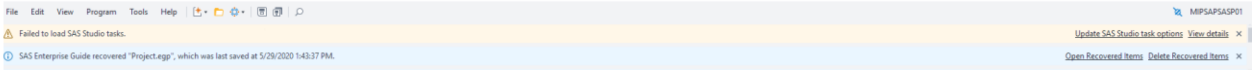
Active profile: MIPSAPSASUSRP01

Name	Server	User
<do not use a profile>		
MDR Legacy	214.49.245.86:8561	epet
MIPSAPMDRSASP01	MIPSAPMDRSASP01:8561	IWA
MIPSAPSASETLP01	MIPSAPSASETLP01:8561	IWA
MIPSAPSASP01	MIPSAPSASP01:8561	IWA
MIPSAPSASUSRP01	MIPSAPSASUSRP01:8561	IWA

Connection is Active (note SAS EG can be Connection Profiles set up to both MDR and MIP SAS servers)

More EG Tips:

To disable the “Failed to load SAS Studio tasks”

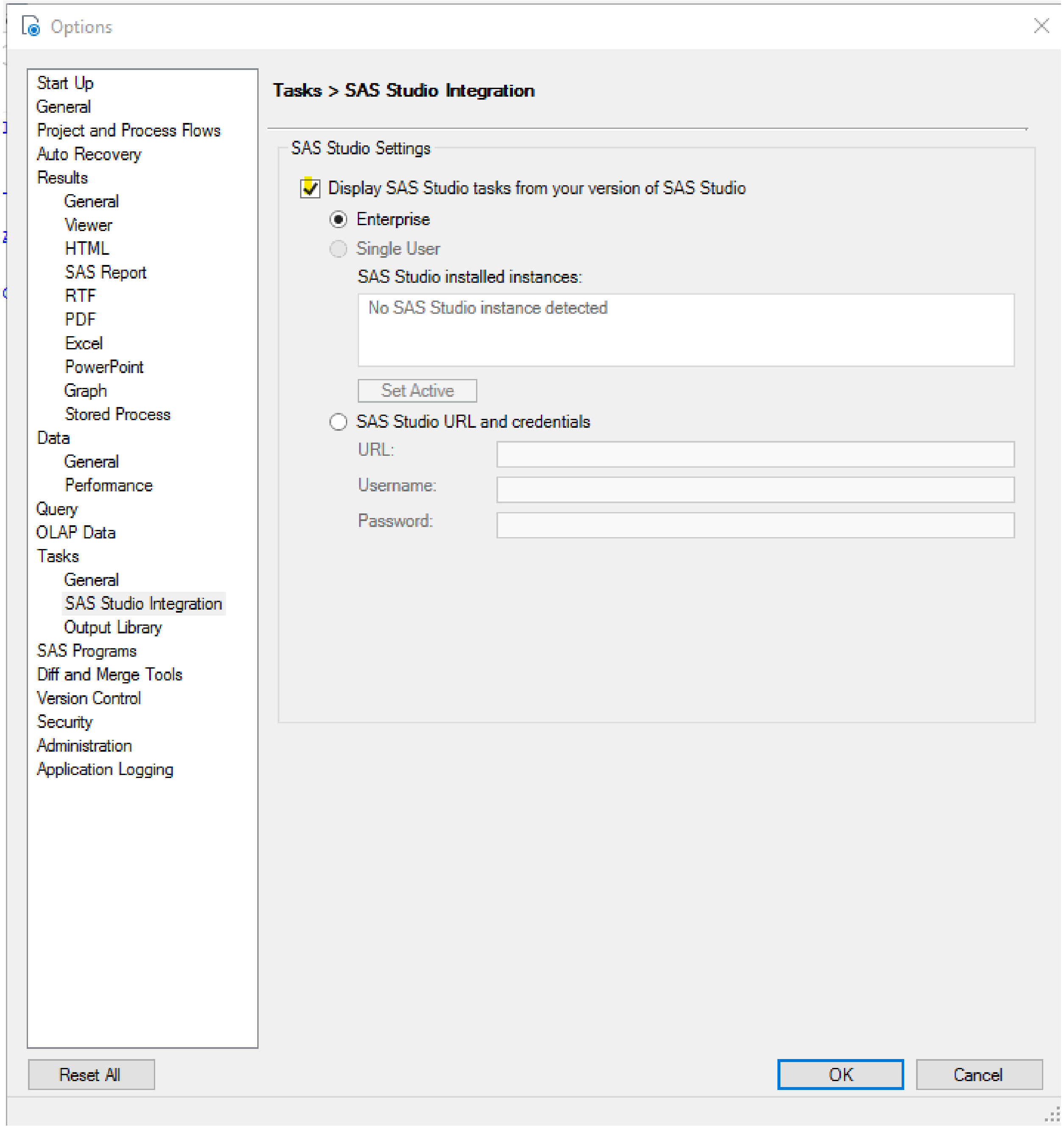


Failed to load SAS Studio tasks.

SAS Enterprise Guide recovered "Project.egp", which was last saved at 5/29/2020 1:43:37 PM.

Tools > Options

Tasks > SAS Studio Integration> Uncheck “Display SAS Studio tasks from you version of SAS Studio”



Tasks > SAS Studio Integration

SAS Studio Settings

☑ Display SAS Studio tasks from your version of SAS Studio

● Enterprise

○ Single User

SAS Studio installed instances:

No SAS Studio instance detected

○ SAS Studio URL and credentials

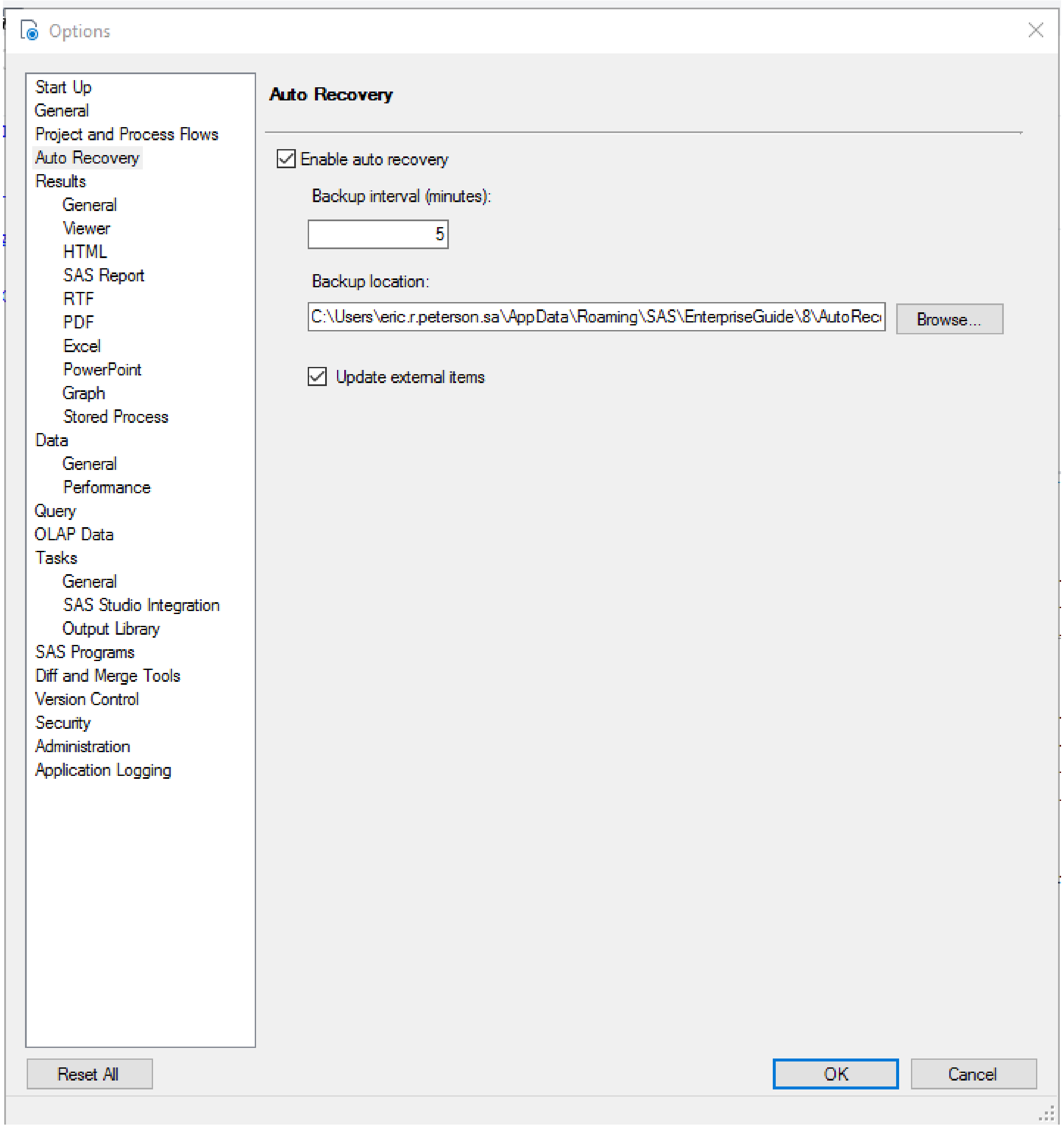
URL:

Username:

Password:

SAS EG has Auto Recovery on by default. You can disable if you want. You can also choose a separate Backup location. The default location is the AVHE desktop you're on.

C:\Users\eric.r.peterson.sa\AppData\Roaming\SAS\EnterpriseGuide\8\AutoRecovery



Auto Recovery

☑ Enable auto recovery

Backup interval (minutes): 5

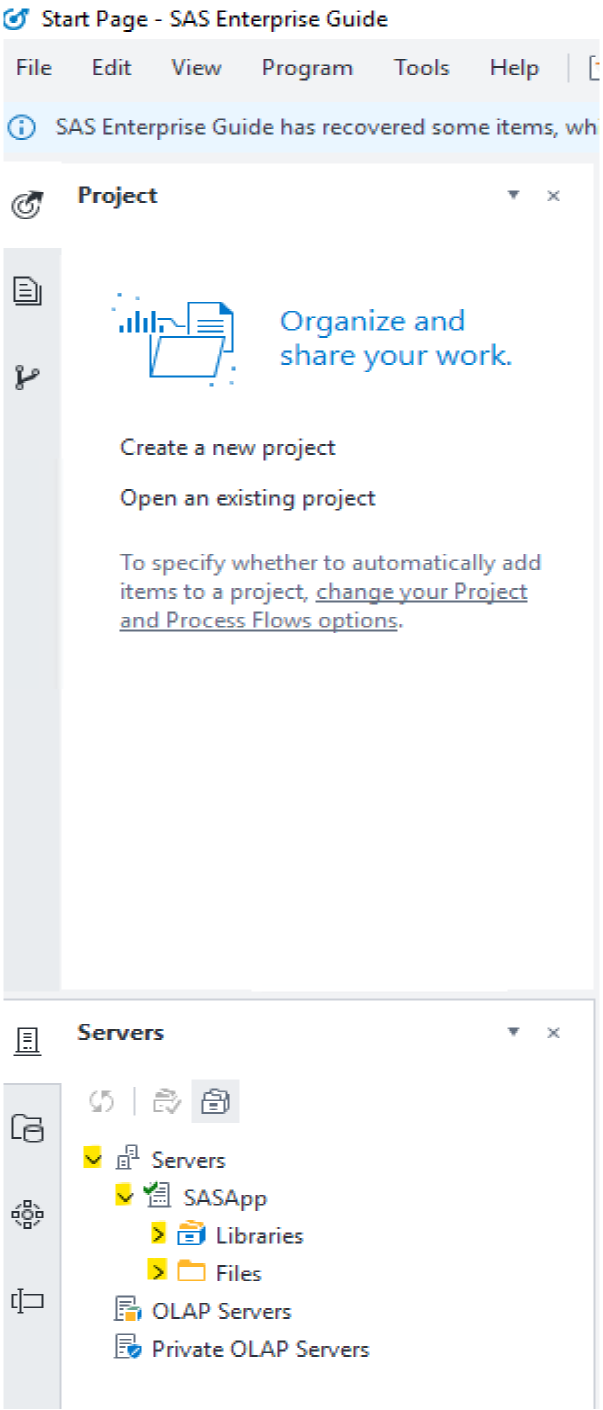
Backup location: C:\Users\eric.r.peterson.sa\AppData\Roaming\SAS\EnterpriseGuide\8\AutoRec

☑ Update external items

You can customize the application with different options

Database Connections

SAS can be connected to any SQL Server or Redshift databases in the MIP enclave. The BI/SAS Admins would need to make the connection definition within SAS. SAS accesses both database platforms using Kerberos authentication. If you have access to the database then you will have access through SAS. If you do not have access to the database then you will not have access using SAS even though you may see the server and database listed.



SAS Enterprise Guide has recovered some items, wh

Organize and share your work.

Create a new project

Open an existing project

To specify whether to automatically add items to a project, change your Project and Process Flows options.

To view the Libraries and Files expand SASApp.

We have pre-defined most of the database on the SQLFARM for SFM-PRDDWSP and SFM-PRDRPT and Redshift_HSDW.