

SAP Analytics Cloud User and Team Provisioning SCIM API Sample Scripts

User Guide



Contents

Introduction.....	6
Business Benefits and Overview.....	9
 Initial Setup with Postman	10
Step A – download and install Postman.....	10
Step B – download the samples.....	11
Step C – import and define the environments into Postman.....	12
Import environment ‘templates’	14
Create OAuth client to get API user/pass	17
Enter values into Postman Environment.....	19
Step D – import sample collections in Postman	22
Step E – Run sample collection 1001 to test the environment has been setup correctly	23
Step F – (optional) Postman Environments for Multiple SAP Analytics Cloud Services.....	25
Step G – Define new user default settings	27
 Introduction to sample scripts (collections).....	30
SAP HANA Academy hands-on video tutorials	30
Sample Collections for a single SAP Analytics Cloud Service.....	30
Test.....	31
Set New User Default Settings.....	31
Creating users (user basis).....	32
Creating and updating users (user basis)	32
Creating and updating users (including updating the email address) (user basis)	32
Creating and updating users (SAML variants) (user basis)	33
Summary for which create/update user scripts to use depending on your SAML SSO setup.....	33
Delete users (user basis).....	34
User Updates (user basis).....	34
User Updates (user basis but given a team)	35
List all Teams.....	35
Create or Update Teams (team basis).....	35
Add/remove users and roles to Teams (team basis).....	35
Teams on Teams (team basis).....	36
Deleting teams (team basis)	36
Admin Tool Kit (whole user/team scan)	37
 Sample Collections for multiple SAP Analytics Cloud Services	39
 Scenarios	41

Prerequisites for all scripts (except 10xx and 2xxx)	42
Teams	42
Why is a change needed?	42
Option 1 (recommended): Ignore Content Namespace For Teams	43
Option 2 (no longer recommended): IMPLEMENT_WORKAROUND_FOR_SCIM_GROUPS	44
Roles	45
Collection behaviour and design	47
Teams created without Team folders	47
NEO platform exception	47
Postman tests	47
General handling of errors	48
Sessions, tokens and sharing of tokens across collections	49
Command line interface	49
Miscellaneous	50
When things go wrong	50
License restrictions causing user and team updates to fail	50
Performance issues	50
Official references and documentation	50
Non-script, non-API errors	51
Bugs	51
API error responses and possible cause	52
Provided 'as is'	52
About and contact	52
Data protection	52
Collection documentation	53
1001-Test-Es-Tests Single Environment Setup	54
1002-Test-Es-Tests Single Environment Advanced Setup	55
1011-Test-Em-Tests Multi Environment Setup	56
1021-U-Ue-Fcj-Update New User Default settings	57
1101-U-C-Fcj-Es-Create users (no roles, no teams)	58
1102-U-C-Cr-Fj-Es-Create users (with roles, no teams)	60
1103-U-C-Crt-Fj-Es-Create users (with roles & teams)	62
1111-U-CU-CUlm-Fcj-Es-Create/update users (no roles, no teams)	65
1112-U-CU-CUlmr-Oarrk-Fcj-Es-Create/update users (roles, no teams)	67
1113-U-CU-CUlmrt-Oarrk-Fcj-Es-Create/update users (roles & teams)	70
1121-U-CU-CUlem-Fcj-Es-Create/update users (no roles, no teams)	75
1122-U-CU-CUlemr-Oarrk-Fcj-Es-Create/update users (roles, no teams)	77
1123-U-CU-CUlemrt-Oarrk-Fcj-Es-Create/update users (roles & teams)	80

1131-U-CU-CUlem-Fcj-Es-SAML Create/update users (no roles, no teams)	85
1132-U-CU-CUlemr-Oarrk-Fcj-Es-SAML Create/update users (roles, no teams).....	87
1133-U-CU-CUlemrt-Oarrk-Fcj-Es-SAML Create/update users (roles & teams).....	90
1201-U-UC-UClem-Oarrk-Fcj-Es-Update/create users (no roles, no teams)	94
1202-U-UC-UClemr-Oarrk-Fj-Es-Update/create users (with roles, no teams)	97
1203-U-UC-UClemrt-Oarrk-Fj-Es-Update/create users (with roles & teams)	100
1231-U-UC-UClem -Fcj-Es-SAML Update/create users (no roles, no teams)	105
1232-U-UC-UClemr-Oarrk-Fj-Es-SAML Update/create users (with roles, no teams).....	107
1233-U-UC-UClemrt-Oarrk-Fj-Es-SAML Update/create users (with roles & teams).....	110
1301-U-D-Du-Fcj-Es-Delete Users	115
1311-U-D-Du-Fcj-Es-Delete Users (by saml mapping)	117
1321-U-D-Du-Fcj-Es-Delete Users (by email).....	118
1401-U-U-UI-Fcj-Es-Update User License.....	119
1403-U-U-Ut-Fj-Es-Update user team membership	121
1404-U-U-Um-Fcj-Es-Update User Manager	123
1406-U-U-Ulppddtn-Fj-Es-Update User DateTimeNumFormat DataAccessLang	124
1407-U-U-Ulppddtn-Fj-Es-Update User License DateTimeNumFormat DataAccessLang.....	126
1408-U-U-Ur-Oarrk-Fj-Es-Update User Role.....	128
1409-U-U-Um-Fcj-Es-Update SAML Mapping.....	130
1418-U-U-Ur-Oarrk-Fj-Es-Update User Role (by saml mapping)	131
1419-U-U-Um-Fcj-Es-Update SAML Mapping (by saml mapping)	132
1428-U-U-Ur-Oarrk-Fj-Es-Update User Role (by email)	133
1429-U-U-Um-Fcj-Es-Update SAML Mapping (by email)	134
1451-TU-U-UI-Fcj-Es-Update Team License	135
1454-TU-U-Um-Fcj-Es-Update Team Manager.....	137
1456-TU-U-Upddtn-Fj-Es-Update Team DateTimeNumFormat DataAccessLang.....	138
1457-TU-U-Ulppddtn-Fj-Es-Update Team License, DateTimeNumFormat, DataAccessLang.....	140
1458-TU-U-Ur-Oarrk-Fj-Es-Update Role for Each User of Team.....	142
1501-T-UC-Ud-Fcj-Es-Update create team	144
1601-All_T-List all teams	145
1602-T-Uc-Uur-Oarrk-B-Fj-Es users/roles actions on Teams.....	146
1612-T-Uc-Uur-Oarrk-Fcj-Es users/roles actions on Teams	150
1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams	152
1654-All_T-Uc-Uur-Oark-Transfer API Hidden Team To API Created Team.....	155
1665-All_U-Uc-Uu-Oarrieei-Fj-Es-AdminToolKit.....	160

1801-T-D-Dt-Fcj-Es-Delete teams (ok to timeout)	173
1802-T-D-Dt-Fcj-Es-Delete teams (empty first)	174
1851-T-D-Dtu-Fcj-Es-Delete teams & users in the team	175
1903-U-UC-Oarrk-Fcj-Em-Transport users	177
1933-U-UC-Oarrk-Fcj-Em-SAML Transport users.....	181
1953-TU-UC-TOarrk-UOarrkie-Fcj-Em-Transport teams and users	182
1983-TU-UC-TOarrk-UOarrkie-Fcj-Em-SAML Transport teams and users.....	189
2601-All_T-List all teams	190
Scenarios	191
Scenario D01 - Delete users then delete managers.....	192
Scenario D02 - Delete dormant users A	193
Scenario D03 - Delete dormant users B	196
Scenario D04 - Delete dormant users C	200
Scenario D05 - Delete dormant users D	204
Scenario D06 - Delete dormant users E	209
Scenario D07 - Delete dormant users F.....	214
Scenario L01 - Managers with BIconcurrent to BInamed license	219
Scenario L02 - Disabled users to BIconcurrent license (now redundant)	219
Scenario L03 - Convert all BIconcurrent to BInamed license.....	220
Scenario M01 - Reassign users of given manager to another	221
Scenario R01 - Swap directly assigned role for a team role	222
Scenario S01 - Assign settings for recently created with default settings	224
Scenario S02 - Assign settings concurrent lic for recently created w default settings	225
Scenario T01 - Transport Managers then Users	226

Introduction



User



Role



Team

These sample scripts have been developed for the SAP Analytics Cloud SCIM (System for Cross-domain Identity Management) API to demonstrate its optimal means of use and remove any barriers to its adoption. These samples are freely available and are provided on an 'as is' basis.

They have been developed with 'best practices' in mind to ensure the greatest throughput and likelihood of a known result. These aspects are critical when performing high-volume administration tasks of the SCIM nature. They are designed to avoid known breaking points of both the API and, where possible, the script itself. These samples have encapsulated intelligence, providing an abstract layer between the API and the various use cases the scripts support. This intelligence can be re-used and adopted immediately without necessarily understanding how the API behaves and without needing to learn from any lessons, as this is done for you.

The sample scripts have been developed in Postman using JavaScript. Postman, which is freely available, is a commonly used tool for implementing API workflows. For those organisations that use Postman, the samples can be used immediately or with some minor modifications. For other organisations, Postman provides an excellent basis to demonstrate how requests are made to the API and optionally export code snippets in various languages, such as Java, Python, C, PHP, Node.js and many others. It means development time can be reduced, in some cases quite dramatically.



POSTMAN

Should your organisation prohibit using Postman, you can still use these samples since a free, open-source command-line option is available. It means:

- You'll need to update the 'environment' by editing a .json file rather than using the user interface.
- You'll need to use the command line to run a script (Collection) rather than the user interface.
- There is no need to download the Postman app, connect to any Postman website or web service or register with Postman. The open-source option doesn't require any internet connectivity to operate.

This document provides step-by-step instructions on using these samples with Postman, making it suitable for API developers and any SAP Analytics Cloud services administrator who wishes to automate administration tasks compared to manually through the user interface.

The sample scripts (called 'collections' in Postman) provide the ability to perform administration tasks including:

- Creating, updating users, or updating all users of a given Team.
 - Includes, for example, updating the BI Licence Type of all users in a Team.
- Deleting users or deleting users of a Team.
- Creating Teams and adding or removing users (and roles) into or out of teams.
- Performing 'team on team' functions such as copying a Team, adding, or removing a Team to or from another.

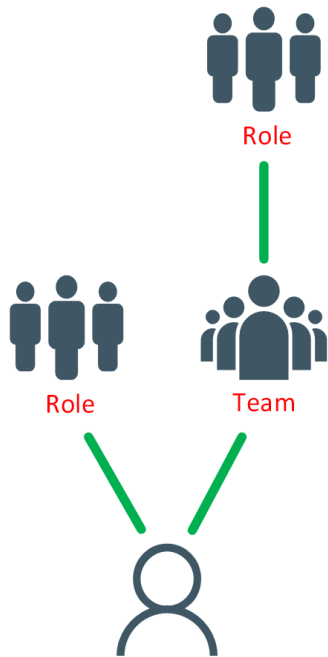
- Identify users by their usage and add them to a Team. For example, dormant users who haven't logged on recently have no personal content and are not the creators of any public content. Another script can then process such teams of users to complete a business requirement. In this example, dormant users can be deleted, saving licenses for other users.
- "Transporting" users, or a team of users, from one SAP Analytics Cloud Service to another (with options to transport the users' Teams and roles)
 - Although strictly speaking, not a single user or team is transported, since the user's personal content or the Team's content isn't transported, it's just their metadata that is 'transported'. Strictly speaking, all the scripts are doing is reading and then creating new or updating existing ones, and it can only do this for the properties the API exposes; this currently excludes the contents of folders, for example. (Another API is available that transports content.)

These samples allow you to benefit from the intelligence built into their design:

- Adding a massive number of users to a Team, for example, becomes complex when the number of users is over 4,500. Updates need to be 'chunked' into multiple calls, each changing in size as the team size changes. Using these scripts means avoiding unnecessary surprises later when the number of users increases over time.
- Some sample scripts provide a different workflow to achieve a similar functional requirement, but each workflow impacts the throughput differently. Guidance is provided to alternate scripts that could be used to achieve a greater throughput. It enables you to design your implementation correctly from the outset without needing to re-do work or conduct a lengthy 'sizing' exercise to determine the best approach. Again, this means you can avoid unnecessary surprises later.

Additionally, all the sample scripts manage the sessions' timeout and automatically re-submit any requests following an error or when a new session is needed. It means session management and most error handling are managed thoroughly. This is particularly helpful, given that some operations can span a considerable amount of time and could frequently span the session timeout of the OAuth client. It also means you benefit from all the known and expected, but perhaps very rare, occurrences of errors from the API without the need to investigate, isolate the cause and resolve the problem yourself.

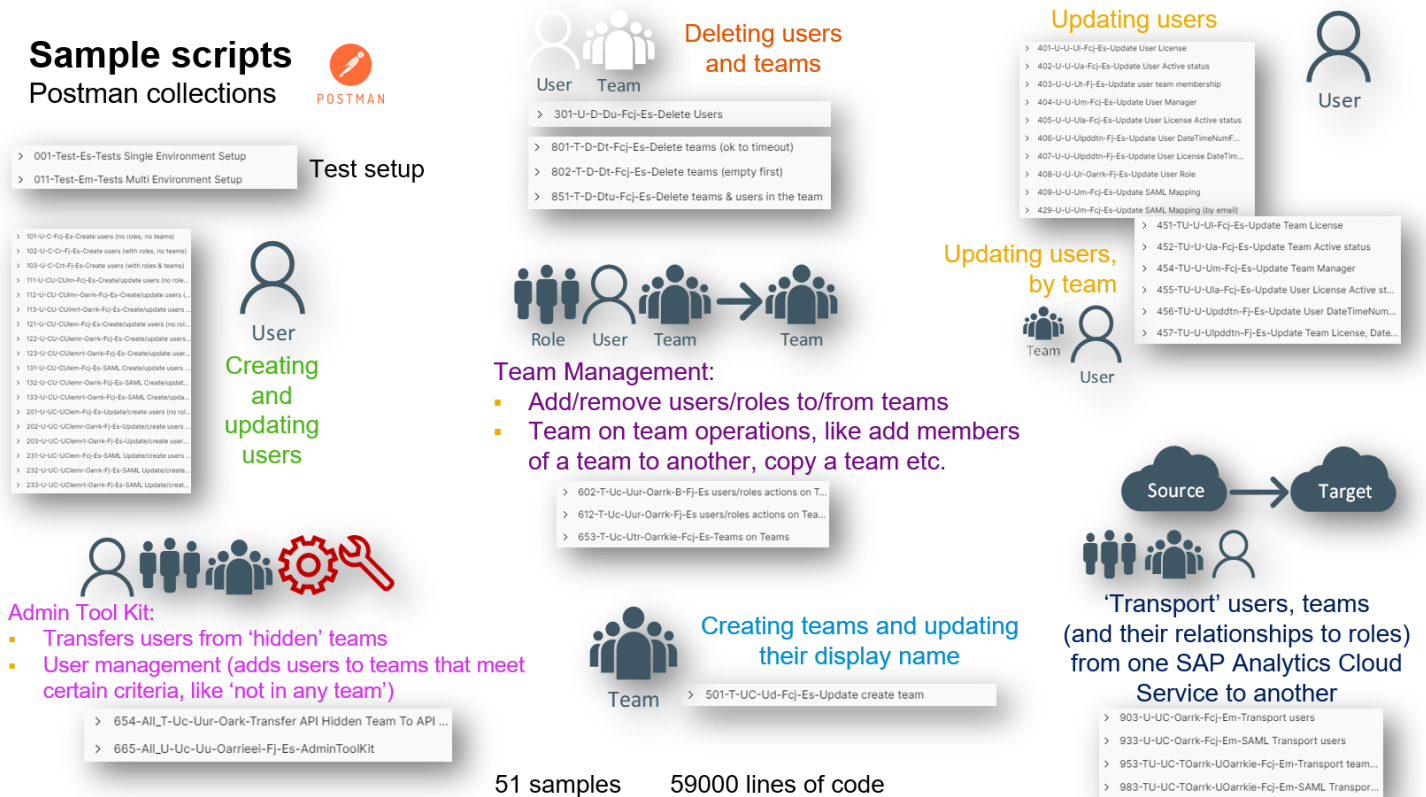
If the reader of this document does not have a comprehensive understanding of the security concepts of SAP Analytics Cloud, then please refer to the articles introduced by these blogs in addition to the official documentation:



- <https://blogs.sap.com/2020/06/19/sap-analytics-cloud-technical-and-administration-overview/>
- <https://blogs.sap.com/2019/06/21/sap-analytics-cloud-security-concepts-and-best-practice/>
- <https://blogs.sap.com/2020/03/10/sap-analytics-cloud-managing-licenses-with-roles-and-teams/>

Understanding the concepts to ensure the proper security model is essential since this will decide which scripts provide the optimal performance. Making the wrong decision could be costly when the number of users increases.

Business Benefits and Overview



The business benefits are summarised in the blog, introducing these sample scripts and the best practices for the SCIM API. Please refer to <https://blogs.sap.com/2021/05/28/sap-analytics-cloud-scim-api-best-practices-and-sample-scripts/>. The blog post also provides a means to ask the community and the author questions about these samples.

A higher-level overview of these sample scripts is available via the [Support Content Help Portal](#).

This user guide is for those keen to use the sample scripts rather than understand what the sample scripts provide regarding business benefits.

Initial Setup with Postman



The initial setup requires the following steps:

- A. Download and install Postman Desktop
- B. Download the samples
- C. Import and define the environments in Postman
- D. Import sample collections into Postman
- E. Run sample collection 1001 to test that the environment is set correctly
- F. (optional) Postman Environments for Multiple SAP Analytics Cloud Services
- G. Set new user default settings

The following sections provide step-by-step instructions on each of these steps.

If you are using the **Embedded Edition** of SAP Analytics Cloud, then:

1. Follow the setup within the Embedded Edition API Sample Scripts for Administration User Guide (accessible via <https://blogs.sap.com/2022/04/07/sap-analytics-cloud-embedded-edition-best-practices-sample-scripts-for-administration/> and [direct download](#))
2. Skip steps A, B and most of step C and go directly to step 24. You will have already completed the equivalent steps in the guide mentioned above. For 'SCIM' workflows, you will share everything from Step 24 onwards with the Enterprise Edition of SAP Analytics Cloud.

Don't forget to review the 'prerequisites for all scripts' section in this user guide.

Step A – download and install Postman

Download Postman from the Postman website. You will need the client <https://www.postman.com/downloads/>. You may also require the Postman Desktop Agent, but that isn't essential and is needed only for the web version of Postman with these scripts.

Postman can be run on a desktop client or via the web using a desktop agent, but Postman also has a command line interface, which is more suitable for enterprise deployments. This document covers only the client setup. For more details on running the Postman command line, please refer to the Postman website.

Step B – download the samples



Download the samples from <https://github.com/SAP-samples/analytics-cloud-scim-api-samples> ([direct zip download](#)).

This GitHub repository contains three sets of samples, each one has its sub-folder:

Description	Subfolder	Documentation	SAP Analytics Cloud Edition
SCIM API for managing users and teams	SCIM	https://d.dam.sap.com/a/kv3cXuB (this document)	Enterprise & Embedded
Administrating the Service	Embedded	https://d.dam.sap.com/a/Wg2pPhA	Embedded only
Export data from acquired data models	Export Data	https://d.dam.sap.com/a/A69ssVS	Enterprise only
Activities Log download	Activities Log	https://d.dam.sap.com/a/L1SUAhp	Enterprise only

The zip download will download all the subfolders, so you only need to work with one of these.

This guide is for the ‘SCIM API’ set of samples in the ‘SCIM’ subfolder.

Within the **SCIM** folder, the samples consist of the following components:

1. The sample scripts themselves. These are Postman collection JSON files; each collection is a sample script. The first release of these samples included 45 collections; thus, each sample has a collection file. Many collections have been added since, and more are likely to be added over time.
2. Environments. These are JSON Postman environment files. The environment defines the username, password, and server details the samples must use. Multiple Postman collections will use a single environment. Two environment files are provided, which act as a ‘template’ for you to complete for your SAP Analytics Cloud Services.
3. Example data files. The Postman collections will use JSON and CSV files as data files. These data files drive the script input. For example, a script that creates users will read a data file containing the users. The example data files provide the structure to which each script expects and provide an excellent basis to amend for your purposes. All but four scripts have data files, meaning there’s at least one example data file for all the other scripts. All scripts can read a JSON file, and a few can read a CSV file. Where the script can read a CSV, an example CSV file is provided in addition to a JSON file.

Step C – import and define the environments into Postman



Before you can use any of the sample scripts, you'll need to import the Postman environments and configure them appropriately.

A summary of the environment variable configuration is:

<u>Variable</u>	<u>Description</u>
SACserviceFQDN	The hostname, fully qualified, of the SAP Analytics Cloud Service
SACtokenFQDN	The hostname, fully qualified, of the OAuth Client Token Service. Importantly, it's the fully qualified hostname, not the full URL.
Username	Username of the OAuth client
Password	Password of the OAuth client without any carriage return at the end!
ContentNamespace	Content Namespace of the SAP Analytics Cloud Service (menu-admin-system configuration)
SACplatform	The platform SAP Analytics Cloud is hosted on. Supported values are "NEO" for SAP Data Centres and "CF" for Cloud Foundry. If you're unsure, use "CF".
SAMLSSO	Defines the Authentication method defined for the SAP Analytics Cloud Service. If you are using the default Identify Provider that comes with SAP Analytics Cloud, then use a value of "default". Otherwise, set this to "userid", "email" or "custom" depending on how you configured SAML Single-Sign-On.
TimeZoneHours	Currently, just 1 sample script (the 'AdminToolKit') uses the 'TimeZone', which can be used in the description of the team it updates. It just makes it more obvious when the team was last updated. You can customise the TimeZone for your organisation with the 'TimeZoneHours' (a number between -23 and 23), the 'TimeZoneMinutes' (either -30, 0 or 30) and the 'TimeZoneDescription' (any text you like). Update these values as you see fit.
TimeZoneMinutes	
TimeZoneDescription	
SCIM_NewUserDefault_preferredlanguage	Default value for new users. Possible values: 'ar', 'bg', 'zh-CN', 'zh-TW', 'hr', 'ca', 'cs', 'cy', 'nl', 'da', 'en', 'en-GB', 'fi', 'fr', 'fr-CA', 'de', 'de-CH', 'el', 'he', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'ms', 'no', 'pl', 'pt', 'pt-PT', 'ro', 'ru', 'sh', 'sk', 'sl', 'es', 'es-MX', 'sv', 'th', 'tr', 'uk', 'vi', 'en-US-sapstd' and possibly more will be added over time!
SCIM_NewUserDefault_active	Default value for new users. Possible values: 'true' or 'false' are valid, unlike 'TRUE' or 'FALSE'
SCIM_NewUserDefault_dataaccesslanguage	Default value for new users. Possible values: 'zz' (default), 'af', 'ar', 'bg', ... many more as shown above in preferredlanguage.
SCIM_NewUserDefault_dateformatting	Default value for new users. Possible values include: 'MMM d, yyyy', 'dd/MM/yyyy', 'MM.dd.yyyy', ... many more!

SCIM_NewUserDefault_timeformatting	Default value for new users. Possible values include: 'H:mm:ss' (24H), 'h:mm:ss A' (12H)
SCIM_NewUserDefault_numberformatting	Default value for new users. Possible values include: '1,234.56', '1.234,56', '1 234,56'
SCIM_NewUserDefault_cleanupnotifications_numberofdays	Default value for new users. Possible values: 0, 1, 2, etc..
SCIM_NewUserDefault_systemnotificationsemailoptin	Default value for new users. Possible values: 'true' or 'false' are valid, unlike 'TRUE' or 'FALSE'
SCIM_NewUserDefault_marketingemailoptin	Default value for new users. Possible values: 'true' or 'false' are valid, unlike 'TRUE' or 'FALSE'

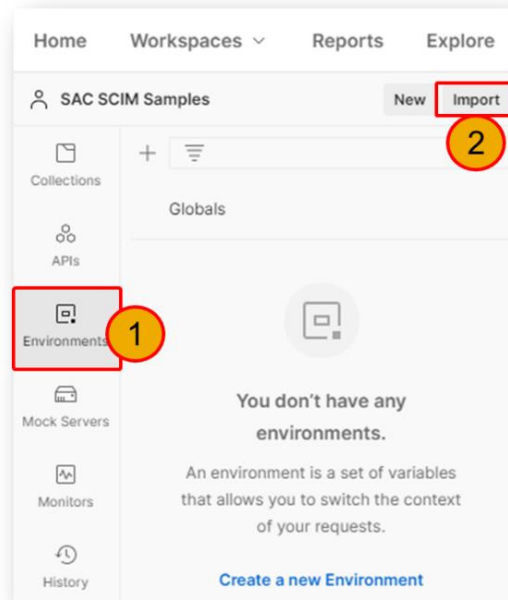
Individual step-by-step instructions are on how to set these variables correctly follows here:



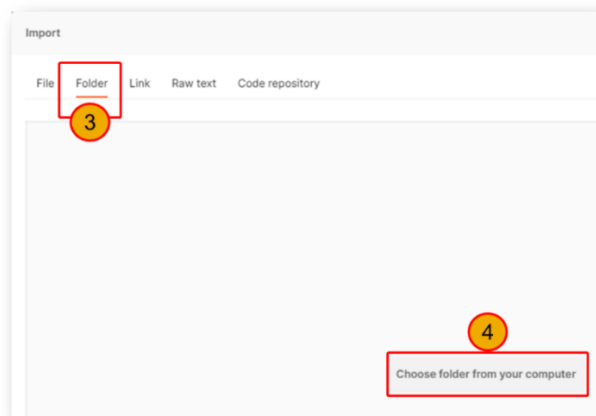
SAP HANA Academy provide a step-by-step tutorial for 'Step C' to 'Step E': <https://youtu.be/4OyLqrJh4ik>

Import environment 'templates'

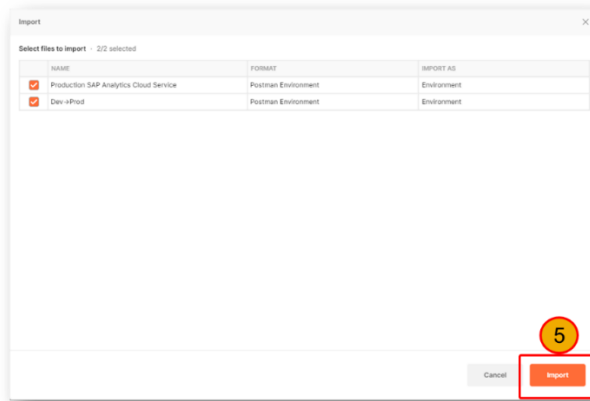
Open Postman client. If you're new to Postman, you'll appreciate to know a 'workspace' has been created for you. You can have multiple workspaces. You'll work with just one workspace for this setup. Environments are stored inside a workspace.



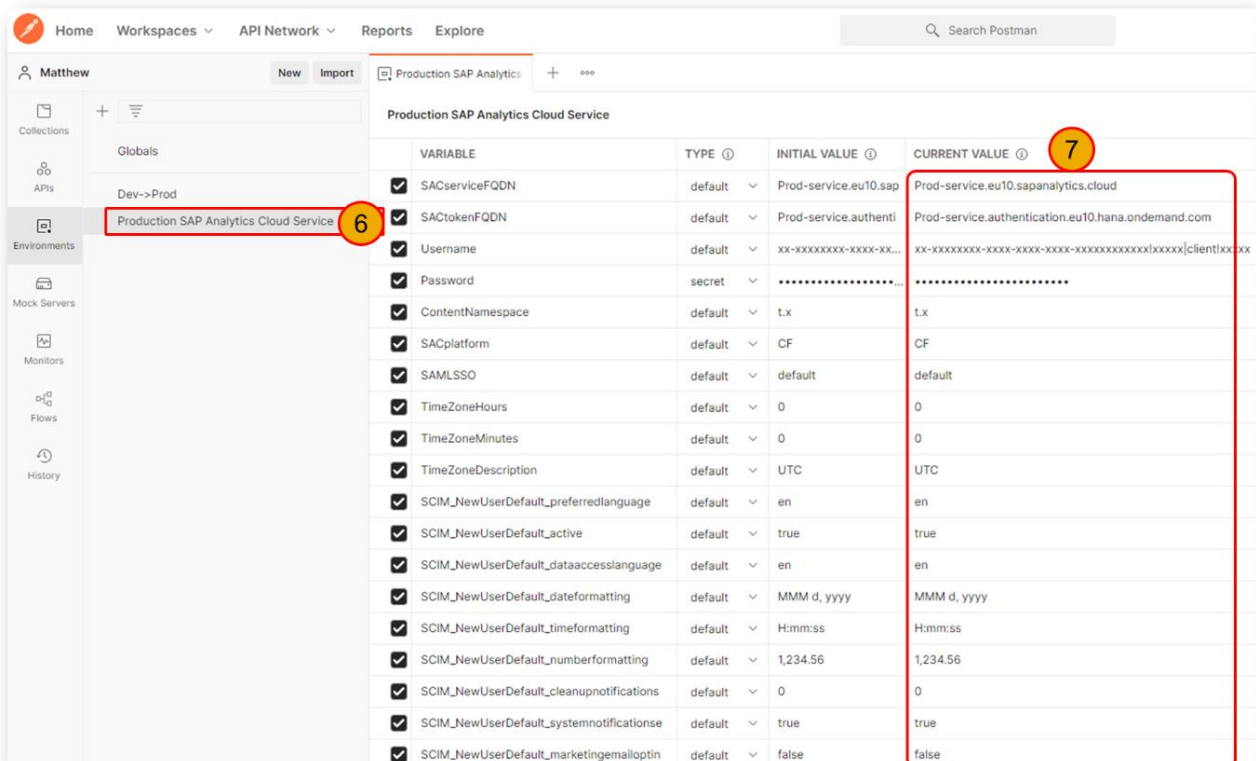
1. Select the 'Environments' tab
2. Select 'Import'



3. Select 'Folder'
4. Select 'Choose folder from your computer'. Then select the folder where you downloaded the environment json files earlier. You need to select the 'SCIM\SCIM Environments' folder and press 'Select Folder'



5. Select Import. This will import the 2 environment json files into your Postman workspace. Each act as a template for you to edit for your SAP Analytics Cloud Service.

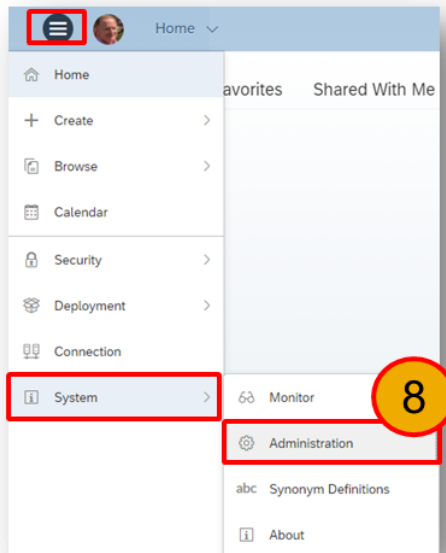


6. Select the environment called 'Production SAP Analytics Cloud Service'
 - a. Optionally, feel free to rename it appropriately.
 - b. If you have more than one SAP Analytics Cloud Service, duplicate this environment, one for each SAP Analytics Cloud Service and name them appropriately. For each of these environments, you'll need to repeat the following steps, one for each SAP Analytics Cloud Service.
7. We need to define this environment for your SAP Analytics Cloud Service(s).
 - a. The environment consists of numerous variables, such as 'Username', 'Password' and many others.
 - b. We need to update the '**Current Value**' of each of these variables. For the moment, ignore the 'Initial Value'
 - c. The following steps will guide you through where to obtain each of these values.

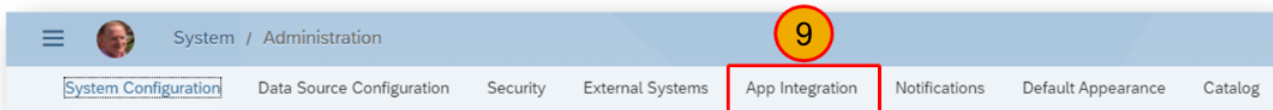
- d. There's nothing to do in this step, but remember that the following steps will require you to update the values shown in box 7.

Create OAuth client to get API user/pass

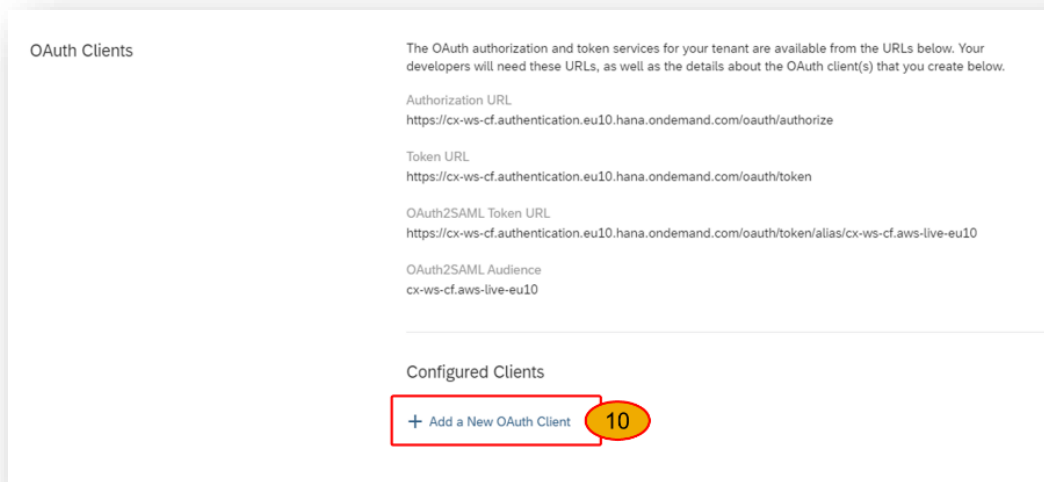
Firstly, we need to create a user in SAP Analytics Cloud that the API will use. This user isn't a regular user but a special 'OAuth' user. We shall create an 'OAuth client', providing us with a username and password. We will then copy these values into the Postman environment (step 7).



8. Login to SAP Analytics Cloud as an administrator and select from the menu, System-Administration



9. Select 'App Integration'



10. Select 'Add a new OAuth Client'

11. Provide a name for the client.
12. Select the Purpose to be 'API Access'
13. Select 'User Provisioning' under 'Access'. For currently just a single sample script, the '1665-AdminToolKit' requires the 'Activities' service. Consider granting this access at this time so save repeating these steps later.
14. Select 'Add'. This will create a new OAuth client, which is a user. It will take a few moments to create the user.

15. The OAuth Client ID is shown, as can be seen in the diagram above. (this is on a Cloud Foundry Platform; for the NEO platform, the client ID is the same as the name you gave in step 11.) Copy this value and paste it into the **Current Value** for the **Username** of the Postman environment from step 7. Be careful to **remove the carriage return** often copied with it when you paste the username!
16. Show the secret, which is the password for the user, and copy this and paste that value as the **Current Value** for the **Password** of the Postman environment from step 7.

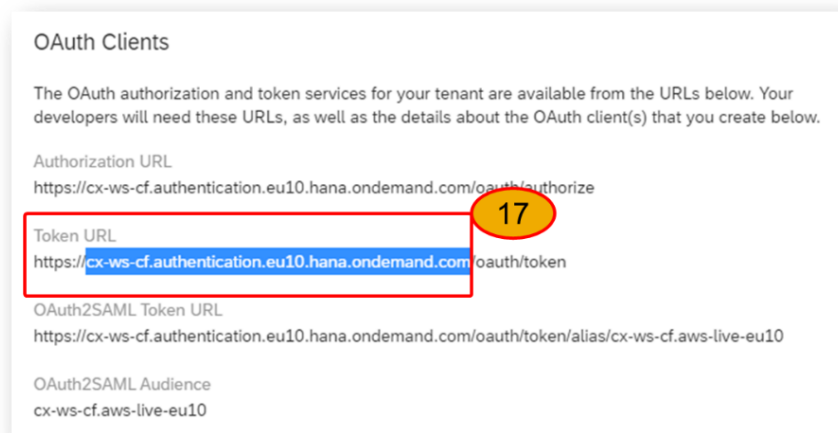
Your Postman environment should now look something like this:

Production SAP Analytics Cloud Service			
	VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ
<input checked="" type="checkbox"/>	SACserviceFQDN	Prod-service.eu10.sapanalytics.cloud	Prod-service.eu10.sapanalytics.cloud
<input checked="" type="checkbox"/>	SACtokenFQDN	Prod-service.authentication.eu10.hana.ondemand.com	Prod-service.authentication.eu10.hana.ondemand.com
<input checked="" type="checkbox"/>	Username	xx-xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxx xxxx client x	sb-c237a359-5b04-48d4-8ec4-c813cd3b854c b43987 client b3650
<input checked="" type="checkbox"/>	Password	xxxxxxxxxxxxxxxxxxxxxx=	58+5s5f4XxbJNgsrwqFga0mK/2Q=
<input checked="" type="checkbox"/>	ContentNamespace	t.x	t.x
<input checked="" type="checkbox"/>	SACplatform	CF	CF
<input checked="" type="checkbox"/>	SAMLSSO	default	default
<input checked="" type="checkbox"/>	TimeZoneHours	0	0
<input checked="" type="checkbox"/>	TimeZoneMinutes	0	0
<input checked="" type="checkbox"/>	TimeZoneDescription	UTC	UTC

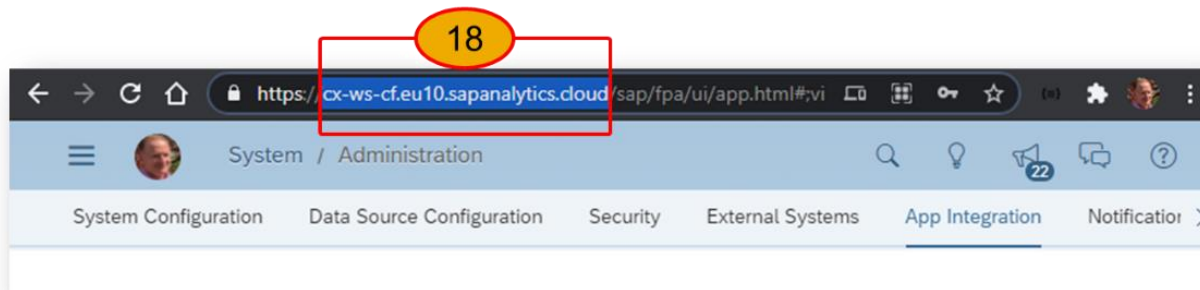
(Other variables starting with the name SCIM_NewUserDefault_ are not shown in the screenshot.)

Enter values into Postman Environment

We have created an OAuth client; we must provide Postman with the correct values to populate the Environment; otherwise, the scripts will not work correctly.



17. Press 'done' to return to the page that shows step 17. Copy the full qualified domain name of the **Token URL** (ignoring the `https://` at the beginning and ignoring the `/oauth/token` at the end. i.e. copy the part highlighted in blue. It will be different for each SAP Analytics Cloud service). Paste this value into the **Current Value** for **SACtokenFQDN** of the Postman environment from step 7.

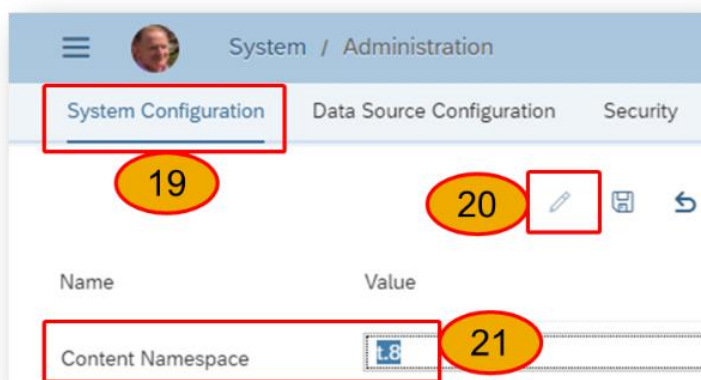


18. Copy the SAP Analytics Cloud service fully qualified domain and paste this value into the **Current Value** for **SACserviceFQDN** of the Postman environment from step 7. Like step 17, ignore the `https://` at the beginning and forget everything at the end.

Your Postman environment should now look something like this:

	VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ
<input checked="" type="checkbox"/>	SACserviceFQDN	Prod-service.eu10.sapanalytics.cloud	c1sacse.eu10.sapanalytics.cloud
<input checked="" type="checkbox"/>	SACtokenFQDN	Prod-service.authentication.eu10.hana.ondemand.	c1sacse.authentication.eu10.hana.ondemand.com
<input checked="" type="checkbox"/>	Username	xx-xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxxxxxxxxxxx c	sb-c237a359-5b04-48d4-8ec4-c813cd3b854c b43987 client b3650
<input checked="" type="checkbox"/>	Password	xxxxxxxxxxxxxxxxxxxxxxxxxxxxx=	58+5s5f4XxbJNgsrwqFga0mK/2Q=
<input checked="" type="checkbox"/>	ContentNamespace	t.x	t.x
<input checked="" type="checkbox"/>	SACplatform	CF	CF
<input checked="" type="checkbox"/>	SAMLSSO	default	default
<input checked="" type="checkbox"/>	TimeZoneHours	0	0
<input checked="" type="checkbox"/>	TimeZoneMinutes	0	0
<input checked="" type="checkbox"/>	TimeZoneDescription	UTC	UTC

(Other variables starting with the name SCIM_NewUserDefault_ are not shown in the screenshot)



19. Select the 'System Configuration' tab.
 20. Select the 'edit' icon (we won't make any changes, but enabling this makes it easy to copy a value!)

21. Copy the 'Content Namespace' value and paste this value into the **Current Value** for **ContentNamespace** of the Postman environment from step 7.

	VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ	
<input checked="" type="checkbox"/>	SACserviceFQDN	Prod-service.eu10.sapanalyti...	c1sacse.eu10.sapanalytics.cloud	
<input checked="" type="checkbox"/>	SACtokenFQDN	Prod-service.authentication....	c1sacse.authentication.eu10.hana.ondemand.com	
<input checked="" type="checkbox"/>	Username	xx-xxxxxxxx-xxxx-xxxx-xxxx...	sb-c237a359-5b04-48d4-8ec4-c813cd3b854c!b43987 client!b3650	
<input checked="" type="checkbox"/>	Password	xxxxxxxxxxxxxxxxxxxxxxxxxx=	58+5s5f4XxbJNgsrwqFga0mK/2Q=	
<input checked="" type="checkbox"/>	ContentNamespace	t.x	t.x	21
<input checked="" type="checkbox"/>	SACplatform	CF	CF	22
<input checked="" type="checkbox"/>	SAMLSSO	default	default	23
<input checked="" type="checkbox"/>	TimeZoneHours	0	0	24
<input checked="" type="checkbox"/>	TimeZoneMinutes	0	0	
<input checked="" type="checkbox"/>	TimeZoneDescription	UTC	UTC	

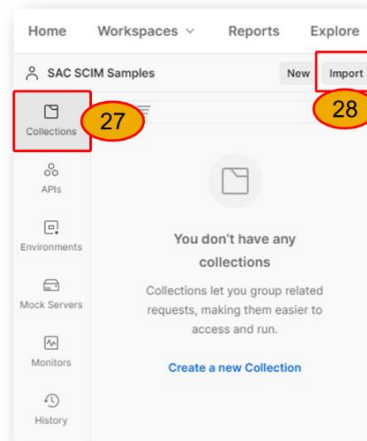
(Other variables starting with the name SCIM_NewUserDefault_ are not shown in the screenshot)

22. The value for '**SACplatform**' will need to be either 'CF' for those on Cloud Foundry or 'NEO' for those running on an SAP Data Centre. If you're unsure, leave the value as 'CF'.
23. The '**SAMLSSO**' defines the Authentication method. If you are using the default Identify Provider that comes with SAP Analytics Cloud, then leave this as 'default'. Otherwise, set this to 'userid' 'email' or 'custom' depending upon how you configured SAML Single-Sign-On.
24. Currently, just 1 sample script (the 'AdminToolKit') uses the 'TimeZone', which can be used in the description of the team it updates. It just makes it more apparent when the team was last updated. You can customise the TimeZone for your organisation with the '**TimeZoneHours**' (a number between -23 and 23), the '**TimeZoneMinutes**' (either -30, 0 or 30) and the '**TimeZoneDescription**' (any text you like). Update these values as you see fit.
25. Now we have set all the Current Value settings; we can persist them. Press 'Persist All'.
26. Press Save

Step D – import sample collections in Postman

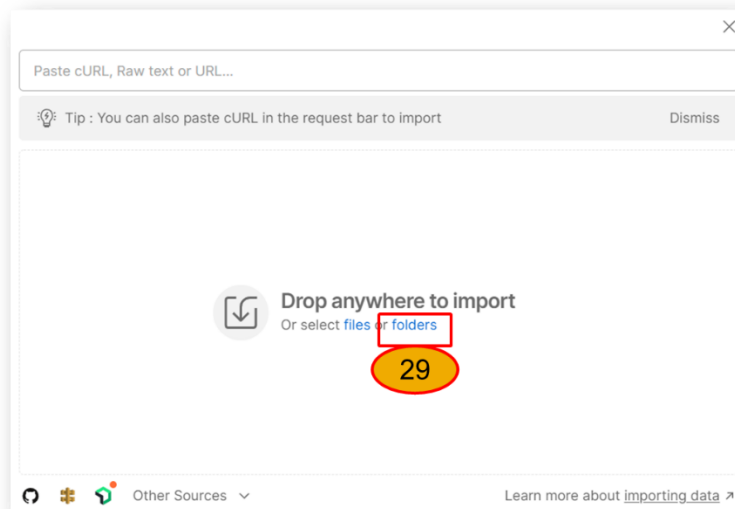


Now the environment has been set up; we can import the sample scripts; each is called a collection in Postman. We can then run a test collection to check the environment we've set is valid.



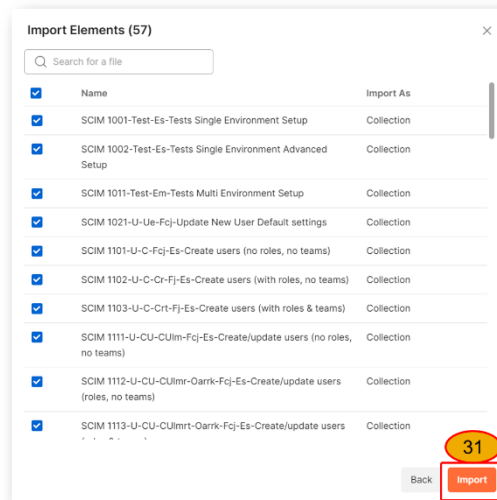
27. In Postman, select the 'Collections' tab.

28. Click 'Import'



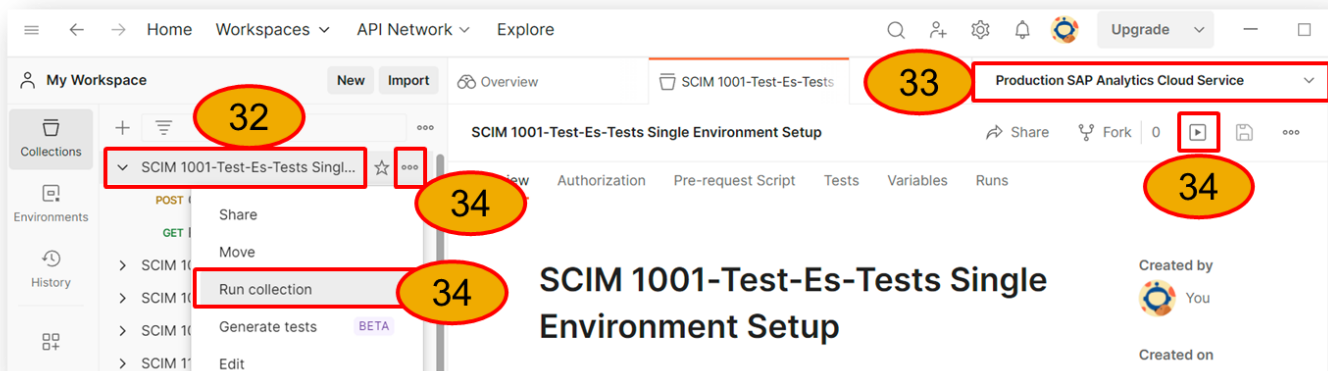
29. Select 'Folder'

30. Select the folder where you downloaded the sample scripts to. You need to choose the 'SCIM\SCIM Samples' folder and press 'Upload'



31. Select 'Import'. This will import all the sample scripts (collections) into your Postman Workspace.

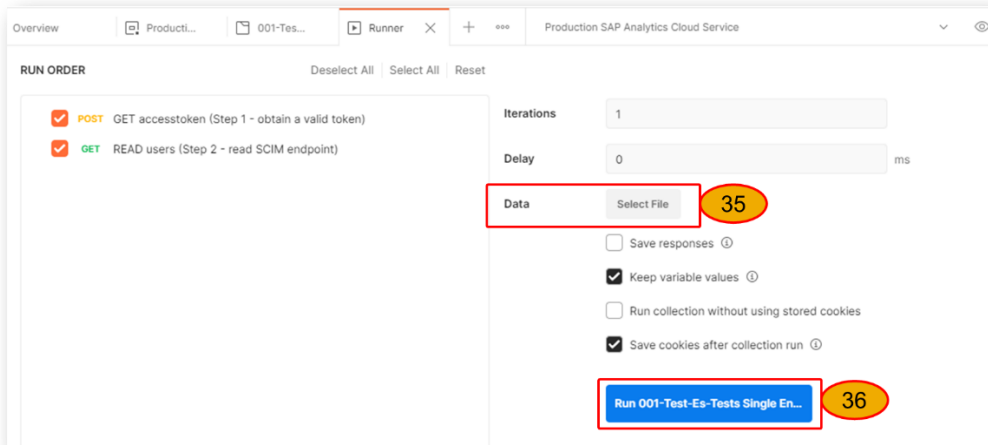
Step E – Run sample collection 1001 to test the environment has been setup correctly



32. Select the collection '1001-Test-Es-Tests Single Environment Setup'. A new window will open for this collection. This collection will perform a straightforward test against the API. It will just log in and then read the number of registered users in SAP Analytics Cloud. The collection is harmless as no updates are performed.

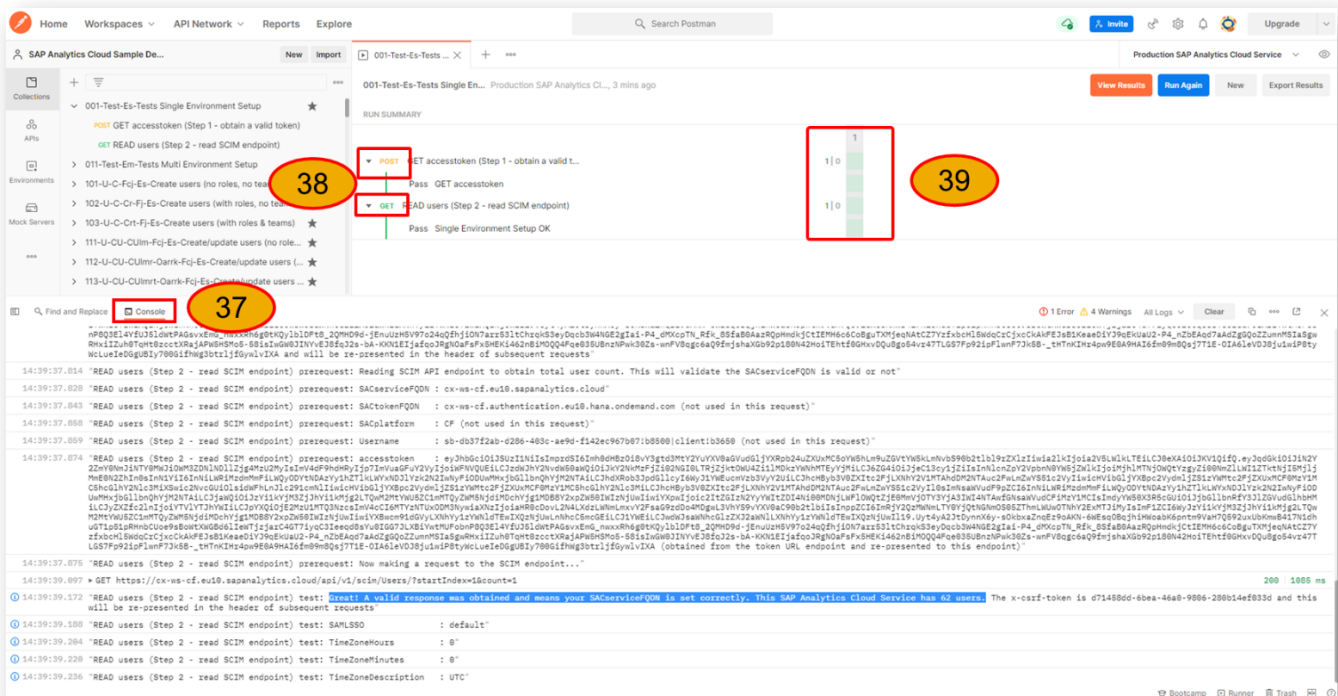
33. Select the environment which was defined in step C. (We need to select the environment before we press run since changing the environment after we press run has no effect).

34. Run collection (having previously selected the correct environment)



35. A new window, called a 'Runner', will open. For this collection, there is no data file; however, this is where you'd select a data file if the collection required it. Step B introduced the example data files. Almost all collections require a data file, so it's important not to miss this step. However, for this collection, no data file is needed.

36. Press Run.



37. Open the Console and enlarge the console window.

- The console shows important information.
- For this collection, the script is particularly verbose and provides a great deal of information to help you determine where any problem may be.

38. Open the 'Run Summary' for each of the requests.

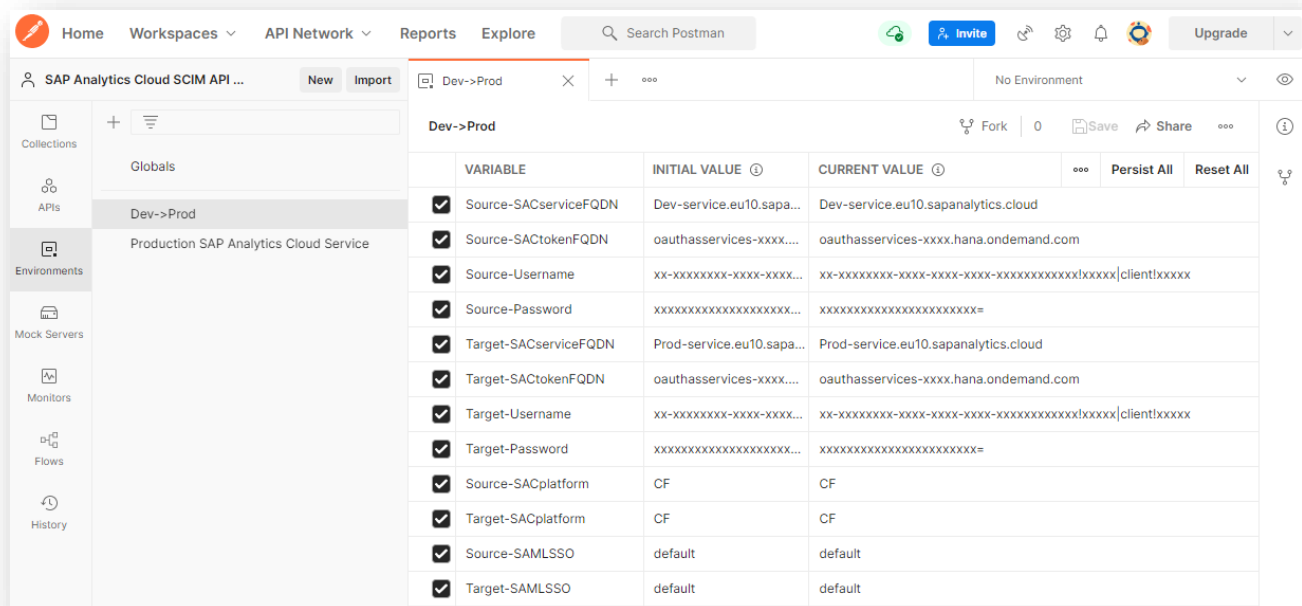
39. You should observe 'pass' tests. If you do, then everything is well, and you've successfully configured the environment and run a simple test. The console will tell you how many users are in your SAP Analytics Cloud Service (highlighted in blue in the screenshot). If either of these tests fail, inspect the console carefully for clues to where you went wrong.

Step F – (optional) Postman Environments for Multiple SAP Analytics Cloud Services



In the previous steps a single environment was configured, or if you have multiple SAP Analytics Cloud Services, then you may have duplicated that environment for each SAP Analytics Cloud and configured its values appropriately.

The second environment, that is provided, is called ‘Dev->Prod’ and has very similar variables to the single environment. However, this environment is for when you have multiple SAP Analytics Cloud Services and you wish to transport meta data about users and teams, using the API, between and across SAP Analytics Cloud Services:



The variable names are duplicated, one set for the Source and another for the Target:

Variable	Description
Source-SACserviceFQDN	The hostname, fully qualified, of the source SAP Analytics Cloud Service
Source-SACtokenFQDN	The hostname, fully qualified, of the source OAuth Client Token Service. Importantly, it's the fully qualified hostname, not the full URL.
Source-Username	Username of the source OAuth client
Source-Password	Password of the source OAuth client without any carriage return at the end!
Source-ContentNamespace	Content Namespace of the source SAP Analytics Cloud Service (menu-admin-system configuration)
Source-SACplatform	The source platform SAP Analytics Cloud is hosted on. Supported values are “NEO” for SAP Data Centres, and “CF” for Cloud Foundry. If you're unsure, use “CF”.

Source-SAMLSSO	“default” for when using the default Identify Provider that comes with SAP Analytics Cloud. “userid”, “email”, or “custom” if using your own Identify Provider for Single Sign-On. This is for the source.
Target-SACserviceFQDN	The hostname, fully qualified, of the target SAP Analytics Cloud Service
Target-SACtokenFQDN	The hostname, fully qualified, of the target OAuth Client Token Service. Importantly, it’s the fully qualified hostname, not the full URL.
Target-Username	Username of the target OAuth client
Target-Password	Password of the target OAuth client without any carriage return at the end!
Target-ContentNamespace	Content Namespace of the target SAP Analytics Cloud Service (menu-admin-system configuration)
Target-SACplatform	The target platform SAP Analytics Cloud is hosted on. Supported values are “NEO” for SAP Data Centres, and “CF” for Cloud Foundry. If you’re unsure, use “CF”.
Target -SAMLSSO	“default” for when using the default Identify Provider that comes with SAP Analytics Cloud. “userid”, “email”, or “custom” if using your own Identify Provider for Single Sign-On. This is for the target.

This means there are scripts that support two SAP Analytics Cloud at the same time and means you can, albeit in a limited way, transport meta data of users and teams from one to another. It also means you can transport between NEO and CF platforms, and any combination.

If you have multiple SAP Analytics Cloud Services and you wish to use scripts to transport users or teams between them, then enter the values appropriately for each of the source and target services into this ‘Dev->Prod’ environment. Rename and duplicate the environment as you see fit. The scripts that require these Source and Target variables are scripts 101x and 19xx.

Once you have defined this environment, test it using sample script 1011.

Step G – Define new user default settings



Before creating users with the script 11xx or 12xx it's important that they are created correctly! This is quite straightforward and does NOT require any code changes (unlike a previous version of these samples that did)

The data files to create users do not specify all the properties of the user that can be set. This was done to keep the data files as simple as possible and to reduce the effort required to create them. It is also likely that all users will share the same initial settings. The settings excluded from the data file are:

- a) Preferred language
- b) Data access language
- c) Date formatting
- d) Time formatting
- e) Number formatting
- f) Clean-up notification number of days
- g) System notification opt-in
- h) Marketing email opt-in

Creating users with the wrong settings would then mean they'd only need to be updated and this could be costly if you have a large user base. It's better to just create the users correctly in the first place.

So, for any script that creates users (except scripts 9xx, which 'transports' a user with all their properties), these settings need to be defined correctly. These new user default settings are stored in the Postman Environment and their names all start with "SCIM_NewUserDefault_":

	VARIABLE	TYPE	INITIAL VALUE	CURRENT VALUE
<input checked="" type="checkbox"/>	SACserviceFQDN	default	Prod-service.eu1...	Prod-service.eu10.sapanalyt
<input checked="" type="checkbox"/>	SACtokenFQDN	default	Prod-service.aut...	Prod-service.authentication.
<input checked="" type="checkbox"/>	Username	default	xx-xxxxxxxx-xxx...	xx-xxxxxxxx-xxxx-xxxx-xxxx
<input checked="" type="checkbox"/>	Password	secret
<input checked="" type="checkbox"/>	ContentNamespace	default	t.x	t.x
<input checked="" type="checkbox"/>	SACplatform	default	CF	CF
<input checked="" type="checkbox"/>	SAMLSSO	default	default	default
<input checked="" type="checkbox"/>	TimeZoneHours	default	0	0
<input checked="" type="checkbox"/>	TimeZoneMinutes	default	0	0
<input checked="" type="checkbox"/>	TimeZoneDescription	default	UTC	UTC
<input checked="" type="checkbox"/>	SCIM_NewUserDefault_preferredlanguage	default	en	en
<input checked="" type="checkbox"/>	SCIM_NewUserDefault_active	default	true	true
<input checked="" type="checkbox"/>	SCIM_NewUserDefault_dataaccesslanguage	default	en	en
<input checked="" type="checkbox"/>	SCIM_NewUserDefault_dateformatting	default	MMM d, yyyy	MMM d, yyyy
<input checked="" type="checkbox"/>	SCIM_NewUserDefault_timeformatting	default	H:mm:ss	H:mm:ss
<input checked="" type="checkbox"/>	SCIM_NewUserDefault_numberformatting	default	1,234.56	1,234.56
<input checked="" type="checkbox"/>	SCIM_NewUserDefault_cleanupnotificationsnumberofdays	default	0	0
<input checked="" type="checkbox"/>	SCIM_NewUserDefault_systemnotificationsemailoptin	default	true	true
<input checked="" type="checkbox"/>	SCIM_NewUserDefault_marketingemailoptin	default	false	false

You have 2 options:

- i. You can manually update the values in the Postman environment. However, please be cautious as they must be valid entries, otherwise an error will occur when trying to create a new user. Please refer to the official documentation for a full list of supported values: [Official reference](#). (See next option to avoid any problems here!)
- ii. Run sample script “1021-U-Ue-Fcj-Update New User Default settings” with a datafile containing the userid, of the user, which has the settings you’d like to copy. This script will then read that user and copy whatever that user’s values are, for all these parameters, and store them inside your Postman Environment. It means there’s a kind of ‘default user’ concept. Before you run this script remember to update the provided sample data file with the userid you’d like the default values to be read from.
 - o Follow the step-by-step instructions you’ve already followed from step 32 onwards but use sample script ‘1021’ and at step 35 provide your updated sample data file.

If you do have groups of users that require different values for these settings, then either duplicate the Postman Environment, one for each group of users and adjust the settings for each environment accordingly, or simply re-run “1021-U-Ue-Fcj-Update New User Default settings” with a data file containing the userid of the user you’d like subsequent new users to inherit.

The settings defined in the template Postman Environment are the default settings of SAP Analytics Cloud.

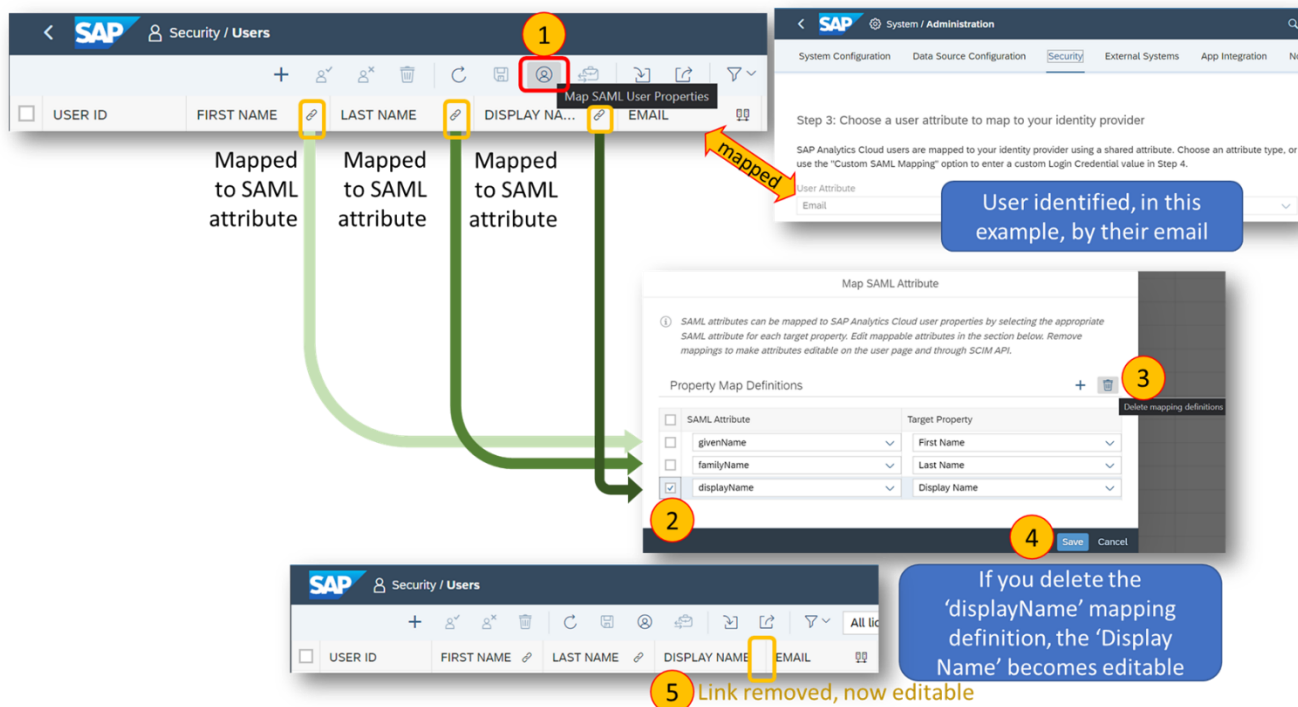
Properties used only for user creation then ignored

Scripts that create users read properties from the data file, and these include:

file_givenname	string	Initial given name
file_familyname	string	Initial family name
file_displayname	string	Initial display name

It’s helpful to note that these properties are the initial values, and once set, they are typically not updated via the API because the Identity Provider (IDP) becomes the authority of these values. These samples do not edit these three values once set. Only the Identity Provider can change them unless they are unmapped. Typically, a user can update these values themselves, but that depends on the IDP. SAML attributes update these mapped properties upon a user login.

If you use your own Identify Provider and configure SAML Single-Sign-On to map on a **User ID** or **custom attribute**, then the e-mail is also not updatable; only the Identify Provider can update the e-mail in this case, again because it’s a mapped value.



The 'Menu-Security-Users' screen shows a 'link' icon for all the mapped attributes.

In this example, the e-mail address is the property the users are identified with, and you can see that by the icon on that column. If you had 'USERID' or 'CUSTOM', that icon will be on a different column and match the 'Menu-System-Administration-Security-SAML SSO-Step 3-User Attribute'. If you are using 'CUSTOM', another column 'SAML MAPPING' will be shown in the menu-security-users screen.

The property you are mapping users on, in this case, e-mail, is a property you can edit if and only if it's e-mail or custom. You can't edit that if it's USERID because user IDs are not editable.

Other properties are 'First Name', 'Last Name', 'Display Name' and, by default, are not editable because they are mapped. The button (1) 'Map SAML User Properties' lets you edit the mapped attributes.

Opening the dialogue to show the 'Property Map Definitions', allow you to delete any or all of these mappings. In this example, the 'displayName' mapping (2) (3) is deleted and saved (4).

As the 'Display Name' is now unmapped, the 'link' icon next to the column is removed (5), which means you can update it. You may re-map the attribute after making an ad-hoc change.

Introduction to sample scripts (collections)

The samples fall into two main categories:

- Samples that work with a single SAP Analytics Cloud Service.
- Samples that work with two SAP Analytics Cloud Services, a source, and a target.

SAP HANA Academy hands-on video tutorials



The SAP HANA Academy has made available videos demonstrating how these sample scripts work.

Please refer to the [introduction blog post](#), which overviews the video series. Please follow the blog post and subscribe to the related YouTube channel for updates.

Sample Collections for a single SAP Analytics Cloud Service

This introduces the scripts that can be used with the environment that is for a single SAP Analytic Cloud Service, i.e., the one where step-by-step instructions were provided earlier in step C:

All scripts require an environment containing at least these variables:

<u>Variable</u>	<u>Description</u>
SACserviceFQDN	The hostname, fully qualified, of the SAP Analytics Cloud Service
SACtokenFQDN	The hostname, fully qualified, of the OAuth Client Token Service. Importantly, is the fully qualified hostname, not the full URL.
Username	Username of the OAuth client
Password	Password of the OAuth client without any carriage return at the end!
ContentNamespace	Content Namespace of the SAP Analytics Cloud Service (menu-admin-system configuration)
SACplatform	The platform SAP Analytics Cloud is hosted on. Supported values are “NEO” for SAP Data Centres, and “CF” for Cloud Foundry. If you’re unsure, use “CF”.
SAMLSSO	Defines the Authentication method defined for the SAP Analytics Cloud Service. If you are using the default Identify Provider that comes with SAP Analytics Cloud, then use a value of “default”. Otherwise, set this to “userid”, “email” or “custom” depending upon how you configured SAML Single-Sign-On.

Additionally, scripts that create users (1xx and 2xx) require 9 variables to be defined. These variables store the default values for new users:

SCIM_NewUserDefault_*	There are 9 variables that store the settings used when new users are created. These variables are documented in “Step G – Define new user default settings”. If your Postman environment does not yet have these variables defined, then sample scripts 1011 and 1021 will create them for you.
-----------------------	--

Apart from collection 1001 (and 1002, 1011 and a few others), all scripts require input data files to drive the script.

For those who are new to Postman, it’s essential to understand that the first request in any one collection is always made for each row of the data file. It means if the first call is a ‘user’ request, then a user request is made for each row/entry of the data file. If the first request is a ‘team’, then a team request is made for each row/entry of the data file, etc. This, in turn, means some collections operate on a ‘user’ basis and some on a ‘team’ basis.

A detailed section for each sample collection is provided later in this document. It contains more details on when each script should be used, how the script works, what the input data file requirements are and much more. However, there are a good number of scripts, so this section provides a slightly higher-level view to help you navigate which script is most suitable for your use case. Please refer to the detailed section for each sample for more details. Remember that for each sample script, sample CSV and JSON data files are provided as a template for you to copy.

Before running any of the non-test scripts, please ensure you’ve read and understood the requisites for creating users and teams and how to manage roles. These are described later in this document.

Test

There is one sample collection to test that your environment is set correctly:

- 1001-Test-Es-Tests Single Environment Setup

This is already introduced in the step-by-step instructions. This is ideal for validating that the environment is correctly set and that you can log in successfully. It won’t perform any updates to your SAP Analytics Cloud Service; it’s completely harmless!

Set New User Default Settings

- 1021-U-Ue-Fcj-Update New User Default settings

Already introduced in the step-by-step instructions in Step G. This script reads an existing user within SAP Analytics Cloud and stores nine user properties (date/time/number formats, etc.) within the Postman Environment. These values are used when creating new users.

Creating users (user basis)

There are 3 sample collections to create users:

- 1101-U-C-Fcj-Es-Create users (no roles, no teams)
- 1102-U-C-Cr-Fj-Es-Create users (with roles, no teams)
- 1103-U-C-Crt-Fj-Es-Create users (with roles & teams)

These are ideal when you want to just create users and you'd like Postman to show a 'fail' test if a duplicate user is found (matched on the user id and the email address).

Creating and updating users (user basis)

There are 6 sample collections to create and update users:

- 1111-U-CU-CUlm-Fcj-Es-Create-update users (no roles, no teams)
- 1112-U-CU-CUlmr-Oarrk-Fcj-Es-Create-update users (roles, no teams)
- 1113-U-CU-CUlmrt-Oarrk-Fcj-Es-Create-update users (roles & teams)
- 1121-U-CU-CUlem-Fcj-Es-Create-update users (no roles, no teams)
- 1122-U-CU-CUlemr-Oarrk-Fcj-Es-Create-update users (roles, no teams)
- 1123-U-CU-CUlemrt-Oarrk-Fcj-Es-Create-update users (roles & teams)

These are ideal when you want to create users and update existing ones, but you're doing more creates than updates. However (for all scripts 11xx), if the user already exists and you are updating the email address, a new 'duplicate' user will* be created (* depends on how SAML SSO is configured). Samples 112x will detect if a duplicate user was created, and it will delete this user before correctly updating the user's email and assigning the user to any teams.

Creating and updating users (including updating the email address) (user basis)

There are 3 sample collections to update and create users. These are useful when you're performing more updates than creates as the throughout will be greater than the 111x scripts:

- 1201-U-UC-UClem-Fcj-Es-Update-create users (no roles, no teams)
- 1202-U-UC-UClemr-Oarrk-Fj-Es-Update-create users (with roles, no teams)
- 1203-U-UC-UClemrt-Oarrk-Fj-Es-Update-create users (with roles & teams)

Scripts 1123 and 1203 are likely to be your go-to scripts for creating and updating users as they complete all the aspects of a user, including updating the email address. Use script 1123 when creating at least 52% of the users in your data file; otherwise, use script 1203. If you're updating emails, then this figure rises to 77%. Do NOT use scripts 1123 or 1203 if you're updating teams. You'd gain significantly more outstanding performance operating scripts 16xx if all you want to do is update which teams a user is to be a member of. The difference in performance can be genuinely remarkable: 0.26 seconds per user compared to 21.9 seconds per user!

Creating and updating users (SAML variants) (user basis)

The scripts mentioned above to create and update users (110x, 111x, 112x, 120x) are generally suitable (and always ideal if you're using the default authentication method or configured a custom SAML SSO and mapping on 'user id'). However, the SAP Analytics Cloud user ID created may not be the one you desire when creating new users, and this is especially true when using SAML SSO and mapping on the email address or a custom property that isn't the same as the user ID. Then, the SAP Analytics Cloud user ID created is derived from either the email or from the user ID. ('MATTHEW' would be the user ID when creating a new user with the email matthew@sap.com, and SAML SSO is mapped to e-mail). If you wish to determine what the SAP Analytics Cloud user ID should be, then use the 'SAML' variants:

- 1131-U-CU-CUlem-Fcj-Es-SAML Create-update users (no roles, no teams)
- 1132-U-CU-CUlemr-Oarrk-Fcj-Es-SAML Create-update users (roles, no teams)
- 1133-U-CU-CUlemrt-Oarrk-Fcj-Es-SAML Create-update users (roles & teams)
- 1231-U-UC-UClem-Fcj-Es-SAML Update-create users (no roles, no teams)
- 1232-U-UC-UClemr-Oarrk-Fj-Es-SAML Update-create users (with roles, no teams)
- 1233-U-UC-UClemrt-Oarrk-Fj-Es-SAML Update-create users (with roles & teams)

These samples will create a user ID of your choice rather than one derived by SAP Analytics Cloud. These 'SAML' variants can be used even when the default Identify Provider is used, though there's no real advantage.

Defining the user ID of your choice can be helpful as these IDs determine which users should be in a Team, and if the ID is only known by SAP Analytics Cloud, then this could be problematic. If you don't yet use a custom identity provider but later wish to, then there are benefits of using the proper user ID from the outset. There are also benefits of consistent user IDs across multiple SAP Analytics Cloud Services rather than occasional mismatches throughout the landscape.

A self-determined user ID has a few other benefits, and these include:

- Users make sense of the user ID in the user interface.
- It is easier to control as row-level security for acquired data models is based on the user ID.
- Gain insights easier as the Activities log identifies the user by the user ID.
- Ensure sharing permissions work as expected when artefacts are transported about the landscape since sharing permissions use the user ID.

Summary for which create/update user scripts to use depending on your SAML SSO setup

This table provides the recommended guidance for which scripts to use:

Authentication Type	Recommended	Try to avoid
Default	1101, 1102, 1103 1121, 1122, 1123 1201, 1202, 1203	1111, 1112, 1113
SAML SSO mapped on 'userid'	1101, 1102, 1103 1111, 1112, 1113 1201, 1202, 1203	
SAML SSO mapped on 'email' or 'custom'	1131, 1132, 1133 1231, 1232, 1233	1101, 1102, 1103 1111, 1112, 1113 1201, 1202, 1203

If you are in doubt, then sample scripts 1231, 1232, and 1233 are always safe, just not necessarily the most efficient.

Delete users (user basis)

There's just one sample collection to delete a user on a user basis:

- 1301-U-D-Du-Fcj-Es-Delete Users
- 1311-U-D-Du-Fcj-Es-Delete Users (by saml mapping)
- 1321-U-D-Du-Fcj-Es-Delete Users (by email)

It just deletes users. Need to delete any manager after having deleted all the users of that manager. 1311 and 1321 do the same as 1301, except you can specify the user by their saml mapping property or email.

User Updates (user basis)

There are 11 sample collections to update user properties:

- 1401-U-U-Ui-Fcj-Es-Update User License
- 1403-U-U-Ut-Fj-Es-Update user team membership
- 1404-U-U-Um-Fcj-Es-Update User Manager
- 1406-U-U-Ulppddtn-Fj-Es-Update User DateTimeNumFormat DataAccessLang
- 1407-U-U-Ulppddtn-Fj-Es-Update User License DateTimeNumFormat DataAccessLang
- 1408-U-U-Ur-Oarrk-Fj-Es-Update User Role
- 1418-U-U-Ur-Oarrk-Fj-Es-Update User Role (by saml mapping)
- 1428-U-U-Ur-Oarrk-Fj-Es-Update User Role (by email)
- 1409-U-U-Um-Fcj-Es-Update SAML Mapping
- 1419-U-U-Um-Fcj-Es-Update SAML Mapping (by saml mapping)
- 1429-U-U-Um-Fcj-Es-Update SAML Mapping (by email)

All these properties, like BI licence, team membership, manager, etc., are all properties of a user, though not all of them can be updated on the user endpoint with version 1 of the API, unlike version 2!

Version 1 cannot update the user's team membership with the user's endpoint. It means a different API version 1 endpoint is needed to update the team membership of a user, although reading a user does show the current user's team membership. To edit a team membership with API version 1, the /groups endpoint is needed, which updates the team, not the user.

Script 1403 may appear to be particularly handy for updating the team membership of a user, given this is most likely the user property that will change with the most significant frequency. However, these scripts, including 1403, are user-based and will require a request per user. Although script 1403 performs an intelligent 'batching' of users when it comes to updating the team itself, it updates the Team every 500 users rather than once per user; the throughput will be significantly lower compared to the team-based equivalent, scripts 16xx. You should avoid this script, 1403, wherever possible if you have large volumes of users to update. However, script 1403 is ideal for removing all teams from a user when you don't know which teams the user is a member of.

For other properties, other than the team membership, you can only update these properties on a per-user basis.

Samples 1418 and 1428 do the same as 1408, except you can identify the user by their SAML mapping or e-mail. It means you could, for example, remove all roles from a user, given their e-mail address.

The same applies to 1409 with 1419 and 1429. It means you could update, for example, a user's SAML mapping given their current SAML mapping or their e-mail.

Version 2 of the API can update the team membership of a user on the user's endpoint; however, the sample scripts for these have yet to be developed and made available. Stay tuned for updates and follow the blog for those updates.

User Updates (user basis but given a team)

There are 4 sample collections to update users, but they get the user list from a team rather than a direct list of users.

- 1451-TU-U-UI-Fcj-Es-Update Team License
- 1454-TU-U-Um-Fcj-Es-Update Team Manager
- 1456-TU-U-Upddtn-Fj-Es-Update User DateTimeNumFormat DataAccessLang
- 1457-TU-U-UIpddtn-Fj-Es-Update Team License, DateTimeNumFormat, DataAccessLang
- 1458-TU-U-Ur-Oarrk-Fj-Es-Update Role for Each User of Team

Scripts 140x are user-based; they require the data file to list each user, one user per entry/row. These scripts remain on a user basis, but instead of reading a data file containing an entry/row per user, the data file provides the team's name. The team members provide the list of users to be updated, and then each user is individually updated as they were for the scripts 140x. There is no script 1453, as this would be a team-on-team operation covered by scripts 16xx.

Scripts 1456 and 1457 should be unnecessary as these properties, like datetime formats and language, should have been set when creating users. See the requisites for creating users later in this document. These scripts are thus handy if you forgot to do that!

List all Teams

- SCIM 1601-All_T-List all teams
- SCIM 2601-All_T-List all teams

Scripts x601 lists all teams to the Postman console.

1601 uses API version 1, and 2601 uses API version 2.

Neither of these scripts make any changes; they are pretty harmless.

Create or Update Teams (team basis)

There's just one sample collection to create a Team:

- 1501-T-UC-Ud-Fcj-Es-Update create team

Create a Team or update a Team with a display name of your choice. All scripts will create a Team if the team doesn't exist. However, all other scripts do not allow you to specify the display name. This was done to keep the data files as simple as possible. Instead, use this script to create the Team or update the display name of an existing Team if you don't want to accept the default display name.

Add/remove users and roles to Teams (team basis)

There are 2 sample collections to assign users and roles to teams:

- 1602-T-Uc-Uur-Oarrk-B-Fj-Es users-roles actions on Teams
- 1612-T-Uc-Uur-Oarrk-Fj-Es users-roles actions on Teams

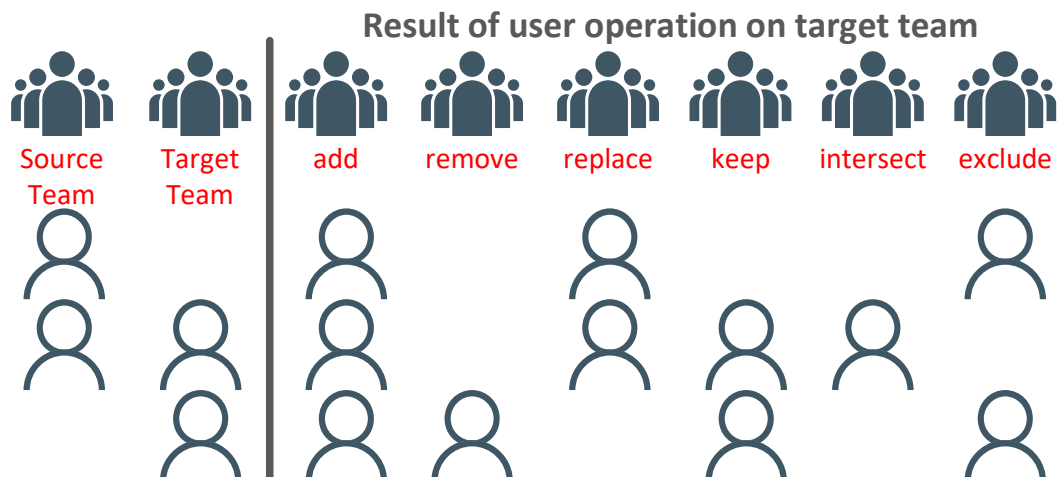
These samples read a list of users (or roles) and add, remove, or replace them on a team. Sample script 1612 'batches' the requests, and 1602 doesn't. Batching is handy when the data file may contain the same team on multiple entries, as it will improve the throughput.

Teams on Teams (team basis)

There is one sample collection for team-on-team operations:

- 1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams

This versatile script enables you to perform many operations and is best described by this image:



The data file specifies a source and a target team. Ignoring the 'keep' operation enables you to perform five different operations that update the target team as appropriate.

The script will intelligently determine what users need to add or remove, removing users first before adding new users to ensure the most significant throughput. The script intelligently chunks the updates, providing the maximum throughput, whilst ensuring each request is completed within the 5-minute timeout window. The script could run for a long time, but the timeouts are entirely managed, including any session timeout. The chunk size is dynamically adjusted as the script progresses depending on the API's performance and whether the users are added or removed. It also works for role membership of the team, and you can mix and match the user and role operations as you please. Any changes to the role assignment are also performed at the optimal time, between users being removed and others being added, to ensure the most significant throughput.

Although not mentioned before, all team updates benefit from the intelligence explained here. However, since these other scripts are user-based and batch the team requests into 500 users at a time, it's unlikely all these features will be utilised, but they all benefit from the resilience of their design.

Deleting teams (team basis)

There are 3 sample collections to delete Teams:

- 1801-T-D-Dt-Fcj-Es-Delete teams (ok to timeout)
- 1802-T-D-Dt-Fcj-Es-Delete teams (empty first)
- 1851-T-D-Dtu-Fcj-Es-Delete teams & users in the team

These sample scripts provide various options to delete teams, including script 1851, which enables you to optionally delete the teams' users and optionally the Team itself or any combination.

Admin Tool Kit (whole user/team scan)

These samples are for when things go wrong or to ease the administration of users by further automation.

2 sample collections perform 'whole' scans of the users or teams:

- 1654-All_T-Uc-Uur-Oark-Transfer API Hidden Team To API Created Team
- 1665-All_U-Uc-Uu-Oarrieei-Fj-Es-AdminToolKit

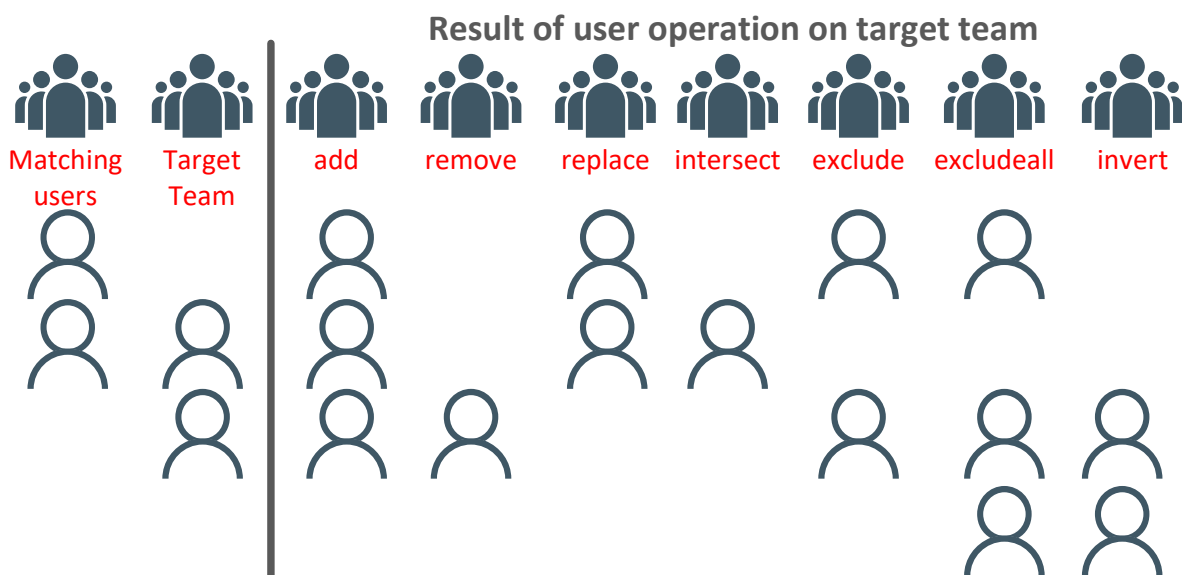
Sample 1654 reads a team that can't be managed by the API and transfers the users and role membership of these Teams to a Team that can be managed via the API. It could save a great deal of re-work, but the script is redundant primarily now, thanks to a new setting 'Ignore Content Namespace For Teams' described later in this document.

Sample 1665 creates teams and adds users to the team that meet specific criteria. It scans all the users defined in SAP Analytics Cloud, identifying all users against criteria. For those users that match the requirements, then the user is added or removed. You can quickly identify users by adding them to a team to update or process them with another sample script! There are some 18 different types of 'tests' and include things like:

- Users with Business Intelligence 'concurrent session' license
- Users who are managers
- Users without a role or a team
- Users with an e-mail domain @sap.com
- Users recently created or users created at least x days ago
- Users not logged on for x days or users that have at least y logins within the last z days
- Users with or without private folder or public folder content
- Etc. etc.

The sample enables both 'and' and 'or' operators between the different tests, allowing for complex logic, such as 'users not in a team AND created in the last week'.

Additionally, there are actions for the matching users upon the team:



These actions include 'replace', which removes all users not fulfilling the defined test, and 'invert', which is simply the opposite of the matching users.

There are many use cases that this AdminToolKit enables on its own, for example, adding a user to a Team where the user isn't in any Teams.

The AdminToolKit enables additional use cases that require the combination of this with other sample collections; for example, users who are managers must be transported before users who have a manager. Conversely, users that are managers must be deleted after users that have a manager. The AdminToolKit also helps manage users' life cycles by identifying dormant users, which can then be deleted by another sample script, saving licenses. For an overview on managing dormant users please visit this [blog](#).

These other use cases are best described in the 'Scenarios' section.

Sample Collections for multiple SAP Analytics Cloud Services

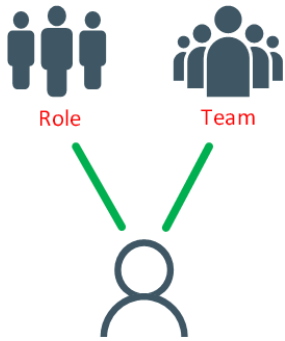
This section introduces scripts with the environment for multiple SAP Analytic Cloud Services. All scripts require an environment containing at least these variables:

<u>Variable:</u>	<u>Description:</u>
Source-SACserviceFQDN	The hostname, fully qualified, of the source SAP Analytics Cloud Service
Source-SACtokenFQDN	The hostname, fully qualified, of the source OAuth Client Token Service. Importantly, it's the fully qualified hostname, not the full URL.
Source-Username	Username of the source OAuth client
Source-Password	Password of the source OAuth client without any carriage return at the end!
Source-ContentNamespace	Content Namespace of the source SAP Analytics Cloud Service (menu-admin-system configuration)
Source-SACplatform	The source platform SAP Analytics Cloud is hosted on. Supported values are "NEO" for SAP Data Centres and "CF" for Cloud Foundry. If you're unsure, use "CF".
Source -SAMLSSO	"default" for when using the default Identify Provider that comes with SAP Analytics Cloud. "userid", "email", or "custom" if using your own Identify Provider for Single Sign-On. This is for the source.
Target-SACserviceFQDN	The hostname, fully qualified, of the target SAP Analytics Cloud Service
Target-SACtokenFQDN	The hostname, fully qualified, of the target OAuth Client Token Service. Importantly, it's the fully qualified hostname, not the full URL.
Target-Username	Username of the target OAuth client
Target-Password	Password of the target OAuth client without any carriage return at the end!
Target-ContentNamespace	Content Namespace of the target SAP Analytics Cloud Service (menu-admin-system configuration)
Target-SACplatform	The target platform SAP Analytics Cloud is hosted on. Supported values are "NEO" for SAP Data Centres, and "CF" for Cloud Foundry. If you're unsure, use "CF".
Target -SAMLSSO	"default" for when using the default Identify Provider that comes with SAP Analytics Cloud. "userid", "email", or "custom" if using your own Identify Provider for Single Sign-On. This is for the target.

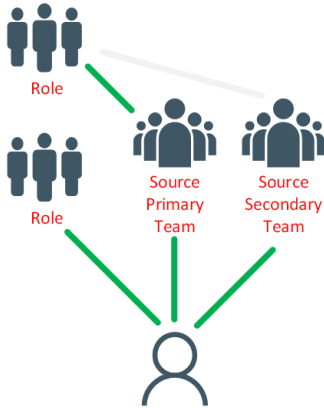
There are just 3 scripts that work across multiple SAP Analytics Cloud Services:

- 1011-Test-Em-Tests Multi Environment Setup
- 1903-U-UC-Oarrk-Fcj-Em-Transport users
- 1953-TU-UC-TOarrk-UOarrkie-Fcj-Em-Transport teams and users

Script 1011 is comparable with 1001 and is ideal for validating the environment is set correctly and that you can login to both source and target successfully.



Script 1903 ‘transports’ users from one SAP Analytics Cloud Service to another, with or without this role and team assignment. This is ideal for any number of users to be ‘transported’.



Script 1953 has the advantage that you can transport a whole team of users, it means the data file specifies just the team’s name, rather than a list of the users as is necessary with script 903.

This script is particularly versatile as it can also transport the team’s membership it has to roles, and it can transport the team membership of a team without necessarily transporting the actual users themselves (number of users doesn’t matter).

For transporting the actual users’, script 1953 should only be used when the number of users is fewer than about 100. Under this condition, you may need to use a combination of 1903 and 1953.

Similarly, to the create/update sample collections (110x, 111x, 112x, 120x) these samples 1903 and 1953 are generally suitable (and always suitable if you’re using SAML SSO and mapping on ‘user id’). However, there are also sample collections for the SAML variant, named 1933 and 1953 that ensure the user id is the same and consistent across your landscape. It’s always safer to use this SAML variant, although the throughput is much lower.

Scenarios

A scenario is a straightforward concept comprising sets of pre-configured sample data files. Each scenario addresses a single use-case by combining different sample scripts (Postman collections) in a particular order.

It means most of the thinking has been done for you. All you need to do is tweak the data files for your needs, such as the team names, roles names, manager ids, etc.

The summary of the scenarios is as follows:

Scenario	Description / use-case
D01 - Delete users then delete managers	Given a team of users to be deleted, it will delete the users first before then deleting the managers so to avoid the problem of not being able to delete a user that is a manager of another user.
D02 - Delete dormant users A	Deletes users that were created over 3 months ago, have not logged in within the last 30 days, and have 2 or fewer logins within the last 90 days
D03 - Delete dormant users B	Same as D02, but users have no private folder content.
D04 - Delete dormant users C	Same as D03, but users did not create public folder content.
D05 - Delete dormant users D	Same as D04, but users are not managers.
D06 - Delete dormant users E	Same as D05, but users are activated.
D07 - Delete dormant users F	Same as D06, but users are with a BI named license
L01 - Managers with BIconcurrent to BInamed license	Identifies which users have a 'Business Intelligence' concurrent session license AND that are also managers. It then updates their license type to 'named user' ensuring all managers are always able to login and don't need to wait for a spare concurrent session to be available.
L02 - Deactivated users to BIconcurrent license	Identifies all deactivated users (with a BI named license) and sets their 'Business Intelligence' license to 'concurrent session' avoiding an unnecessarily consumption of a 'named user' license once the user is re-activated.
L03 - Convert all BIconcurrent to BInamed license	Updates all users that have a Business Intelligence 'concurrent session' license to a Business Intelligence 'named user' license.
M01 - Reassign users of given manager to another	Re-assigns all users of the given manager(s) to another manager which is helpful when the manager is unavailable or needs to be deleted.
R01- Swap directly assigned role for a team role	Re-assigns all users which have the 'BI_Admin' role directly assigned to them, adds these users into a team, adds the team to be a member of the 'BI_Admin' role before then removing the directly assigned role from all the users. This enables you to adopt the best practice for assigning roles to users, which is with the help of inheritance and teams.
S01 - Assign settings for recently created with default settings	Assigns the right date/time/number and language settings for those users created in last week with the default settings. It means your users will then be set correctly avoiding each user to update their own settings which is an unnecessary task.
S02 - Assign settings concurrent lic for recently created w default settings	Same as S01, only it also assigns the user with a 'Business Intelligence' 'concurrent session' license.
T01 - Transport Managers then Users	Given a team of users to be transported, it transports the managers first, and then all the users avoiding the problem that you can't create a user with a manager, if the manager doesn't already exist.

Like all the sample collections, these sample scenarios are also documented in detail.

Prerequisites for all scripts (except 10xx and 2xxx)

It is best to complete several essential requirements before running these scripts. These will help prevent rework and ensure they run successfully.

The prerequisites fall into a few categories:

- Teams
- Roles

Teams

The SAP Analytics Cloud SCIM API provides two versions; both are SCIM version 2 compliant:

1. Version 1 uses a root endpoint `/api/v1/scim/`. For example, a GET request to `/api/v1/scim/Users` will return the users.
2. Version 2 uses a root endpoint `/api/v1/scim2/`. For example, a GET request to `/api/v1/scim2/Users` would also return the users, albeit with a different schema than version 1.

All the samples that start with a 1 use version 1 of the API, for example, sample scripts **1001**, **1101**, **1401**, **1983**, etc.

Similarly, scripts that start with a 2, use version 2 of the API. However, there is currently just one sample script, **2601**, which lists the teams to the console log.

It means these samples predominantly use version 1. However, you cannot update Teams with version 1 until a setting is changed, unlike version 2.

This restriction to version 1 is not applicable when SAP Analytics Cloud is hosted on an SAP Data Centre (NEO). However, most customers are hosted on a non-SAP Data Centre, so it's almost certain this restriction applies to your organisation. It means you almost certainly need to change one of these settings.

Only 1 of these two settings is needed, but changing both is no harm if you have previously enabled the other. The settings are:

1. Ignore Content Namespace For Teams
2. IMPLEMENT_WORKAROUND_FOR_SCIM_GROUPS

Why is a change needed?

A setting change is needed so teams created by the user interface or imported Teams via the Content Network can be read and updated via the API. Failure to make this change will mean Teams cannot be updated, and an error will be returned from the API, complaining that a Team can't be found even though it exists. Version 2 of the API does not have this issue, meaning neither of the above settings needs to be changed so that Teams can be updated, unlike with version 1.

If you only use version 2 of the SCIM API, then no change to these settings is needed.

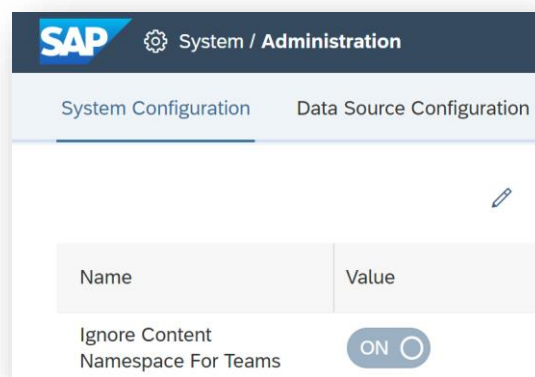
Option 1 (recommended): Ignore Content Namespace For Teams

This is the recommended setting to change.

Once this setting has been changed, all teams can be used by version 1 of the API and the user interface, regardless of how or when they were created. There are no* restrictions. Version 2 of the SCIM API isn't restricted this way.

* However, there's a problem when two teams share the same name! It is rare, but it can happen when a Team is created manually, and another Team of the same name is imported via the Content Network. In such cases, two Teams of the same name can exist. Technically, they will have different namespaces. If you have two Teams of the same name, then version 1 of the API will return an error, complaining the Team doesn't exist, even when this setting is ON. It does this because it doesn't know which Team to use, so it throws an error because version 1 of the API isn't aware of namespaces! Delete the duplicate Team and the problem is resolved. Consider using a sample script to copy team members from one Team to another to save some rework. However, deleting a Team is problematic if the Team is used to grant access rights to resources, as extra rework is needed to use the other Team in those sharing settings.

How to change the 'Ignore Content Namespace For Teams' setting



Log in as an administrator for each SAP Analytics Cloud Service and select Menu-System-Administration-System Configuration. Then, search for the setting 'Ignore Content Namespace For Teams' and press the edit button to change the value from OFF to ON, as shown in the image above. Save the change.

Option 2 (no longer recommended): IMPLEMENT_WORKAROUND_FOR_SCIM_GROUPS

Changing this setting from off/disabled to on/enabled is not recommended because there is a complication with all teams previously created manually or via importing them from the Content Network.

The difficulty is that the API cannot update any previously created teams, so you must delete all those existing teams if you want to use the API to manage them. Though, confusingly, a sample script provided here can read, though not manage, these teams to transfer the user (and role) membership to another Team, saving much rework, namely sample script “1654-All_T-Uc-Uur-Oark-Transfer API Hidden Team To API Created Team”. Please refer to this sample’s notes section for further understanding and explanation.

Once the setting has changed to on/enabled, the API and the user interface can manage Teams created manually or by importing them from the Content Network. You can mix the management of the Team through both the API and the user interface as you please.

Given the abovementioned complications, avoiding creating teams before changing this setting to on/enabled makes sense.

How to change the IMPLEMENT_WORKAROUND_FOR_SCIM_GROUPS setting

Log an incident with SAP Support, quoting your SAP Analytics Cloud URL and requesting the business toggle IMPLEMENT_WORKAROUND_FOR_SCIM_GROUPS turned on/enabled as mentioned in [SAP KBA 2857395](#).

Before submitting your request to have each SAP Analytics Cloud Service updated with this change, please consider updating your SAP Analytics Cloud ‘Content Namespace’ to be the same consistent value across all your SAP Analytics Cloud Services.

It is not essential to use the same Content Namespace throughout your landscape. However, there are a few benefits to doing so. With the business toggle IMPLEMENT_WORKAROUND_FOR_SCIM_GROUPS enabled, you will not be able to change the Content Namespace (until it is disabled/turned off)

Already enabled IMPLEMENT_WORKAROUND_FOR_SCIM_GROUPS?

If you have already enabled IMPLEMENT_WORKAROUND_FOR_SCIM_GROUPS, you could enable the other option and disable IMPLEMENT_WORKAROUND_FOR_SCIM_GROUPS. The result would be the same, but you would have lifted the restriction associated with IMPLEMENT_WORKAROUND_FOR_SCIM_GROUPS regarding changing the Namespace, and it would also put your organisation back in control of the setting rather than requiring SAP to make the change for you.

Roles

1. Roles must pre-exist.

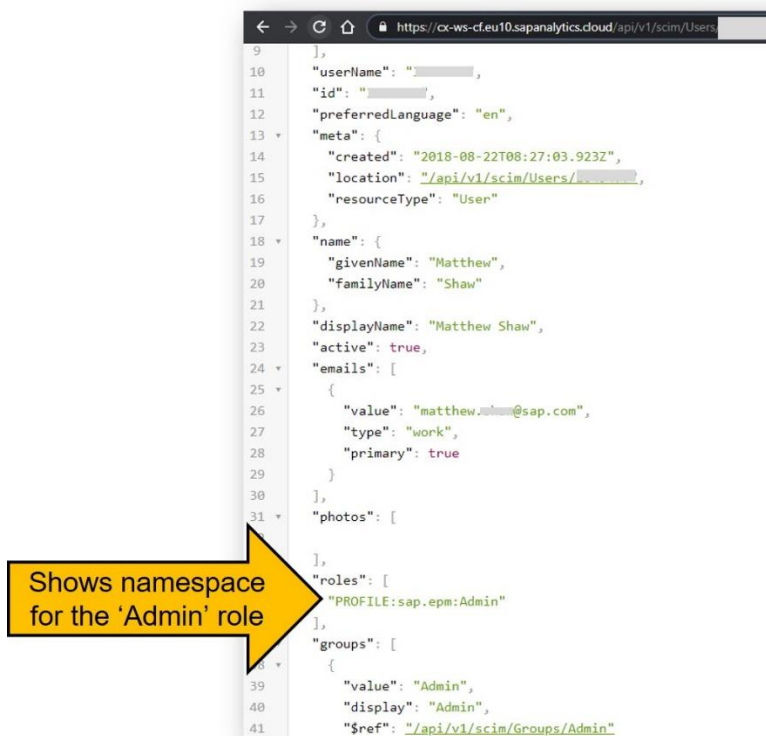
The API cannot create roles, which means if you want to add a user or a team to be a member of a role, those roles must pre-exist.

Content Namespace aware

The scripts are 'Content Namespace' aware. It means when a role is provided as a value in the data file, if the role shares the same content namespace as the environment, there's no need to specify the PROFILE: Namespace: as part of its name. For example, if a Role called 'Role1' exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only 'Role1' needs to be specified. For any role that doesn't match the ContentNamespace, then PROFILE: Namespace: must be specified with Namespace set accordingly. For example, the provided 'BI_Admin' role would be PROFILE:sap.epm:BI_Admin.

How to determine the 'Namespace' name of a role

Make an API call to GET a user's definition. Use your browser and use the URL (replacing the fully qualified domain name and the userID appropriately): <http://servicename.sapanalytics.cloud/api/v1/scim/Users/userID>



The Namespace will be shown as part of the **roles** as shown above.

2. Namespace of source and target Roles must match

It means that for the scripts that work across multiple environments (19xx), the Roles in both the source and the target must match. They will match when either:

- The Role has been transported using the service's tools between source and target services. These tools are the legacy deployment-export/import or via the Content Network. You could also use the SAP Analytics Cloud Content Network REST API ([doc link](#)).
- The Roles have been created individually within each environment when, at the time of the role creation, the Content Namespace setting in the 'System-Administration-System Configuration' was the same for both source and target.

Generally, it is best practice to keep the Content Namespace the same and consistent across the landscape. For more details, please refer to the article introduced by this [blog](#), and for a demonstration of how to transport content between environments, please refer to the article introduced by this [blog](#).

If the Content Namespaces for Roles do not match and when any user or Team references the Role, that operation will fail. The operation will be to either create or update a user or update a team.

Collection behaviour and design

The samples enjoy a standard design, and it's worth understanding these aspects to appreciate their behaviour.

Teams created without Team folders

Version 2 of the SCIM API enables creating teams, with and without creating a team folder. All the sample scripts provided create teams with the setting, so a team folder is not created, and it does this by mixing version 1 and version 2 of the API. So, whilst earlier it was mentioned all samples that start with 1xxx use version 1 of the SCIM API, this isn't entirely true as they all use version 2 albeit limited to only team creation to resolve the problem of too many team folders, which then need to be manually deleted.

Thus, all sample scripts that create teams will do so by creating the Team **without** a team folder, **unlike the earlier versions of the samples**. It means that if you would like a team with a team folder, you must either create the Team manually or adjust the code in these samples.

NEO platform exception

The only restriction is that version 2 of the SCIM API is unavailable for SAP Analytics Cloud Services hosted on an SAP Data Centre (NEO). It means that these samples will use version 1 of the API to create a team, in such cases, which will always create a team **with** a team folder. This is unlike the user interface, where a team folder is an option. The feature of an optional team folder was provided after the release of SCIM API version 1. It means that when a team is created, you may wish to delete that team folder, and this is a manual task via the user interface.

Postman tests

The sample collections come with Postman tests; for example, the 'CREATE user' test will pass once the user creation is successful.

Depending upon the sample, the Postman tests vary depending upon the use case. For example, the scripts (10x) that only create users will show a 'CREATE user' fail if the user already exists. Compared to the other scripts that perform an update, the 'CREATE user' will neither pass nor fail if the user already exists.

You may depend on the result of these Postman Tests, and it is certainly possible you may wish to make alterations to this aspect of the samples to tailor them to your needs.

General handling of errors

The scripts handle different, sometimes unexpected, responses from the API; for example, reading a user may return a 404 status “not found”, and they cater for such conditions. However, they manage many other errors, too:

- Session timeouts. Each access token has a maximum lifetime. All the collections capture and manage the expiry of the access token. When access tokens expire, the scripts automatically generate a new one, and the original request is re-submitted. It means just because the sessions timed out mid-operation, it does not affect the net result. The console log will report a session timeout, and you’ll see from the console a new session will be established.
- Any unexpected error will typically result in 2 re-attempts of the same request, which is helpful when there’s an occasional ‘wobble’ somewhere, and nothing is wrong with the request itself, but something else went wrong. Sometimes, an HTTP 500 status ‘bad gateway’ is returned; they are rare but occur occasionally. In such cases, the request will be re-attempted. The console log will report such errors; typically, the re-attempt will recover from the error.
- Endless loops. There are two primary endpoints: the SCIM API and the token endpoints. Each endpoint is somewhat independent, meaning one could be responsive and the other not. One may be non-responsive for all manner of reasons. Just because one endpoint returns unexpected results could trigger a call to the other endpoint, for example, to obtain a new access token. The fetch of a new access token is likely valid, but that may not resolve the error on the SCIM endpoint. In such conditions, this ‘loop’ is detected, and the script will exit that loop and move on to the following user or team. The console log will report such errors, including when a ‘loop’ of errors is detected. Endless loops are very rare.

Sessions, tokens and sharing of tokens across collections

Two tokens are managed by the samples:

1. **accesstoken**: this token is obtained from the token endpoint and is represented in the authorisation header of the SCIM endpoint.
2. **x-csrf-token**: this token is obtained from the SCIM endpoint by performing a 'fetch' during the first 'GET' request. Once that first 'GET' request is made, the token is re-presented in subsequent requests. Whilst performing a 'fetch' for every 'GET' is technically possible, this isn't optimal since the 'fetch' consumes resources on the SAC API and delays the request, albeit very slightly, by about 0.5 seconds. If you're making thousands and thousands of calls, this half-a-second soon adds up to many hours. For this reason, this token is only fetched once per access token session. Some scripts don't perform a GET at all, or if, during multiple PUT requests, the session times out, a separate GET request is made to obtain a valid x-csrf-token since the token cannot be 'fetched' on a non-GET call. The console log will report such events.
 - A new endpoint `/api/v1/csrf` is now available since the original samples were developed for obtaining the x-csf-token. This new endpoint simplifies the logic, and newly developed samples use this endpoint immediately after getting a new access token rather than performing a 'fetch' on the first 'GET' request. Recently developed samples use this workflow, leaving the older samples unchanged using the workflow described above.

In all cases, the access token and the x-csrf-token values are stored in the Postman environment, which means if you run multiple Collections, the tokens are re-used across them. This saves time as there's often no need to re-establish a session if the existing one can be re-used. It means if you run a collection and it suddenly times out, even if you've only just started it a few moments earlier. This will happen when you previously ran another Collection earlier that day but perhaps forgot the session would still be valid. In all cases, you don't need to worry about the session timeout of either of these tokens; the samples manage their life cycle and recover when they expire, regardless of when or where it occurs in the script.

Command line interface

Postman provides a command line interface allowing these scripts to run without the user interface. This command line option should generally be used as it allows for improved stability and potential performance improvements. Please visit the Postman website and its associated community pages for more details.



SAP HANA Academy provide a step-by-step tutorial for the command line option
<https://youtu.be/UfmauS2xagk>

Miscellaneous

When things go wrong

There are some best practices when things appear to be misbehaving! Typically, and in almost every case, it will be a client-side issue.

Recommendations:

1. Recheck you've followed the instructions in this user guide carefully, and this includes
 - how do you run the samples by running the collection with the collection runner and **not** just pressing the 'Send' button on each request
 - how did you enter the values into the template Postman Environment(s) provided
2. Check you're using the correct data file for the collection (prevalent mistake!)
3. Check you've selected the correct environment before you start the Postman runner (prevalent mistake!)
4. Inspect the console log for clues as to what is wrong
5. Restart Postman
6. Use the option in Postman User Interface Help-Clear Cache and Reload
7. Check you're on the latest version of Postman
8. Run the 1001 or 1010 collections to validate that the SAP Analytics Cloud endpoints are responding correctly
9. Check your data file is valid
 - simplify your data file to a single entry
 - revert to the sample data files
10. Go back to step 1!

License restrictions causing user and team updates to fail

Creating users or updating users or teams will fail without appropriate license availability. The whole operation will fail, which means if 100 users are added to a team but there are only sufficient licenses for 99, then all 100 users will not be added. For more details, please visit the article <https://blogs.sap.com/2020/03/10/sap-analytics-cloud-managing-licenses-with-roles-and-teams/>

Performance issues

Even though Postman reports how long the API took to respond, these figures may not be reliable. The apparent duration of the SAP Analytics Cloud API may not be the actual performance as reported in Postman. A 70% performance improvement of server-side API times has been observed by re-installing the Postman client and removing all local associated files. It also appears running Postman without the console shown improves the server-side API times!

Official references and documentation

- [SCIM API guide](#)
- [REST API Guide](#)
- [Story URL API Guide](#)

Non-script, non-API errors

Postman collections can fail due to errors caused neither by the SAP Analytics Cloud API, nor the collection script code or its design. Rather these are caused by either client or network issues.

An example includes:

- 'Error: socket hang up'.
 - This can be observed in the Postman console. Often the API call will have been successful. Sometimes though, Postman stops processing at this point.

Under these conditions, Postman recommend use of the web-based version of Postman over the client desktop application. Postman are likely to recommend the command-line version over the web-based version.

Bugs

These samples span multiple technologies, and it may not be obvious which technology is the cause of your puzzlement! Do take your time to isolate where the issue is. There are multiple channels of support and assistance:

- Postman support and its associated community forum. These channels should be used when you believe there's an issue with Postman, for example if Postman has stability issues, or misbehaves in some way.
- SAP Product Support should be used when you believe there is a defect with the SAP Analytics Cloud SCIM API itself. SAP Product Support would likely wish to see the issue reproduced with a single request, or minimal number of requests. Remember, SAP Product Support don't support these sample scripts, they are provided 'as is'.
- The SAP community <https://community.sap.com/> is likely to be helpful with progressing general questions or observations of the API when you're not convinced the issue is a defect.
- Finally, the author of these scripts, whilst is unlikely to help with individual issues, would be pleased to hear of any bugs. The total number of lines of code across all the samples amounts to over 59,000. So, there's most likely a few bugs somewhere! When you find a bug please report it to the author and share it for others to benefit. As these samples are provided 'as is' there is no official support or channel by which bugs will be resolved. The source code is open for a good reason.

API error responses and possible cause

This section captures, perhaps least common errors from the API and their possible cause when the response isn't entirely obvious:

<u>Endpoint</u>	<u>Response</u>	<u>Possible cause</u>
PUT /Users/userid	{"status":400,"message":"User/group operation not completed: Failed to create/migrate user on Cloud ID tenant"}	The email address is already in use by another user
PUT /Groups/groupid	{"status":500,"message":"Error executing the user/group operation."}	The Role the team is a member of is being updated elsewhere
PUT /Users/userid	{"status":400,"message":"User/group operation not completed: User XXXXX does not exist"}	Updating the userName (SAML Mapping) property is already in use by another user. If so, the user will then be assigned a null value for userName

Provided 'as is'

These samples are being provided "AS IS" without any warranty obligations whatsoever on the part of SAP. SAP makes no express or implied warranties of any type, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose.

About and contact

For related wiki content, PPT download, video, feedback, questions & answers please visit

<https://blogs.sap.com/2021/05/28/sap-analytics-cloud-scim-api-best-practices-and-sample-scripts>

Sample scripts created by Matthew Shaw, SAP.

<https://people.sap.com/matthew.shaw/>

<https://twitter.com/MattShaw> On BI

Data protection


Many data files you create for these scripts may contain personal data. Please ensure you follow your organisation's guidelines regarding the processing and handling this sensitive data to ensure compliance with the law.

Collection documentation

Each sample script is documented with the following properties, and these are described here:

Sample	The name of the sample collection.
Script Basis	It will typically be either 'User' or 'Team' basis. It means the collection will call either 'per user' or 'per team. Some scripts are 'TU: Team User' based, which means they read the team for a list of users and then iterate every team user.
OAuth Access	The list of OAuth Access services needed typically is 'User Provisioning', though some require additional access.
Basic description	A high-level description of what the use-case the sample provides
Ideal for	Description of when the sample collection would be suitable
Not suitable when	Description of when the sample isn't suitable and this is important to note as although a script may be functionally valid, it may not scale due to either performance issues of the API or stability issues of the client.
Notes	Handy notes that typically compare one sample with another or share personal information or advice.
Data file syntax	<p>Defines the column names or the properties of the csv or JSON file.</p> <p>The notation "F: .csv and .json" means that both csv and JSON files can be used, whereas "F: .json" means only a JSON file can be used.</p> <p>Csv files can only be used when the data file doesn't contain a ',' as part of the data or when an array of values is used.</p> <p>For all samples, for example, csv and JSON files are provided as a template for you to copy to help ensure you define the data file correctly.</p>
Environment	Specifies the variables that need to be set for the sample to work
How the script works	Describes at a high level how the sample works and the logic behind the sample.
Known script volume limitations	<p>Some scripts are entirely suitable for the Postman application; however, some scripts perform many looped requests per iteration, which isn't what Postman is designed for. It means the stability of Postman could become an issue due to this factor. Typically speaking, the Java Heap of Postman can become problematic in some situations. Whilst great care has been taken over managing that Java Heap, for example, by declaring variables using 'const' over 'let' and 'let' over 'var', there's a limit to what can be achieved.</p> <p>It will always be recommended to use the Postman command line over the use of the Desktop application, as this tends to provide more stability and resilience.</p> <p>This section describes if an alternative sample script can be used that is more favourable to the design of Postman, as well as describing the likely limit to when that would be needed.</p>
Script throughput	<p>This section presents the end-to-end throughput time of the script, helping you to determine which script would be suitable for your organisation as the number of users increases over time.</p> <p>There are a significant number of factors that affect performance, so the figures provided are only for guidance purposes to help you determine and compare one sample script with another.</p> <p>A 'script overhead' percentage is shown in (brackets). This is the overhead the script has for client-side processing, and the throughput figures include this overhead.</p>
Video Tutorial	Any relevant HANA Academy step-by-step tutorial YouTube URL will be shared.
Sample Data Files	For a few of the collections, the sample data files require additional explanation.
Related Scenarios	Related Scenarios will be listed if applicable.

1001-Test-Es-Tests Single Environment Setup

Sample	1001-Test-Es-Tests Single Environment Setup
Script Basis	None; it will perform just 1 iteration of 2 requests
OAuth Access	'User Provisioning'
Basic description	Tests the environment is set correctly for scripts that work with a single SAC service
Ideal for	<ol style="list-style-type: none"> 1. Checking the environment is set correctly for all scripts except 9xx. 2. Checking the username and password remains valid 3. Checking both API endpoints are functional (access token and the SCIM) 4. Isolating where any misconfiguration may exist 5. Checking how many users registered in SAP Analytics Cloud
Not suitable when	
Notes	It's a harmless script; it will only read from the API, and no updates are performed.
Data file syntax	There are no data file requirements.
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password, TimeZoneHours, TimeZoneMinutes, TimeZoneDescription, SAMLSSO
How the script works	<p>It reads variable values from the environment and makes two API requests based on their settings.</p> <p>The first API request obtains an access token. The access token is stored as another environmental variable and represented as part of the subsequent request's header.</p> <p>The second API request is to the SCIM endpoint and reads the first user of all the users registered in SAC. The request also returns the total number of registered users, which is output to the console.</p> <p>During both calls, the console log is particularly verbose and writes out constructive feedback as to what variables are set to what values and what the response was from the API for each of the two calls.</p>
Known script volume limitations	None
Script throughput	n/a
Video tutorial	 <p>SAP HANA Academy provide a step-by-step tutorial for the initial setup and running this script 1001 https://youtu.be/4OyLqrJh4ik</p> <p>(Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)</p>


1002-Test-Es-Tests Single Environment Advanced Setup

Sample	1002-Test-Es-Tests Single Environment Advanced Setup
Script Basis	None; it will perform just 1 iteration of 5 requests
OAuth Access	'User Provisioning', 'Activities'
Basic description	Tests the environment is set correctly for scripts that work with a single SAC service The tests include validating both the SCIM and Activities endpoints. The Activities endpoint is needed for just 1 sample script, '1665-All_U-Uc-Uu-Oarrieei-Fj-Es-AdminToolKit', though others may be added over time.
Ideal for	<ol style="list-style-type: none"> 1. Checking the environment is set correctly for the AdminToolKit sample. 2. Checking the username and password remains valid. 3. Checking API endpoints are functional (access token, SCIM, Activities and Repository) 4. Isolating where any misconfiguration may exist 5. Checking how many users registered in SAP Analytics Cloud 6. Checking how many resources are stored and accessible via the Repositories API 7. Checking how many activity logs are stored in SAP Analytics Cloud and when the date of the oldest log
Not suitable when	There is no need to use sample script, '1665-All_U-Uc-Uu-Oarrieei-Fj-Es-AdminToolKit'
Notes	It's a harmless script; it will only read from the API, and no updates are performed. For step-by-step instructions please refer earlier in this guide to 'Initial Setup with Postman' and Step E and follow the steps there, but just use this sample instead of 1001.
Data file syntax	There are no data file requirements.
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password, TimeZoneHours, TimeZoneMinutes, TimeZoneDescription, SAMLSSO
How the script works	<p>It reads variable values from the environment and makes five API requests based on their settings.</p> <ol style="list-style-type: none"> 1. The first API request obtains an access token. The access token is stored as another environmental variable and represented as part of the subsequent request's header. 2. A csrf-token is obtained using the dedicated endpoint for the csrf-token /api/v1/csrf. This endpoint is for all APIs, including SCIM version 1 and version 2. An error from this endpoint would suggest a problem with the SAP API. 3. The SCIM endpoint reads the first user of all the users registered in SAC. The request also returns the total number of registered users, which is output to the console. 4. The repositories endpoint is called with a single request to return all the IDs for all resources. A request parameter \$select is used to limit the response to only the ID. Feel free to remove this parameter to return all available fields. 5. Finally, the activities endpoint is called to return the oldest log entry, which validates if the Activities access is granted to the OAuth client. For more details about this endpoint, please visit https://blogs.sap.com/2023/01/18/sap-analytics-cloud-activities-log-command-line-interface-cli-to-automate-downloads-associated-best-practices/ <p>During all calls, the console log is particularly verbose and writes out constructive feedback as to what variables are set to what values and what the response was from the API for each call.</p>
Known script volume limitations	None
Script throughput	n/a

1011-Test-Em-Tests Multi Environment Setup


Sample	1011-Test-Em-Tests Multi Environment Setup
Script Basis	None, it will perform just 1 iteration of 4 requests
OAuth Access	'User Provisioning'
Basic description	Tests the environment is set correctly for scripts that work with two SAC services
Ideal for	<ol style="list-style-type: none"> 1. Checking the environments are set correctly for scripts 19xx. 2. Checking the usernames and passwords remains valid 3. Checking both API endpoints are functional (access token and the SCIM) for both source and target services 4. Isolating where any misconfiguration may exist 5. Checking how many users are registered in each SAP Analytics Cloud Service
Not suitable when	
Notes	It's a harmless script; it will only read from the API, and no updates are performed.
Data file syntax	There are no data file requirements.
Environment	Two Services, source, and target: (Source-SACserviceFQDN, Source-SACtokenFQDN, Source-Username, Source-Password, Source-ContentNamespace, Source-SACplatform, Source-SAMLSSO, Target-SACserviceFQDN, Target, SACTokenFQDN, Target-Username, Target-Password, Target-ContentNamespace, Target-SACplatform, Target-SAMLSSO)
How the script works	It works like script 1001, only for two SAC services, source and target.
Known script volume limitations	None
Script throughput	n/a

1021-U-Ue-Fcj-Update New User Default settings

Sample	1021-U-Ue-Fcj-Update New User Default settings		
Script Basis	U: User		
OAuth Access	'User Provisioning'		
Basic description	Ue: Updates the Postman Environment with settings read from an existing user. These settings are used as the default values for users when they are created.		
Ideal for	1. Setting the default values for new users and avoiding making mistakes by updating the Postman Environment manually		
Not suitable when			
Notes			
Data file syntax	F: .csv and .json		
	Field	Type	Description
	file_userid	string	UserID
Environment	<p>Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password, SAMLSSO</p> <p>Updates the following: SCIM_NewUserDefault_preferredlanguage, SCIM_NewUserDefault_dataaccesslanguage, SCIM_NewUserDefault_dateformatting, SCIM_NewUserDefault_timeformatting, SCIM_NewUserDefault_numberformatting, SCIM_NewUserDefault_cleanupnotificationsnumberofdays, SCIM_NewUserDefault_systemnotificationsemailoptin, SCIM_NewUserDefault_marketingemailoptin</p>		
How the script works	<p>Reads the data file and for the userid specified that the user is read from SAP Analytics Cloud.</p> <p>The script is only expecting 1 userid (1 row) in the data file since multiple doesn't make sense.</p> <p>Once the user is read, 9 of the user's properties are then stored as variables in the Postman Environment. These Postman Environment variables are then used as the default values for any new users that are created subsequently.</p> <p>The script doesn't make any changes to any existing users.</p>		
Known script volume limitations	None		
Script throughput	n/a		
Video tutorial	 <p>SAP HANA Academy provide a step-by-step tutorial for this sample script 1101 https://youtu.be/W2vislXKVUY</p> <p>(Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)</p>		

1101-U-C-Fcj-Es-Create users (no roles, no teams)

Sample	1101-U-C-Fcj-Es-Create users (no roles, no teams)																										
Script Basis	U: User																										
OAuth Access	'User Provisioning'																										
Basic description	C: Creates Users																										
Ideal for	<div><div></div><div><div>2.</div><div>Creating new users when the most significant throughput is desired.</div></div><div><div>3.</div><div>Creating a .csv file is easier compared to creating a .json.</div></div><div><div>4.</div><div>Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file</div></div><div><div>5.</div><div>You are using the default Identity Provider that comes with SAP Analytics Cloud, OR SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'USER ID'</div></div></div>																										
Not suitable when	<div><div></div><div><div>1.</div><div>Any user might already exist with the same or different e-mail than already registered, i.e. the user needs to be updated.</div></div><div><div>2.</div><div>Users need roles assigned to them directly.</div></div><div><div>3.</div><div>Running another script to assign users to teams isn't possible (i.e. given a list of users, add the users to a team), but users still need to be in a team.</div></div><div><div>4.</div><div>You need to ensure the 'user id' is determined by you and not derived from the e-mail because SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'e-mail' or a custom property.</div></div></div>																										
Notes	Use this script and assign users to teams with scripts 160x Update users with scripts 111x, 112x or 12xx Rarely an error (502 response) can occur during the user creation, which means it will appear as though the user creation was unsuccessful, but almost always, it will have created the user successfully. This sample will show an error for such users.																										
Data file syntax	F: .csv and .json <div><table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_givename</td><td>string</td><td>Initial given name</td></tr><tr><td>file_familyname</td><td>string</td><td>Initial family name</td></tr><tr><td>file_displayname</td><td>string</td><td>Initial display name</td></tr><tr><td>file_email</td><td>string</td><td>Email address</td></tr><tr><td>file_managerid</td><td>string</td><td>Optional managers ID</td></tr></table></div>			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_givename	string	Initial given name	file_familyname	string	Initial family name	file_displayname	string	Initial display name	file_email	string	Email address	file_managerid	string	Optional managers ID
Field	Type	Description																									
file_userid	string	UserID																									
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																									
file_givename	string	Initial given name																									
file_familyname	string	Initial family name																									
file_displayname	string	Initial display name																									
file_email	string	Email address																									
file_managerid	string	Optional managers ID																									
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.																										
How the script works	<div>Reads the data file, and for each entry (row) a user is created in SAP Analytics Cloud.</div> <div>Be careful with the file_isconcurrent value as it must be a lowercase true or false and not TRUE OR FALSE.</div> <div>Creating Users If using the default authentication method (i.e., not using a custom Identity Provider) then: If the user already exists, but the e-mail does not match that user, SAC will create a user of a</div>																										

	<p>different ID than requested, a ‘_1’ will be added to the userid. A warning will be shown in the console.</p> <p>If the e-mail address already exists, but the username does not, a 400 error will be returned, and the script will record a ‘CREATE user’ failure.</p> <p>A new user will not be created if the e-mail address provided belongs to an existing user.</p>			
Known script volume limitations	None			
Script throughput		Service with 0 users	Service with 80,000 users	
	Creating users	0.54 users / sec 1.85 secs / user (4.6%)	0.34 users / sec 2.96 secs / user (2.9%)	
Video tutorial	<div>  <p>SAP HANA Academy provide a step-by-step tutorial for this sample script 1101 https://youtu.be/W2vislXKVUY</p> <p>(Since the recording, the script numbers are now preceded with a ‘1’, 001 becomes 1001, etc.)</p> </div>			

1102-U-C-Cr-Fj-Es-Create users (with roles, no teams)

Sample	1102-U-C-Cr-Fj-Es-Create users (with roles, no teams)																													
Script Basis	U: User																													
OAuth Access	'User Provisioning'																													
Basic description	Cr: Creates users with roles assigned to the user directly (not generally best practice)																													
Ideal for	<div><div>1.</div><div>Creating new users when the most significant throughput is desired.</div></div> <div><div>2.</div><div>Users need roles assigned to them directly.</div></div> <div><div>3.</div><div>Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file</div></div> <div><div>4.</div><div>SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'USER ID'</div></div>																													
Not suitable when	<div><div>1.</div><div>Any user might already exist with the same or different e-mail than already registered, i.e. the user needs to be updated.</div></div> <div><div>2.</div><div>Running another script to assign users to teams isn't possible (i.e. given a list of users, add the users to a team), but users still need to be in a team.</div></div> <div><div>3.</div><div>You need to ensure the 'user ID' is determined by you and not derived from the e-mail because SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'e-mail' or a custom property.</div></div>																													
Notes	<div>Does the same as sample 1101; only this script enables roles to be assigned to users.</div> <div>Assign users to teams with scripts 160x</div> <div>Update users with scripts 111x, 112x or 12xx</div> <div>Rarely an error (502 response) can occur during the user creation, which means it will appear as though the user creation was unsuccessful, but almost always, it will have created the user successfully. This sample will show an error for such users.</div> <div>If the file_isconcurrent setting is set to true (for concurrent), but should the user be assigned roles directly (rather than indirectly via a team) and if any of those directly assigned roles are planning roles, then the isConcurrent setting will not be respected, i.e., the isConcurrent will be set to false (a named user).</div>																													
Data file syntax	<div>F: .json</div> <table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_givename</td><td>string</td><td>Initial given name</td></tr><tr><td>file_familyname</td><td>string</td><td>Initial family name</td></tr><tr><td>file_displayname</td><td>string</td><td>Initial display name</td></tr><tr><td>file_email</td><td>string</td><td>Email address</td></tr><tr><td>file_managerid</td><td>string</td><td>Optional ID of the manager's ID</td></tr><tr><td>file_JSON_roles</td><td>JSON values of roles</td><td>Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]</td></tr></table>			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_givename	string	Initial given name	file_familyname	string	Initial family name	file_displayname	string	Initial display name	file_email	string	Email address	file_managerid	string	Optional ID of the manager's ID	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]
Field	Type	Description																												
file_userid	string	UserID																												
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																												
file_givename	string	Initial given name																												
file_familyname	string	Initial family name																												
file_displayname	string	Initial display name																												
file_email	string	Email address																												
file_managerid	string	Optional ID of the manager's ID																												
file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]																												
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, ContentNamespace, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.																													
How the script works	Reads the data file, and for each entry (row), a user is created in SAP Analytics Cloud.																													

	<p>Creating Users</p> <p>If using the default authentication method (i.e., not using a custom Identity Provider) then: If the user already exists, but the e-mail does not match that user, SAC will create a user of a different id than requested, a ‘_1’ will be added to the userid. A warning will be shown in the console.</p> <p>If the e-mail address already exists, but the username does not, a 400 error will be returned, and the script will record a ‘CREATE user’ failure.</p> <p>A new user will not be created if the email address provided belongs to a user that already exists.</p> <p>Roles</p> <p>Roles are assigned directly to the user if specified, as the user is created in the single POST call that creates the user. There is no separate call to assign the user to roles.</p> <p>The roles with the same content namespace as the environment don’t need to specify the PROFILE:namespace: as part of its name. For example, if a Role called ‘Role1’ exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only ‘Role1’ needs to be specified. For any role that doesn’t match the ContentNamespace, then PROFILE:namespace: must be specified with namespace set accordingly. For example, the provided ‘BI_Admin’ role would be PROFILE:sap.epm:BI_Admin</p> <p>The file_JSON_roles can be an empty array, i.e. [] If the role doesn’t already exist a 400 status error will be returned, and the ‘UPDATE user’ will not pass its test.</p>			
Known script volume limitations	None			
Script throughput		Service with 0 users	Service with 80,000 users	
	Creating users	0.54 users / sec 1.85 secs / user (4.6%)	0.34 users / sec 2.96 secs / user (2.9%)	

1103-U-C-Crt-Fj-Es-Create users (with roles & teams)

Sample	1103-U-C-Crt-Fj-Es-Create users (with roles & teams)																										
Script Basis	U: User																										
OAuth Access	'User Provisioning'																										
Basic description	Crt: Creates users with roles assigned to the user directly (not generally best practice) and assigns the users to teams																										
Ideal for	<div><div>1.</div><div>Creating new users when the most significant throughput is desired</div></div> <div><div>2.</div><div>Users need a role assigned to them directly</div></div> <div><div>3.</div><div>Users need to be members of teams</div></div> <div><div>4.</div><div>Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file</div></div> <div><div>5.</div><div>You are using the default Identity Provider that comes with SAP Analytics Cloud, OR SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'USER ID.'</div></div>																										
Not suitable when	<div><div>1.</div><div>Any user might already exist with the same or different e-mail than already registered, i.e. the user needs to be updated.</div></div> <div><div>2.</div><div>Running another script to assign users to teams is possible (i.e. given a list of users, add the users to a Team), and users still need to be in a Team.</div></div> <div><div>3.</div><div>You need to ensure the 'user ID' is determined by you and not derived from the e-mail because SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'e-mail' or a custom property.</div></div>																										
Notes	<p>It does the same as sample 1102, only this script enables users to be assigned to Teams. Update users with scripts 111x, 112x or 12xx</p> <p>Rarely an error (502 response) can occur during the user creation, which means it will appear as though the user creation was unsuccessful, but almost always, it will have created the user successfully. This sample will show a warning for such users, and the team assignment will still be made; this will be problem-free so long as the e-mail address doesn't clash with another user (if that was the case, a duplicate user (with _1 at the end) would have been created). When this scarce error occurs, the number of users added to a team may report that more users were expected than present. You can safely ignore this warning, and it only occurs because the user is logically added to the team twice, once on the first request and again on its re-try, and this sample doesn't check for this condition to improve script throughput (unlike the other scripts that do check for this condition).</p> <p>If the file_isconcurrent setting is set to true (for concurrent), but should the user be assigned roles directly (rather than indirectly via a team) and if any of those directly assigned roles are planning roles, then the isConcurrent setting will not be respected, i.e., the isConcurrent will be set to false (a named user).</p>																										
Data file syntax	<div>F: .json</div> <table><tr><td>Field</td><td>Type</td><td>Description</td></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_givename</td><td>string</td><td>Initial given name</td></tr><tr><td>file_familyname</td><td>string</td><td>Initial family name</td></tr><tr><td>file_displayname</td><td>string</td><td>Initial display name</td></tr><tr><td>file_email</td><td>string</td><td>Email address</td></tr><tr><td>file_managerid</td><td>string</td><td>Optional ID of the manager's ID</td></tr></table>			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_givename	string	Initial given name	file_familyname	string	Initial family name	file_displayname	string	Initial display name	file_email	string	Email address	file_managerid	string	Optional ID of the manager's ID
Field	Type	Description																									
file_userid	string	UserID																									
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																									
file_givename	string	Initial given name																									
file_familyname	string	Initial family name																									
file_displayname	string	Initial display name																									
file_email	string	Email address																									
file_managerid	string	Optional ID of the manager's ID																									

	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]
	file_JSON_teams	JSON values of teams	Optional JSON array of keys "value" and string values of teams. Example: [{"value":"Team1"}, {"value":"Team2"}]
Environment	Single Service: SACserviceFQDN, SACplatform, SACtokenFQDN, ContentNamespace, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.		
How the script works	<p>Reads the data file, and for each entry (row), a user is created in SAP Analytics Cloud.</p> <p>Creating Users</p> <p>If using the default authentication method (i.e., not using a custom Identity Provider) then:</p> <p>If the user already exists, but the e-mail does not match that user, SAC will create a user of a different ID than requested, a '_1' will be added to the userid. A warning will be shown in the console.</p> <p>If the e-mail address already exists, but the username does not, a 400 error will be returned, and the script will record a 'CREATE user' failure.</p> <p>A new user will not be created if the email address provided belongs to a user that already exists.</p> <p>Roles</p> <p>Roles are assigned directly to the user if specified, as the user is created in the single POST call that creates the user. There is no separate call to assign the user to roles.</p> <p>The roles with the same content namespace as the environment don't need to specify the PROFILE:namespace: as part of its name. For example, if a Role called 'Role1' exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only 'Role1' needs to be specified. For any role that doesn't match the ContentNamespace, then PROFILE:namespace: must be specified with namespace set accordingly. For example, the provided 'BI_Admin' role would be PROFILE:sap.epm:BI_Admin</p> <p>The file_JSON_roles can be an empty array, i.e. []</p> <p>If the role doesn't already exist, a 400 status error will be returned, and the 'UPDATE user' will not pass its test.</p> <p>Teams</p> <p>User-to-team assignment cannot be performed using the version 1 API endpoint that is used to create or update a user; instead, a different endpoint must be used. This endpoint is for the Teams (Groups) themselves. It means more than one user can be specified per team, but if a single user requires membership changes to multiple teams, it will require multiple GET/PUT pair calls, one GET/PUT pair per team.</p> <p>User membership to teams is performed every 500 entries (users), and at the end of the file, i.e. they are 'batched' together to achieve the greatest throughput. The users, for each team, are batched together to form a single GET/PUT pair per team rather than a GET/PUT pair per user, per team.</p>		

	<p>If the team does not exist, it will be created without a team folder, and the display name will match its name; otherwise, the existing team's display name is respected.</p> <p>(When SAP Analytics Cloud is hosted on an SAP Data Centre (NEO), then teams are created with a team folder, and the display name for the team will have the text “with Team Folder”. E.g. “Team1 with Team Folder”. If you’d like the team folder description to share the same team display name, use sample script 1501 before running this sample script. Alternatively, update the team folder description manually.)</p> <p>Users added into teams are ‘chunked’ into multiple PUT calls. However, given the batch size is 500 users, it’s unlikely any chunking will be applied, and it would only occur when the team has 29225 users!</p> <p>The file_JSON_teams can be an empty array, i.e. []</p>			
Known script volume limitations	None			
Script throughput		Service with 0 users	Service with 80,000 users	
	Creating users and users being added into 3 teams	0.40 users / sec 2.48 secs / user (7 %)	0.26 users / sec 3.78 secs / user (4.6 %)	

1111-U-CU-CULm-Fcj-Es-Create/update users (no roles, no teams)

Sample	1111-U-CU-CULm-Fcj-Es-Create/update users (no roles, no teams)																										
Script Basis	U: User																										
OAuth Access	'User Provisioning'																										
Basic description	CU: Creates new users and updates existing users																										
Ideal for	<div><div></div><div><div><div>1.</div><div>Creating new users when the most significant throughput is desired, and some of those users already exist and may require updating.</div></div><div><div>2.</div><div>For any user that exists and, therefore, may require updating, you can be sure the e-mail address has not changed.</div></div><div><div>3.</div><div>For existing users, you only wish to update the 'BI Licence Type' and the users' manager.</div></div><div><div>4.</div><div>Creating a .csv file is easier compared to creating a .json.</div></div><div><div>5.</div><div>You are using the default Identity Provider that comes with SAP Analytics Cloud, OR SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'USER ID'</div></div></div></div>																										
Not suitable when	<div><div></div><div><div><div>1.</div><div>Any user might already exist with a different e-mail than already registered.</div></div><div><div>2.</div><div>Users need roles assigned to them directly.</div></div><div><div>3.</div><div>Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file</div></div><div><div>4.</div><div>You need to ensure the 'user ID' is determined by you and not derived from the e-mail because SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'email' or a custom property.</div></div></div></div>																										
Notes	Use this script and assign users to teams with scripts 1160x Update users with scripts 12xx if you need to update users' e-mail addresses																										
Data file syntax	F: .csv and .json <div><table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_givename</td><td>string</td><td>Initial given name</td></tr><tr><td>file_familyname</td><td>string</td><td>Initial family name</td></tr><tr><td>file_displayname</td><td>string</td><td>Initial display name</td></tr><tr><td>file_email</td><td>string</td><td>Email address</td></tr><tr><td>file_managerid</td><td>string</td><td>Optional managers ID</td></tr></table></div>			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_givename	string	Initial given name	file_familyname	string	Initial family name	file_displayname	string	Initial display name	file_email	string	Email address	file_managerid	string	Optional managers ID
Field	Type	Description																									
file_userid	string	UserID																									
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																									
file_givename	string	Initial given name																									
file_familyname	string	Initial family name																									
file_displayname	string	Initial display name																									
file_email	string	Email address																									
file_managerid	string	Optional managers ID																									
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.																										
How the script works	<div><div></div><div><div>Reads the data file, and for each entry (row) a user is created in SAP Analytics Cloud.</div><div>Be careful with the file_isconcurrent value as it must be a lowercase true or false and not TRUE OR FALSE.</div><div><div>Creating Users</div><div>If using the default authentication method (i.e., not using a custom Identity Provider) then: If the user already exists, but the e-mail does not match that user, SAC will create a user of a different ID than requested, a '_1' will be added to the userid. A warning will be shown in the console. Use sample script 1121, which will delete this 'duplicate' user in these circumstances and continue to update the user's e-mail address.</div><div>If the e-mail address already exists, but the username does not, a 400 error will be returned, and the script will record a 'CREATE user' failure.</div></div></div></div>																										

	<p>A new user will not be created if the email address provided belongs to a user that already exists.</p> <p>Updating Users (Im) The user will only be updated if it needs to be, and it will only be updated if both the username and e-mail match correctly. The following properties can be updated: isconcurrent, managerid</p>			
Known script volume limitations	None			
Script throughput		Service with 0 users	Service with 80,000 users	
	Creating users	0.53 users / sec 1.92 secs / user (4.7 %)	0.33 users / sec 3.01 secs / user (2.8 %)	

1112-U-CU-CULmr-Oarrk-Fcj-Es-Create/update users (roles, no teams)

Sample	112-U-CU-CULmr-Oarrk-Fcj-Es-Create/update users (roles, no teams)																																
Script Basis	U: User																																
OAuth Access	'User Provisioning'																																
Basic description	CU: Creates new users and updates existing users with roles assigned to the user directly (not generally best practice) (roles can be added/removed/replaced)																																
Ideal for	<div><div></div><div><div><div>1.</div><div>Creating new users when the most significant throughput is desired, and some of those users already exist and may require updating</div></div><div><div>2.</div><div>For any user that exists and, therefore, may require updating, you can be sure the e-mail address has not changed.</div></div><div><div>3.</div><div>Users need roles assigned to them directly.</div></div><div><div>4.</div><div>For existing users, you only wish to update the 'BI Licence Type' and the users' manager and their role assignment.</div></div><div><div>5.</div><div>You are using the default Identity Provider that comes with SAP Analytics Cloud, OR SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'USER ID'</div></div></div></div>																																
Not suitable when	<div><div></div><div><div><div>1.</div><div>Any user might already exist with a different e-mail than already registered.</div></div><div><div>2.</div><div>You are generally performing more updates than creating new users</div></div><div><div>3.</div><div>Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file</div></div><div><div>4.</div><div>You need to ensure the 'user ID' is determined by you and not derived from the e-mail because SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'e-mail' or a custom property</div></div></div></div>																																
Notes	<div><div></div><div><div>It does the same as sample 1111, only this script enables roles to be assigned to users.</div><div>Assign users to teams with scripts 160x</div><div>Update users with scripts 12xx if you need to update users' e-mail addresses.</div><div>If the file_isconcurrent setting is set to true (for concurrent), but should the user be assigned roles directly (rather than indirectly via a team) and if any of those directly assigned roles are planning roles, then the isConcurrent setting will not be respected, i.e., the isConcurrent will be set to false (a named user).</div></div></div>																																
Data file syntax	<div><div>F: .json</div><table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_givename</td><td>string</td><td>Initial given name</td></tr><tr><td>file_familyname</td><td>string</td><td>Initial family name</td></tr><tr><td>file_displayname</td><td>string</td><td>Initial display name</td></tr><tr><td>file_email</td><td>string</td><td>Email address</td></tr><tr><td>file_managerid</td><td>string</td><td>Optional ID of the manager's ID</td></tr><tr><td>file_JSON_roles</td><td>JSON values of roles</td><td>Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]</td></tr><tr><td>file_roles_action</td><td>string</td><td>Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used to assign the role for any new user.</td></tr></table></div>			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_givename	string	Initial given name	file_familyname	string	Initial family name	file_displayname	string	Initial display name	file_email	string	Email address	file_managerid	string	Optional ID of the manager's ID	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]	file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used to assign the role for any new user.
Field	Type	Description																															
file_userid	string	UserID																															
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																															
file_givename	string	Initial given name																															
file_familyname	string	Initial family name																															
file_displayname	string	Initial display name																															
file_email	string	Email address																															
file_managerid	string	Optional ID of the manager's ID																															
file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]																															
file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used to assign the role for any new user.																															

Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, ContentNamespace, Username, Password, SAMLSSO
How the script works	<p>Reads the data file, and for each entry (row), a user is created in SAP Analytics Cloud.</p> <p>Creating Users</p> <p>If using the default authentication method (i.e., not using a custom Identity Provider) then: If the user already exists, but the email does not match that user, SAC will create a user of a different ID than requested, a ‘_1’ will be added to the userid. A warning will be shown in the console. Use sample script 1122, which will delete this ‘duplicate’ user in these circumstances and continue to update the user’s e-mail address.</p> <p>If the e-mail address already exists, but the username does not, a 400 error will be returned, and the script will record a ‘CREATE user’ failure.</p> <p>A new user will not be created if the email address provided belongs to a user that already exists.</p> <p>Roles</p> <p>Roles are assigned directly to the user if specified, as the user is created in the single POST call that creates the user. There is no separate call to assign the user to roles.</p> <p>The roles with the same content namespace as the environment don’t need to specify the PROFILE:namespace: as part of its name. For example, if a Role called ‘Role1’ exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only ‘Role1’ needs to be specified. For any role that doesn’t match the ContentNamespace then PROFILE:namespace: must be specified with namespace set accordingly. For example, the provided ‘BI_Admin’ role would be PROFILE:sap.epm:BI_Admin</p> <p>Roles are only added to new users if the roles_action is “add” or “replace”, otherwise, the roles are ignored, and the user will not be assigned to any roles.</p> <p>Any default roles will be added after the user is created. In a rare event, an update to the user could follow the user creation (this can happen on session recovery or a false negative response from the API, a 502 error). Should this occur, and if the roles_action is ‘replace’, then any default roles not specified will be removed from the user. To avoid this rare event with the use of default roles, use ‘add’ for roles_action when creating users. The same event could also change the ‘Business Intelligence license type’ from the one defined in the default role and would, instead, set it to the setting as read from the data file.</p> <p>The file_JSON_roles can be an empty array, i.e. [] If the role doesn’t already exist, a 400 status error will be returned, and the ‘UPDATE user’ will not pass its test.</p> <p>Updating Users (lmr)</p> <p>The user will only be updated if it needs to be, and it will only be updated if both the username and e-mail match correctly. The following properties can be updated: isconcurrent, managerid, roles</p> <p>Updating Operations (arrk)</p> <p>For updating existing users, the roles_action can be “add”, “remove”, “replace” and “keep”: add: will add the roles to the user</p>

	<div><div>remove: will remove the roles from the user, if the user has any of the roles listed</div><div>replace: will replace (set) the roles for the user (i.e. remove and add accordingly)</div><div>keep: will respect any roles the user has already been assigned, no changes will be made</div></div> <div><div><div><div><div><div></div><div>from file</div><div><div><div><div></div><div></div><div></div></div><div><div><div></div><div></div><div></div></div></div></div><div><div><div><div></div><div></div><div></div></div><div><div><div></div><div></div><div></div></div></div></div></div><div><div><div></div><div>user</div><div><div><div><div></div><div></div><div></div></div><div><div><div></div><div></div><div></div></div></div></div><div><div><div><div></div><div></div><div></div></div><div><div><div></div><div></div><div></div></div></div></div></div></div><div><div><div><div><div><div></div><div>add</div><div><div><div><div></div><div></div><div></div></div><div><div><div></div><div></div><div></div></div></div></div><div><div><div><div></div><div></div><div></div></div><div><div><div></div><div></div><div></div></div></div></div><div><div><div><div></div><div></div><div></div></div><div><div><div></div><div></div><div></div></div></div></div></div><div><div><div></div><div>remove</div><div><div><div><div></div><div></div><div></div></div></div></div><div><div><div><div></div><div></div><div></div></div></div></div></div><div><div><div></div><div>replace</div><div><div><div><div></div><div></div><div></div></div><div><div><div></div><div></div><div></div></div></div></div><div><div><div><div></div><div></div><div></div></div><div><div><div></div><div></div><div></div></div></div></div></div><div><div><div></div><div>keep</div><div><div><div><div></div><div></div><div></div></div><div><div><div></div><div></div><div></div></div></div></div><div><div><div><div></div><div></div><div></div></div><div><div><div></div><div></div><div></div></div></div></div></div></div></div><div><div><div><div><div><div></div><div>Result of role operation on user</div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div></div>								
Known script volume limitations	None								
Script throughput	<table><tr><td></td><td>Service with 0 users</td><td>Service with 80,000 users</td></tr><tr><td rowspan="2">Creating users</td><td>0.53 users / sec</td><td>0.33 users / sec</td></tr><tr><td>1.92 secs / user (4.7 %)</td><td>3.01 secs / user (2.8 %)</td></tr></table>		Service with 0 users	Service with 80,000 users	Creating users	0.53 users / sec	0.33 users / sec	1.92 secs / user (4.7 %)	3.01 secs / user (2.8 %)
	Service with 0 users	Service with 80,000 users							
Creating users	0.53 users / sec	0.33 users / sec							
	1.92 secs / user (4.7 %)	3.01 secs / user (2.8 %)							

1113-U-CU-CULmrt-Oarrk-Fcj-Es-Create/update users (roles & teams)

Sample	1113-U-CU-CULmrt-Oarrk-Fcj-Es-Create/update users (roles & teams)																										
Script Basis	U: User																										
OAuth Access	'User Provisioning'																										
Basic description	CU: Creates new users and updates existing users with roles assigned to the user directly (not generally best practice) and assigns the users to teams (roles and teams can be added/removed/replaced)																										
Ideal for	<ol style="list-style-type: none">1. Creating new users when the most significant throughput is desired, and some of those users already exist and may require updating2. For any user that exists and, therefore, may require updating, you can be sure the e-mail address has not changed.3. Users need roles assigned to them directly4. Users need to be members of teams5. For existing users, you only wish to update the 'BI Licence Type' and the users' manager, their role assignment and their team assignment.6. You are using the default Identity Provider that comes with SAP Analytics Cloud, OR SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'USER ID'																										
Not suitable when	<ol style="list-style-type: none">1. Any user might already exist with a different e-mail than already registered.2. Running another script to assign users to teams is possible (i.e. given a list of users, add the users to a team), and users still need to be in a team.3. You are generally performing more updates than creating new users4. Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file5. You need to ensure the 'user id' is determined by you and not derived from the e-mail because SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'e-mail' or a custom property																										
Notes	<p>It does the same as sample 1112, only this script enables users to be assigned to teams. Update users with scripts 12xx if you need to update users' e-mail addresses. If the file_isconcurrent setting is set to true (for concurrent), but should the user be assigned roles directly (rather than indirectly via a team) and if any of those directly assigned roles are planning roles, then the isConcurrent setting will not be respected, i.e., the isConcurrent will be set to false (a named user).</p> <p>If you are using your own Identity Provider with SAML SSO mapping on the 'e-mail' and you wish to update the email, then you can also use sample scripts 1409 and 1429</p>																										
Data file syntax	<div>F: .json</div> <table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_givename</td><td>string</td><td>Initial given name</td></tr><tr><td>file_familyname</td><td>string</td><td>Initial family name</td></tr><tr><td>file_displayname</td><td>string</td><td>Initial display name</td></tr><tr><td>file_email</td><td>string</td><td>Email address</td></tr><tr><td>file_managerid</td><td>string</td><td>Optional ID of the manager's ID</td></tr></table>			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_givename	string	Initial given name	file_familyname	string	Initial family name	file_displayname	string	Initial display name	file_email	string	Email address	file_managerid	string	Optional ID of the manager's ID
Field	Type	Description																									
file_userid	string	UserID																									
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																									
file_givename	string	Initial given name																									
file_familyname	string	Initial family name																									
file_displayname	string	Initial display name																									
file_email	string	Email address																									
file_managerid	string	Optional ID of the manager's ID																									

	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]
	file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used to assign the role for any new user.
	file_JSON_teams	JSON values of teams	Optional JSON array of keys "value" and string values of teams. Example: [{"value":"Team1"}, {"value":"Team2"}]
	file_teams_action	string	Defines the action to perform on the users' team membership. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used for any new user to be a member of a team.
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, ContentNamespace, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.		
How the script works	<p>Reads the data file, and for each entry (row), a user is created in SAP Analytics Cloud.</p> <p>Creating Users</p> <p>If using the default authentication method (i.e., not using a custom Identity Provider) then: If the user already exists, but the e-mail does not match that user, SAC will create a user of a different ID than requested, a '_1' will be added to the userid. A warning will be shown in the console. Use sample script 1123 to delete this 'duplicate' user and continue to update the user's e-mail address.</p> <p>If the e-mail address already exists, but the username does not, a 400 error will be returned, and the script will record a 'CREATE user' failure.</p> <p>A new user will not be created if the email address provided belongs to a user that already exists.</p> <p>Roles</p> <p>Roles are assigned directly to the user if specified, as the user is created in the single POST call that creates the user. There is no separate call to assign the user to roles.</p> <p>The roles with the same content namespace as the environment don't need to specify the PROFILE:namespace: as part of its name. For example, if a Role called 'Role1' exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only 'Role1' needs to be specified. For any role that doesn't match the ContentNamespace, then PROFILE:namespace: must be specified with namespace set accordingly. For example, the provided 'BI_Admin' role would be PROFILE:sap.epm:BI_Admin</p> <p>Roles are only added to new users if the roles_action is "add" or "replace"; otherwise, the roles are ignored, and the user will not be assigned to any roles.</p> <p>Any default roles will be added after the user is created. In a rare event, an update to the user could follow the user creation (this can happen on session recovery or a false negative response from the API, a 502 error). Should this occur, and if the roles_action is 'replace', then any default roles not specified will be removed from the user. To avoid this rare event with the use of default roles, use 'add' for roles_action when creating users. The same event</p>		

could also change the 'Business Intelligence license type' from the one defined in the default role and would, instead, set it to the setting as read from the data file.

The file_JSON_roles can be an empty array, i.e. []

If the role doesn't already exist, a 400 status error will be returned, and the 'UPDATE user' will not pass its test.

Teams

User-to-team assignment cannot be performed using the version 1 API endpoint that is used to create or update a user; instead, a different endpoint must be used. This endpoint is for the Teams (Groups) themselves. It means more than one user can be specified per team, but if a single user requires membership changes to multiple teams, it will require multiple GET/PUT pair calls, one GET/PUT pair per team.

User membership to teams is performed every 500 entries (users), and at the end of the file, i.e. they are 'batched' together to achieve the most significant throughput. The users for each team are batched together to form a single GET/PUT pair per team rather than a GET/PUT pair per user per team.

If the team does not exist, it will be created **without** a team folder, and the display name will match its name; otherwise, the existing team's display name is respected.

(When SAP Analytics Cloud is hosted on an SAP Data Centre (NEO), then teams are created **with** a team folder, and the display name for the team will have the text "with Team Folder". E.g. "Team1 with Team Folder". If you'd like the team folder description to share the same team display name, use sample script 1501 before running this sample script. Alternatively, update the team folder description manually.)

Users added into teams are 'chunked' into multiple PUT calls. However, given the batch size is 500 users, it's unlikely any chunking will be applied, and it would only occur when the team has 29225 users!

The file_JSON_teams can be an empty array, i.e. []

Teams are only added to new users if the teams_action is "add" or "replace", otherwise, the teams are ignored and the user will not be assigned to any teams.

Updating Users (lmrt)

The user will only be updated if it needs to be and it will only be updated if both the username and email match correctly.

The following properties can be updated: isconcurrent, managerid, roles and teams assignment.

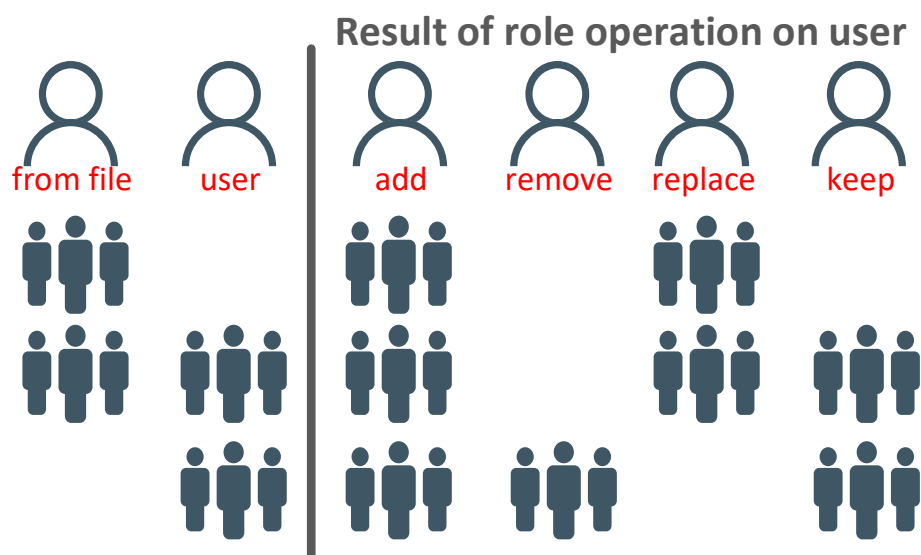
Updating Operations (arrk)

For updating existing users, the roles_action can be "add", "remove", "replace" and "keep":
add: will add the roles to the user.

remove: will remove the roles from the user, if the user has any of the roles listed.

replace: will replace (set) the roles for the user (i.e. remove and add accordingly).

keep: will respect any roles the user has already been assigned, no changes will be made.

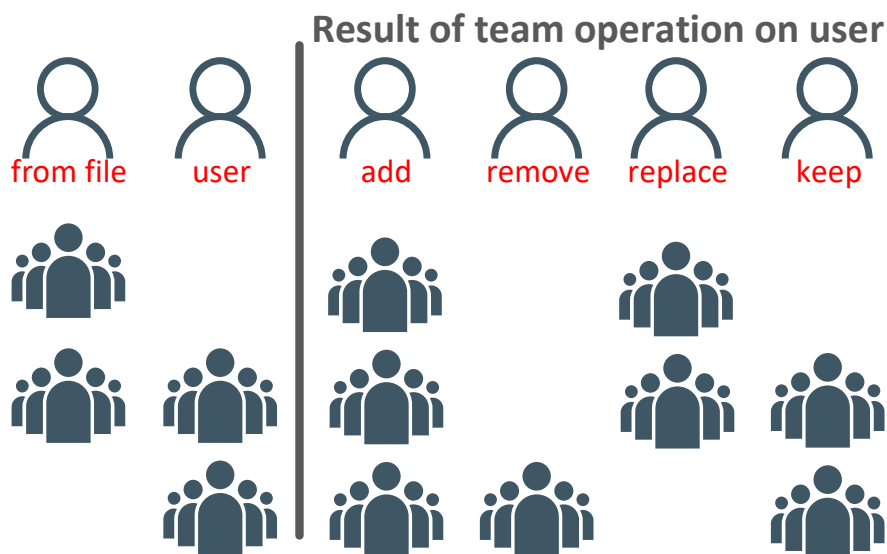


For updating existing users, the `teams_action` can be “add”, “remove”, “replace” and “keep”:
add: will add the user to the team.

remove: will remove the user from the team, if the user is a member of any of the teams listed.

replace: will replace (set) the team membership of the user (i.e. remove and add accordingly).

keep: will respect any teams the user is already a member of, no changes will be made.



Known script volume limitations	None			
Script throughput		Service with 0 users	Service with 80,000 users	
	Creating users and users being added into 3 teams	0.43 users / sec 2.31 secs / user (7.9 %)	0.28 users / sec 3.6 secs / user (5.5 %)	

	Updating users (not email, not team)	0.20 users / sec 4.93 secs / users (5.7%)	0.12 users / sec 8.33 secs / user (3.2%)	
	Updating users (that don't need an update)	0.33 users / sec 2.99 secs / users (5.6%)	0.16 users / sec 6.53 secs / user (3.0%)	
	Updating user and team (removing 3 teams and adding 3 teams)	0.33 users / sec 2.99 secs / user (16.9%)	0.15 users / sec 6.88 secs / user (7.4%)	

1121-U-CU-CUlem-Fcj-Es-Create/update users (no roles, no teams)

Sample	1121-U-CU-CUlm-Fcj-Es-Create/update users (no roles, no teams)																										
Script Basis	U: User																										
OAuth Access	'User Provisioning'																										
Basic description	CU: Creates new users and updates existing users.																										
Ideal for	<div><div></div><div><div><div>1.</div><div>Creating new users when the most significant throughput is desired; some already exist and may require updating.</div></div><div><div>2.</div><div>For existing users, you only wish to update the 'BI Licence Type', the users' manager or e-mail address*. (* Updating the e-mail isn't possible when SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on 'userid' or 'custom')</div></div><div><div>3.</div><div>Creating a .csv file is easier compared to creating a .json.</div></div><div><div>4.</div><div>You are using the default Identity Provider that comes with SAP Analytics Cloud, OR SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'USER ID'</div></div></div></div>																										
Not suitable when	<div><div></div><div><div><div>1.</div><div>Users need roles assigned to them directly</div></div><div><div>2.</div><div>Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file.</div></div><div><div>3.</div><div>You need to ensure the 'user id' is determined by you and not derived from the e-mail because SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'email' or a custom property.</div></div></div></div>																										
Notes	<div><div></div><div><div>Use this script and assign users to teams with scripts 160x It does the same as sample script 1111, except e-mail addresses can also be updated (though a new user will be created and subsequently deleted in the process)</div><div>The samples 112x have extra logic in them, where they delete a 'duplicated' user. This is only applicable using the default authentication method with SAP Analytics Cloud. If you are using your own Identity Provider, then these samples 112x do not add anything over the 111x samples.</div><div>If you are using your own Identity Provider with SAML SSO mapping on the 'e-mail' and you wish to update the e-mail, then you can also use sample scripts 1409 and 1429</div></div></div>																										
Data file syntax	<div><div></div><div><div>F: .csv and .json</div><table><tr><td>Field</td><td>Type</td><td>Description</td></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_givename</td><td>string</td><td>Initial given name</td></tr><tr><td>file_familyname</td><td>string</td><td>Initial family name</td></tr><tr><td>file_displayname</td><td>string</td><td>Initial display name</td></tr><tr><td>file_email</td><td>string</td><td>Email address</td></tr><tr><td>file_managerid</td><td>string</td><td>Optional managers ID</td></tr></table></div></div>			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_givename	string	Initial given name	file_familyname	string	Initial family name	file_displayname	string	Initial display name	file_email	string	Email address	file_managerid	string	Optional managers ID
Field	Type	Description																									
file_userid	string	UserID																									
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																									
file_givename	string	Initial given name																									
file_familyname	string	Initial family name																									
file_displayname	string	Initial display name																									
file_email	string	Email address																									
file_managerid	string	Optional managers ID																									
Environment	<div><div></div><div><div>Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.</div></div></div>																										
How the script works	<div><div></div><div><div>Reads the data file, and for each entry (row), a user is created in SAP Analytics Cloud.</div><div>Be careful with the file_isconcurrent value as it must be a lowercase true or false and not TRUE OR FALSE.</div><div>Creating Users</div></div></div>																										

	<p>If using the default authentication method (i.e., not using a custom Identity Provider) then: If the user already exists, but the e-mail does not match that user, SAC will create a user of a different ID than requested, a '_1' will be added to the userid. This duplicate user will then be deleted before the user is read, allowing the user's e-mail to be updated. A 'DELETE duplicate user' test will pass.</p> <p>If the email address already exists and is assigned to a different user, the script will record an 'UPDATE user' failure.</p> <p>Updating Users (lme) The user will only be updated if it needs to be and it will only be updated if both the username and email match correctly. The following properties can be updated: isconcurrent, managerid, email</p>			
Known script volume limitations	None			
Script throughput		Service with 0 users	Service with 80,000 users	
	Creating users	0.53 users / sec 1.92 secs / user (4.7 %)	0.33 users / sec 3.01 secs / user (2.8 %)	
	Updating users (not email, not team)	0.20 users / sec 4.93 secs / users (5.7%)	0.12 users / sec 8.33 secs / user (3.2%)	
	Updating users' email	0.11 users / sec 9.23 secs / user (4.0%)	0.07 users / sec 13.56 secs / user (2.8%)	
	Updating users (that don't need an update)	0.33 users / sec 2.99 secs / users (5.6%)	0.16 users / sec 6.53 secs / user (3.0%)	
	Updating user and team (removing 3 teams and adding 3 teams)	0.33 users / sec 2.99 secs / user (16.9%)	0.15 users / sec 6.88 secs / user (7.4%)	

1122-U-CU-CUImr-Oarrk-Fcj-Es-Create/update users (roles, no teams)

Sample	1122-U-CU-CULmr-Oarrk-Fcj-Es-Create/update users (roles, no teams)																										
Script Basis	U: User																										
OAuth Access	'User Provisioning'																										
Basic description	CU: Creates new users and updates existing users with roles assigned to the user directly (not generally best practice) (roles can be added/removed/replaced)																										
Ideal for	<div><div>1.</div><div>Creating new users when the greatest throughput is desired, and some of those users already exist and may require updating</div></div> <div><div>2.</div><div>Users need roles assigned to them directly.</div></div> <div><div>3.</div><div>For existing users, you only wish to update the 'BI Licence Type', the users' manager or e-mail address*. (* Updates to the e-mail are not possible when SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on 'userid' or 'custom')</div></div> <div><div>4.</div><div>You are using the default Identity Provider that comes with SAP Analytics Cloud, OR, SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'USER ID'</div></div>																										
Not suitable when	<div><div>1.</div><div>You are generally performing more updates than creating new users</div></div> <div><div>2.</div><div>Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file</div></div> <div><div>3.</div><div>You need to ensure the 'user id' is determined by you and not derived from the e-mail because SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'e-mail' or a custom property</div></div>																										
Notes	<div>It does the same as sample 1121, only this script enables roles can be assigned to users Assign users to teams with scripts 160x.</div> <div>The samples 112x have extra logic in them, where they delete a 'duplicated' user. This is only applicable if using the default authentication method with SAP Analytics Cloud. If you are using your own Identity Provider, then these samples 112x do not add anything over the 111x samples.</div> <div>If the file_isconcurrent setting is set to true (for concurrent), but should the user be assigned roles directly (rather than indirectly via a team) and if any of those directly assigned roles are planning roles, then the isConcurrent setting will not be respected, i.e., the isConcurrent will be set to false (a named user).</div> <div>If you are using your own Identity Provider with SAML SSO mapping on the 'email' and you wish to update the email, then you can also use sample scripts 1409 and 1429</div>																										
Data file syntax	<div>F: .json</div> <table><tr><td>Field</td><td>Type</td><td>Description</td></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_givenname</td><td>string</td><td>Initial given name</td></tr><tr><td>file_familyname</td><td>string</td><td>Initial family name</td></tr><tr><td>file_displayname</td><td>string</td><td>Initial display name</td></tr><tr><td>file_email</td><td>string</td><td>Email address</td></tr><tr><td>file_managerid</td><td>string</td><td>Optional ID of the manager's ID</td></tr></table>			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_givenname	string	Initial given name	file_familyname	string	Initial family name	file_displayname	string	Initial display name	file_email	string	Email address	file_managerid	string	Optional ID of the manager's ID
Field	Type	Description																									
file_userid	string	UserID																									
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																									
file_givenname	string	Initial given name																									
file_familyname	string	Initial family name																									
file_displayname	string	Initial display name																									
file_email	string	Email address																									
file_managerid	string	Optional ID of the manager's ID																									

	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]
	file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used to assign the role for any new user.
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, ContentNamespace, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.		
How the script works	<p>Reads the data file, and for each entry (row) a user is created in SAP Analytics Cloud.</p> <p>Creating Users</p> <p>If using the default authentication method (i.e., not using a custom Identity Provider) then: If the user already exists, but the email does not match that user, SAC will create a user of a different id than requested, a '_1' will be added to the userid. This duplicate user will then be deleted before the user is read, allowing the user's e-mail to be updated. A 'DELETE duplicate user' test will pass.</p> <p>If the e-mail address already exists and is assigned to a different user, the script will record an 'UPDATE user' failure.</p> <p>Roles</p> <p>Roles are assigned directly to the user if specified, as the user is created in the single POST call that creates the user. There is no separate call to assign the user to roles.</p> <p>The roles with the same content namespace as the environment don't need to specify the PROFILE:namespace: as part of its name. For example, if a Role called 'Role1' exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only 'Role1' needs to be specified. For any role that doesn't match the ContentNamespace, then PROFILE:namespace: must be specified with namespace set accordingly. For example, the provided 'BI_Admin' role would be PROFILE:sap.epm:BI_Admin</p> <p>Roles are only added to new users if the roles_action is "add" or "replace", otherwise, the roles are ignored, and the user will not be assigned to any roles.</p> <p>Any default roles will be added after the user is created. In a rare event, an update to the user could follow the user creation (this can happen on session recovery or a false negative response from the API, a 502 error). Should this occur, and if the roles_action is 'replace', then any default roles not specified will be removed from the user. To avoid this rare event with the use of default roles, use 'add' for roles_action when creating users. The same event could also change the 'Business Intelligence license type' from the one defined in the default role and would, instead, set it to the setting as read from the data file.</p> <p>The file_JSON_roles can be an empty array, i.e. [] If the role doesn't already exist a 400 status error will be returned, and the 'UPDATE user' will not pass its test.</p> <p>Updating Users (lcmr)</p>		

The user will only be updated if it needs to be, and it will only be updated if both the username and email match correctly.

The following properties can be updated: isconcurrent, email, managerid, roles

Updating Operations (arrk)

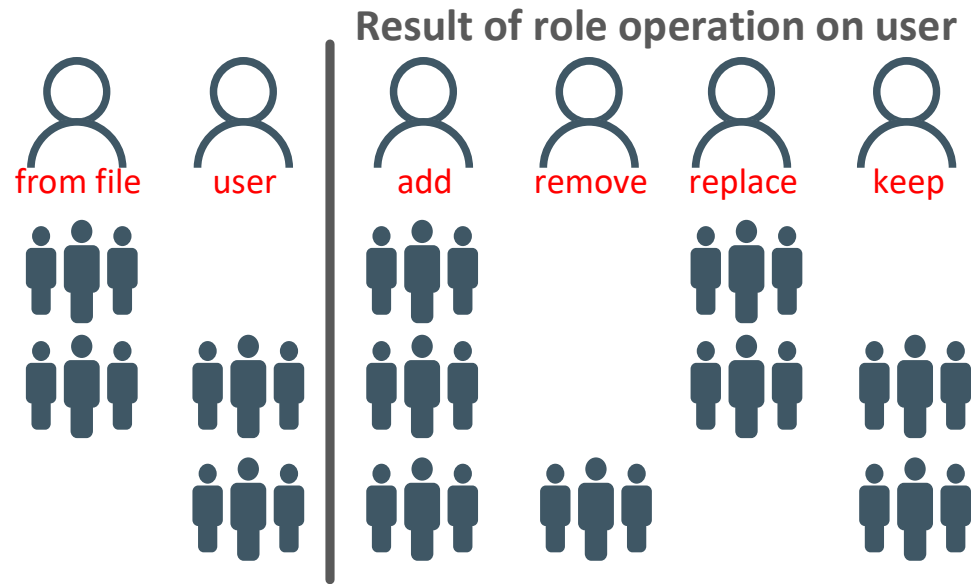
For updating existing users, the roles_action can be “add”, “remove”, “replace” and “keep”:

add: will add the roles to the user

remove: will remove the roles from the user if the user has any of the roles listed

replace: will replace (set) the roles for the user (i.e. remove and add accordingly)

keep: will respect any roles the user has already been assigned, no changes will be made



Known script volume
limitations

None

Script throughput

	Service with 0 users	Service with 80,000 users
Creating users	0.53 users / sec 1.92 secs / user (4.7 %)	0.33 users / sec 3.01 secs / user (2.8 %)
Updating users (not email, not team)	0.20 users / sec 4.93 secs / users (5.7%)	0.12 users / sec 8.33 secs / user (3.2%)
Updating users' email	0.11 users / sec 9.23 secs / user (4.0%)	0.07 users / sec 13.56 secs / user (2.8%)
Updating users (that don't need an update)	0.33 users / sec 2.99 secs / users (5.6%)	0.16 users / sec 6.53 secs / user (3.0%)
Updating user and team (removing 3 teams and adding 3 teams)	0.33 users / sec 2.99 secs / user (16.9%)	0.15 users / sec 6.88 secs / user (7.4%)

1123-U-CU-CUlemrt-Oarrk-Fcj-Es-Create/update users (roles & teams)

Sample	1123-U-CU-CULmrt-Oarrk-Fcj-Es-Create/update users (roles & teams)																							
Script Basis	U: User																							
OAuth Access	'User Provisioning'																							
Basic description	CU: Creates new users and updates existing users with roles assigned to the user directly (not generally best practice) and assigns the users to teams (roles and teams can be added/removed/replaced)																							
Ideal for	<ol style="list-style-type: none">1. Creating new users when the most significant throughput is desired, and some of those users already exist and may require updating2. Users need roles assigned to them directly3. Users need to be members of teams4. For existing users, you only wish to update the 'BI Licence Type', the users' manager or e-mail address* or their team assignment (* Updates to the e-mail is not possible when SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on 'userid' or 'custom')5. You are using the default Identity Provider that comes with SAP Analytics Cloud, OR SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'USER ID'																							
Not suitable when	<ol style="list-style-type: none">1. Running another script to assign users to teams is possible (i.e. given a list of users, add the users to a team), and users still need to be in a team.2. You are generally performing more updates than creating new users3. Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file4. You need to ensure the 'user id' is determined by you and not derived from the e-mail because SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'e-mail' or a custom property																							
Notes	<p>It does the same as sample 1122; only this script enables users to be assigned to teams.</p> <p>The samples 112x have extra logic in them, where they delete a 'duplicated' user. This is only applicable if using the default authentication method with SAP Analytics Cloud. If you are using your own Identity Provider, then these samples 112x do not add anything over the 111x samples.</p> <p>If the file_isconcurrent setting is set to true (for concurrent), but should the user be assigned roles directly (rather than indirectly via a team) and if any of those directly assigned roles are planning roles, then the isConcurrent setting will not be respected, i.e., the isConcurrent will be set to false (a named user).</p> <p>If you are using your own Identity Provider with SAML SSO mapping on the 'e-mail' and you wish to update the e-mail, then you can also use sample scripts 1409 and 1429</p>																							
Data file syntax	<table><tr><td colspan="3">F: .json</td></tr><tr><td>Field</td><td>Type</td><td>Description</td></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_givenname</td><td>string</td><td>Initial given name</td></tr><tr><td>file_familyname</td><td>string</td><td>Initial family name</td></tr><tr><td>file_displayname</td><td>string</td><td>Initial display name</td></tr></table>			F: .json			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_givenname	string	Initial given name	file_familyname	string	Initial family name	file_displayname	string	Initial display name
F: .json																								
Field	Type	Description																						
file_userid	string	UserID																						
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																						
file_givenname	string	Initial given name																						
file_familyname	string	Initial family name																						
file_displayname	string	Initial display name																						

	file_email	string	Email address
	file_managerid	string	Optional ID of the manager's ID
	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]
	file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used to assign the role for any new user.
	file_JSON_teams	JSON values of teams	Optional JSON array of keys "value" and string values of teams. Example: [{"value":"Team1"}, {"value":"Team2"}]
	file_teams_action	string	Defines the action to perform on the users' team membership. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used for any new user to be a member of a team.
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, ContentNamespace, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.		
How the script works	<p>Reads the data file, and for each entry (row) a user is created in SAP Analytics Cloud.</p> <p>Creating Users</p> <p>If using the default authentication method (i.e., not using a custom Identity Provider) then: If the user already exists, but the e-mail does not match that user, SAC will create a user of a different id than requested, a '_1' will be added to the userid. This duplicate user will then be deleted before the user is read, allowing the user's e-mail to be updated. A 'DELETE duplicate user' test will pass.</p> <p>If the email address already exists and is assigned to a different user, the script will record an 'UPDATE user' failure.</p> <p>Roles</p> <p>Roles are assigned directly to the user if specified, as the user is created in the single POST call that creates the user. There is no separate call to assign the user to roles.</p> <p>The roles with the same content namespace as the environment don't need to specify the PROFILE:namespace: as part of its name. For example, if a Role called 'Role1' exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only 'Role1' needs to be specified. For any role that doesn't match the ContentNamespace, then PROFILE:namespace: must be specified with namespace set accordingly. For example, the provided 'BI_Admin' role would be PROFILE:sap.epm:BI_Admin</p> <p>Roles are only added to new users if the roles_action is "add" or "replace"; otherwise, the roles are ignored, and the user will not be assigned to any roles.</p> <p>Any default roles will be added after the user is created. On a rare event, an update to the user could follow the user creation (this can happen on session recovery, or a false negative response from the API, a 502 error). Should this occur, and if the roles_action is 'replace', then any default roles not specified will be removed from the user. To avoid this rare event with the use of default roles, use 'add' for roles_action when creating users. The same event</p>		

could also change the 'Business Intelligence license type' from the one defined in the default role and would, instead, set it to the setting as read from the data file.

The file_JSON_roles can be an empty array, i.e. []

If the role doesn't already exist a 400 status error will be returned and the 'UPDATE user' will not pass its test.

Teams

User-to-team assignment cannot be performed using the version 1 API endpoint that is used to create or update a user; instead, a different endpoint must be used. This endpoint is for the Teams (Groups) themselves. It means more than one user can be specified per team, but if a single user requires membership changes to multiple teams, it will need multiple GET/PUT pair calls, one GET/PUT pair per team.

User membership to teams is performed every 500 entries (users), and at the end of the file, i.e. they are 'batched' together to achieve the greatest throughput. The users for each team, are batched together to form a single GET/PUT pair per team rather than a GET/PUT pair per user, per team.

If the team does not exist, it will be created **without** a team folder, and the display name will match its name; otherwise, the existing team's display name is respected.

(When SAP Analytics Cloud is hosted on an SAP Data Centre (NEO), then teams are created **with** a team folder, and the display name for the team will have the text "with Team Folder". E.g. "Team1 with Team Folder". If you'd like the team folder description to share the same team display name, use sample script 1501 before running this sample script. Alternatively, update the team folder description manually.)

Users added into teams are 'chunked' into multiple PUT calls. However, given the batch size is 500 users, it's unlikely any chunking will be applied, and it would only occur when the team has 29225 users!

The file_JSON_teams can be an empty array, i.e. []

Teams are only added to new users if the teams_action is "add" or "replace", otherwise, the teams are ignored, and the user will not be assigned to any teams.

Updating Users (lemrt)

The user will only be updated if it needs to be and it will only be updated if both the username and email match correctly.

The following properties can be updated: isconcurrent, email, managerid, roles and teams assignment.

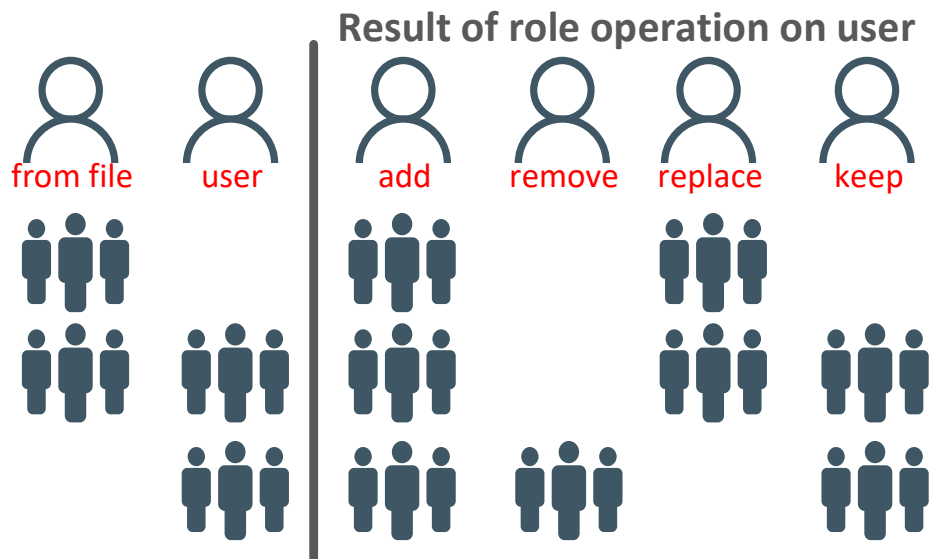
Updating Operations (arrk)

For updating existing users, the roles_action can be "add", "remove", "replace" and "keep":
add: will add the roles to the user.

remove: will remove the roles from the user, if the user has any of the roles listed.

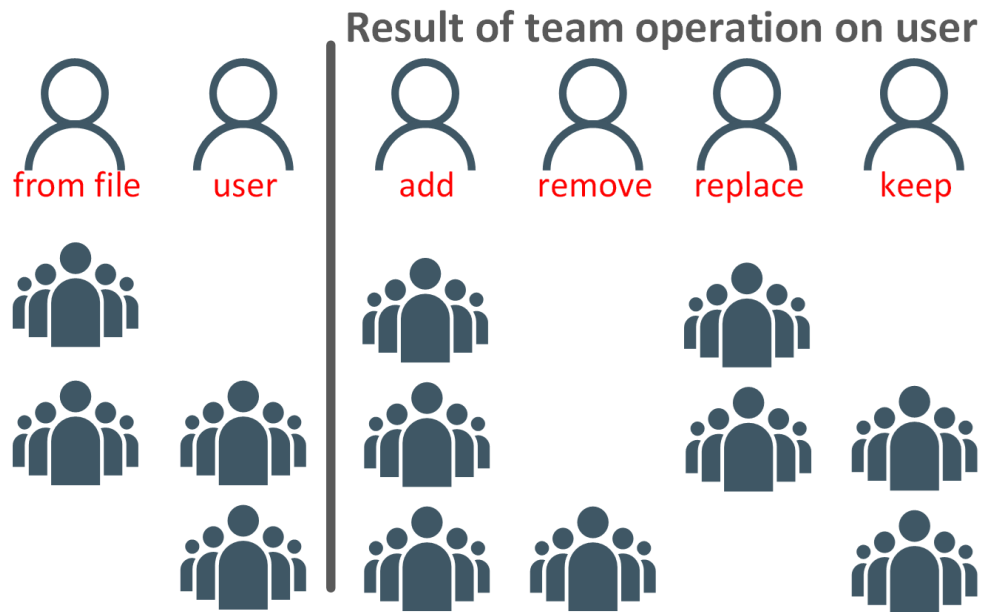
replace: will replace (set) the roles for the user (i.e. remove and add accordingly).

keep: will respect any roles the user has already been assigned, no changes will be made.



For updating existing users, the `teams_action` can be “add”, “remove”, “replace” and “keep”:

- `add`: will add the user to the team.
- `remove`: will remove the user from the team, if the user is a member of any of the teams listed.
- `replace`: will replace (set) the team membership of the user (i.e. remove and add accordingly).
- `keep`: will respect any teams the user is already a member of, and no changes will be made.



Known script volume limitations	None			
Script throughput		Service with 0 users	Service with 80,000 users	
	Creating users and users being added into 3 teams	0.43 users / sec 2.31 secs / user (7.9 %)	0.28 users / sec 3.6 secs / user (5.5 %)	

	Updating users (not email, not team)	0.20 users / sec 4.93 secs / users (5.7%)	0.12 users / sec 8.33 secs / user (3.2%)	
	Updating users' email	0.11 users / sec 9.23 secs / user (4.0%)	0.07 users / sec 13.56 secs / user (2.8%)	
	Updating users (that don't need an update)	0.33 users / sec 2.99 secs / users (5.6%)	0.16 users / sec 6.53 secs / user (3.0%)	
	Updating user and team (removing 3 teams and adding 3 teams)	0.33 users / sec 2.99 secs / user (16.9%)	0.15 users / sec 6.88 secs / user (7.4%)	

1131-U-CU-CUlem-Fcj-Es-SAML Create/update users (no roles, no teams)

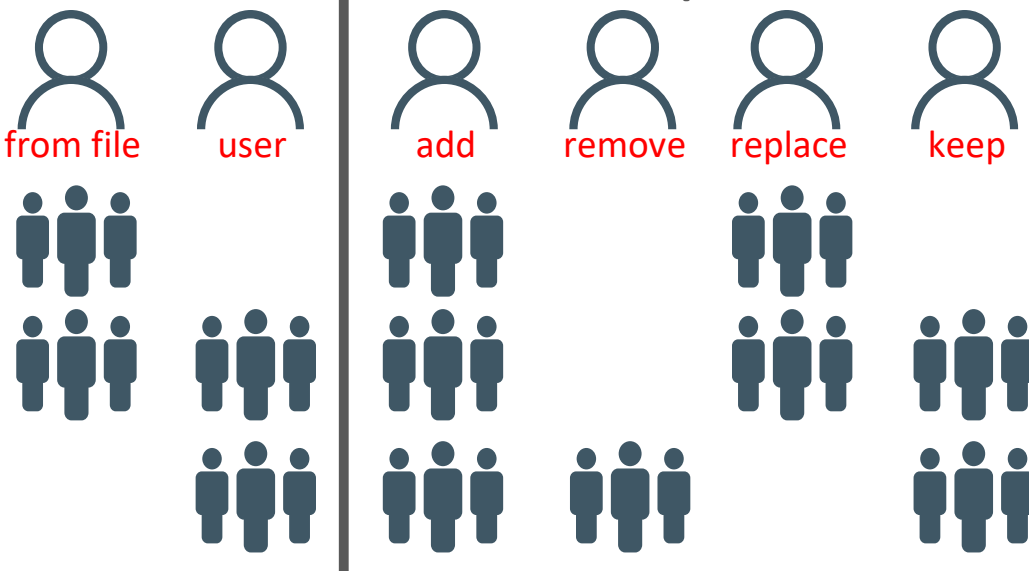
Sample	1131-U-CU-CULm-Fcj-Es-SAML Create/update users (no roles, no teams)																													
Script Basis	U: User																													
OAuth Access	'User Provisioning'																													
Basic description	CU: Creates new users and updates existing users.																													
Ideal for	<ol style="list-style-type: none">1. Assigning a user ID of your choice rather than one based on the e-mail (because you are using SAML SSO and mapping on a user's e-mail)2. Creating new users when the most significant throughput is desired, and some of those users already exist and may require updating.3. For existing users, you only wish to update the 'BI Licence Type', the users' manager or e-mail address*. (* Updates to the e-mail are impossible when SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on 'userid' or 'custom', and the e-mail remains a mapped property.)4. Creating a .csv file is easier compared to creating a .json.5. SAP Analytics Cloud is configured with SAML Single-Sign-On mapping on the e-mail, but you wish to create users with an ID determined by you rather than one derived from the e-mail address.6. SAP Analytics Cloud is set up for Single-Sign-On using SAML and mapping the user on a custom property different from the userid. For example, you can create a user with userid M_SHAW, with e-mail matthew@sap.com and custom property M-SHAW																													
Not suitable when	<ol style="list-style-type: none">1. Users need roles assigned to them directly2. Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file																													
Notes	<ul style="list-style-type: none">• Use this script and assign users to teams with scripts 1160x• It does the same as sample script 1121, except this is designed for when SAP Analytics Cloud is setup for SAML Single-Sign-On and mapping the user on the e-mail or a custom property. This script will create the userid with the desired setting, by creating a user with an e-mail address that contains the userid, before then going on to update the user's e-mail with the desired e-mail setting, and if using a custom property, the SAML mapping userName property is then also updated• If you are using your own Identity Provider with SAML SSO mapping on the 'e-mail' or 'custom' and you wish to update the e-mail or the custom property, then you can also use sample scripts 1409, 1419 and 1429																													
Data file syntax	F: .csv and .json <table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_givename</td><td>string</td><td>Initial given name</td></tr><tr><td>file_familyname</td><td>string</td><td>Initial family name</td></tr><tr><td>file_displayname</td><td>string</td><td>Initial display name</td></tr><tr><td>file_email</td><td>string</td><td>Email address</td></tr><tr><td>file_managerid</td><td>string</td><td>Optional managers ID</td></tr><tr><td>file_samlmapping</td><td>string</td><td>Needed when SAMLSSO is set to custom, it is the custom SAML property</td></tr></table>			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_givename	string	Initial given name	file_familyname	string	Initial family name	file_displayname	string	Initial display name	file_email	string	Email address	file_managerid	string	Optional managers ID	file_samlmapping	string	Needed when SAMLSSO is set to custom, it is the custom SAML property
Field	Type	Description																												
file_userid	string	UserID																												
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																												
file_givename	string	Initial given name																												
file_familyname	string	Initial family name																												
file_displayname	string	Initial display name																												
file_email	string	Email address																												
file_managerid	string	Optional managers ID																												
file_samlmapping	string	Needed when SAMLSSO is set to custom, it is the custom SAML property																												
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.																													
How the script works	Reads the data file and for each entry (row) a user is created in SAP Analytics Cloud.																													

	<p>Be careful with the <code>file_isconcurrent</code> value as it must be a lowercase <code>true</code> or <code>false</code> and not <code>TRUE</code> OR <code>FALSE</code>.</p> <p>Creating Users</p> <p>A user will initially be created with the wrong e-mail address! This is to ensure the <code>userid</code> is created with the one desired and not one derived from the e-mail (if SAMLSSO is mapped on e-mail) or from the <code>userName</code> (if SAMLSSO is mapped on a custom property). The script will then go on to read the user (to obtain any default roles assigned) and then update the user's email with the desired setting from <code>file_email</code>. And if SAMLSSO is set to custom, then the <code>userName</code> property is updated with the <code>file_samlmapping</code> value.</p> <p>Updating Users (Iem)</p> <p>A new user will always be created, even when updating an existing user. This new duplicate user will be identified and then deleted before going on to update the 'right' users' properties.</p> <p>The following properties can be updated: <code>isconcurrent</code>, <code>email</code>, <code>managerid</code></p>
Known script volume limitations	None
Script throughput	Data not collected.
Sample Data Files	<ul style="list-style-type: none"> • Example 1 is suitable for the default authentication method and for all SAMLSSO configurations other than custom. • Example 2 is suitable for SAMLSSO configurations using a custom property. The 'file_samlmapping' contains that custom property. • Example 3 is suitable for SAMLSSO configurations using a custom property, but that custom property contains/is the e-mail address. The 'file_samlmapping' contains the e-mail address, and this is needed when the e-mail is case-sensitive.

1132-U-CU-CUleMr-Oarrk-Fcj-Es-SAML Create/update users (roles, no teams)

Sample	1132-U-CU-CULmr-Oarrk-Fcj-Es-SAML Create/update users (roles, no teams)		
Script Basis	U: User		
OAuth Access	'User Provisioning'		
Basic description	CU: Creates new users and updates existing users with roles assigned to the user directly (not generally best practice) (roles can be added/removed/replaced)		
Ideal for	<div><div></div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></</div></div></div></div>		

	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]
	file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used to assign the role for any new user.
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, ContentNamespace, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.		
How the script works	<p>Reads the data file, and for each entry (row), a user is created in SAP Analytics Cloud.</p> <p>Creating Users A user will initially be created with the wrong e-mail address! This is to ensure the userid is created with the one desired and not one derived from the e-mail (if SAMLSSO is mapped on e-mail) or from the userName (if SAMLSSO is mapped on a custom property). The script will then go on to read the user (to obtain any default roles assigned) and then update the user's e-mail with the desired setting from file_email. And if SAMLSSO is set to custom, then the userName property is updated with the file_samlmapping value.</p> <p>Roles Roles are assigned directly to the user if specified, as the user is created in the single POST call that creates the user. There is no separate call to assign the user to roles.</p> <p>The roles with the same content namespace as the environment don't need to specify the PROFILE:namespace: as part of its name. For example, if a Role called 'Role1' exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only 'Role1' needs to be specified. For any role that doesn't match the ContentNamespace then PROFILE:namespace: must be specified with namespace set accordingly. For example, the provided 'BI_Admin' role would be PROFILE:sap.epm:BI_Admin</p> <p>Roles are only added to new users if the roles_action is "add" or "replace"; otherwise, the roles are ignored, and the user will not be assigned to any roles.</p> <p>Any default roles will be added after the user is created. In a rare event, an update to the user could follow the user creation (this can happen on session recovery or a false negative response from the API, a 502 error). Should this occur, and if the roles_action is 'replace', then any default roles not specified will be removed from the user. To avoid this rare event with the use of default roles, use 'add' for roles_action when creating users. The same event could also change the 'Business Intelligence license type' from the one defined in the default role and would, instead, set it to the setting as read from the data file.</p> <p>The file_JSON_roles can be an empty array, i.e. [] If the role doesn't already exist, a 400 status error will be returned, and the 'UPDATE user' will not pass its test.</p> <p>Updating Users (lemr) A new user will always be created, even when updating an existing user. This new duplicate user will be identified and then deleted before going on to update the 'right' users' properties.</p>		

	<p>The following properties can be updated: isconcurrent, e-mail, managerid, roles</p> <p>Updating Operations (arrk)</p> <p>For updating existing users, the roles_action can be “add”, “remove”, “replace” and “keep”:</p> <p>add: will add the roles to the user</p> <p>remove: will remove the roles from the user if the user has any of the roles listed</p> <p>replace: will replace (set) the roles for the user (i.e. remove and add accordingly)</p> <p>keep: will respect any roles the user has already been assigned, no changes will be made</p> <div><p>Result of role operation on user</p></div>
Known script volume limitations	None
Script throughput	Data not collected.
Sample Data Files	<ul style="list-style-type: none">• Example 1 is suitable for the default authentication method and for all SAMLSSO configurations other than custom.• Example 2 is suitable for SAMLSSO configurations using a custom property. The ‘file_samlmapping’ contains that custom property.• Example 3 is suitable for SAMLSSO configurations using a custom property, but that custom property contains/is the e-mail address. The ‘file_samlmapping’ contains the email address, and this is needed when the e-mail is case-sensitive.

1133-U-CU-CUlemrt-Oarrk-Fcj-Es-SAML Create/update users (roles & teams)

Sample	1133-U-CU-CULmrt-Oarrk-Fcj-Es-SAML Create/update users (roles & teams)		
Script Basis	U: User		
OAuth Access	'User Provisioning'		
Basic description	CU: Creates new users and updates existing users with roles assigned to the user directly (not generally best practice) and assigns the users to teams (roles and teams can be added/removed/replaced)		
Ideal for	<div><div></div><div><div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><</div></div></div>		

	file_managerid	string	Optional ID of the manager's ID
	file_samlmapping	string	Needed when SAMLSSO is set to custom, it is the custom SAML property
	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]
	file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used to assign the role for any new user.
	file_JSON_teams	JSON values of teams	Optional JSON array of keys "value" and string values of teams. Example: [{"value":"Team1"}, {"value":"Team2"}]
	file_teams_action	string	Defines the action to perform on the users' team membership. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used for any new user to be a member of a team.
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, ContentNamespace, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.		
How the script works	<p>Reads the data file, and for each entry (row), a user is created in SAP Analytics Cloud.</p> <p>Creating Users</p> <p>A user will initially be created with the wrong e-mail address! This is to ensure the userid is created with the one desired and not one derived from the email (if SAMLSSO is mapped on e-mail) or from the userName (if SAMLSSO is mapped on a custom property). The script will then read the user (to obtain any default roles assigned) and then update the user's e-mail with the desired setting from file_email. And if SAMLSSO is set to custom, then the userName property is updated with the file_samlmapping value.</p> <p>Roles</p> <p>Roles are assigned directly to the user if specified, as the user is created in the single POST call that creates the user. There is no separate call to assign the user to roles.</p> <p>The roles with the same content namespace as the environment don't need to specify the PROFILE:namespace: as part of its name. For example, if a Role called 'Role1' exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only 'Role1' needs to be specified. For any role that doesn't match the ContentNamespace, then PROFILE:namespace: must be specified with namespace set accordingly. For example, the provided 'BI_Admin' role would be PROFILE:sap.epm:BI_Admin</p> <p>Roles are only added to new users if the roles_action is "add" or "replace"; otherwise, the roles are ignored, and the user will not be assigned to any roles.</p> <p>Any default roles will be added after the user is created. In a rare event, an update to the user could follow the user creation (this can happen on session recovery or a false negative response from the API, a 502 error). Should this occur, and if the roles_action is 'replace', then any default roles not specified will be removed from the user. To avoid this rare event with the use of default roles, use 'add' for roles_action when creating users. The same event could also change the 'Business Intelligence license type' from the one defined in the default role and would, instead, set it to the setting as read from the data file.</p>		

The file_JSON_roles can be an empty array, i.e. []
If the role doesn't already exist a 400 status error will be returned and the 'UPDATE user' will not pass its test.

Teams

User-to-team assignment cannot currently be performed using the API endpoint that is used to create or update a user; instead, a different endpoint must be used. This endpoint is for the Teams (Groups) themselves. It means more than one user can be specified per team, but if a single user requires membership changes to multiple teams, it will require multiple GET/PUT pair calls, one GET/PUT pair per team.

User membership to teams is performed every 500 entries (users), and at the end of the file, i.e. they are 'batched' together to achieve the greatest throughput. The users for each team are batched together to form a single GET/PUT pair per team rather than a GET/PUT pair per user, per team.

If the team does not exist, it will be created **without** a team folder, and the display name will match its name; otherwise, the existing team's display name is respected.

(When SAP Analytics Cloud is hosted on an SAP Data Centre (NEO), then teams are created **with** a team folder, and the display name for the team will have the text "with Team Folder". E.g. "Team1 with Team Folder". If you'd like the team folder description to share the same team display name, use sample script 1501 before running this sample script. Alternatively, update the team folder description manually.)

Users added into teams are 'chunked' into multiple PUT calls. However, given the batch size is 500 users, it's unlikely any chunking will be applied, and it would only occur when the team has 29225 users!

The file_JSON_teams can be an empty array, i.e. []

Teams are only added to new users if the teams_action is "add" or "replace"; otherwise, the teams are ignored, and the user will not be assigned to any teams.

Updating Users (lemrt)

A new user will always be created, even when updating an existing user. This new duplicate user will be identified and then deleted before going on to update the 'right' users' properties.

The following properties can be updated: isconcurrent, email, managerid, roles and teams assignment.

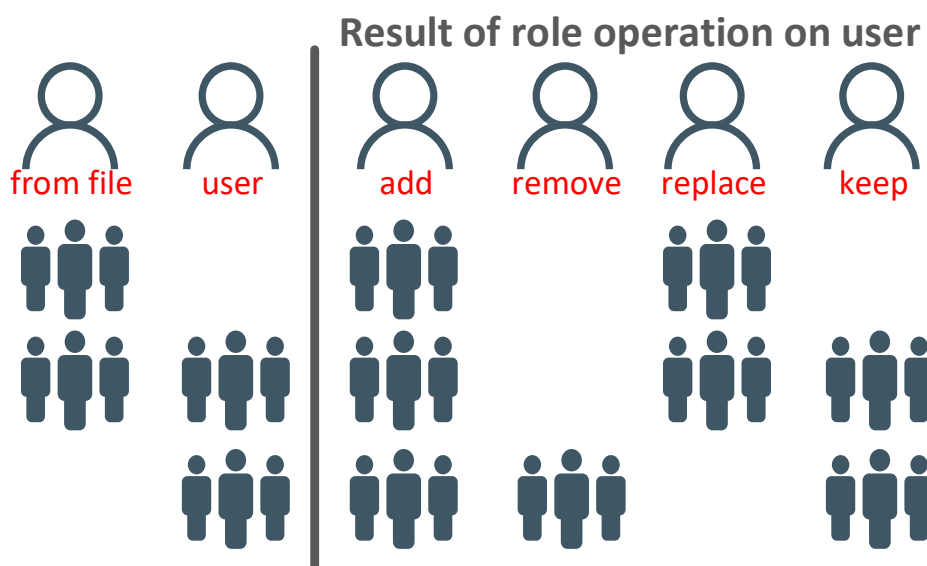
Updating Operations (arrk)

For updating existing users, the roles_action can be "add", "remove", "replace" and "keep":
add: will add the roles to the user.

remove: will remove the roles from the user if the user has any of the roles listed.

replace: will replace (set) the roles for the user (i.e. remove and add accordingly).

keep: will respect any roles the user has already been assigned, no changes will be made.



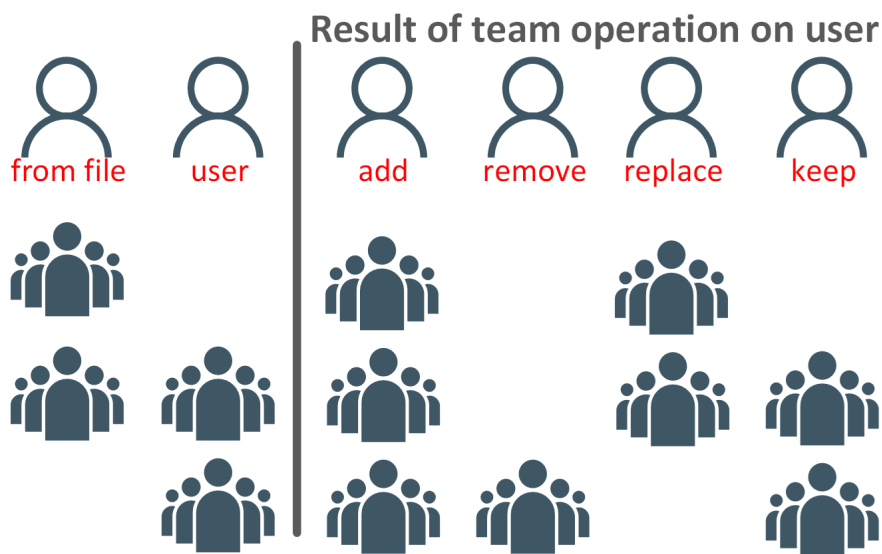
For updating existing users, the `teams_action` can be “add”, “remove”, “replace” and “keep”:

add: will add the user to the team.

remove: will remove the user from the team, if the user is a member of any of the teams listed.

replace: will replace (set) the team membership of the user (i.e. remove and add accordingly).

keep: will respect any teams the user is already a member of; no changes will be made.



Known script volume limitations	None
Script throughput	Data not collected.
Sample Data Files	<ul style="list-style-type: none"> • Example 1 is suitable for the default authentication method and for all SAMLSSO configurations other than custom. • Example 2 is suitable for SAMLSSO configurations using a custom property. The 'file_samlmapping' contains that custom property. • Example 3 is suitable for SAMLSSO configurations using a custom property, but that custom property contains/is the e-mail address. The 'file_samlmapping' contains the e-mail address, which is needed when the e-mail is case-sensitive.

1201-U-UC-UClem-Oarrk-Fcj-Es-Update/create users (no roles, no teams)

Sample	1201-U-UC-UClem-Oarrk-Fcj-Es-Update/create users (no roles, no teams)																										
Script Basis	U: User																										
OAuth Access	'User Provisioning'																										
Basic description	UC: Updates existing users and creates new users																										
Ideal for	<div><div><div>1.</div><div>Creating new users when some existing users have a different e-mail address than those already registered. This avoids duplicate '_1' users being created, as is the problem with scripts 11xx.</div></div><div><div>2.</div><div>Updating users' e-mail addresses (i.e. a user might already exist with a different e-mail than already registered, and this avoids duplicate '_1' users being created, as is the problem with scripts 11xx)</div></div><div><div>3.</div><div>For existing users, you only wish to update the 'BI Licence Type', the users' manager, and the user's e-mail address *. (* Updates to the e-mail are not possible when SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on 'userid' or 'custom'.)</div></div><div><div>4.</div><div>Creating a .csv file is easier compared to creating a .json.</div></div><div><div>5.</div><div>Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file</div></div><div><div>6.</div><div>You are using the default Identity Provider that comes with SAP Analytics Cloud, OR SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'USER ID'</div></div></div>																										
Not suitable when	<div><div><div>1.</div><div>The very greatest throughout of creating users is desired (use scripts 11xx instead)</div></div><div><div>2.</div><div>Users need roles assigned to them directly.</div></div><div><div>3.</div><div>You need to ensure the 'user id' is determined by you and not derived from the e-mail because SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'e-mail' or on a custom property that is different from the userid.</div></div></div>																										
Notes	<div>Use this script and assign users to teams with scripts 1160x</div> <div>If you are using your own Identity Provider with SAML SSO mapping on the 'e-mail' and you wish to update the e-mail, then you can also use sample scripts 1409 and 1429</div>																										
Data file syntax	<div>F: .csv and .json</div> <table><tr><td>Field</td><td>Type</td><td>Description</td></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_givename</td><td>string</td><td>Initial given name</td></tr><tr><td>file_familyname</td><td>string</td><td>Initial family name</td></tr><tr><td>file_displayname</td><td>string</td><td>Initial display name</td></tr><tr><td>file_email</td><td>string</td><td>Email address</td></tr><tr><td>file_managerid</td><td>string</td><td>Optional managers ID</td></tr></table>			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_givename	string	Initial given name	file_familyname	string	Initial family name	file_displayname	string	Initial display name	file_email	string	Email address	file_managerid	string	Optional managers ID
Field	Type	Description																									
file_userid	string	UserID																									
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																									
file_givename	string	Initial given name																									
file_familyname	string	Initial family name																									
file_displayname	string	Initial display name																									
file_email	string	Email address																									
file_managerid	string	Optional managers ID																									
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.																										
How the script works	<div>Reads the data file, and for each entry (row), the user is read from SAP Analytics Cloud. If the user is not found, the user is created; otherwise, a comparison is made of the existing user's properties and compared with the provided properties from the file. If an update is necessary, the user is appropriately updated; otherwise, the user is skipped.</div> <div>Be careful with the file_isconcurrent value as it must be a lowercase true or false and not TRUE OR FALSE.</div>																										

	<p>Creating Users</p> <p>If the e-mail address already exists, but the username does not, a 400 error will be returned, and the script will record a 'CREATE user' failure.</p> <p>A new user will not be created if the email address provided belongs to a user that already exists.</p> <p>Updating Users (Iem)</p> <p>The user will only be updated if it needs to be.</p> <p>The following properties can be updated: isconcurrent, email and managerid</p>
Known script volume limitations	None

Script throughput		Service with 0 users	Service with 80,000 users
	Creating users	0.26 users / sec 3.87 secs / user (4.3%)	0.13 users / sec 7.77 secs / user (2.3%)
	Updating users (not email, not team)	0.33 users / sec 3.01 secs / users (5.7%)	0.11 users / sec 8.88 secs / user (2/1%)
	Updating users' email	0.33 users / sec 3.01 secs / users (5.7%)	0.11 users / sec 8.88 secs / user (2/1%)
	Updating users (that don't need an update)	0.82 users / sec 1.23 secs / users (6.4%)	0.17 users / sec 5.81 secs / user (1.4%)
	Updating user and team (removing 3 teams and adding 3 teams)	0.3 users / sec 3.28 secs / user (10.9%)	0.12 users / sec 8.09 secs / user (4.3%)

1202-U-UC-UCleMr-Oarrk-Fj-Es-Update/create users (with roles, no teams)

Sample	1202-U-UC-UCleMr-Oarrk-Fj-Es-Update/create users (with roles, no teams)																										
Script Basis	U: User																										
OAuth Access	'User Provisioning'																										
Basic description	UC: Updates existing users and creates new users with roles assigned to the user directly (not generally best practice) (roles can be added/removed/replaced)																										
Ideal for	<div><div><div>1.</div><div>Creating new users when some existing users have a different email address than those already registered. This avoids duplicate ' _1' users being created, as is the problem with scripts 11xx.</div></div><div><div>2.</div><div>Updating users' e-mail addresses (i.e. a user might already exist with a different e-mail than already registered, and this avoids duplicate ' _1' users being created, as is the problem with scripts 11xx)</div></div><div><div>3.</div><div>For existing users, you only wish to update the 'BI Licence Type', the users' manager, the user's e-mail address* and their role assignment. (* Updates to the email are not possible when SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on 'userid' or 'custom'.)</div></div><div><div>4.</div><div>Users need roles assigned to them directly.</div></div><div><div>5.</div><div>Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file</div></div><div><div>6.</div><div>You are using the default Identity Provider that comes with SAP Analytics Cloud, OR SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'USER ID'</div></div></div>																										
Not suitable when	<div><div><div>1.</div><div>The very greatest throughout of creating users is desired (use scripts 11xx instead)</div></div><div><div>2.</div><div>You need to ensure the 'user id' is determined by you and not derived from the e-mail because SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'e-mail' or on a custom property that is different from the userid.</div></div></div>																										
Notes	<div><div><div>It does the same as sample 1201, only this script enables roles to be assigned to users. Assign users to teams with scripts 160x.</div><div>If the file_isconcurrent setting is set to true (for concurrent), but should the user be assigned roles directly (rather than indirectly via a team) and if any of those directly assigned roles are planning roles, then the isConcurrent setting will not be respected, i.e., the isConcurrent will be set to false (a named user).</div><div>If you are using your own Identity Provider with SAML SSO mapping on the 'e-mail' and you wish to update the e-mail, then you can also use sample scripts 1409 and 1429</div></div></div>																										
Data file syntax	<div><div>F: .json</div><table><tr><td>Field</td><td>Type</td><td>Description</td></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_givenname</td><td>string</td><td>Initial given name</td></tr><tr><td>file_familyname</td><td>string</td><td>Initial family name</td></tr><tr><td>file_displayname</td><td>string</td><td>Initial display name</td></tr><tr><td>file_email</td><td>string</td><td>Email address</td></tr><tr><td>file_managerid</td><td>string</td><td>Optional ID of the manager's ID</td></tr></table></div>			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_givenname	string	Initial given name	file_familyname	string	Initial family name	file_displayname	string	Initial display name	file_email	string	Email address	file_managerid	string	Optional ID of the manager's ID
Field	Type	Description																									
file_userid	string	UserID																									
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																									
file_givenname	string	Initial given name																									
file_familyname	string	Initial family name																									
file_displayname	string	Initial display name																									
file_email	string	Email address																									
file_managerid	string	Optional ID of the manager's ID																									

	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]
	file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used to assign the role for any new user.
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, ContentNamespace, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.		
How the script works	<p>Reads the data file, and for each entry (row), the user is read from SAP Analytics Cloud. If the user is not found, the user is created; otherwise, a comparison is made of the existing user's properties and compared with the provided properties from the file. If an update is necessary, then the user is appropriately updated; otherwise, the user is skipped.</p> <p>Creating Users If the e-mail address already exists, but the username does not, a 400 error will be returned, and the script will record a 'CREATE user' failure.</p> <p>A new user will not be created if the email address provided belongs to a user that already exists.</p> <p>Roles Roles are assigned directly to the user if specified, as the user is created in the single POST call that creates the user. There is no separate call to assign the user to roles.</p> <p>The roles with the same content namespace as the environment don't need to specify the PROFILE:namespace: as part of its name. For example, if a Role called 'Role1' exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only 'Role1' needs to be specified. For any role that doesn't match the ContentNamespace then PROFILE:namespace: must be specified with namespace set accordingly. For example, the provided 'BI_Admin' role would be PROFILE:sap.epm:BI_Admin</p> <p>Roles are only added to new users if the roles_action is "add" or "replace"; otherwise, the roles are ignored, and the user will not be assigned to any roles.</p> <p>Any default roles will be added after the user is created. In a rare event, an update to the user could follow the user creation (this can happen on session recovery or a false negative response from the API, a 502 error). Should this occur, and if the roles_action is 'replace', then any default roles not specified will be removed from the user. To avoid this rare event with the use of default roles, use 'add' for roles_action when creating users. The same event could also change the 'Business Intelligence license type' from the one defined in the default role and would, instead, set it to the setting as read from the data file.</p> <p>The file_JSON_roles can be an empty array, i.e. [] If the role doesn't already exist a 400 status error will be returned, and the 'UPDATE user' or 'CREATE user' will not pass its test.</p> <p>Updating Users (lemr) The user will only be updated if it needs to be.</p>		

The following properties can be updated: isconcurrent, email, managerid, roles.

Updating Operations (arrk)

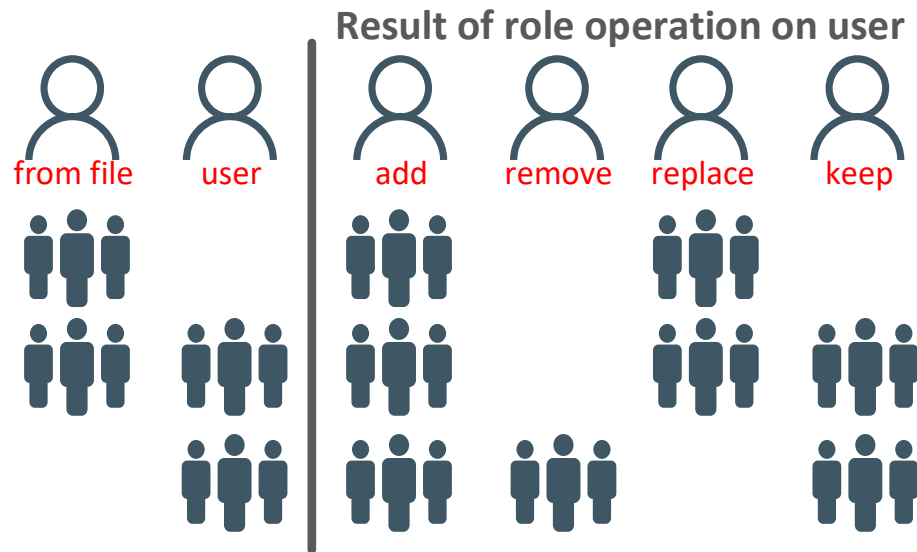
For updating existing users, the roles_action can be “add”, “remove”, “replace” and “keep”:

add: will add the roles to the user

remove: will remove the roles from the user if the user has any of the roles listed

replace: will replace (set) the roles for the user (i.e. remove and add accordingly)

keep: will respect any roles the user has already been assigned; no changes will be made



Known script volume
limitations

None

Script throughput

	Service with 0 users	Service with 80,000 users
Creating users	0.26 users / sec 3.87 secs / user (4.3%)	0.13 users / sec 7.77 secs / user (2.3%)
Updating users (not email, not team)	0.33 users / sec 3.01 secs / users (5.7%)	0.11 users / sec 8.88 secs / user (2/1%)
Updating users' email	0.33 users / sec 3.01 secs / users (5.7%)	0.11 users / sec 8.88 secs / user (2/1%)
Updating users (that don't need an update)	0.82 users / sec 1.23 secs / users (6.4%)	0.17 users / sec 5.81 secs / user (1.4%)
Updating user and team (removing 3 teams and adding 3 teams)	0.3 users / sec 3.28 secs / user (10.9%)	0.12 users / sec 8.09 secs / user (4.3%)

1203-U-UC-UClemrt-Oarrk-Fj-Es-Update/create users (with roles & teams)

Sample	1203-U-UC-UClemrt-Oarrk-Fj-Es-Update/create users (with roles & teams)														
Script Basis	U: User														
OAuth Access	'User Provisioning'														
Basic description	UC: Updates existing users and creates new users with roles assigned to the user directly (not generally best practice) and assigns the users to teams (roles and teams can be added/removed/replaced)														
Ideal for	<ol style="list-style-type: none">1. Creating new users when some existing users have a different e-mail address than those already registered. This avoids duplicate '_1' users being created, as is the problem with scripts 11xx.2. Updating users' e-mail addresses (i.e. a user might already exist with a different e-mail than already registered, and this avoids duplicate '_1' users being created, as is the problem with scripts 11xx)3. For existing users, you wish to update the 'BI Licence Type', the users' manager, the user's e-mail address*, their role assignment and their team assignment (* Updates to the e-mail is not possible when SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on 'userid' or 'custom'.)4. Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file5. You are using the default Identity Provider that comes with SAP Analytics Cloud, OR SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'USER ID'														
Not suitable when	<ol style="list-style-type: none">1. The very greatest throughout of creating users is desired (use scripts 11xx instead)2. Running another script to assign users to teams is possible (i.e. given a list of users, add the users to a team), and users still need to be in a team.3. You need to ensure the 'user id' is determined by you and not derived from the e-mail because SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'e-mail' or on a custom property that is different from the userid.														
Notes	<p>It does the same as sample 1202; only this script enables users to be assigned to teams. In the case when:</p> <ol style="list-style-type: none">1. team updates are the only thing that needs to be updated,2. or team updates can be separated from user updates/creation <p>then use script 1201 or 1202 to update and create users and scripts 6xx to assign users to teams. This will provide the greatest throughput and is likely to be far more reliable for large volumes. Use this script, 1203, for only lower volumes of users and updates (typically, say less than 500 users and less than 50 teams).</p> <p>If the file_isconcurrent setting is set to true (for concurrent), but should the user be assigned roles directly (rather than indirectly via a team) and if any of those directly assigned roles are planning roles, then the isConcurrent setting will not be respected, i.e., the isConcurrent will be set to false (a named user).</p> <p>If you are using your own Identity Provider with SAML SSO mapping on the 'e-mail' and you wish to update the e-mail, then you can also use sample scripts 1409 and 1429</p>														
Data file syntax	<table><tr><td colspan="3">F: .json</td></tr><tr><td>Field</td><td>Type</td><td>Description</td></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr></table>			F: .json			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user
F: .json															
Field	Type	Description													
file_userid	string	UserID													
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user													

	file_givename	string	Initial given name
	file_familyname	string	Initial family name
	file_displayname	string	Initial display name
	file_email	string	Email address
	file_managerid	string	Optional ID of the manager's ID
	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]
	file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used to assign the role for any new user.
	file_JSON_teams	JSON values of teams	Optional JSON array of keys "value" and string values of teams. Example: [{"value":"Team1"}, {"value":"Team2"}]
	file_teams_action	string	Defines the action to perform on the users' team membership. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used for any new user to be a member of a team.
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, ContentNamespace, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.		
How the script works	<p>Reads the data file, and for each entry (row), the user is read from SAP Analytics Cloud. If the user is not found, the user is created; otherwise, a comparison is made of the existing user's properties and compared with the provided properties from the file. If a non-team assignment update is necessary, then the user is appropriately updated; otherwise, the user is skipped. Teams are appropriately updated every 500 users and at the end of the file.</p> <p>Creating Users If the e-mail address already exists, but the username does not, a 400 error will be returned, and the script will record a 'CREATE user' failure.</p> <p>A new user will not be created if the email address provided belongs to a user that already exists.</p> <p>Roles Roles are assigned directly to the user if specified, as the user is created in the single POST call that creates the user. There is no separate call to assign the user to roles.</p> <p>The roles with the same content namespace as the environment don't need to specify the PROFILE:namespace: as part of its name. For example, if a Role called 'Role1' exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only 'Role1' needs to be specified. For any role that doesn't match the ContentNamespace, then PROFILE:namespace: must be specified with namespace set accordingly. For example, the provided 'BI_Admin' role would be PROFILE:sap.epm:BI_Admin</p> <p>Roles are only added to new users if the roles_action is "add" or "replace", otherwise, the roles are ignored, and the user will not be assigned to any roles.</p>		

Any default roles will be added after the user is created. In a rare event, an update to the user could follow the user creation (this can happen on session recovery or a false negative response from the API, a 502 error). Should this occur, and if the roles_action is 'replace', then any default roles not specified will be removed from the user. To avoid this rare event with the use of default roles, use 'add' for roles_action when creating users. The same event could also change the 'Business Intelligence license type' from the one defined in the default role and would, instead, set it to the setting as read from the data file.

The file_JSON_roles can be an empty array, i.e. []

If the role doesn't already exist, a 400 status error will be returned, and the 'UPDATE user' will not pass its test.

Teams

User-to-team assignment cannot currently be performed using the API endpoint that is used to create or update a user; instead, a different endpoint must be used. This endpoint is for the Teams (Groups) themselves. It means more than one user can be specified per team, but if a single user requires membership changes to multiple teams, it will require multiple GET/PUT pair requests, one GET/PUT pair per team.

User membership to teams is performed every 500 entries (users), and at the end of the file, i.e. they are 'batched' together to achieve the most significant throughput. The users for each team are batched together to form a single GET/PUT pair per team rather than a GET/PUT pair per user, per team.

If the team does not exist, it will be created **without** a team folder, and the display name will match its name; otherwise, the existing team's display name is respected.

(When SAP Analytics Cloud is hosted on an SAP Data Centre (NEO), then teams are created **with** a team folder, and the display name for the team will have the text "with Team Folder". E.g. "Team1 with Team Folder". If you'd like the team folder description to share the same team display name, use sample script 1501 before running this sample script. Alternatively, update the team folder description manually.)

Users added into teams are 'chunked' into multiple PUT calls. However, given the batch size is 500 users, it's unlikely any chunking will be applied, and it would only occur when the team has 29225 users!

The file_JSON_teams can be an empty array, i.e. []

Teams are only added to new users if the teams_action is "add" or "replace"; otherwise, the teams are ignored, and the user will not be assigned to any teams.

Updating Users (lemrt)

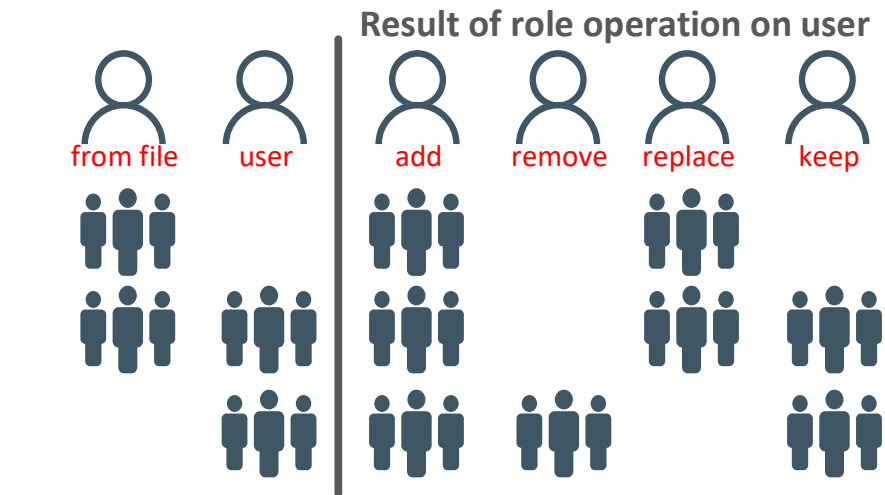
The following properties can be updated: isconcurrent, email, managerid, roles and teams assignment.

Updating Operations (arrk)

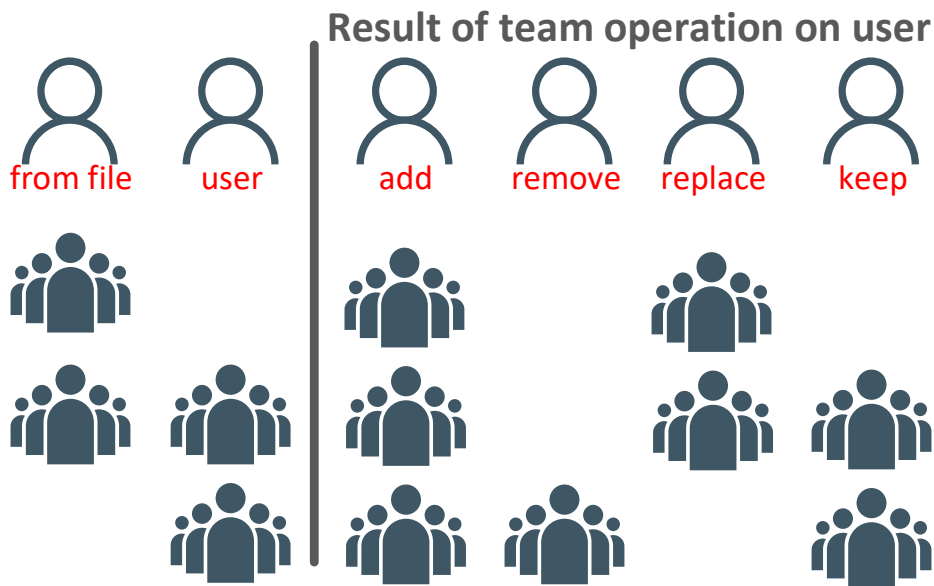
For updating existing users, the roles_action can be "add", "remove", "replace" and "keep":
add: will add the roles to the user.

remove: will remove the roles from the user if the user has any of the roles listed.

replace: will replace (set) the roles for the user (i.e. remove and add accordingly).
keep: will respect any roles the user has already been assigned, no changes will be made.



For updating existing users, the teams_action can be “add”, “remove”, “replace” and “keep”:
add: will add the user to the team.
remove: will remove the user from the team, if the user is a member of any of the teams listed.
replace: will replace (set) the team membership of the user (i.e. remove and add accordingly).
keep: will respect any teams the user is already a member of; no changes will be made.



Known script volume limitations

If the number of team updates (number of teams and the number of ‘chunk’ updates per team) is large, typically many hundred, then the script will use a large memory set, which could cause instability issues in the client. Running this script in the command line rather than the user interface is recommended to reduce script failures.

The batch size of 500 to update teams has been explicitly set to allow team updates to be processed mid-file. This is to avoid the situation where no team assignments are updated when the script fails mid-operation. If the number of users and team assignments is large (roughly over, say, 2000 users and over 100 teams, each team of over 2000 members), then

	this script is likely to be problematic. In such cases, use script 1201 with scripts 16xx to assign users to teams.		
Script throughput		Service with 0 users	Service with 80,000 users
	Creating users and users being added into 3 teams	0.30 users / sec 3.28 secs / user	0.12 users / sec 8.10 secs / user
	Updating users (not email, not team)	0.33 users / sec 3.01 secs / users (5.7%)	0.11 users / sec 8.88 secs / user (2/1%)
	Updating users' email	0.33 users / sec 3.01 secs / users (5.7%)	0.11 users / sec 8.88 secs / user (2/1%)
	Updating users (that don't need an update)	0.82 users / sec 1.23 secs / users (6.4%)	0.17 users / sec 5.81 secs / user (1.4%)
	Updating user and team (removing 3 teams and adding 3 teams)	0.3 users / sec 3.28 secs / user (10.9%)	0.12 users / sec 8.09 secs / user (4.3%)

1231-U-UC-UClem -Fcj-Es-SAML Update/create users (no roles, no teams)

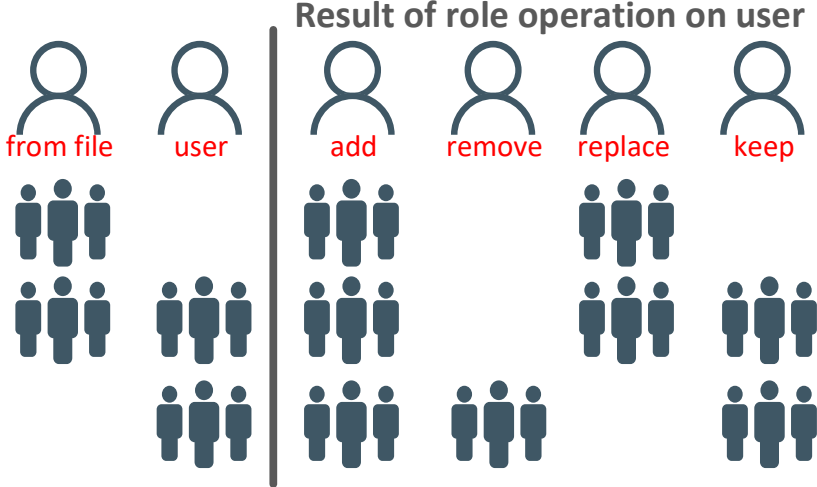
Sample	1231-U-UC-UClem -Fcj-Es-SAML Update/create users (no roles, no teams)																													
Script Basis	U: User																													
OAuth Access	'User Provisioning'																													
Basic description	UC: Updates existing users and creates new users																													
Ideal for	<div><div></div><div><div><div>1.</div><div>Assigning a user ID of your choice rather than one based on the e-mail (because you are using SAML SSO and mapping on a user's e-mail or a custom property)</div></div><div><div>2.</div><div>Updating users' e-mail addresses (i.e. a user might already exist with a different e-mail than already registered).</div></div><div><div>3.</div><div>For existing users, you only wish to update the 'BI Licence Type', the users' manager, and the user's e-mail address*. (* Updates to the email are not possible when SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on 'userid' or 'custom'.)</div></div><div><div>4.</div><div>Creating a .csv file is easier compared to creating a .json.</div></div><div><div>5.</div><div>Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file</div></div><div><div>6.</div><div>SAP Analytics Cloud is set up for Single-Sign-On with SAML and mapping the user on the e-mail and to ensure the user is created with an ID determined by you rather than one derived from the e-mail address.</div></div><div><div>7.</div><div>SAP Analytics Cloud is set up for Single-Sign-On using SAML and mapping the user on a custom property which is different from the userid. For example, you can create a user with userid M_SHAW, with e-mail matthew@sap.com and custom property M-SHAW</div></div></div></div>																													
Not suitable when	<div><div></div><div><div><div>1.</div><div>The very greatest throughout of creating users is desired (use scripts 11xx instead, if possible)</div></div><div><div>2.</div><div>Users need roles assigned to them directly.</div></div></div></div>																													
Notes	<div><div></div><div><div>Use this script and assign users to teams with scripts 160x</div><div>If you are using your own Identity Provider with SAML SSO mapping on the 'e-mail' or 'custom' and you wish to update the e-mail or the custom property, then you can also use sample scripts 1409, 1419 and 1429</div></div></div>																													
Data file syntax	<div><div></div><div><div>F: .csv and .json</div><table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_givename</td><td>string</td><td>Initial given name</td></tr><tr><td>file_familyname</td><td>string</td><td>Initial family name</td></tr><tr><td>file_displayname</td><td>string</td><td>Initial display name</td></tr><tr><td>file_email</td><td>string</td><td>Email address</td></tr><tr><td>file_managerid</td><td>string</td><td>Optional managers ID</td></tr><tr><td>file_samlmapping</td><td>string</td><td>Needed when SAMLSSO is set to custom, it is the custom SAML property</td></tr></table></div></div>			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_givename	string	Initial given name	file_familyname	string	Initial family name	file_displayname	string	Initial display name	file_email	string	Email address	file_managerid	string	Optional managers ID	file_samlmapping	string	Needed when SAMLSSO is set to custom, it is the custom SAML property
Field	Type	Description																												
file_userid	string	UserID																												
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																												
file_givename	string	Initial given name																												
file_familyname	string	Initial family name																												
file_displayname	string	Initial display name																												
file_email	string	Email address																												
file_managerid	string	Optional managers ID																												
file_samlmapping	string	Needed when SAMLSSO is set to custom, it is the custom SAML property																												
Environment	<div><div></div><div><div>Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.</div></div></div>																													
How the script works	<div><div></div><div><div>Reads the data file, and for each entry (row), the user is read from SAP Analytics Cloud. If the user is not found, the user is created, otherwise, a comparison is made of the existing user's properties and compared with the provided properties from the file. If an update is necessary, then the user is appropriately updated, otherwise, the user is skipped.</div></div></div>																													

	<p>Be careful with the <code>file_isconcurrent</code> value as it must be a lowercase <code>true</code> or <code>false</code> and not <code>TRUE</code> OR <code>FALSE</code>.</p> <p>Creating Users A user will initially be created with the wrong e-mail address! This is to ensure the <code>userid</code> is created with the one desired and not one derived from the e-mail (if SAMLSSO is mapped on e-mail) or from the <code>userName</code> (if SAMLSSO is mapped on a custom property). The script will then go on to read the user (to obtain any default roles assigned) and then update the user's e-mail with the desired setting from <code>file_email</code>. And if SAMLSSO is set to custom, then the <code>userName</code> property is updated with the <code>file_samlmapping</code> value.</p> <p>Updating Users (Iem) A new user will always be created, even when updating an existing user. This new duplicate user will be identified and then deleted before updating the 'right' users' properties. The following properties can be updated: <code>isconcurrent</code>, <code>email</code> and <code>managerid</code></p>
Known script volume limitations	None
Script throughput	Data not collected.
Sample Data Files	<ul style="list-style-type: none"> • Example 1 is suitable for the default authentication method and for all SAMLSSO configurations other than custom. • Example 2 is suitable for SAMLSSO configurations using a custom property. The 'file_samlmapping' contains that custom property. • Example 3 is suitable for SAMLSSO configurations using a custom property, but that custom property contains/is the e-mail address. The 'file_samlmapping' contains the e-mail address, and this is needed when the e-mail is case-sensitive.

1232-U-UC-UCleMr-Oarrk-Fj-Es-SAML Update/create users (with roles, no teams)

Sample	1232-U-UC-UCleMr-Oarrk-Fj-Es-SAML Update/create users (with roles, no teams)																							
Script Basis	U: User																							
OAuth Access	'User Provisioning'																							
Basic description	UC: Updates existing users and creates new users with roles assigned to the user directly (not generally best practice) (roles can be added/removed/replaced)																							
Ideal for	<div><div><div>1.</div><div>Assigning a user ID of your choice rather than one based on the e-mail (because you are using SAML SSO and mapping on a user's e-mail)</div></div><div><div>2.</div><div>Updating users' e-mail addresses (i.e. a user might already exist with a different e-mail than already registered, and this avoids duplicate '_1' users being created, as is the problem with scripts 11xx)</div></div><div><div>3.</div><div>For existing users, you only wish to update the 'BI Licence Type', the users' manager, the user's e-mail address* and their role assignment. (* Updates to the e-mail is not possible when SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on 'userid' or 'custom'.)</div></div><div><div>4.</div><div>Users need roles assigned to them directly.</div></div><div><div>5.</div><div>Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file</div></div><div><div>6.</div><div>SAP Analytics Cloud is set up for Single-Sign-On using SAML and mapping the user on the e-mail to ensure the user is created with an ID determined by you rather than one derived from the e-mail address.</div></div><div><div>7.</div><div>SAP Analytics Cloud is set up for Single-Sign-On using SAML and mapping the user on a custom property different from the userid. For example, you can create a user with userid M_SHAW, with e-mail matthew@sap.com and custom property M-SHAW</div></div></div>																							
Not suitable when	<div><div><div>1.</div><div>The very greatest throughout of creating users is desired (use scripts 11xx instead, if possible)</div></div></div>																							
Notes	<div><div><div>It does the same as sample 1201, only this script enables roles to be assigned to users. Assign users to teams with scripts 160x.</div><div>If the file_isconcurrent setting is set to true (for concurrent), but should the user be assigned roles directly (rather than indirectly via a team) and if any of those directly assigned roles are planning roles, then the isConcurrent setting will not be respected, i.e., the isConcurrent will be set to false (a named user).</div><div>If you are using your own Identity Provider with SAML SSO mapping on the 'e-mail' or 'custom' and you wish to update the e-mail or the custom property, then you can also use sample scripts 1409, 1419 and 1429</div></div></div>																							
Data file syntax	<div><div><div>F: .json</div><table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_givename</td><td>string</td><td>Initial given name</td></tr><tr><td>file_familyname</td><td>string</td><td>Initial family name</td></tr><tr><td>file_displayname</td><td>string</td><td>Initial display name</td></tr><tr><td>file_email</td><td>string</td><td>Email address</td></tr></table></div></div>			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_givename	string	Initial given name	file_familyname	string	Initial family name	file_displayname	string	Initial display name	file_email	string	Email address
Field	Type	Description																						
file_userid	string	UserID																						
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																						
file_givename	string	Initial given name																						
file_familyname	string	Initial family name																						
file_displayname	string	Initial display name																						
file_email	string	Email address																						

	file_managerid	string	Optional ID of the manager's ID
	file_samlmapping	string	Needed when SAMLSSO is set to custom, it is the custom SAML property
	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value": "Role1"}, {"value": "PROFILE:sap.epm:BI_Admin"}]
	file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used to assign the role for any new user.
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, ContentNamespace, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.		
How the script works	<p>Reads the data file, and for each entry (row), the user is read from SAP Analytics Cloud. If the user is not found, the user is created; otherwise, a comparison is made of the existing user's properties and compared with the provided properties from the file. If an update is necessary, the user is appropriately updated; otherwise, the user is skipped.</p> <p>Creating Users</p> <p>A user will initially be created with the wrong e-mail address! This is to ensure the userid is created with the one desired and not one derived from the e-mail (if SAMLSSO is mapped on e-mail) or from the userName (if SAMLSSO is mapped on a custom property). The script will then read the user (to obtain any default roles assigned) and then update the user's e-mail with the desired setting from file_email. And if SAMLSSO is set to custom, then the userName property is updated with the file_samlmapping value.</p> <p>Roles</p> <p>Roles are assigned directly to the user if specified, as the user is created in the single POST call that creates the user. There is no separate call to assign the user to roles.</p> <p>The roles with the same content namespace as the environment don't need to specify the PROFILE:namespace: as part of its name. For example, if a Role called 'Role1' exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only 'Role1' needs to be specified. For any role that doesn't match the ContentNamespace, then PROFILE:namespace: must be specified with namespace set accordingly. For example, the provided 'BI_Admin' role would be PROFILE:sap.epm:BI_Admin</p> <p>Roles are only added to new users if the roles_action is "add" or "replace", otherwise, the roles are ignored, and the user will not be assigned to any roles.</p> <p>Any default roles will be added after the user is created. In a rare event, an update to the user could follow the user creation (this can happen on session recovery, or a false negative response from the API, a 502 error). Should this occur, and if the roles_action is 'replace', then any default roles not specified will be removed from the user. To avoid this rare event with the use of default roles, use 'add' for roles_action when creating users. The same event could also change the 'Business Intelligence license type' from the one defined in the default role and would, instead, set it to the setting as read from the data file.</p> <p>The file_JSON_roles can be an empty array, i.e. []</p>		

	<p>If the role doesn't already exist, a 400 status error will be returned, and the 'UPDATE user' or 'CREATE user' will not pass its test.</p> <p>Updating Users (lemr)</p> <p>A new user will always be created, even when updating an existing user. This new duplicate user will be identified and then deleted before updating the 'right' users' properties. The following properties can be updated: isconcurrent, email, managerid, roles.</p> <p>Updating Operations (arrk)</p> <p>For updating existing users, the roles_action can be "add", "remove", "replace" and "keep":</p> <p>add: will add the roles to the user</p> <p>remove: will remove the roles from the user if the user has any of the roles listed</p> <p>replace: will replace (set) the roles for the user (i.e. remove and add accordingly)</p> <p>keep: will respect any roles the user has already been assigned; no changes will be made</p> <p>Result of role operation on user</p> 
Known script volume limitations	None
Script throughput	Data not collected.
Sample Data Files	<ul style="list-style-type: none">• Example 1 is suitable for the default authentication method and all SAMLSSO configurations other than custom.• Example 2 is suitable for SAMLSSO configurations using a custom property. The 'file_samlmapping' contains that custom property.• Example 3 is suitable for SAMLSSO configurations using a custom property, but that custom property contains/is the e-mail address. The 'file_samlmapping' contains the e-mail address, which is needed when the e-mail is case-sensitive.

1233-U-UC-UClemrt-Oarrk-Fj-Es-SAML Update/create users (with roles & teams)

Sample	1233-U-UC-UClemrt-Oarrk-Fj-Es-SAML Update/create users (with roles & teams)		
Script Basis	U: User		
OAuth Access	'User Provisioning'		
Basic description	UC: Updates existing users and creates new users with roles assigned to the user directly (not generally best practice) and assigns the users to teams (roles and teams can be added/removed/replaced)		
Ideal for	<ol style="list-style-type: none"> 1. Assigning a user ID of your choice rather than one based on the e-mail (because you are using SAML SSO and mapping on a user's email or on a custom property) 2. Updating users' e-mail addresses (i.e. a user might already exist with a different e-mail than already registered, and this avoids duplicate '_1' users being created, as is the problem with scripts 11xx) 3. For existing users, you wish to update the 'BI Licence Type', the users' manager, the user's e-mail address*, their role assignment and their team assignment. (* Updates to the e-mail is not possible when SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on 'userid' or 'custom'.) 4. Relying on any default role to assign the user to a 'Business Intelligence' license type that isn't set correctly as part of the data file. 5. SAP Analytics Cloud is set up for Single-Sign-On using SAML and mapping the user on the e-mail to ensure the user is created with an ID determined by you rather than one derived from the e-mail address. 6. SAP Analytics Cloud is set up for Single-Sign-On using SAML and mapping the user on a custom property which is different from the userid. For example, you can create a user with userid M_SHAW, with e-mail matthew@sap.com and custom property M-SHAW 		
Not suitable when	<ol style="list-style-type: none"> 1. The very greatest throughout of creating users is desired (use scripts 11xx instead, if possible) 2. Running another script to assign users to teams is possible (i.e. given a list of users, add the users to a team), and users still need to be in a team. 		
Notes	<p>Does the same as sample 1202, only this script enables users to be assigned to teams. In the case when:</p> <ol style="list-style-type: none"> 1. team updates are the only thing that needs to be updated, 2. or team updates can be separated from user updates/creation <p>then use script 1201 or 1202 to update and create users and scripts 6xx to assign users to teams. This will provide the greatest throughput and will likely be far more reliable for large volumes. Use this script, 1203, for only lower volumes of users and updates (typically, say, less than 500 users and less than, say, 50 teams).</p> <p>If the file_isconcurrent setting is set to true (for concurrent), but should the user be assigned roles directly (rather than indirectly via a team) and if any of those directly assigned roles are planning roles, then the isConcurrent setting will not be respected, i.e., the isConcurrent will be set to false (a named user).</p> <p>If you are using your own Identity Provider with SAML SSO mapping on the 'e-mail' or 'custom' and you wish to update the e-mail or the custom property, then you can also use sample scripts 1409, 1419 and 1429</p>		
Data file syntax	F: .json		
	Field	Type	Description
	file_userid	string	UserID

	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user
	file_givename	string	Initial given name
	file_familyname	string	Initial family name
	file_displayname	string	Initial display name
	file_email	string	Email address
	file_managerid	string	Optional ID of the manager's ID
	file_samlmapping	string	Needed when SAMLSSO is set to custom, it is the custom SAML property
	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]
	file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used to assign the role for any new user.
	file_JSON_teams	JSON values of teams	Optional JSON array of keys "value" and string values of teams. Example: [{"value":"Team1"}, {"value":"Team2"}]
	file_teams_action	string	Defines the action to perform on the users' team membership. Possible values are: "add", "remove", "replace" and "keep". "add" or "replace" must be used for any new user to be a member of a team.
Environment	Single Service: SACserviceFQDN, SACplatform, SACTokenFQDN, ContentNamespace, Username, Password, SAMLSSO plus all 9 of the "SCIM_NewUserDefault_" variables as defined in Step C of the initial setup.		
How the script works	<p>Reads the data file and for each entry (row) the user is read from SAP Analytics Cloud. If the user is not found, the user is created, otherwise a comparison is made of the existing user's properties and compared with the provided properties from the file. If a non-team assignment update is necessary, then the user is appropriately updated, otherwise the user is skipped. Teams are appropriately updated every 500 users and at the end of the file.</p> <p>Creating Users</p> <p>A user will initially be created with the wrong email address! This is to ensure the userid is created with the one desired and not one derived from the email (if SAMLSSO is mapped on email) or from the userName (if SAMLSSO is mapping on a custom property). The script will then go on to read the user (to obtain any default roles assigned) and then update the users' email with the desired setting from file_email. And if SAMLSSO is set to custom, then the userName property is updated with the file_samlmapping value.</p> <p>Roles</p> <p>Roles are assigned directly to the user, if specified, as the user is created in the single POST call that creates the user. There is no separate call to assign the user to roles.</p> <p>The roles with the same content namespace as the environment don't need to specify the PROFILE:namespace: as part of its name. For example, if a Role called 'Role1' exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only 'Role1' needs to be specified. For any role that doesn't match the ContentNamespace then PROFILE:namespace: must be</p>		

specified with namespace set accordingly. For example, the provided 'BI_Admin' role would be PROFILE:sap.epm:BI_Admin

Roles are only added to new users if the roles_action is "add" or "replace", otherwise the roles are ignored and the user will not be assigned to any roles.

Any default roles will be added after the user is created. On a rare event, an update to the user could follow the user creation (this can happen on session recovery, or a false negative response from the API, a 502 error). Should this occur, and if the roles_action is 'replace', then any default roles not specified will be removed from the user. To avoid this rare event with the use of default roles, use 'add' for roles_action when creating users. The same event could also change the 'Business Intelligence license type' from the one defined in the default role and would, instead, set it to the setting as read from the data file.

The file_JSON_roles can be an empty array, i.e. []

If the role doesn't already exist a 400 status error will be returned and the 'UPDATE user' will not pass its test.

Teams

User to team assignment cannot currently be performed using the API endpoint that is used to create or update a user, instead a different endpoint must be used. This endpoint is for the Teams (Groups) themselves. It means more than one user can be specified per team, but if a single user requires membership changes to multiple teams it will require multiple GET/PUT pair calls, one GET/PUT pair per team.

User membership to teams is performed every 500 entries (users) and at the end of the file, i.e. they are 'batched' together to achieve the greatest throughput. The users, for each team, are batched together to form a single GET/PUT pair per team rather than a GET/PUT pair per user, per team.

If the team does not exist, it will be created **without** a team folder, and the display name will match its name; otherwise, the existing team's display name is respected.

(When SAP Analytics Cloud is hosted on an SAP Data Centre (NEO), then teams are created **with** a team folder, and the display name for the team will have the text "with Team Folder". E.g. "Team1 with Team Folder". If you'd like the team folder description to share the same team display name, use sample script 1501 before running this sample script. Alternatively, update the team folder description manually.)

Users added into teams are 'chunked' into multiple PUT calls. However, given the batch size is 500 users, it's unlikely any chunking will be applied, and it would only occur when the team has 29225 users!

The file_JSON_teams can be an empty array, i.e. []

Teams are only added to new users if the teams_action is "add" or "replace", otherwise the teams are ignored and the user will not be assigned to any teams.

Updating Users (lemrt)

A new user will always be created, even when updating an existing user. This new duplicate user will be identified and then deleted, before going on to update the 'right' users' properties.

The following properties can be updated: isconcurrent, email, managerid, roles and teams assignment.

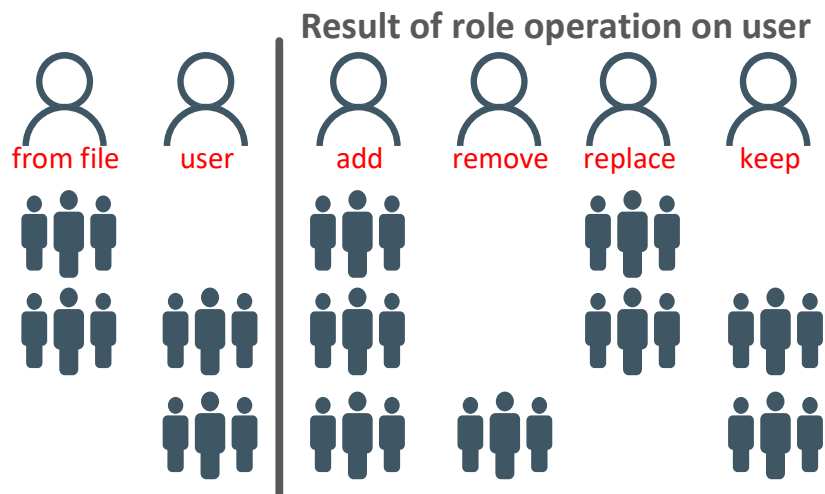
Updating Operations (arrk)

For updating existing users, the roles_action can be "add", "remove", "replace" and "keep":
add: will add the roles to the user.

remove: will remove the roles from the user, if the user has any of the roles listed.

replace: will replace (set) the roles for the user (i.e. remove and add accordingly).

keep: will respect any roles the user has already been assigned, no changes will be made.

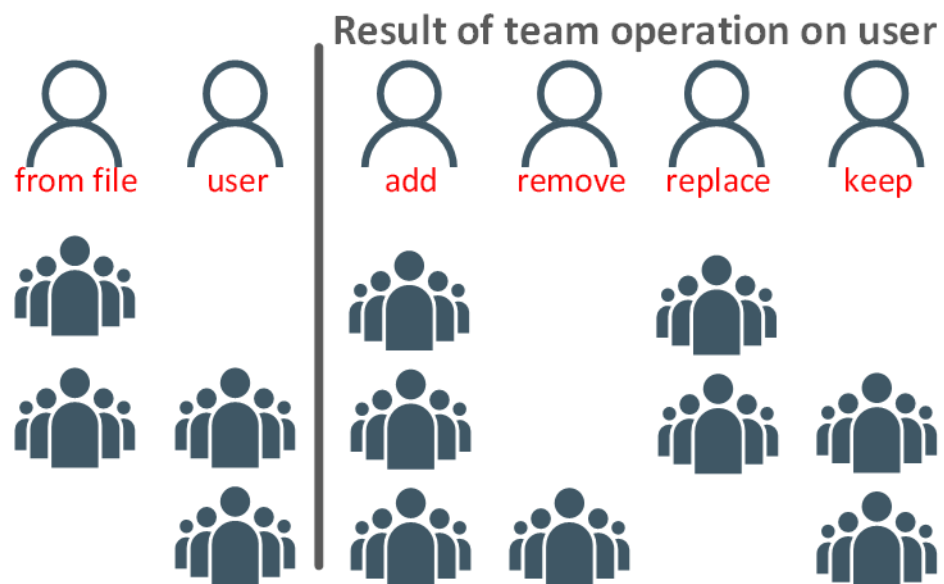


For updating existing users, the teams_action can be "add", "remove", "replace" and "keep":
add: will add the user to the team.

remove: will remove the user from the team, if the user is a member of any of the teams listed.


replace: will replace (set) the team membership of the user (i.e. remove and add accordingly).

keep: will respect any teams the user has already a member of, no changes will be made.




<p>Known script volume limitations</p>	<p>If the number of team updates (number of teams and the number of 'chunk' updates per team) is large, typically many hundred, then the script will use a large memory set, and this could cause instability issues in the client. To reduce script failures, running this script in the command line rather than the user interface is recommended.</p> <p>The batch size of 500 to update teams has been explicitly set to allow team updates to be processed mid-file. This is to avoid the situation where no team assignments are updated at all when the script fails mid-operation. If the number of users and team assignments is large (roughly over 2000 users and over 100 teams, each team of over 2000 members), then this script is likely problematic. In such cases, use script 1201 with scripts 16xx to assign users to teams.</p>
<p>Script throughput</p>	<p>Data not collected.</p>
<p>Sample Data Files</p>	<ul style="list-style-type: none"> • Example 1 is suitable for the default authentication method and for all SAMLSSO configurations other than custom. • Example 2 is suitable for SAMLSSO configurations using a custom property. The 'file_samlmapping' contains that custom property. • Example 3 is suitable for SAMLSSO configurations using a custom property, but that custom property contains/is the e-mail address. The 'file_samlmapping' contains the email address, and this is needed when the email is case-sensitive.

1301-U-D-Du-Fcj-Es-Delete Users

Sample	1301-U-D-Du-Fcj-Es-Delete Users		
Script Basis	U: User		
OAuth Access	'User Provisioning'		
Basic description	Du: Deletes Users		
Ideal for	<ul style="list-style-type: none"> Deleting specific users or when the number of users to be deleted is very large (over about 1000) When you know the userid. 		
Not suitable when	Deleting a team of users when the team size is less than about 1000 users, as script 1851 would be more suitable. Script 1851 requires just the team name to be provided rather than the list of all the usernames.		
Notes	<p>Deleting a user that is a manager requires special consideration – see below.</p> <p>When deleting a user, all of their personal content stored in their personal folder will be transferred to the system owner's personal folder, and the System Owner will become the owner of that content.</p> <p>There is a rare condition where the script might report a failure because the user doesn't exist, but the user did exist, and it was successfully deleted. This is a rare event that occurs when the session expires mid-request or when there's a 'wobble' on the SAP Analytics Cloud API.</p>		
Data file syntax	F: .csv and .json		
	Field	Type	Description
	file_delete_userid	string	UserID
Environment	Single Service (SACserviceFQDN, SACplatform, SACtokenFQDN, Username, Password)		
How the script works	<p>Deleting a user that is a manager of an existing user will fail the 'DELETE user' test. You can</p> <ol style="list-style-type: none"> 1. change the manager of the other users first 2. or delete the manager last, having deleted all the other users first. A variation of this is to run the script twice, assuming the file contains all the users of the manager and the manager itself <p>Alternatively, you can use "Scenario D01 - Delete users then delete managers" in the scenarios section to perform a 'clean' delete.</p>		
Known script volume limitations	None		
Script throughput		Service with 500 users	Service with 80,000 users
	Deleting users	0.54 users / sec 1.85 secs / user (4.2%)	0.37 users / sec 2.74 secs / user (2.9%)
Video tutorial	 <p>SAP HANA Academy provide a step-by-step tutorial for this sample script 1301 https://youtu.be/PaDAR7BQi_w</p> <p>(Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)</p>		


Related Scenarios	D01 - Delete users then delete managers	Given a team of users to be deleted, it will delete the users first before deleting the managers to avoid the problem of not being able to delete a user who is a manager of another user.
	M01 - Reassign users of given manager to another	Re-assign all users of the given manager(s) to another manager, which is helpful when the manager is unavailable or needs to be deleted.

1311-U-D-Du-Fcj-Es-Delete Users (by saml mapping)

Sample	1311-U-D-Du-Fcj-Es-Delete Users (by saml mapping)								
Script Basis	U: User								
OAuth Access	'User Provisioning'								
Basic description	Du: Deletes Users								
Ideal for	<ul style="list-style-type: none">Deleting specific users or when the number of users to be deleted is very large (over about 1000)When you do not necessarily know the userid, you're either using SAMLSSO based on e-mail or a custom property. Instead, you'd prefer to reference the users to be deleted by the SAML mapping property (userName).								
Not suitable when	Deleting a team of users when the team size is less than about 1000 users, as the script 1851 would be more suitable. Script 1851 requires just the team name to be provided rather than the list of all the usernames.								
Notes	<p>Deleting a user that is a manager requires special consideration – see below.</p> <p>When deleting a user, all their personal content stored in their personal folder will be transferred to the system owner's personal folder, and the System Owner will become the owner of that content.</p> <p>There is a rare condition where the script might report a failure because the user doesn't exist, but the user did exist, and it was successfully deleted. This is a rare event that occurs when the session expires mid-request or when there's a 'wobble' on the SAP Analytics Cloud API.</p>								
Data file syntax	F: .csv and .json <table><tr><td>Field</td><td>Type</td><td>Description</td></tr><tr><td>file_delete_samlmapping</td><td>string</td><td>SAML mapping userName property</td></tr></table>			Field	Type	Description	file_delete_samlmapping	string	SAML mapping userName property
Field	Type	Description							
file_delete_samlmapping	string	SAML mapping userName property							
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password)								
How the script works	<p>Deleting a user that is a manager of an existing user will fail the 'DELETE user' test. You can</p> <ol style="list-style-type: none">change the manager of the other users firstor delete the manager last, having deleted all the other users first. A variation of this is to run the script twice, assuming the file contains all the users of the manager and the manager itself <p>Alternatively, you can use "Scenario D01 - Delete users then delete managers" in the scenarios section to perform a 'clean' delete.</p>								
Known script volume limitations	None								
Script throughput	Not yet calculated								
Video tutorial	 SAP HANA Academy provide a step-by-step tutorial for this sample script 1301 https://youtu.be/PaDAR7BQi_w (Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)								
Related Scenarios	D01 - Delete users then delete managers	Given a team of users to be deleted, it will delete the users first before deleting the managers to avoid the problem of not being able to delete a user who is a manager of another user.							
	M01 - Reassign users of given	Re-assign all users of the given manager(s) to another manager, which is helpful when the manager is unavailable or needs to be deleted.							

	manager to another	
--	--------------------	--


1321-U-D-Du-Fcj-Es-Delete Users (by email)

Sample	1321-U-D-Du-Fcj-Es-Delete Users (by email)							
Script Basis	U: User							
OAuth Access	'User Provisioning'							
Basic description	Du: Deletes Users							
Ideal for	<ul style="list-style-type: none"> Deleting specific users or when the number of users to be deleted is very large (over about 1000) When you do not necessarily know the userid, you're either using SAMLSSO based on e-mail or a custom property. Instead, you'd prefer to reference the users to be deleted by their e-mail address. 							
Not suitable when	Deleting a team of users when the team size is less than about 1000 users, as script 1851 would be more suitable. Script 1851 requires just the team name to be provided rather than the list of all the usernames.							
Notes	<p>Deleting a user that is a manager requires special consideration – see below.</p> <p>When deleting a user, all their personal content stored in their personal folder will be transferred to the system owner's personal folder, and the System Owner will become the owner of that content.</p> <p>There is a rare condition where the script might report a failure because the user doesn't exist, but the user did exist, and it was successfully deleted. This is a very rare event and occurs when the session expires mid-request or when there's a 'wobble' on the SAP Analytics Cloud API.</p>							
Data file syntax	F: .csv and .json <table border="1"> <thead> <tr> <th>Field</th><th>Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>file_delete_email</td><td>string</td><td>Users email address</td></tr> </tbody> </table>		Field	Type	Description	file_delete_email	string	Users email address
Field	Type	Description						
file_delete_email	string	Users email address						
Environment	Single Service (SACserviceFQDN, SACplatform, SACtokenFQDN, Username, Password)							
How the script works	<p>Deleting a user that is a manager of an existing user will fail the 'DELETE user' test. You can</p> <ol style="list-style-type: none"> change the manager of the other users first or delete the manager last, having deleted all the other users first. A variation of this is to run the script twice, assuming the file contains all the users of the manager and the manager itself <p>Alternatively, you can use "Scenario D01 - Delete users then delete managers" in the scenarios section to perform a 'clean' delete.</p>							
Known script volume limitations	None							
Script throughput	Not yet calculated							
Video tutorial	 <p>SAP HANA Academy provide a step-by-step tutorial for this sample script 1301 https://youtu.be/PaDAR7BQi_w</p> <p>(Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)</p>							
Related Scenarios	D01 - Delete users then delete managers	Given a team of users to be deleted, it will delete the users first before deleting the managers to avoid the problem of not being able to delete a user who is a manager of another user.						

	M01 - Reassign users of given manager to another	Re-assign all users of the given manager(s) to another manager, which is helpful when the manager is unavailable or needs to be deleted.
--	--	--

1401-U-U-UI-Fcj-Es-Update User License

Sample	1401-U-U-UI-Fcj-Es-Update User License		
Script Basis	U: User		
OAuth Access	'User Provisioning'		
Basic description	UI: Updates the users' BI license type property		
Ideal for	Updating specific users or when the number of users to be updated is very large (over about 1000)		
Not suitable when	Updating a team of users and when the team size is less than about 1000 users, as the script 1451 would be more suitable. Script 1451 requires just the team name to be provided rather than the list of all the usernames.		
Notes	<p>The BI license type is only applicable when a Business Intelligence concurrent user license has been purchased, and the user is consuming only that license type, other than, say, an Analytics Hub or Planning license. This topic is discussed in great detail in an article introduced by this blog https://blogs.sap.com/2020/03/10/sap-analytics-cloud-managing-licenses-with-roles-and-teams/. Please refer to this article for more information.</p> <p>The 'concurrent-session' setting is only a 'request' for a concurrent-session license, and it does not mean the user <u>will</u> consume a concurrent-session license. The user <u>could</u> consume a named-user license because they were assigned a Planning Role (or an Analytics Hub Role). For more details on this, please visit the blog just mentioned.</p>		
Data file syntax	F: .csv and .json		
	Field	Type	Description
	file_userid	string	UserID
	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user
Environment	Single Service (SACserviceFQDN, SACplatform, SAcTokenFQDN, Username, Password)		
How the script works	<p>Reads the data file, and for each entry (row), the user is read from SAP Analytics Cloud. If the user is not found, the test 'READ user' fails; otherwise, a comparison is made of the existing user's properties and compared with the provided property from the file. If an update is necessary, the user is appropriately updated; otherwise, the user is skipped. If the user doesn't need to be updated, the test 'UPDATE user' will neither pass nor fail.</p> <p>Be careful with the file_isconcurrent value as it must be a lowercase true or false and not TRUE OR FALSE.</p> <p>If the file_isconcurrent setting is set to true (for concurrent), but should the user be assigned roles directly (rather than indirectly via a team) and if any of those directly assigned roles are planning roles, then the isConcurrent setting will not be respected, i.e., the isConcurrent will be set to false (a named user).</p>		
Known script volume limitations	None		
Script throughput		Service with 500 users	Service with 80,000 users

	Updating user	0.39 users / sec 2.57 secs / user (7.0%)	0.14 users / sec 6.90 secs / user (2.56%)	
Video tutorial		SAP HANA Academy provide a step-by-step tutorial for sample scripts 140x and 145x https://youtu.be/emDiUFCi_vk (Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)		
Related Scenarios	L01 - Managers with Blconcurrent to Blnamed license	Identifies which users have a 'Business Intelligence' concurrent session license AND which are also managers. It then updates their license type to 'named user', ensuring all managers are always able to log in and don't need to wait for a spare concurrent session to be available.		
	S02 - Assign settings concurrent lic for recently created w default settings	Assign the correct date/time/number and language settings for those users created in the last week with the default settings and a 'BI concurrent session' license. It means your users will then be set correctly, avoiding each user updating their settings, which is an unnecessary task.		

1403-U-U-Ut-Fj-Es-Update user team membership

Sample	1403-U-U-Ut-Fj-Es-Update user team membership		
Script Basis	U: User		
OAuth Access	'User Provisioning'		
Basic description	Ua: Updates the users' team membership.		
Ideal for	<ol style="list-style-type: none"> 1. Updating a small number of users and the number of teams to be updated is small (say about less than 1000 users and less than, say, 10 teams). 2. Running another script to assign users to teams isn't possible (i.e. given a list of users, add the users to a team), i.e. a pure 'user event' script is mandated. 3. Removing a user from all teams when you don't know the teams they're a member of 		
Not suitable when	A large number of users needs to be updated, and then a greater throughput and stability can be achieved using scripts 16xx for the same task; only these scripts are team-based.		
Notes	Use this script with caution; wherever and whenever possible, use scripts 16xx.		
Data file syntax	F: .json		
	Field	Type	Description
	file_userid	string	UserID
	file_JSON_teams	JSON values of teams	JSON array of keys "value" and string values of teams. Example: [{"value": "Team1"}, {"value": "Team2"}]
	file_teams_action	string	Defines the action to perform on the assignment of the team to roles. Possible values are: "add", "remove", "replace" and "keep".
Environment	Single Service (SACserviceFQDN, SACplatform, SAcTokenFQDN, Username, Password)		
How the script works	<p>Reads the data file and for each entry (row), the user is read from SAP Analytics Cloud. If the user is not found, a Postman test 'READ user' will fail, and the following user is read. If the user is found, a comparison is made of the existing user's team membership (possible by reading the user properties) and compared with the provided properties from the file. If any team membership updates are required, they are batched together; teams are only appropriately updated every 500 users and at the end of the file.</p> <p>Teams</p> <p>User-to-team assignment cannot currently be performed using the version 1 API endpoint that is used to create or update a user; instead, a different endpoint must be used. This endpoint is for the Teams (Groups) themselves. It means more than one user can be specified per team, but if a single user requires membership changes to multiple teams, it will need multiple GET/PUT pair requests, one GET/PUT pair per team.</p> <p>User membership to teams is performed every 500 entries (users), and at the end of the file, i.e. they are 'batched' together to achieve the most significant throughput. The users for each team are batched together to form a single GET/PUT pair per team rather than a GET/PUT pair per user per team.</p> <p>If the team does not exist, it will be created without a team folder, and the display name will match its name; otherwise, the existing teams' display name is respected along with any roles the team is already a member of.</p>		

(When SAP Analytics Cloud is hosted on an SAP Data Centre (NEO), then teams are created **with** a team folder, and the display name for the team will have the text “with Team Folder”. E.g. “Team1 with Team Folder”. If you’d like the team folder description to share the same team display name, use sample script 1501 before running this sample script. Alternatively, update the team folder description manually.)

Users added into teams are ‘chunked’ into multiple PUT calls. However, given the batch size is 500 users, it’s unlikely any chunking will be applied, and it would only occur when the team has 29225 users!

The file_JSON_teams can be an empty array, i.e. [], and this is useful if you’d like to remove the user from all the teams they are a member of. This is particularly helpful when you don’t know the teams the user is already a member of.

Updating Operations (arrk)

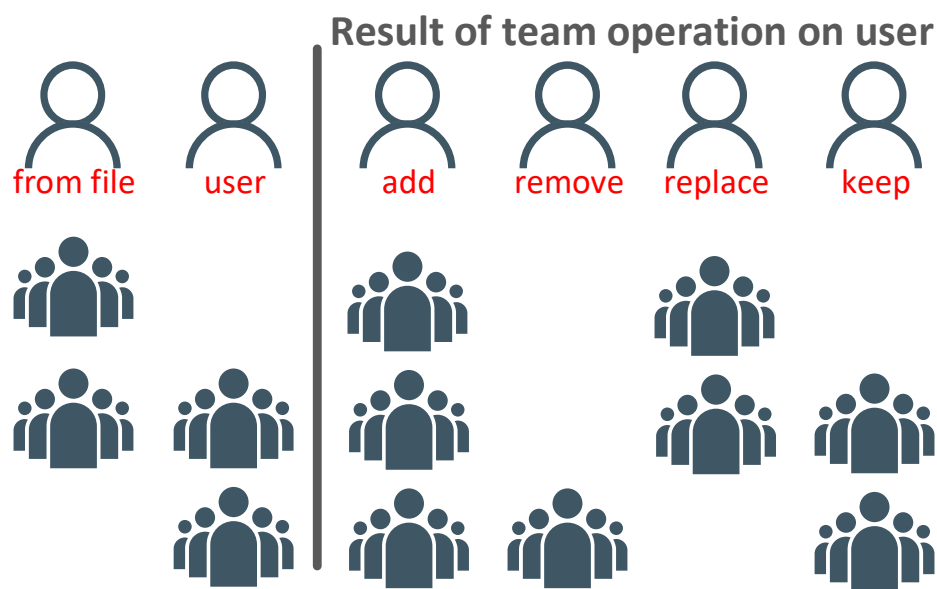
The teams_action can be “add”, “remove”, “replace” and “keep”:

add: will add the user to the team.

remove: will remove the user from the team, if the user is a member of any of the teams listed.

replace: will replace (set) the team membership of the user (i.e. remove and add accordingly).

keep: will respect any teams the user has already a member of, no changes will be made. This script with an action of ‘keep’ is quite wasteful, it will perform a read request of the user and no action could every occur.




Known script volume limitations


None

Script throughput

	Service with 500 users	Service with 80,000 users
Adding user into 1, 2 or 3 teams	0.79 users / sec 1.26 secs / user (9.1% to 13.5% - 9.1%)	0.19 users / sec 5.22 secs / user (2.3% to 3.7% - 2.3% overhead)


		overhead with 1 team, 13.5% with 3 teams)	with 1 team, 3.7% with 3 teams)	
Video tutorial		SAP HANA Academy provide a step-by-step tutorial for sample scripts 140x and 145x https://youtu.be/emDiUFCi_vk (Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)		

1404-U-U-Um-Fcj-Es-Update User Manager

Sample	1404-U-U-Um-Fcj-Es-Update User Manager			
Script Basis	U: User			
OAuth Access	'User Provisioning'			
Basic description	Um: Updates the users' manager.			
Ideal for	Updating specific users or when the number of users to be updated is very large (over about 1000)			
Not suitable when	Updating a team of users when the team size is less than about 1000 users, as the script 1454 would be more suitable. Script 1454 requires just the team name to be provided rather than the list of all the usernames.			
Notes	Cyclic loops of users to managers aren't allowed. It means user A can't be a manager of user B when user B is the manager of user A. Users are not allowed to be a manager of themselves.			
Data file syntax	F: .csv and .json			
	Field	Type	Description	
	file_userid	string	UserID	
	file_managerid	string	Optional ID of the manager's ID	
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password)			
How the script works	Reads the data file, and for each entry (row), the user is read from SAP Analytics Cloud. If the user is not found, the test 'READ user' fails; otherwise, a comparison is made of the existing user's properties and compared with the provided property from the file. If an update is necessary, the user is appropriately updated; otherwise, the user is skipped. If the user doesn't need to be updated, the test 'UPDATE user' will neither pass nor fail.			
Known script volume limitations	None			
Script throughput		Service with 500 users	Service with 80,000 users	
	Updating user	0.39 users / sec 2.57 secs / user (7.0%)	0.14 users / sec 6.90 secs / user (2.56%)	
Video tutorial	 SAP HANA Academy provide a step-by-step tutorial for sample scripts 140x and 145x https://youtu.be/emDiUFCi_vk (Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)			
Related Scenarios	M01 - Reassign users of given manager to another	Re-assign all users of the given manager(s) to another manager, which is helpful when the manager is unavailable or needs to be deleted.		


1406-U-U-Ulpddtn-Fj-Es-Update User DateTimeNumFormat DataAccessLang

Sample	1406-U-U-Ulpddtn-Fj-Es-Update User DateTimeNumFormat DataAccessLang																							
Script Basis	U: User																							
OAuth Access	'User Provisioning'																							
Basic description	Ulpddtn: Updates the users' properties: <div><div>1. Preferredlanguage</div><div>2. Dataaccesslanguage</div><div>3. Dateformatting</div><div>4. Timeformatting</div><div>5. Numberformatting</div></div>																							
Ideal for	Updating specific users or when the number of users to be updated is very large (over about 1000)																							
Not suitable when	Updating a team of users and when the team size is less than about 1000 users, as the script 1453 would be more suitable. Script 1453 requires just the team name to be provided rather than the list of all the usernames.																							
Notes	This script is typically only necessary when users have been created incorrectly. To avoid the need to perform any user updates on newly created users, edit the 'CREATE user' requests in the scripts that create users (scripts 11xx and 12xx) with the correct values for the preferredlanguage, dataaccesslanguage, dateformatting, timeformatting and numberformatting. These values are 'hard coded' in the script. Set these values before running the scripts to creating users. This way, the users will have the correct settings for these properties, and there'll be no need to update them afterwards.																							
Data file syntax	<div>F: .json</div> <table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_preferredlanguage</td><td>string</td><td>possible values: 'ar', 'bg', 'zh-CN', 'zh-TW', 'hr', 'ca', 'cs', 'cy', 'nl', 'da', 'en', 'en-GB', 'fi', 'fr', 'fr-CA', 'de', 'de-CH', 'el', 'he', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'ms', 'no', 'pl', 'pt', 'pt-PT', 'ro', 'ru', 'sh', 'sk', 'sl', 'es', 'es-MX', 'sv', 'th', 'tr', 'uk', 'vi', 'en-US-sapbsd' and possibly more will be added over time!</td></tr><tr><td>file_dataaccesslanguage</td><td>string</td><td>possible values: 'zz' (default), 'af', 'ar', 'bg', ... many more as shown above in preferredlanguage.</td></tr><tr><td>file_dateformatting</td><td>string</td><td>possible values include: 'MMM d, yyyy', 'dd/MM/yyyy', 'MM.dd.yyyy', ... many more!</td></tr><tr><td>file_timeformatting</td><td>string</td><td>possible values include: 'H:mm:ss' (24H), 'h:mm:ss A' (12H)</td></tr><tr><td>file_numberformatting</td><td>string</td><td>possible values include: '1,234.56', '1.234,56', '1 234,56'</td></tr></table>			Field	Type	Description	file_userid	string	UserID	file_preferredlanguage	string	possible values: 'ar', 'bg', 'zh-CN', 'zh-TW', 'hr', 'ca', 'cs', 'cy', 'nl', 'da', 'en', 'en-GB', 'fi', 'fr', 'fr-CA', 'de', 'de-CH', 'el', 'he', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'ms', 'no', 'pl', 'pt', 'pt-PT', 'ro', 'ru', 'sh', 'sk', 'sl', 'es', 'es-MX', 'sv', 'th', 'tr', 'uk', 'vi', 'en-US-sapbsd' and possibly more will be added over time!	file_dataaccesslanguage	string	possible values: 'zz' (default), 'af', 'ar', 'bg', ... many more as shown above in preferredlanguage.	file_dateformatting	string	possible values include: 'MMM d, yyyy', 'dd/MM/yyyy', 'MM.dd.yyyy', ... many more!	file_timeformatting	string	possible values include: 'H:mm:ss' (24H), 'h:mm:ss A' (12H)	file_numberformatting	string	possible values include: '1,234.56', '1.234,56', '1 234,56'
Field	Type	Description																						
file_userid	string	UserID																						
file_preferredlanguage	string	possible values: 'ar', 'bg', 'zh-CN', 'zh-TW', 'hr', 'ca', 'cs', 'cy', 'nl', 'da', 'en', 'en-GB', 'fi', 'fr', 'fr-CA', 'de', 'de-CH', 'el', 'he', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'ms', 'no', 'pl', 'pt', 'pt-PT', 'ro', 'ru', 'sh', 'sk', 'sl', 'es', 'es-MX', 'sv', 'th', 'tr', 'uk', 'vi', 'en-US-sapbsd' and possibly more will be added over time!																						
file_dataaccesslanguage	string	possible values: 'zz' (default), 'af', 'ar', 'bg', ... many more as shown above in preferredlanguage.																						
file_dateformatting	string	possible values include: 'MMM d, yyyy', 'dd/MM/yyyy', 'MM.dd.yyyy', ... many more!																						
file_timeformatting	string	possible values include: 'H:mm:ss' (24H), 'h:mm:ss A' (12H)																						
file_numberformatting	string	possible values include: '1,234.56', '1.234,56', '1 234,56'																						
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password)																							
How the script works	Reads the data file, and for each entry (row), the user is read from SAP Analytics Cloud. If the user is not found, the test 'READ user' fails; otherwise, a comparison is made of the existing user's properties and compared with the provided property from the file. If an update is necessary, the user is appropriately updated; otherwise, the user is skipped. If the user doesn't need to be updated, the test 'UPDATE user' will neither pass nor fail.																							
Known script volume limitations	None																							

Script throughput		Service with 500 users	Service with 80,000 users	
	Updating user	0.39 users / sec 2.57 secs / user (7.0%)	0.14 users / sec 6.90 secs / user (2.56%)	
Video tutorial		SAP HANA Academy provide a step-by-step tutorial for sample scripts 140x and 145x https://youtu.be/emDiUFCi_vk (Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)		
Related Scenarios	S01 - Assign settings for recently created with default settings	Assign the correct date/time/number and language settings for those users created last week with the default settings. It means your users will then be set correctly, avoiding each user from updating their settings, which is an unnecessary task.		
	S02 - Assign settings concurrent lic for recently created w default settings	It is the same as S01, only it assigns the user a 'Business Intelligence' 'concurrent session' license.		

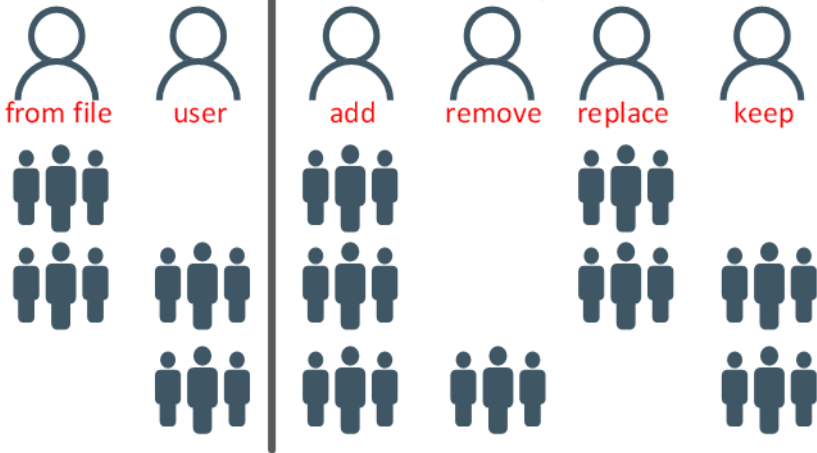

1407-U-U-Ulppdtn-Fj-Es-Update User License DateTimeNumFormat DataAccessLang

Sample	1407-U-U-Ulppdtn-Fj-Es-Update User License DateTimeNumFormat DataAccessLang																							
Script Basis	U: User																							
OAuth Access	'User Provisioning'																							
Basic description	Ulppdtn: Updates the users' properties: <ul style="list-style-type: none">1. BI license type2. Preferredlanguage3. Dataaccesslanguage4. Dateformatting5. Timeformatting6. Numberformatting																							
Ideal for	Updating specific users or when the number of users to be updated is very large (over about 1000)																							
Not suitable when	Updating a team of users and when the team size is less than about 1000 users, as the script 1457 would be more suitable. Script 1457 requires just the team name to be provided rather than the list of all the usernames.																							
Notes	<p>The BI license type is only applicable when a Business Intelligence concurrent user license has been purchased, and the user is consuming only that license type, other than, say, an Analytics Hub or Planning license. This topic is discussed in great detail in an article introduced by this blog https://blogs.sap.com/2020/03/10/sap-analytics-cloud-managing-licenses-with-roles-and-teams/. Please refer to this article for more information.</p> <p>Be careful with the <code>file_isconcurrent</code> value as it must be a lowercase <code>true</code> or <code>false</code> and not <code>TRUE</code> OR <code>FALSE</code>.</p> <p>This script is typically only necessary when users have been created incorrectly. To avoid the need to perform any user updates on newly created users, edit the 'CREATE user' requests in the scripts that create users (scripts 11xx and 12xx) with the correct values for the preferredlanguage, dataaccesslanguage, dateformatting, timeformatting and numberformatting. These values are 'hard coded' in the script. Set these values before running the scripts to create users. This way, the users will have the correct settings for these properties, and there'll be no need to update them afterwards.</p>																							
Data file syntax	F: .json <table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_userid</td><td>string</td><td>UserID</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td>file_preferredlanguage</td><td>string</td><td>possible values: 'ar', 'bg', 'zh-CN', 'zh-TW', 'hr', 'ca', 'cs', 'cy', 'nl', 'da', 'en', 'en-GB', 'fi', 'fr', 'fr-CA', 'de', 'de-CH', 'el', 'he', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'ms', 'no', 'pl', 'pt', 'pt-PT', 'ro', 'ru', 'sh', 'sk', 'sl', 'es', 'es-MX', 'sv', 'th', 'tr', 'uk', 'vi', 'en-US-sappsd' and possibly more will be added over time!</td></tr><tr><td>file_dataaccesslanguage</td><td>string</td><td>possible values: 'zz' (default), 'af', 'ar', 'bg', ... many more as shown above in preferredlanguage.</td></tr><tr><td>file_dateformatting</td><td>string</td><td>possible values include: 'MMM d, yyyy', 'dd/MM/yyyy', 'MM.dd.yyyy', ... many more!</td></tr><tr><td>file_timeformatting</td><td>string</td><td>possible values include: 'H:mm:ss' (24H), 'h:mm:ss A' (12H)</td></tr></table>			Field	Type	Description	file_userid	string	UserID	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user	file_preferredlanguage	string	possible values: 'ar', 'bg', 'zh-CN', 'zh-TW', 'hr', 'ca', 'cs', 'cy', 'nl', 'da', 'en', 'en-GB', 'fi', 'fr', 'fr-CA', 'de', 'de-CH', 'el', 'he', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'ms', 'no', 'pl', 'pt', 'pt-PT', 'ro', 'ru', 'sh', 'sk', 'sl', 'es', 'es-MX', 'sv', 'th', 'tr', 'uk', 'vi', 'en-US-sappsd' and possibly more will be added over time!	file_dataaccesslanguage	string	possible values: 'zz' (default), 'af', 'ar', 'bg', ... many more as shown above in preferredlanguage.	file_dateformatting	string	possible values include: 'MMM d, yyyy', 'dd/MM/yyyy', 'MM.dd.yyyy', ... many more!	file_timeformatting	string	possible values include: 'H:mm:ss' (24H), 'h:mm:ss A' (12H)
Field	Type	Description																						
file_userid	string	UserID																						
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user																						
file_preferredlanguage	string	possible values: 'ar', 'bg', 'zh-CN', 'zh-TW', 'hr', 'ca', 'cs', 'cy', 'nl', 'da', 'en', 'en-GB', 'fi', 'fr', 'fr-CA', 'de', 'de-CH', 'el', 'he', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'ms', 'no', 'pl', 'pt', 'pt-PT', 'ro', 'ru', 'sh', 'sk', 'sl', 'es', 'es-MX', 'sv', 'th', 'tr', 'uk', 'vi', 'en-US-sappsd' and possibly more will be added over time!																						
file_dataaccesslanguage	string	possible values: 'zz' (default), 'af', 'ar', 'bg', ... many more as shown above in preferredlanguage.																						
file_dateformatting	string	possible values include: 'MMM d, yyyy', 'dd/MM/yyyy', 'MM.dd.yyyy', ... many more!																						
file_timeformatting	string	possible values include: 'H:mm:ss' (24H), 'h:mm:ss A' (12H)																						

	file_numberformatting	string	possible values include: '1,234.56', '1.234,56', '1 234,56'	
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password)			
How the script works	Reads the data file, and for each entry (row) the user is read from SAP Analytics Cloud. If the user is not found, the test 'READ user' fails; otherwise, a comparison is made of the existing user's properties and compared with the provided property from the file. If an update is necessary, the user is appropriately updated; otherwise, the user is skipped. If the user doesn't need to be updated, the test 'UPDATE user' will neither pass nor fail.			
Known script volume limitations	None			
Script throughput		Service with 500 users	Service with 80,000 users	
	Updating user	0.39 users / sec 2.57 secs / user (7.0%)	0.14 users / sec 6.90 secs / user (2.56%)	
Video tutorial		SAP HANA Academy provide a step-by-step tutorial for sample scripts 140x and 145x https://youtu.be/emDiUFCi_vk (Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)		
Related Scenarios	S01 - Assign settings for recently created with default settings	Assign the right date/time/number and language settings for those users created last week with the default settings. It means your users will then be set correctly, avoiding each user from updating their settings, which is an unnecessary task.		
	S02 - Assign settings concurrent lic for recently created w default settings	Same as S01, it only assigns the user a 'Business Intelligence' 'concurrent session' license.		

1408-U-U-Ur-Oarrk-Fj-Es-Update User Role


Sample	1408-U-U-Ur-Oarrk-Fj-Es-Update User Role														
Script Basis	U: User														
OAuth Access	'User Provisioning'														
Basic description	Ur: Updates the user's role assignment when the role is directly assigned to the user (not generally best practice)														
Ideal for	<ul style="list-style-type: none">Updating the directly assigned roles for a user when you already know the user ID assigned by SAP Analytics CloudUpdating specific users or when the number of users to be updated is very large (over about 1000)														
Not suitable when	Updating a team of users when the team size is less than about 1000 users, as script 1458, would be more suitable. Script 1458 requires just the team name to be provided rather than the list of all the usernames.														
Notes	<p>The BI license type will change to 'named user' from 'concurrent session' if a Planning Role is assigned directly to the user (unlike when a user inherits a Planning role through inheritance of team membership).</p> <p>Use sample 1418 or 1428 to achieve the same goal, but instead, reference the user to be updated by either their SAML mapping or their e-mail address.</p>														
Data file syntax	F: .json <table><tr><td>Field</td><td>Type</td><td>Description</td></tr><tr><td>file_userid</td><td>string</td><td>Username</td></tr><tr><td>file_roles_action</td><td>string</td><td>Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep".</td></tr><tr><td>file_JSON_roles</td><td>JSON values of roles</td><td>Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]</td></tr></table>			Field	Type	Description	file_userid	string	Username	file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep".	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]
Field	Type	Description													
file_userid	string	Username													
file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep".													
file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]													
Environment	Single Service (SACserviceFQDN, SACplatform, SAcTokenFQDN, Username, Password)														
How the script works	<p>Reads the data file, and for each entry (row), the user is read from SAP Analytics Cloud. If the user is not found, the test 'READ user' fails; otherwise, a comparison is made of the existing user's roles and compared with the provided property from the file. If an update is necessary, the user is appropriately updated; otherwise, the user is skipped. If the user doesn't need to be updated, the test 'UPDATE user' will neither pass nor fail.</p> <p>The roles with the same content namespace as the environment don't need to specify the PROFILE:namespace: as part of its name. For example, if a Role called 'Role1' exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only 'Role1' needs to be specified. For any role that doesn't match the ContentNamespace, then PROFILE:namespace: must be specified with namespace set accordingly. For example, the provided 'BI_Admin' role would be PROFILE:sap.epm:BI_Admin</p> <p>The file_JSON_roles can be an empty array, i.e. [], and this is useful if you'd like to remove all roles directly assigned to the user.</p> <p>If the role doesn't already exist, a 400 status error will be returned, and the 'UPDATE user' will not pass its test.</p>														

	<p>Updating Operations (arrk)</p> <p>For updating existing users, the roles_action can be “add”, “remove”, “replace” and “keep”:</p> <p>add: will add the roles to the user.</p> <p>remove: will remove the roles from the user if the user has any of the roles listed.</p> <p>replace: will replace (set) the roles for the user (i.e. remove and add accordingly).</p> <p>keep: will respect any roles the user has already been assigned; no changes will be made.</p>		
	<p>Result of role operation on user</p> 		
Known script volume limitations	None		
Script throughput		Service with 500 users	Service with 80,000 users
	Updating user	0.39 users / sec 2.57 secs / user (7.0%)	0.14 users / sec 6.90 secs / user (2.56%)
Video tutorial	 SAP HANA Academy provide a step-by-step tutorial for sample scripts 140x and 145x https://youtu.be/emDiUFCi_vk (Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)		
Related Scenarios	R01- Swap directly assigned role for a team role	Re-assign all users with the 'BI_Admin' role directly assigned to them, add these users to a team, and add the team to be a member of the 'BI_Admin' role before removing the directly assigned role from all the users. This enables you to adopt the best practice for assigning roles to users with the help of inheritance and teams.	

1409-U-U-Um-Fcj-Es-Update SAML Mapping

Sample	1409-U-U-Um-Fcj-Es-Update SAML Mapping		
Script Basis	U: User		
OAuth Access	'User Provisioning'		
Basic description	Um: Updates the user SAML mapping property (userName)		
Ideal for	<ul style="list-style-type: none">Updating the SAML mapping property when you already know the user ID assigned by SAP Analytics CloudWhen you are using your own custom Identity Provider, and you have set up SSO to map on a custom attribute, which is the SAML mapping propertyWhen using your own custom Identity Provider, and you have set up SSO to map on e-mail, the SAML mapping property. i.e., you want to update the user's e-mail addressAssisting in the process of switching the SAP Analytics Cloud Identity Provider SSO from mapping on userid to mapping on a custom property (such as your own organisation ID)		
Not suitable when	<ul style="list-style-type: none">You don't know, or can't easily determine, the user ID assigned by SAP Analytics Cloud. You would rather identify the user by their e-mail address. If this is the case, use sample script 1429When you are using the default IDPWhen you are using your own custom Identity Provider and you have set up SSO to map on the User ID.		
Notes	<p>The user property 'userName' holds the SAML Mapping value.</p> <p>This property is NOT changeable when using the default IDP with SAP Analytics Cloud.</p> <p>It is also not changeable when using your custom Identity Provider, and you have set SSO to map on the User ID.</p> <p>Use sample 1419 or 1429 to achieve the same goal, but instead, reference the user to be updated by either their SAML mapping or their e-mail address.</p>		
Data file syntax	F: .csv and .json		
	Field	Type	Description
	file_userid	string	UserID
	file_samlmapping	string	The new value for the SAML Mapping
Environment	Single Service (SACserviceFQDN, SACplatform, SAcTokenFQDN, Username, Password, SAMLSSO)		
How the script works	Reads the data file, and for each entry (row), the user is read from SAP Analytics Cloud. If the user is not found, the test 'READ user' fails; otherwise, a comparison is made of the existing user's userName (SAML mapping) and compared with the provided property from the file. If an update is necessary, the user is appropriately updated; otherwise, the user is skipped. If the user doesn't need to be updated, the test 'UPDATE user' will neither pass nor fail.		
Known script volume limitations	None		
Script throughput		Service with 500 users	Service with 80,000 users
	Updating user	0.39 users / sec	0.14 users / sec
		2.57 secs / user (7.0%)	6.90 secs / user (2.56%)


1418-U-U-Ur-Oarrk-Fj-Es-Update User Role (by saml mapping)

Sample	1418-U-U-Ur-Oarrk-Fj-Es-Update User Role (by saml mapping)		
Script Basis	U: User		
OAuth Access	'User Provisioning'		
Basic description	Ur: Updates the user's role assignment when the role is directly assigned to the user (not generally best practice)		
Ideal for	<ul style="list-style-type: none"> Updating specific users or when the number of users to be updated is very large (over about 1000) When you do not necessarily know the userid, you use SAMLSSO based on e-mail or a custom property. Instead, you'd prefer to reference the users to be updated by the SAML mapping property (userName). 		
Not suitable when	Updating a team of users when the team size is less than about 1000 users, as the script 1458 would be more suitable. Script 1458 requires just the team's name to be provided rather than the list of all the usernames.		
Notes	The BI license type will change to 'named user' from 'concurrent session' if a Planning Role is assigned directly to the user (unlike when a user inherits a Planning role through inheritance of team membership).		
Data file syntax	F: .json		
	Field	Type	Description
	file_samlmapping	string	SAML mapping userName property
	file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep".
	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value": "Role1"}, {"value": "PROFILE:sap.epm:BI_Admin"}]
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password)		
How the script works	Same as sample script 1408		
Known script volume limitations	None		
Script throughput	Not yet calculated		
Video tutorial	 SAP HANA Academy provide a step-by-step tutorial for sample scripts 140x and 145x https://youtu.be/emDiUFCi_vk (Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)		
Related Scenarios	R01- Swap directly assigned role for a team role		
	Re-assign all users with the 'BI_Admin' role directly assigned to them, add these users to a team, and add the team to be a member of the 'BI_Admin' role before removing the directly assigned role from all the users. This enables you to adopt the best practice for assigning roles to users with the help of inheritance and teams.		

1419-U-U-Um-Fcj-Es-Update SAML Mapping (by saml mapping)

Sample	1419-U-U-Um-Fcj-Es-Update SAML Mapping (by saml mapping)		
Script Basis	U: User		
OAuth Access	'User Provisioning'		
Basic description	Um: Updates the user SAML mapping property (userName)		
Ideal for	<ul style="list-style-type: none">When you do not necessarily know the userid, you use SAMLSSO based on e-mail or a custom property. Instead, you'd prefer to reference the users to be updated by the SAML mapping property (userName).When you are using your own custom Identity Provider, and you have set up SSO to map on a custom attribute, which is the SAML mapping propertyWhen using your own custom Identity Provider, and you have set up SSO to map on e-mail, the SAML mapping property. i.e., you want to update the user's e-mail addressAssisting in the process of switching the SAP Analytics Cloud Identity Provider SSO from mapping on userid to mapping on a custom property (such as your own organisation ID)		
Not suitable when	<ul style="list-style-type: none">When you are using the default IDPWhen you are using your own custom Identity Provider and you have set up SSO to map on the User ID.		
Notes	<p>The user property 'userName' holds the SAML Mapping value.</p> <p>This property is NOT changeable when using the default IDP that comes with SAP Analytics Cloud.</p> <p>It is also not changeable when using your custom Identity Provider, and you have set up SSO to map on the User ID.</p>		
Data file syntax	F: .csv and .json		
	Field	Type	Description
	file_current_samlmapping	string	The currently assigned SAML Mapping for the user
	file_samlmapping	string	The new value for the SAML Mapping
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password, SAMLSSO)		
How the script works	Reads the data file, and for each entry (row), the user is read from SAP Analytics Cloud. If the user is not found, the test 'READ user' fails; otherwise, a comparison is made of the existing user's userName (SAML mapping) and compared with the provided property from the file. If an update is necessary, the user is appropriately updated; otherwise, the user is skipped. If the user doesn't need to be updated, the test 'UPDATE user' will neither pass nor fail.		
Known script volume limitations	None		
Script throughput	Not yet calculated		

1428-U-U-Ur-Oarrk-Fj-Es-Update User Role (by email)


Sample	1428-U-U-Ur-Oarrk-Fj-Es-Update User Role (by email)		
Script Basis	U: User		
OAuth Access	'User Provisioning'		
Basic description	Ur: Updates the user's role assignment when the role is directly assigned to the user. (not generally best practice)		
Ideal for	<ul style="list-style-type: none"> Updating specific users or when the number of users to be updated is very large (over about 1000) When you do not necessarily know the userid, you use SAMLSSO based on e-mail or a custom property. Instead, you'd prefer to reference the users to be updated by their e-mail address. 		
Not suitable when	Updating a team of users when the team size is less than about 1000 users, as script 1458, would be more suitable. Script 1458 requires just the team name to be provided rather than the list of all the usernames.		
Notes	The BI license type will change to 'named user' from 'concurrent session' if a Planning Role is assigned directly to the user (unlike when a user inherits a Planning role through inheritance of team membership).		
Data file syntax	F: .json		
	Field	Type	Description
	file_email	string	Users email address
	file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace" and "keep".
	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value": "Role1"}, {"value": "PROFILE:sap.epm:BI_Admin"}]
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password)		
How the script works	Same as sample script 1408		
Known script volume limitations	None		
Script throughput	Not yet calculated		
Video tutorial	 SAP HANA Academy provide a step-by-step tutorial for sample scripts 140x and 145x https://youtu.be/emDiUFCi_vk (Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)		
Related Scenarios	R01- Swap directly assigned role for a team role		
	Re-assign all users with the 'BI_Admin' role directly assigned to them, add these users to a team, and add the team to be a member of the 'BI_Admin' role before removing the directly assigned role from all the users. This enables you to adopt the best practice for assigning roles to users with the help of inheritance and teams.		

1429-U-U-Um-Fcj-Es-Update SAML Mapping (by email)


Sample	1429-U-U-Um-Fcj-Es-Update SAML Mapping (by email)											
Script Basis	U: User											
OAuth Access	'User Provisioning'											
Basic description	Um: Updates the user's SAML mapping property (userName) but identifies the user by their email address rather than the SAC user ID.											
Ideal for	<ul style="list-style-type: none">Updating the SAML mapping property when you don't know the user ID assigned by SAP Analytics Cloud; you'd rather identify the user by their e-mail address.When you are using your custom Identity Provider, and you have set up SSO to map on a custom attribute, which is the SAML mapping propertyWhen using your own custom Identity Provider, and you have set up SSO to map on e-mail, the SAML mapping property. i.e., you want to update the user's e-mail addressAssisting in the process of switching the SAP Analytics Cloud Identity Provider SSO from mapping on userid to mapping on a custom property (such as your organisation ID)											
Not suitable when	<ul style="list-style-type: none">You can easily determine the user ID assigned by SAP Analytics Cloud. If this is the case, use sample script 1409, as the throughput is likely to be slightly better compared to this.When you are using the default IDPWhen you are using your own custom Identity Provider and have set up SSO to map on the User ID.											
Notes	The user property 'userName' holds the SAML Mapping value. This property is NOT changeable when using the default IDP with SAP Analytics Cloud. It is also not changeable when using your custom Identity Provider, and you have set up SSO to map on the User ID.											
Data file syntax	F: .csv and .json <table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_email</td><td>string</td><td>Users email address</td></tr><tr><td>file_samlmapping</td><td>string</td><td>The new value for the SAML Mapping</td></tr></table>			Field	Type	Description	file_email	string	Users email address	file_samlmapping	string	The new value for the SAML Mapping
Field	Type	Description										
file_email	string	Users email address										
file_samlmapping	string	The new value for the SAML Mapping										
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password, SAMLSSO)											
How the script works	Reads the data file, and for each entry (row), the user is read from SAP Analytics Cloud. If the user is not found, the test 'READ user' fails; otherwise, a comparison is made of the existing user's userName (SAML mapping) and compared with the provided property from the file. If an update is necessary, the user is appropriately updated; otherwise, the user is skipped. If the user doesn't need to be updated, the test 'UPDATE user' will neither pass nor fail.											
Known script volume limitations	None											
Script throughput	Not recorded.											

1451-TU-U-UI-Fcj-Es-Update Team License

Sample	1451-TU-U-UI-Fcj-Es-Update Team License											
Script Basis	TU: Team and then each user of the team											
OAuth Access	'User Provisioning'											
Basic description	UI: For the given teams, each user of each team is updated. The users' BI license type property is updated.											
Ideal for	<div><div>1.</div><div>When you'd like to update whole teams at a time (providing just the team names rather than a list of users)</div></div> <div><div>2.</div><div>Updating whole teams of users and when the number of users in any given team isn't much greater than 1000 users.</div></div>											
Not suitable when	When any of the given teams have a team size greater than 1000 users, then script 1401 would be more suitable to update teams of users. Script 1401 requires each user to be provided but is likely more stable when processing many iterations.											
Notes	<p>The BI license type only applies when a Business Intelligence concurrent user license has been purchased, and the user is consuming only that license type, other than an Analytics Hub or Planning license. This topic is discussed in great detail in an article introduced by this blog https://blogs.sap.com/2020/03/10/sap-analytics-cloud-managing-licenses-with-roles-and-teams/. Please refer to this article for more information.</p> <p>The 'concurrent-session' setting is only a 'request' for a concurrent-session license, and it does not mean the user <u>will</u> consume a concurrent-session license. The user <u>could</u> consume a named-user license because they were assigned a Planning Role (or an Analytics Hub Role). For more details on this, please visit the blog just mentioned.</p>											
Data file syntax	<div>F: .csv and .json</div> <table><tr><td>Field</td><td>Type</td><td>Description</td></tr><tr><td>file_team</td><td>string</td><td>Team name</td></tr><tr><td>file_isconcurrent</td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr></table>			Field	Type	Description	file_team	string	Team name	file_isconcurrent	boolean	Sets the BI license type to concurrent or named user
Field	Type	Description										
file_team	string	Team name										
file_isconcurrent	boolean	Sets the BI license type to concurrent or named user										
Environment	Single Service (SACserviceFQDN, SACplatform, SACtokenFQDN, Username, Password)											
How the script works	<p>Reads the data file, and the team is read for each entry (row). Then, for each user of that team, the user is read. If the user is not found, the test 'READ user' fails; otherwise, a comparison is made of the existing user's properties and compared with the provided property from the file. If an update is necessary, the user is appropriately updated; otherwise, the user is skipped. If the user doesn't need to be updated, the test 'UPDATE user' will neither pass nor fail.</p> <p>Once all the team users are processed, the next team is read, and the script continues.</p> <p>Be careful with the file_isconcurrent value as it must be a lowercase true or false and not TRUE OR FALSE.</p> <p>If the file_isconcurrent setting is set to true (for concurrent), but should the user be assigned roles directly (rather than indirectly via a team) and if any of those directly assigned roles are planning roles, then the isConcurrent setting will not be respected, i.e., the isConcurrent will be set to false (a named user).</p>											
Known script volume limitations	The maximum team size is around 1000 users. It means stability issues should be expected around this size of team, and the alternative script of 1401 would provide greater stability.											
Script throughput		Service with 500 users	Service with 80,000 users									


	Updating user	0.33 users / sec 2.99 secs / user (8.6%)	0.14 users / sec 7.13 secs / user (3.5%)	
Video tutorial		SAP HANA Academy provide a step-by-step tutorial for sample scripts 140x and 145x https://youtu.be/emDiUFCi_vk		
Related Scenarios	L01 - Managers with Blconcurrent to Blnamed license	Identifies which users have a 'Business Intelligence' concurrent session license AND which are also managers. It then updates their license type to 'named-user', ensuring all managers can always log in without waiting for a spare concurrent session.		
	L03 - Convert all Blconcurrent to Blnamed license	Updates all users with a Business Intelligence 'concurrent session' license to a Business Intelligence 'named user' license.		
	S02 - Assign settings concurrent lic for recently created w default settings	Assign the correct date/time/number and language settings for those users created in the last week with the default settings and a 'BI concurrent session' license. It means your users will then be set correctly, avoiding each user updating their settings, which is an unnecessary task.		

1454-TU-U-Um-Fcj-Es-Update Team Manager

Sample	1454-TU-U-Um-Fcj-Es-Update Team Manager											
Script Basis	TU: Team and then each user of the team											
OAuth Access	'User Provisioning'											
Basic description	Um: For the given teams, each user of each team is updated. The users' manager is updated.											
Ideal for	<div><div>1.</div><div>When you'd like to update whole teams at a time (providing just the team names rather than a list of users)</div></div> <div><div>2.</div><div>Updating whole teams of users and when the number of users in any given team isn't much greater than 1000 users.</div></div>											
Not suitable when	When any of the given teams has a team size greater than 1000 users, then script 1404 would be more suitable to update teams of users. Script 1404 requires each user to be provided but is likely to be more stable when processing many iterations.											
Notes	Cyclic loops of users to managers aren't allowed. It means user A can't be a manager of user B when user B is the manager of user A. Users are not allowed to be a manager of themself.											
Data file syntax	F: .csv and .json <table><tr><td>Field</td><td>Type</td><td>Description</td></tr><tr><td>file_team</td><td>string</td><td>Team name</td></tr><tr><td>file_managerid</td><td>string</td><td>Optional ID of the manager's ID</td></tr></table>			Field	Type	Description	file_team	string	Team name	file_managerid	string	Optional ID of the manager's ID
Field	Type	Description										
file_team	string	Team name										
file_managerid	string	Optional ID of the manager's ID										
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password)											
How the script works	<div>Reads the data file, and the team is read for each entry (row). Then, for each user of that team, the user is read. If the user is not found, the test 'READ user' fails; otherwise, a comparison is made of the existing user's properties and compared with the provided property from the file. If an update is necessary, the user is appropriately updated; otherwise, the user is skipped. If the user doesn't need to be updated, the test 'UPDATE user' will neither pass nor fail.</div> <div>Once all the users of the team are processed, the next team is read, and the script continues.</div>											
Known script volume limitations	The maximum team size is around 1000 users. It means stability issues should be expected around this size of the team, and the alternative script of 1404 would provide greater stability.											
Script throughput		Service with 500 users	Service with 80,000 users									
	Updating user	0.33 users / sec 2.99 secs / user (8.6%)	0.14 users / sec 7.13 secs / user (3.5%)									
Video tutorial	<div> SAP HANA Academy provide a step-by-step tutorial for sample scripts 140x and 145x https://youtu.be/emDiUFCi_vk</div> <div>(Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)</div>											
Related Scenarios	M01 - Reassign users of given manager to another	Re-assign all users of the given manager(s) to another manager, which is helpful when the manager is unavailable or needs to be deleted.										


1456-TU-U-Upddtn-Fj-Es-Update Team DateTimeNumFormat DataAccessLang

Sample	1456-TU-U-Upddtn-Fj-Es-Update Team DateTimeNumFormat DataAccessLang																							
Script Basis	TU: Team and then each user of the team																							
OAuth Access	'User Provisioning'																							
Basic description	Upddtn: For the given teams, each user of each team is updated. Updates the users' properties: <div><div>1.</div><div>Preferredlanguage</div></div> <div><div>2.</div><div>Dataaccesslanguage</div></div> <div><div>3.</div><div>Dateformatting</div></div> <div><div>4.</div><div>Timeformatting</div></div> <div><div>5.</div><div>Numberformatting</div></div>																							
Ideal for	<div><div>1.</div><div>When you'd like to update whole teams at a time (providing just the team names rather than a list of users)</div></div> <div><div>2.</div><div>Updating whole teams of users and when the number of users in any given team isn't much greater than 1000 users.</div></div>																							
Not suitable when	Updating teams of users and when any of the given teams have a team size greater than about 1000 users, then script 1404 would be more suitable. Script 1404 requires each user to be provided but is likely to be more stable when processing a significant number of iterations.																							
Notes	This script is typically only necessary when users have been created incorrectly. To avoid the need to perform any user updates on newly created users, edit the 'CREATE user' requests in the scripts that create users (scripts 11xx and 12xx) with the correct values for the preferredlanguage, dataaccesslanguage, dateformatting, timeformatting and numberformatting. These values are 'hard coded' in the script. Set these values before running the scripts to create users. This way, the users will have the correct settings for these properties, and there'll be no need to update them afterwards.																							
Data file syntax	<div>F: .json</div> <table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_team</td><td>string</td><td>Team name</td></tr><tr><td>file_preferredlanguage</td><td>string</td><td>possible values: 'ar', 'bg', 'zh-CN', 'zh-TW', 'hr', 'ca', 'cs', 'cy', 'nl', 'da', 'en', 'en-GB', 'fi', 'fr', 'fr-CA', 'de', 'de-CH', 'el', 'he', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'ms', 'no', 'pl', 'pt', 'pt-PT', 'ro', 'ru', 'sh', 'sk', 'sl', 'es', 'es-MX', 'sv', 'th', 'tr', 'uk', 'vi', 'en-US-sapssd' and possibly more will be added over time!</td></tr><tr><td>file_dataaccesslanguage</td><td>string</td><td>possible values: 'zz' (default), 'af', 'ar', 'bg', ... many more as shown above in preferredlanguage.</td></tr><tr><td>file_dateformatting</td><td>string</td><td>possible values include: 'MMM d, yyyy', 'dd/MM/yyyy', 'MM.dd.yyyy', ... many more!</td></tr><tr><td>file_timeformatting</td><td>string</td><td>possible values include: 'H:mm:ss' (24H), 'h:mm:ss A' (12H)</td></tr><tr><td>file_numberformatting</td><td>string</td><td>possible values include: '1,234.56', '1.234,56', '1 234,56'</td></tr></table>			Field	Type	Description	file_team	string	Team name	file_preferredlanguage	string	possible values: 'ar', 'bg', 'zh-CN', 'zh-TW', 'hr', 'ca', 'cs', 'cy', 'nl', 'da', 'en', 'en-GB', 'fi', 'fr', 'fr-CA', 'de', 'de-CH', 'el', 'he', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'ms', 'no', 'pl', 'pt', 'pt-PT', 'ro', 'ru', 'sh', 'sk', 'sl', 'es', 'es-MX', 'sv', 'th', 'tr', 'uk', 'vi', 'en-US-sapssd' and possibly more will be added over time!	file_dataaccesslanguage	string	possible values: 'zz' (default), 'af', 'ar', 'bg', ... many more as shown above in preferredlanguage.	file_dateformatting	string	possible values include: 'MMM d, yyyy', 'dd/MM/yyyy', 'MM.dd.yyyy', ... many more!	file_timeformatting	string	possible values include: 'H:mm:ss' (24H), 'h:mm:ss A' (12H)	file_numberformatting	string	possible values include: '1,234.56', '1.234,56', '1 234,56'
Field	Type	Description																						
file_team	string	Team name																						
file_preferredlanguage	string	possible values: 'ar', 'bg', 'zh-CN', 'zh-TW', 'hr', 'ca', 'cs', 'cy', 'nl', 'da', 'en', 'en-GB', 'fi', 'fr', 'fr-CA', 'de', 'de-CH', 'el', 'he', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'ms', 'no', 'pl', 'pt', 'pt-PT', 'ro', 'ru', 'sh', 'sk', 'sl', 'es', 'es-MX', 'sv', 'th', 'tr', 'uk', 'vi', 'en-US-sapssd' and possibly more will be added over time!																						
file_dataaccesslanguage	string	possible values: 'zz' (default), 'af', 'ar', 'bg', ... many more as shown above in preferredlanguage.																						
file_dateformatting	string	possible values include: 'MMM d, yyyy', 'dd/MM/yyyy', 'MM.dd.yyyy', ... many more!																						
file_timeformatting	string	possible values include: 'H:mm:ss' (24H), 'h:mm:ss A' (12H)																						
file_numberformatting	string	possible values include: '1,234.56', '1.234,56', '1 234,56'																						
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password)																							
How the script works	Reads the data file, and for each entry (row), the team is read. Then, for each user of that team, the user is read. If the user is not found, the test 'READ user' fails; otherwise, a comparison is made of the existing user's properties and compared with the provided property from the file. If an update is necessary, the user is appropriately updated;																							

	<p>otherwise, the user is skipped. If the user doesn't need to be updated, the test 'UPDATE user' will neither pass nor fail.</p> <p>Once all the users of the team are processed, the next team is read, and the script continues.</p>		
Known script volume limitations	The maximum team size is around 1000 users. It means stability issues should be expected around this size of team, and the alternative script of 1406 would provide greater stability.		
Script throughput		Service with 500 users	Service with 80,000 users
	Updating user	0.33 users / sec 2.99 secs / user (8.6%)	0.14 users / sec 7.13 secs / user (3.5%)
Video tutorial	 SAP HANA Academy provide a step-by-step tutorial for sample scripts 140x and 145x https://youtu.be/emDiUFCi_vk (Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)		
Related Scenarios	S01 - Assign settings for recently created with default settings	Assign the correct date/time/number and language settings for those users created last week with the default settings. It means your users will then be set correctly, avoiding each user from updating their settings, which is an unnecessary task.	
	S02 - Assign settings concurrent lic for recently created w default settings	It is the same as S01; only it also assigns the user with a 'Business Intelligence' 'concurrent session' license.	

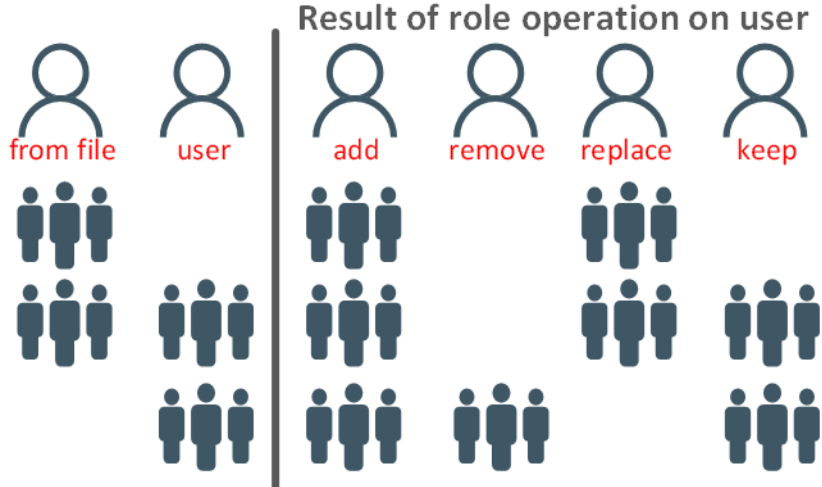

1457-TU-U-Ulpddtn-Fj-Es-Update Team License, DateTimeNumFormat, DataAccessLang

Sample	1457-TU-U-Ulpddtn-Fj-Es-Update Team License, DateTimeNumFormat, DataAccessLang																	
Script Basis	TU: Team and then each user of the team																	
OAuth Access	'User Provisioning'																	
Basic description	Ulpddtn: For the given teams, each user of each team is updated. Updates the users' properties: <ul style="list-style-type: none">1. BI license type2. Preferredlanguage3. Dataaccesslanguage4. Dateformatting5. Timeformatting6. Numberformatting																	
Ideal for	<ul style="list-style-type: none">3. When you'd like to update whole teams at a time (providing just the team names rather than a list of users)4. Updating whole teams of users and when the number of users in any given team isn't much greater than 1000 users.																	
Not suitable when	When any of the given teams have a team size greater than 1000 users, then script 1404 would be more suitable to update teams of users. Script 1404 requires each user to be provided but is likely more stable when processing a significant number of iterations.																	
Notes	<p>The BI license type is only applicable when a Business Intelligence concurrent user license has been purchased, and the user is consuming only that license type, other than, say an Analytics Hub or Planning license. This topic is discussed in great detail in an article introduced by this blog https://blogs.sap.com/2020/03/10/sap-analytics-cloud-managing-licenses-with-roles-and-teams/. Please refer to this article for more information.</p> <p>Be careful with the <code>file_isconcurrent</code> value as it must be a lowercase <code>true</code> or <code>false</code> and not <code>TRUE</code> OR <code>FALSE</code>.</p> <p>This script is typically only necessary when users have been created incorrectly. To avoid the need to perform any user updates on newly created users, edit the 'CREATE user' requests in the scripts that create users (scripts 11xx and 12xx) with the correct values for the <code>preferredlanguage</code>, <code>dataaccesslanguage</code>, <code>dateformatting</code>, <code>timeformatting</code> and <code>numberformatting</code>. These values are 'hard coded' in the script. Set these values before running the scripts to create users. This way, the users will have the correct settings for these properties, and there'll be no need to update them afterwards.</p>																	
Data file syntax	F: .json <table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td><code>file_team</code></td><td>string</td><td>Team name</td></tr><tr><td><code>file_isconcurrent</code></td><td>boolean</td><td>Sets the BI license type to concurrent or named user</td></tr><tr><td><code>file_preferredlanguage</code></td><td>string</td><td>possible values: 'ar', 'bg', 'zh-CN', 'zh-TW', 'hr', 'ca', 'cs', 'cy', 'nl', 'da', 'en', 'en-GB', 'fi', 'fr', 'fr-CA', 'de', 'de-CH', 'el', 'he', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'ms', 'no', 'pl', 'pt', 'pt-PT', 'ro', 'ru', 'sh', 'sk', 'sl', 'es', 'es-MX', 'sv', 'th', 'tr', 'uk', 'vi', 'en-US-sapds' and possibly more will be added over time!</td></tr><tr><td><code>file_dataaccesslanguage</code></td><td>string</td><td>possible values: 'zz' (default), 'af', 'ar', 'bg', ... many more as shown above in preferredlanguage.</td></tr></table>			Field	Type	Description	<code>file_team</code>	string	Team name	<code>file_isconcurrent</code>	boolean	Sets the BI license type to concurrent or named user	<code>file_preferredlanguage</code>	string	possible values: 'ar', 'bg', 'zh-CN', 'zh-TW', 'hr', 'ca', 'cs', 'cy', 'nl', 'da', 'en', 'en-GB', 'fi', 'fr', 'fr-CA', 'de', 'de-CH', 'el', 'he', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'ms', 'no', 'pl', 'pt', 'pt-PT', 'ro', 'ru', 'sh', 'sk', 'sl', 'es', 'es-MX', 'sv', 'th', 'tr', 'uk', 'vi', 'en-US-sapds' and possibly more will be added over time!	<code>file_dataaccesslanguage</code>	string	possible values: 'zz' (default), 'af', 'ar', 'bg', ... many more as shown above in preferredlanguage.
Field	Type	Description																
<code>file_team</code>	string	Team name																
<code>file_isconcurrent</code>	boolean	Sets the BI license type to concurrent or named user																
<code>file_preferredlanguage</code>	string	possible values: 'ar', 'bg', 'zh-CN', 'zh-TW', 'hr', 'ca', 'cs', 'cy', 'nl', 'da', 'en', 'en-GB', 'fi', 'fr', 'fr-CA', 'de', 'de-CH', 'el', 'he', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'ms', 'no', 'pl', 'pt', 'pt-PT', 'ro', 'ru', 'sh', 'sk', 'sl', 'es', 'es-MX', 'sv', 'th', 'tr', 'uk', 'vi', 'en-US-sapds' and possibly more will be added over time!																
<code>file_dataaccesslanguage</code>	string	possible values: 'zz' (default), 'af', 'ar', 'bg', ... many more as shown above in preferredlanguage.																


	file_dateformatting	string	possible values include: 'MMM d, yyyy', 'dd/MM/yyyy', 'MM.dd.yyyy', ... many more!
	file_timeformatting	string	possible values include: 'H:mm:ss' (24H), 'h:mm:ss A' (12H)
	file_numberformatting	string	possible values include: '1,234.56', '1.234,56', '1 234,56'
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password)		
How the script works	<p>Reads the data file, and for each entry (row), the team is read. Then, for each user of that team, the user is read. If the user is not found, the test 'READ user' fails; otherwise, a comparison is made of the existing user's properties and compared with the provided property from the file. If an update is necessary, the user is appropriately updated; otherwise, the user is skipped. If the user doesn't need to be updated, the test 'UPDATE user' will neither pass nor fail.</p> <p>Once all the users of the team are processed, the next team is read, and the script continues.</p>		
Known script volume limitations	The maximum team size is around 1000 users. It means stability issues should be expected around this team size, and the alternative script of 1407 would provide greater stability.		
Script throughput		Service with 500 users	Service with 80,000 users
	Updating user	0.33 users / sec 2.99 secs / user (8.6%)	0.14 users / sec 7.13 secs / user (3.5%)
Video tutorial	 <p>SAP HANA Academy provide a step-by-step tutorial for sample scripts 140x and 145x https://youtu.be/emDiUFCi_vk</p> <p>(Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)</p>		
Related Scenarios	S01 - Assign settings for recently created with default settings	Assign the correct date/time/number and language settings for those users created last week with the default settings. It means your users will then be set correctly, avoiding each user from updating their settings, which is an unnecessary task.	
	S02 - Assign settings concurrent lic for recently created w default settings	It is the same as S01; only it also assigns the user with a 'Business Intelligence' 'concurrent session' license.	

1458-TU-U-Ur-Oarrk-Fj-Es-Update Role for Each User of Team


Sample	1458-TU-U-Ur-Oarrk-Fj-Es-Update Role for Each User of Team														
Script Basis	TU: Team and then each user of the team														
OAuth Access	'User Provisioning'														
Basic description	Ur: For the given teams, each user of each team is updated. The user's role assignment is updated when the role is directly assigned to the user. (not generally best practice)														
Ideal for	<div><div>1.</div><div>When you'd like to update whole teams at a time (providing just the team names rather than a list of users)</div></div> <div><div>2.</div><div>Updating whole teams of users and when the number of users in any given team isn't much greater than 1000 users.</div></div>														
Not suitable when	Updating teams of users and when any of the given teams have a team size greater than about 1000 users, then script 1408 would be more suitable. Script 1408 requires each user to be provided but is likely to be more stable when processing a great number of iterations.														
Notes	The BI license type will change to 'named user' from 'concurrent session' if a Planning Role is assigned directly to the user (unlike when a user inherits a Planning role through inheritance of team membership).														
Data file syntax	F: .json <table><tr><td>Field</td><td>Type</td><td>Description</td></tr><tr><td>file_team</td><td>string</td><td>Team name</td></tr><tr><td>file_roles_action</td><td>string</td><td>Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace", "keep".</td></tr><tr><td>file_JSON_roles</td><td>JSON values of roles</td><td>Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]</td></tr></table>			Field	Type	Description	file_team	string	Team name	file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace", "keep".	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]
Field	Type	Description													
file_team	string	Team name													
file_roles_action	string	Defines the action to perform on the user's assignment to roles. Possible values are: "add", "remove", "replace", "keep".													
file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]													
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password)														
How the script works	<div><div>Reads the data file and for each entry (row) the team is read. Then for each user of that team, the user is read. If the user is not found, the test 'READ user' fails, otherwise a comparison is made of the existing user's properties and compared with the provided property from the file. If an update is necessary, then the user is appropriately updated, otherwise the user is skipped. If the user doesn't need to be updated the test 'UPDATE user' will neither pass nor fail.</div><div>Once all the users of the team are processed, the next team is read, and the script continues.</div><div>The roles with the same content namespace as the environment don't need to specify the PROFILE:namespace: as part of its name. For example, if a Role called 'Role1' is exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only 'Role1' needs to be specified. For any role that doesn't match the ContentNamespace then PROFILE:namespace: must be specified with namespace set accordingly. For example, the provided 'BI_Admin' role would be PROFILE:sap.epm:BI_Admin</div><div>The file_JSON_roles can be an empty array, i.e. [] and this is useful if you'd like to remove all roles directly assigned to the user</div></div>														

	<p>If the role doesn't already exist a 400 status error will be returned and the 'UPDATE user' will not pass its test.</p> <p>Updating Operations (arrk)</p> <p>For updating existing users, the roles_action can be "add", "remove", "replace" and "keep":</p> <p>add: will add the roles to the user.</p> <p>remove: will remove the roles from the user, if the user has any of the roles listed.</p> <p>replace: will replace (set) the roles for the user (i.e. remove and add accordingly).</p> <p>keep: will respect any roles the user has already been assigned, no changes will be made.</p> <div style="text-align: center;"> <p>Result of role operation on user</p>  </div>		
Known script volume limitations	The maximum team size is around 1000 users. It means stability issues should be expected around this size of team and the alternative script of 1407 would provide greater stability.		
Script throughput		Service with 500 users	Service with 80,000 users
	Updating user	0.33 users / sec 2.99 secs / user (8.6%)	0.14 users / sec 7.13 secs / user (3.5%)
Video tutorial	 SAP HANA Academy provide a step-by-step tutorial for sample scripts 140x and 145x https://youtu.be/emDiUFCi_vk (Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)		
Related Scenarios	R01- Swap directly assigned role for a team role	Re-assign all users with the 'BI_Admin' role directly assigned to them, add these users to a team, and add the team to be a member of the 'BI_Admin' role before removing the directly assigned role from all the users. This enables you to adopt the best practice for assigning roles to users with the help of inheritance and teams.	

1501-T-UC-Ud-Fcj-Es-Update create team

Sample	1501-T-UC-Ud-Fcj-Es-Update create team		
Script Basis	T: Team		
OAuth Access	'User Provisioning'		
Basic description	UC: Updates existing or creates a new team with a displayname. No changes are made to the user membership of the team or the team's membership of roles.		
Ideal for	1. Creating a new team with a particular displayname 2. Updating a team with a new displayname		
Not suitable when			
Notes	<p>Other sample scripts that create teams, because they don't exist, are created without a team folder, and the team will have a display name as set in the file.</p> <p>This script can be used before or after other scripts to replace the displayname with the desired string.</p>		
Data file syntax	F: .csv and .json		
	Field	Type	Description
	file_team	string	Team name
	file_displayname	string	Display name
Environment	Single Service (SACserviceFQDN, SACplatform, SAcTokenFQDN, Username, Password)		
How the script works	<p>Uc: Reads the data file, and for each entry (row), the team is read. If the team is found, a comparison of the existing display name with the value from the file. If an update is necessary, the team display name is updated. The existing team membership, for both users and roles, is respected.</p> <p>If the team does not exist, it is created without a team folder, and the team will have the display name value from the file.</p> <p>(When SAP Analytics Cloud is hosted on an SAP Data Centre (NEO), teams are created with a team folder, and the description for the team and the description of the team folder will use the value from the file.)</p> <p>Should the display name be over 1024 characters, a console error is shown, and the display name is truncated to 1024 characters as this is the maximum allowed.</p>		
Known script volume limitations	None		
Script throughput		Service with 500 users	Service with 80,000 users
	Creating team	0.55 teams / sec 1.82 secs / team (8.9%)	0.52 teams / sec 1.91 secs / team (8.3%)
Video tutorial		SAP HANA Academy provide a step-by-step tutorial for this sample script 1501 https://youtu.be/mLmKlqrNqOU (Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)	

1601-All_T-List all teams

Sample	1601-All_T-List all teams
Script Basis	All Teams, one or multiple requests to return all the Teams
OAuth Access	'User Provisioning'
Basic description	List: Lists all the teams found, and for each team, the number of members of the team and the number of roles the team is a member of
Ideal for	Listing all the teams, understanding how many teams there are, and how many of them have users. Understanding the team ID, which is needed when using API version 1.
Not suitable when	
Notes	This is a read-only script and outputs to the console. No harm can be done by running this script! The same sample script is available using API version 2; please see script 2601 . Version 2 identifies teams by the UUID, unlike version 1, which uses the team name. Two teams with the same name may exist, which is problematic for the version 1 API, unlike with version 2.
Data file syntax	There are no data file requirements.
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password)
How the script works	Sends a request using the version 1 API endpoint /api/v1/scim/Groups to retrieve all the teams. Multiple requests will be made if pages of teams need to be returned due to the number of them being limited per page. As each page of teams is returned, the name of the team and the number of users and roles are output to the console. 
Known script volume limitations	None
Script throughput	

1602-T-Uc-Uur-Oarrk-B-Fj-Es users/roles actions on Teams

Sample	1602-T-Uc-Uur-Oarrk-B-Fj-Es users/roles actions on Teams																				
Script Basis	T: Team																				
OAuth Access	'User Provisioning'																				
Basic description	Uur: Updates, or creates a new team, with users and assigns the team to roles																				
Ideal for	<div><div>1.</div><div>Bulk operations on teams for adding/removing or replacing users in the team or the teams' role membership.</div></div> <div><div>2.</div><div>Highest throughput is needed, but an entry in the file per team isn't guaranteed. (use script 1612 when an entry per team is guaranteed).</div></div>																				
Not suitable when	If the number of teams (typically over 30) and the average number of users in a team is large (typically over 5000), then script 1612 should be used, as this sample script will 'batch' the entries and update periodically rather than update per entry.																				
Notes	This sample script will provide a significantly greater throughput to update teams than any scripts on a user basis. There should be a single entry per team with a JSON list of users/roles for the most significant throughput.																				
Data file syntax	<div>F: .json</div> <table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_team</td><td>string</td><td>Team name</td></tr><tr><td>file_users_action</td><td>string</td><td>Defines the action to perform on the team membership of users. Possible values are: "add", "remove", "replace" and "keep".</td></tr><tr><td>file_roles_action</td><td>string</td><td>Defines the action to perform on the assignment of the team to roles. Possible values are: "add", "remove", "replace" and "keep".</td></tr><tr><td>file_JSON_users</td><td>JSON values of users</td><td>Optional JSON array of keys "value" and string values of users. Example: [{"value": "MATTHEW1"}, {"value": "MATTHEW2"}, {"value": "MATTHEW3"}]</td></tr><tr><td>file_JSON_roles</td><td>JSON values of roles</td><td>Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value": "Role1"}, {"value": "PROFILE:sap.epm:BI_Admin"}]</td></tr></table>			Field	Type	Description	file_team	string	Team name	file_users_action	string	Defines the action to perform on the team membership of users. Possible values are: "add", "remove", "replace" and "keep".	file_roles_action	string	Defines the action to perform on the assignment of the team to roles. Possible values are: "add", "remove", "replace" and "keep".	file_JSON_users	JSON values of users	Optional JSON array of keys "value" and string values of users. Example: [{"value": "MATTHEW1"}, {"value": "MATTHEW2"}, {"value": "MATTHEW3"}]	file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value": "Role1"}, {"value": "PROFILE:sap.epm:BI_Admin"}]
Field	Type	Description																			
file_team	string	Team name																			
file_users_action	string	Defines the action to perform on the team membership of users. Possible values are: "add", "remove", "replace" and "keep".																			
file_roles_action	string	Defines the action to perform on the assignment of the team to roles. Possible values are: "add", "remove", "replace" and "keep".																			
file_JSON_users	JSON values of users	Optional JSON array of keys "value" and string values of users. Example: [{"value": "MATTHEW1"}, {"value": "MATTHEW2"}, {"value": "MATTHEW3"}]																			
file_JSON_roles	JSON values of roles	Optional JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value": "Role1"}, {"value": "PROFILE:sap.epm:BI_Admin"}]																			
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password)																				
How the script works	<div>For each entry in the file, the team is read from SAP Analytics Cloud.</div> <div>A comparison is then made of the existing users and their roles' assignments. The net difference is then calculated to determine which users and roles should be added or removed. This includes any previously read entries for the same team. i.e. the same team can have multiple entries in the file, and the net effect is calculated.</div> <div>When any of the following is met:</div> <div><div><div>•</div><div>number of teams read more than 30 teams</div></div><div><div>•</div><div>number of teams read is at least 4, and the number of users in all the teams read so far has reached 40,000 users</div></div><div><div>•</div><div>the end of the file</div></div></div> <div>then, the teams read so far are updated with the necessary changes if any changes are needed. This sample script has thus 'batched' the teams together (B)</div>																				

(Sample script 1612 is identical to this sample script, except the above conditions are not tested; instead, the team update is always performed for every row of the file read. There is no batching of teams like there is with this script 1602)

Only the 'net' difference between the teams is made. This means only the users/roles that need to be added or removed are added or removed. Users and roles are not just added or removed without intelligently inspecting the current assignment. This could mean no updates are performed against a team. In such cases, the 'UPDATE team' test will neither pass nor fail.

Users and roles are updated in 'chunks', meaning multiple PUT calls are made one after the other until all changes have been completed. The 'chunk' size is automatically determined by the existing number of users in the team. For an empty team, the chunk size will be 4590 users and 50 roles; otherwise, the chunk size is determined by these formulas (rounding when necessary):

User chunk size = (4590 - (number_of_users_in_the_team x 0.14))

Role chunk size = ((210 / (number_of_users_in_the_team x 46 / 210)) x 10)

After the initial PUT call has been made, subsequent chunk sizes are dynamically and automatically adjusted as users and roles are added and removed from the team and are no longer determined by the above formula. Instead, the chunk sizes are based on the response time of the API itself. Changes to the chunk sizes are only performed when the last PUT used the 'full' chunk size, and it wasn't a repeated call attempting to recover due to a previous error or failure. This ensures the most significant throughput possible and allows for any performance optimisation of the API to be realised. The console output reports the throughput and any chunk size changes. Chunk sizes are always at least 1.

The maximum number of users in a team is 32,767, and this sample script will ignore any users over this limit to avoid an error from the API.

The console log will report any users or roles that were expected to be added or removed but were not. For example, the console error will log users who don't exist in the team because they're not registered users.

The file_JSON_users can be an empty array, i.e. []

Roles


The roles with the same content namespace as the environment don't need to specify the PROFILE:namespace: as part of its name. For example, if a Role called 'Role1' exists in the SAP Analytics Cloud Service and the environment ContentNamespace is set to the same Namespace as the Service the role was created in, then only 'Role1' needs to be specified. For any role that doesn't match the ContentNamespace, then PROFILE:namespace: must be specified with namespace set accordingly. For example, the provided 'BI_Admin' role would be PROFILE:sap.epm:BI_Admin

Roles are only added to new teams if the roles_action is "add" or "replace"; otherwise, the roles are ignored, and the user will not be assigned to any roles.

The file_JSON_roles can be an empty array, i.e. []

If the role doesn't already exist, a 400 status error will be returned, and the 'UPDATE team' will not pass its test.

	<div data-bbox="578 138 1312 590"> <p>Roles read from file</p> <p>Team</p> <p>Result of role operation on team</p> <p>add remove replace keep</p> </div> <p>Teams</p> <p>If the team does not exist, it will be created without a team folder, and the display name will match its name; otherwise, the existing team's display name is respected.</p> <p>(When SAP Analytics Cloud is hosted on an SAP Data Centre (NEO), then teams are created with a team folder, and the display name for the team will have the text “with Team Folder”. E.g. “Team1 with Team Folder”).</p> <p>Updating Operations (Oarrk)</p> <p>The users_action can be “add”, “remove”, “replace” and “keep”:</p> <p>add: will add the users to the team.</p> <p>remove: will remove the users from the team, if the user is in the team.</p> <p>replace: will replace (set) the users in the team (i.e. remove and add accordingly).</p> <p>keep: will respect any existing users in the team, and no changes will be made to the user membership.</p> <div data-bbox="597 1203 1291 1617"> <p>Users read from file</p> <p>Team</p> <p>Result of user operation on team</p> <p>add remove replace keep</p> </div> <p>The roles_action can be “add”, “remove”, “replace”, and “keep”:</p> <p>add: will add the team to the role.</p> <p>remove: will remove the team from the role if the team is a member of any of the roles listed.</p> <p>replace: will replace (set) the team assignment to roles (i.e. remove and add accordingly).</p> <p>keep: will respect any roles the team is already a member of; no changes will be made.</p>
Known script volume limitations	Stability issues are expected when the number of teams (typically over 30) and the average number of users in a team is large (typically over 5000). For improved stability, use script 1612.

Script throughput	<p>See sample script 1612.</p> <p>Script 1602 will have additional overhead to manage and process the team membership compared to script 1612. The performance of 1602 heavily depends upon the structure of the .json data file.</p>
Video tutorial	<div>  <p>SAP HANA Academy provide a step-by-step tutorial for this sample script 1602 https://youtu.be/o-gmtDKilOw</p> <p>(Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)</p> </div>

1612-T-Uc-Uur-Oarrk-Fcj-Es users/roles actions on Teams

Sample	1612-T-Uc-Uur-Oarrk-Fcj-Es users/roles actions on Teams			
Script Basis	T: Team			
OAuth Access	'User Provisioning'			
Basic description	Uur: Updates or creates a new team with users and assigns the team to roles			
Ideal for	<ol style="list-style-type: none"> 1. Bulk operations on teams for adding/removing or replacing users in the team or the teams' role membership. 2. The highest throughput is needed, and an entry per team is guaranteed. 			
Not suitable when	A team has multiple entries in the file, as this will cause unnecessary updates and reduce the throughput.			
Notes	<p>This sample script will provide a significantly greater throughput to update teams than any scripts on a user basis.</p> <p>Compared to script 1602, this script will likely be more stable and support more teams and users.</p>			
Data file syntax	Same as sample script 1602			
Environment	Single Service (SACserviceFQDN, SACplatform, SACtokenFQDN, Username, Password)			
How the script works	It is the same as sample script 1602, except the teams are not batched; instead, the team update is always performed for every row of the file read.			
Known script volume limitations	None			
Script throughput		Service with 500 users	Service with 80,000 users	
	Adding 1 users to a team (of average 250 users)	0.06 teams / sec	0.04 teams / sec	
		15.89 secs / team	22.63 secs / team	
		0.63 users / sec	0.44 users / sec	
		1.589 secs / user (12.3%)	2.263 secs / user (8.3%)	
	Adding 500 users into an empty team	0.08 teams / sec 13.10 secs / team 38.17 users / sec 0.026 secs / user (2.7%)	0.04 teams / sec 26.05 secs / team 19.20 users / sec 0.052 secs / user (1.3%)	
	Adding 10,000 users into an empty team	n/a	0.0015 teams / sec 660 secs / team 15.15 users / sec 0.066 secs / user (unrecorded)	
	Adding 500 users to team of 9,500 users	n/a	0.01 teams / sec 51.52 secs / team 9.71 users / sec 0.103 secs / user (8.5%)	
	Adding 500 users to team of 19,500 users	n/a	0.02 teams / sec 81.13 secs / team 6.163 users / sec 0.162 secs / user (13.5%)	

	Removing 500 users from a team of 10,000 users	n/a	0.02 teams / sec 41.78 secs / team 11.97 users / sec 0.084 secs / user (10.6%)	
	Removing 500 users from a team of 20,000 users	n/a	0.01 teams / sec 70.45 secs / team 7.10 users / sec 0.141 secs / user (16.9%)	

1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams

Sample	1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams																	
Script Basis	T: Team																	
OAuth Access	'User Provisioning'																	
Basic description	Utr: For performing team-on-team operations. Operations are for users and roles specifying a source and target team. Supported operations: add, remove, replace, keep, intersect, and exclude. A simple example is to add a team to a team, copy a team, remove a team from a team, etc.																	
Ideal for	All operations of teams on teams within a single SAP Analytics Cloud Service																	
Not suitable when	If you wish to perform team operations across different SAP Analytics Cloud Services, use sample scripts 19xx.																	
Notes	<ul style="list-style-type: none">Teams cannot be members of other teams, so this script performs operations solely on the team membership of users and roles. For example, adding users from a source team to a target team is adding the users of the source team to be members of the target team rather than the source team itself becoming a member of the target team.The same applies to roles; only teams are members of roles, rather than roles being members of teams.Perhaps confusingly, users can be direct members of roles, but this isn't recommended or considered best practice. This script doesn't perform direct user-to-role assignment; that's performed by scripts 1102, 1103, 1112, 1113, 1202 and 1203 (these scripts are on a 'user' basis, which means an update per user).User-to-role assignment is best performed indirectly via the team, i.e. a user is a member of the team, and the team is a member of the required roles. Changes to derived indirect role assignment require only a team modification to role assignment rather than to each user. Use scripts 1602 and 1612 to assign users to teams. Use this script to manipulate user-to-team assignments.The contents of any team folder and any security associated with the teams are ignored.																	
Data file syntax	F: .csv and .json <table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_source_team</td><td>string</td><td>Team name</td></tr><tr><td>file_target_team</td><td>string</td><td>Team name, changes are only made to the target</td></tr><tr><td>file_users_action</td><td>string</td><td>Defines the action to perform on the team's membership. Possible values are: "add", "remove", "replace", "keep" "intersect" and "exclude".</td></tr><tr><td>file_roles_action</td><td>string</td><td>Defines the action to perform on the team's roles' assignment. Possible values are: "add", "remove", "replace", "keep" "intersect" and "exclude".</td></tr></table>			Field	Type	Description	file_source_team	string	Team name	file_target_team	string	Team name, changes are only made to the target	file_users_action	string	Defines the action to perform on the team's membership. Possible values are: "add", "remove", "replace", "keep" "intersect" and "exclude".	file_roles_action	string	Defines the action to perform on the team's roles' assignment. Possible values are: "add", "remove", "replace", "keep" "intersect" and "exclude".
Field	Type	Description																
file_source_team	string	Team name																
file_target_team	string	Team name, changes are only made to the target																
file_users_action	string	Defines the action to perform on the team's membership. Possible values are: "add", "remove", "replace", "keep" "intersect" and "exclude".																
file_roles_action	string	Defines the action to perform on the team's roles' assignment. Possible values are: "add", "remove", "replace", "keep" "intersect" and "exclude".																
Environment	Single Service (SACserviceFQDN, SACplatform, SACtokenFQDN, Username, Password)																	
How the script works	It behaves similarly to sample script 1612, except the file does not provide a list of individual users; another team provides the list of users. However, this script is 'team' to 'team'. It																	

means you can copy the team members of one team, add them to another, and perform other operations.

The team is read from SAP Analytics Cloud for each entry in the file.

Then, the existing users and their roles' assignments are compared. The net difference is then calculated to determine which users and roles should be added or removed.

Only the 'net' difference between the teams is made. This means only the users/roles that need to be added or removed are added or removed. Users and roles are not just added or removed without intelligently inspecting the current assignment. This could mean no updates are performed against a team. The 'UPDATE team' test will neither pass nor fail in such cases.

Users and roles are updated in 'chunks', meaning multiple PUT calls are made one after the other until all changes have been completed. The 'chunk' size is automatically determined by the existing number of users in the team. For an empty team, the chunk size will be 4590 users and 50 roles; otherwise, the chunk size is determined by these formulas (rounding when necessary):

User chunk size = (4590 - (number_of_users_in_the_team x 0.14))

Role chunk size = ((210 / (number_of_users_in_the_team x 46 / 210)) x 10)

After the initial PUT call has been made, subsequent chunk sizes are dynamically and automatically adjusted as users and roles are added and removed from the team and are no longer determined by the above formula. Instead, the chunk sizes are based on the response time of the API itself. Changes to the chunk sizes are only performed when the last PUT used the 'full' chunk size, and it wasn't a repeated call attempting to recover due to a previous error or failure since such calls tend to respond very quickly. This ensures the most significant throughput possible and allows for any performance optimisation of the API to be realised. The console output reports the throughput and any chunk size changes. Chunk sizes are always at least 1.

The maximum number of users in a team is 32,767, and this sample script will ignore any users over this limit to avoid an error from the API.

The console log will report any users or roles that were expected to be added or removed but were not. For example, the console error will log users who don't exist in the team because they're not registered.

Teams

If the team does not exist, it will be created **without** a team folder, and the display name will match its name; otherwise, the existing team's display name is respected.

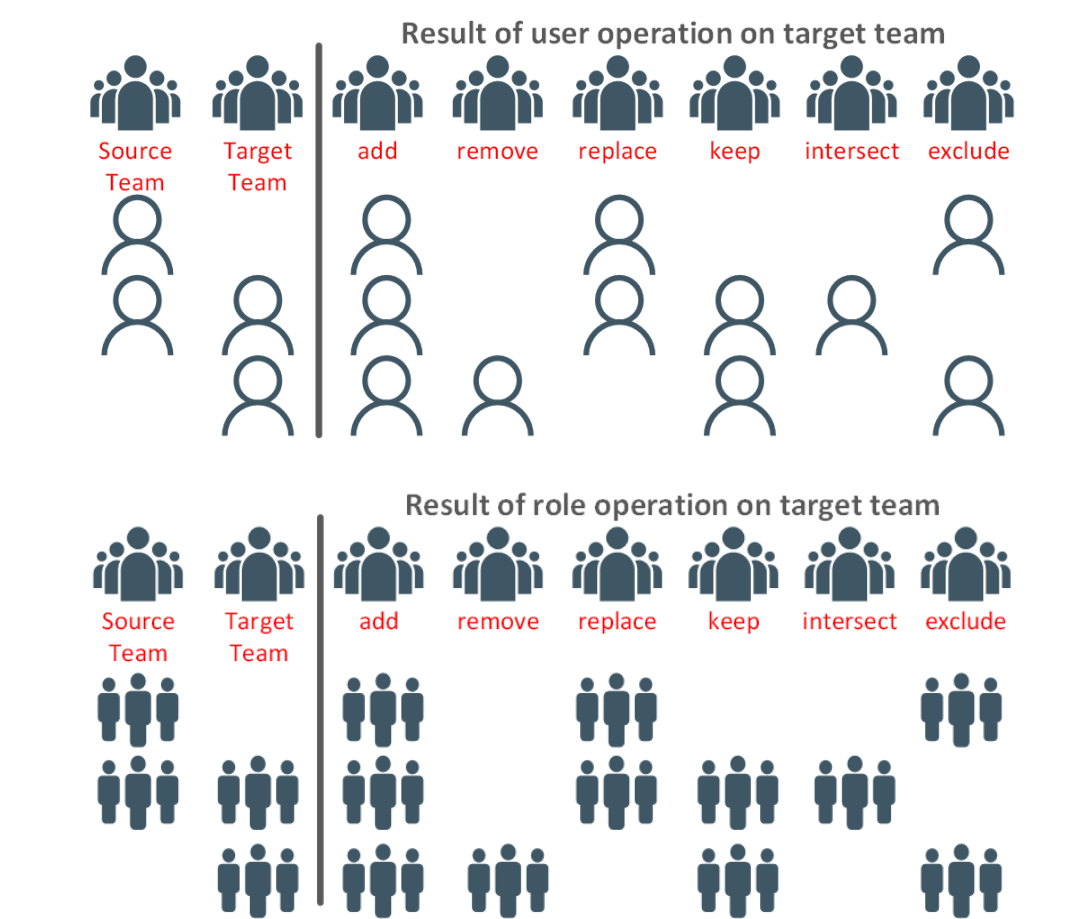
(When SAP Analytics Cloud is hosted on an SAP Data Centre (NEO), then teams are created **with** a team folder, and the display name for the team will have the text "with Team Folder". E.g. "Team1 with Team Folder". If you'd like the team folder description to share the same team display name, use sample script 1501 before running this sample script. Alternatively, update the team folder description manually.)

Operations (Oarrkie)

Since there are separate operations for ‘user actions’ and ‘role actions’ you can mix the actions as you please:

- For example, a ‘user_action’ of ‘remove’ and a ‘role_action’ of ‘keep’ will remove the users of one team from the users of another without changing the team’s role membership.
- Another example is a ‘user_action’ of ‘keep’ and a ‘role_action’ of ‘add’; this would add the source team’s role membership to the target team’s role membership without changing the user membership.

This sample script also provides additional operations of ‘intersect’ and ‘exclude’ that allow for ‘set’ like operations on teams. This will enable you to determine which users are in one team but not another. It also allows you to determine who isn’t in either team.



If both the user and role action is ‘keep’ and the target team doesn’t exist, the target team will be created without any user membership and won’t be a member of any roles.

Known script volume limitations	None
Script throughput	Throughput and performance will be very similar to sample script 1612. No specific information has been collected for this 1653 sample script.

1654-All_T-Uc-Uur-Oark-Transfer API Hidden Team To API Created Team

Sample	1654-All_T-Uc-Uur-Oark-Transfer API Hidden Team To API Created Team
Script Basis	All_T: All Teams
OAuth Access	'User Provisioning'
Basic description	<p>Uc: Transfers the users and roles of a team that can't be read by the API to another team, that can be read by the API.</p> <p>This sample script is now redundant thanks to a setting called 'Ignore Content Namespace For Teams'.</p>
Ideal for	<ol style="list-style-type: none"> 1. When SAP Analytics Cloud services hosted on Cloud Foundry rather than NEO. If you are unsure, you are almost certainly hosted on Cloud Foundry. 2. You have decided not to change the recommended setting of 'Ignore Content Namespace For Teams' from off to on, enabling all teams to be read and updatable by version 1 of the API. 3. Using version 1 of the API since version 2 of the API does not have the same restriction when updating teams. <p>And either points 4 or 5 below apply:</p> <ol style="list-style-type: none"> 4. When the business toggle IMPLEMENT_WORKAROUND_FOR_SCIM_GROUPS is off/disabled (as is the default, see SAP KBA 2857395): <ol style="list-style-type: none"> a. You have created teams manually or transported the team (using Content Network or Export/Import), and you'd like to transfer (or copy) the users from that team to another team, since the team can't be managed via the API. A GET /Groups/TeamID returns 404. b. You would like to prevent re-work by copying the users from the team (that can't be managed or read by the API) into a team that can be read and managed by the API. c. You are happy to reapply all permissions on folders and other content where the original team is specified in access rights (assuming you plan to use the 'new' team rather than the current team). d. You understand the original team, which can't be read by the API, remains unreadable by the API, and it is only the users and roles of the team that are copied to another team. 5. When the business toggle IMPLEMENT_WORKAROUND_FOR_SCIM_GROUPS is on/enabled (this needs to be requested via Product Support, see SAP KBA 2857395): <ol style="list-style-type: none"> a. You have created teams via the API <u>but only when</u> the toggle was <u>previously off/disabled</u>, and as such, these teams can <u>no longer</u> be managed via the API. A GET /Groups/TeamID returns 404. b. You would like to prevent re-work by copying the users from the team (that can't be managed or read by the API anymore) into a team that can be read and managed by the API. c. You are happy to reapply all permissions on folders and other content where the original team is specified in access rights (assuming you plan to use the 'new' team rather than the current team). d. You understand the original team, which can't be read by the API, remains unreadable by the API, and it is only the users and roles of the team that are copied to another team.

Not suitable when	<ul style="list-style-type: none">• The setting ‘Ignore Content Namespace For Teams’ is on.• The team can already be read by the API using the regular /Groups/TeamID endpoint.															
Notes	<p>The script reads all the users (and roles) from the team that can’t be read by version 1 of the API and adds them to another, different team, that can be read by version 1 of the API.</p> <p>For those that have:</p> <ul style="list-style-type: none">• <u>Not yet had</u> the business toggle IMPLEMENT_WORKAROUND_FOR_SCIM_GROUPS switched on/enabled• <u>AND</u> you have already created Teams via the API <p>then you can resolve this issue by changing the setting ‘Ignore Content Namespace For Teams’ to on. Please refer to the section in this guide on prerequisites for teams.</p> <p>If you wish to keep the setting ‘Ignore Content Namespace For Teams’ as off, then you avoid deleting any teams prior to the toggle is enabled/turned on (contrary to the description in SAP KBA mentioned). Instead, consider the following workflow, which could save re-work for those teams you’ve already created via the API:</p> <ol style="list-style-type: none">1. Have the toggle enabled/turned on (log a support incident to request this)<ul style="list-style-type: none">• <i>At this point, your manually created teams will be accessible via the API, but your teams created previously by the API will not be accessible.</i>2. Use this script to copy users+roles (from the previous API created team) to a new team<ul style="list-style-type: none">• <i>This script performs a little magic by reading teams that can’t be read via the API! However the API will now allow manually created teams to be updated.</i>3. Delete previous API created team manually.<ul style="list-style-type: none">• <i>You can only manage the previously API created teams, through the API, when the toggle is disabled/off.</i> <p>This will mean you can copy the team membership with this sample script, prior to deleting them, saving re-work.</p>															
Data file syntax	<p>F: .csv and .json (identical to script 1653, except the possible actions are fewer)</p> <table><tr><th>Field</th><th>Type</th><th>Description</th></tr><tr><td>file_source_team</td><td>string</td><td>Team name</td></tr><tr><td>file_target_team</td><td>string</td><td>Team name, changes are only made to the target</td></tr><tr><td>file_users_action</td><td>string</td><td>Defines the action to perform on the team’s membership. Possible values are: “add”, “replace”, “keep”</td></tr><tr><td>file_roles_action</td><td>string</td><td>Defines the action to perform on the team’s roles’ assignment. Possible values are: “add”, “replace”, “keep”</td></tr></table>	Field	Type	Description	file_source_team	string	Team name	file_target_team	string	Team name, changes are only made to the target	file_users_action	string	Defines the action to perform on the team’s membership. Possible values are: “add”, “replace”, “keep”	file_roles_action	string	Defines the action to perform on the team’s roles’ assignment. Possible values are: “add”, “replace”, “keep”
Field	Type	Description														
file_source_team	string	Team name														
file_target_team	string	Team name, changes are only made to the target														
file_users_action	string	Defines the action to perform on the team’s membership. Possible values are: “add”, “replace”, “keep”														
file_roles_action	string	Defines the action to perform on the team’s roles’ assignment. Possible values are: “add”, “replace”, “keep”														
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password)															
How the script works	<p>Some teams are not readable by version 1 of the API using the endpoint /Groups/TeamID, and such teams will return a 404 not found. However, the endpoint /Groups does return a complete list of all teams, including those teams that return a 404 using the /Groups/TeamID endpoint.</p> <p>This script reads all the teams via the /Groups endpoint to find the team that needs to be transferred to another. The users and roles are then extracted and added to the target team. (The /Groups endpoint only allows us to read teams; we can’t do other functions like update or delete with this endpoint.)</p>															

The `users_action` and `roles_action` can both be 'keep' (see later for 'operations') if you'd like to run the sample collection without making any actual changes to the target team (though any target team not found will be created). This may be helpful to demonstrate the correct team has been identified, as console logs will show the number of users in the source team once it finds the 'hidden' team.

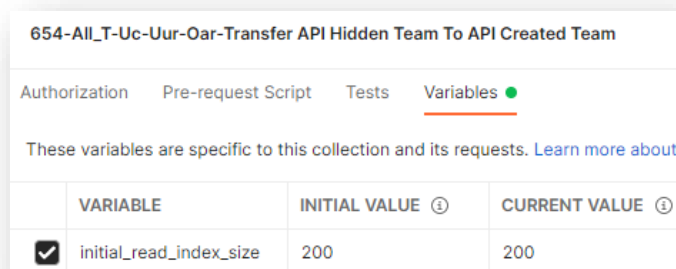
The `source_team` and `target_team` can share the same name! This is somewhat confusing, but each team has a (currently) hidden content namespace, which is different when created via the API than manually or via a transport. This means two teams sharing the same name can co-exist.

The script checks the `source_team` is unreadable, and it should return a 404 'not found' unless the `source_team` and `target_team` share the same name, in which case the team that is read must be the target team, and the script remembers this.

The target team is then read. A new team will be created if the target team does not exist. The team will be created **without** a team folder, and the display name will match its name; otherwise, the existing team's display name is respected.

(When SAP Analytics Cloud is hosted on an SAP Data Centre (NEO), then teams are created **with** a team folder, and the display name for the team will have the text "with Team Folder". E.g. "Team1 with Team Folder". If you'd like the team folder description to share the same team display name, use sample script 1501 before running this sample script. Alternatively, update the team folder description manually.)

The `/Groups` endpoint is then called. This returns all the teams, page by page, where each page has an 'index size' number of teams. The initial index size for each page is 200, which can be adjusted via the Collection Variables, but it's doubtful this is necessary. (it would only be required if the page request of 200 teams took longer than 5 minutes to complete!)



Authorization	Pre-request Script	Tests	Variables ●
These variables are specific to this collection and its requests. Learn more about			
	VARIABLE	INITIAL VALUE ⓘ	CURRENT VALUE ⓘ
<input checked="" type="checkbox"/>	initial_read_index_size	200	200

As each page of teams is returned, the index size is adjusted dynamically and programmatically depending upon the response time. If more than 1 page (i.e., over 200 teams) is returned, the index size will almost certainly be increased. However, the maximum index size set by SAP Analytics Cloud (currently 200) is identified by the script and subsequently set not to exceed this.

Once the source team has been identified, the target team is updated; if the 'user_action' and 'role_action' are different from 'keep', then the target team is updated.

The target team update behaves similarly to sample script 1653:

A comparison is made of the existing users and their roles' assignments. The net difference is then calculated to determine which users and roles should be added or removed.

Only the 'net' difference to the team is made. This means only the users/roles that need to be added or removed are added or removed. Users and roles are not just added or removed without intelligently inspecting the current assignment. This could mean no updates are performed against a team. In such cases, the 'UPDATE team' test will neither pass nor fail.

Users and roles are updated in 'chunks', meaning multiple PUT calls are made one after the other until all changes have been completed. The 'chunk' size is automatically determined by the existing number of users in the team. For an empty team, the chunk size will be 4590 users and 50 roles, otherwise, the chunk size is determined by these formulas (rounding when necessary):

User chunk size = (4590 - (number_of_users_in_the_team x 0.14))

Role chunk size = ((210 / (number_of_users_in_the_team x 46 / 210)) x 10)

After the initial PUT call has been made, subsequent chunk sizes are dynamically and automatically adjusted as users and roles are added and removed from the team and are no longer determined by the above formula. Instead, the chunk sizes are based on the response time of the API itself. Changes to the chunk sizes are only performed when the last PUT used the 'full' chunk size, and it wasn't a repeated call attempting to recover due to a previous error or failure since such calls tend to respond very quickly. This ensures the most significant throughput possible and allows for any performance optimisation of the API to be realised. The console output reports the throughput and any chunk size changes. Chunk sizes are always at least 1.

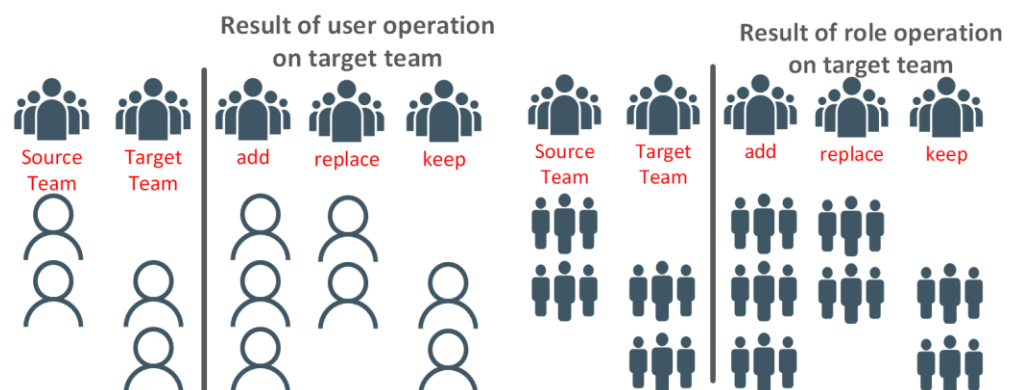
The maximum number of users in a team is 32,767, and this sample script will ignore any users over this limit to avoid an error from the API.

Operations (Oark)

Since there are separate operations for 'user actions' and 'role actions' you can mix the actions as you please:

- For example, a 'user_action' of 'add', and a 'role_action' of 'keep' will add the users of one team, from the users of another without changing the team's role membership.

The target team user and role assignments are best described by the following diagrams:



	<p>If both the user and role action is 'keep' and the target team doesn't exist, the target team will be created without any user membership, and it won't be a member of any roles.</p> <p>For feedback and Q&A on this particular topic and script, please visit https://blogs.sap.com/2021/11/03/making-manually-created-sap-analytics-cloud-teams-readable-by-the-scim-api/</p>
Known script volume limitations	None
Script throughput	No specific information has been collected for this 1654 sample script.

1665-All_U-Uc-Uu-Oarrieei-Fj-Es-AdminToolKit

Sample	1665-All_U-Uc-Uu-Oarrieei-Fj-Es-AdminToolKit
Script Basis	All_U: All Users
OAuth Access	'User Provisioning', ' Activities '
Basic description	<p>Uc: Scans:</p> <ul style="list-style-type: none"> All the users registered in SAP Analytics Cloud and adds/removes users to/from teams based upon criteria set in the data file. The script may also read <ul style="list-style-type: none"> the Activities log API to obtain the number of logins by users and their last login dates. (this requires Activities OAuth Access) The repository API to get resource metadata. It is a versatile script allowing "and" and "or" operators between 18 different types of tests. Example 'tests' include: 'user is not in a team', 'user not in a role', 'user has BI concurrent session', 'user not logged-in for X days', 'Users with private folder content', 'Users without public folder content' etc.
Ideal for	<ol style="list-style-type: none"> Identifying dormant users to run another sample script which deletes them. When things go wrong, you'd like to identify certain users by adding them to a team to update and process them with another sample script that are team-based, such as 145x sample scripts that update users, sample script 1653 that perform team-on-team operations or 195x that transport teams of users. To automate tasks, for example, to re-assign a set of users to a different manager or to process certain users before others to avoid errors. <p>For example, users who are managers must be transported before users that have a manager. Conversely, managers' users must be deleted after users with a manager.</p> <p>There are many use cases that this AdminToolKit enables on its own, and these are best described by looking at the preconfigured sample data files described below.</p> <p>The AdminToolKit also enables other use cases, best described in the 'Scenarios' section. The Scenarios provide preconfigured sample data files for various sample scripts to fulfil a use-case end-to-end.</p>
Not suitable when	For team-on-team operations. For example, if you want to add the users of one team to another, then use sample script 1653. This saves scanning the entire user population when a sample script performs team-on-team operations.
Notes	<ul style="list-style-type: none"> Unlike most samples, this requires the 'Activities' OAuth service access for some of the 'tests'. Please use the sample script '1002-Test-Es-Tests Single Environment Advanced Setup' to validate that the OAuth client has the necessary access. Generally, the easiest and quickest way to use this sample script is to use the preconfigured sample data files provided. These are described below. Many of these sample data files are designed to be used with other sample scripts. As mentioned, the 'Scenarios' section of this document provides preconfigured example data files to match a variety of use cases or scenarios that these scripts don't fulfil on their own. Please refer to the Scenarios section for more details. For an overview on managing dormant users please visit this blog.

Data file syntax F: .json

Field (one word, no spaces)	Type, Mandatory / Optional (default)	Description
file_team	String, mandatory	Team name
file_team_displayname	string, mandatory	Team Display name (rather than Team Folder description). This name can contain the text 'TIMESTAMP' which is replaced with the actual date/time the script is run.
file_users_action	string, mandatory	Defines the action to perform on the team membership of users. Possible values are: "add", "remove", "replace", "intersect", "exclude", "excludeall" and "invert".
file_JSON_users_to_exclude	String, optional ([])	JSON array of keys "value" and string values of users. Example: [{"value": "SAP_SUPPORTXXXXXXXX"}, {"value": "MATTHEW2"}] When used, the user(s) are excluded from all tests, except if they are identified by 'file_action_users_that_are_managers'
file_multiple_action_users_operator_is_AND	Boolean, optional (false)	When true : then, for all the tests that are enabled (set to true), all* the tests must be met for the user to be positively identified. * however, the test 'file_action_users_that_are_managers' is always OR'ed with all other tests, even when this value is true. When false : then, for all the tests that are enabled (set to true) any one of the tests must be true for the user to be positively identified, even if some tests for the user are not met. It performs an 'OR' operation.
file_action_users_match_all_users	Boolean, optional (false)	When true : all users are positively identified. When false : the test is ignored.
file_action_users_with_bi_concurrent_license	Boolean, optional (false)	When true : users are positively identified when they have a Business Intelligence concurrent session license. When false : the test is ignored.
file_action_users_with_bi_named_user_license	Boolean, optional (false)	When true : users are positively identified when they have a Business Intelligence named user license. When false : the test is ignored.
file_action_users_with_a_manager	Boolean, optional (false)	When true : users are positively identified when they have a manager. When false : the test is ignored.
file_action_users_with_named_managerids	Boolean, optional (false)	When true : users are positively identified when they have a manager and the managerid is one of the users listed in file_JSON_named_managerids When false : the test is ignored.
file_JSON_named_managerids	JSON values of users, optional ([])	Applicable only when file_action_users_with_named_managerids is true : JSON array of keys "value" and string values of users. Example: [{"value": "MANAGERID1"}, {"value": "MANAGERID2"}]
file_action_users_without_manager	Boolean, optional (false)	When true : users are positively identified when they have no manager. When false : the test is ignored.

file_action_users_that_are_managers	Boolean, optional (false)	When true : users are positively identified when they are a manager of another user. When false : the test is ignored.
file_action_users_active	Boolean, optional (false)	When true : users are positively identified when they have an Active property of true, which means the user is 'enabled'. When false : the test is ignored.
file_action_users_not_active	Boolean, optional (false)	When true : users are positively identified when they have an Active property of false, which means the user is 'disabled'. When false : the test is ignored.
file_action_users_with_userid_ending_underscore_1	Boolean, optional (false)	When true : users are positively identified when they have a user id which ends _1 or _2. (if using SAML SSO on e-mail – see notes above) When false : the test is ignored.
file_action_users_with_email_domain_matching	Boolean, optional (false)	When true : users are positively identified when the user has an e-mail with the @domain matching the domain as specified by file_email_domain When false : the test is ignored.
file_action_users_with_email_domain_not_matching	Boolean, optional (false)	When true : users are positively identified when the user has an email with the @domain different from the domain as specified by file_email_domain When false : the test is ignored.
file_email_domain	String, optional (sap.com)	Applicable only when file_action_users_with_email_domain_matching is true or file_action_users_with_email_domain_not_matching is true : The e-mail domain without the @, for example 'sap.com'
file_action_users_with_a_role_or_a_team	Boolean, optional (false)	When true : users are positively identified when the user has any role or any team (except the teams listed in file_JSON_teams_to_exclude or team names that start with the values listed in file_JSON_teamnames_starts_with_to_exclude, or roles that are listed in file_JSON_roles_to_exclude) When false : the test is ignored.
file_action_users_with_no_teams	Boolean, optional (false)	When true : users are positively identified when the user is not a member of any team (except the teams listed in file_JSON_teams_to_exclude or team names that start with the values listed in file_JSON_teamnames_starts_with_to_exclude) When false : the test is ignored.
file_JSON_teams_to_exclude	JSON values of teams, optional ([])	Applicable only when file_action_users_with_a_role_or_a_team is true or file_action_users_with_no_teams is true : JSON array of keys "value" and string values of teams. Example: [{"value": "Team1"}, {"value": "Team2"}]
file_JSON_teamnames_starts_with_to_exclude	JSON values of teams, optional ([])	JSON array of keys "value" and string values of teams. Example: [{"value": "AdminToolKit_"}]
file_action_users_with_no_roles	Boolean, optional (false)	When true : users are positively identified when the user is not directly assigned any role (except the roles listed in file_JSON_roles_to_exclude) When false : the test is ignored.
file_JSON_roles_to_exclude	JSON values of roles	Applicable only when file_action_users_with_a_role_or_a_team is true or file_action_users_with_no_roles is true : JSON array of keys "value" and string values

		of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]
file_action_users_with_all_named_roles	Boolean, optional (false)	When true : users are positively identified when they have a role directly assigned to them, listed in file_JSON_named_roles When false : the test is ignored.
file_JSON_named_roles	JSON values of roles, optional ([])	Applicable only when file_action_users_with_all_named_roles is true : JSON array of keys "value" and string values of roles. Roles with the same content namespace as the environment don't need to specify the "PROFILE:namespace:". Example: [{"value":"Role1"}, {"value":"PROFILE:sap.epm:BI_Admin"}]
file_actions_users_with_LangDateTimeNumberFormats_different_from	Boolean #1, optional (false)	When true : users are positively identified when the user's properties do not match ALL the setting of preferredlanguage, dataaccesslanguage, dateformatting, timeformatting and numberformatting When false : the test is ignored.
file_preferredlanguage	String, optional unless #1 is true	Applicable only when file_actions_users_with_LangDateTimeNumberFormats_different_from is true (same for the next entries listed below): possible values: 'ar', 'bg', 'zh-CN', 'zh-TW', 'hr', 'ca', 'cs', 'cy', 'nl', 'da', 'en', 'en-GB', 'fi', 'fr', 'fr-CA', 'de', 'de-CH', 'el', 'he', 'hi', 'hu', 'id', 'it', 'ja', 'ko', 'ms', 'no', 'pl', 'pt', 'pt-PT', 'ro', 'ru', 'sh', 'sk', 'sl', 'es', 'es-MX', 'sv', 'th', 'tr', 'uk', 'vi', 'en-US-sapstd' and possibly more will be added over time!
file_dataaccesslanguage		possible values: 'zz' (default), 'af', 'ar', 'bg', ... many more as shown above in preferredlanguage.
file_dateformatting		possible values include: 'MMM d, yyyy', 'dd/MM/yyyy', 'MM.dd.yyyy', ... many more!
file_timeformatting		possible values include: 'H:mm:ss' (24H), 'h:mm:ss A' (12H)
file_numberformatting		possible values include: '1,234.56', '1.234,56', '1 234,56'
file_action_users_created_recently	Boolean, optional (false)	When true : users are positively identified when created less than or equal to the number of days specified by file_users_created_recently_in_days. When false : the test is ignored.
file_users_created_recently_in_days	Number, optional (7)	Applicable only when file_action_users_created_recently is true ; A value of 1 will identify all users created in the last 24 hours (or one day).
file_action_users_created_more_than_days_ago	Boolean, optional (false)	When true : users are positively identified when created greater than (equal to will not match) the number of days specified by file_users_created_more_than_days. When false : the test is ignored.
file_users_created_more_than_days	Number, optional (30)	Applicable only when file_action_users_created_more_than_days_ago is true ; A value of 1 will identify all users created at least 24 hours ago (or one day).
file_action_users_with_most_recent_login_at_least_days_ago	Boolean, optional (false)	When true : users are positively identified when their most recent login was greater than or equal to the number of days specified by file_users_with_most_recent_login_at_least_days.

		When false : the test is ignored.
file_users_with_with_most_recent_login_at_least_days	Number, optional (30)	Applicable only when file_action_users_with_most_recent_login_at_least_days_ago is true ; A value of 1 will identify all users with the most recent login at least 24 hours ago (or one day), for example, users whose last login was a week ago.
file_action_users_with_most_recent_login_within_last_days	Boolean, optional (false)	When true : users are positively identified when their most recent login was less than (equal to will not match) number of days specified by file_users_with_with_most_recent_login_within_last_days. When false : the test is ignored.
file_users_with_with_most_recent_login_within_last_days	Number, optional (30)	Applicable only when file_action_users_with_most_recent_login_within_last_days is true ; A value of 1 will identify all users with the most recent login within the last 24 hours (or one day), for example, users whose last login was within the last few hours.
file_action_users_with_fewer_logins_than	Boolean, optional (false)	When true : users are positively identified when their total number of logins was less than or equal to the number of days specified by file_users_with_fewer_logins_than. Typically and strongly recommended to be used with file_action_users_with_fewer_logins_within_last_days to limit the time to which the number of logins is counted and also to ensure the activities log has at least log entries of that age. When false : the test is ignored.
file_users_with_fewer_logins_than	Number, optional (5)	Applicable only when file_action_users_with_fewer_logins_than is true ; A value of 0 will identify all users that have no login event (login events are recorded in the activities log, which can be purged)
file_action_users_with_greater_logins_than	Boolean, optional (false)	When true : users are positively identified when their The total number of logins was greater than (equal to will not match) the number of days specified by file_users_with_greater_logins_than. Typically and strongly recommended to be used with file_action_users_with_greater_logins_within_last_days to limit the time to which the number of logins is counted and also to ensure the activities log has at least log entries of that age. When false : the test is ignored.
file_users_with_greater_logins_than	Number, optional (5)	Applicable only when file_action_users_with_greater_logins_than is true ; A value of 0 will identify all users that have at least 1 login event (login events are recorded in the activities log, which can be purged)
file_action_users_with_fewer_logins_within_last_days	Boolean, optional (false)	Applicable only when file_action_users_with_fewer_logins_than is true , and when true : Limits the logins counted to login events that occurred within the last number of days as specified by file_users_with_logins_within_last_days

	<code>file_action_users_with_greater_logins_within_last_days</code>	Boolean, optional (false)	Applicable only when <code>file_action_users_with_greater_logins_than</code> is true , and when true : Limits the logins counted to login events that occurred within the last number of days as specified by <code>file_users_with_logins_within_last_days</code>
	<code>file_users_with_logins_within_last_days</code>	Number, optional (60)	Applicable only when <code>file_action_users_with_fewer_logins_within_last_days</code> is both applicable and true OR when <code>file_action_users_with_greater_logins_within_last_days</code> is both applicable and true A value of 60 would limit the login count to only login events within the last 60 days. If the activities log does not have 60 days' records, the script will identify this and abort. It means should the activities log be purged, users will not be incorrectly identified, since a purged log will not hold any login events and thus could identify users incorrectly as any login events could not be identified.
	<code>file_action_users_with_private_folder_content</code>	Boolean, optional (false)	When true : users are positively identified when their private folder content has any of the following artefacts: Stories, Applications, Digital Boardrooms, Templates, Insights and Folders. It will not identify if the folder has any other types of artefacts, including models, datasets, etc. When false : the test is ignored.
	<code>file_action_users_without_private_folder_content</code>	Boolean, optional (false)	When true : users are positively identified when their private folder content does not have the following artefacts: Stories, Applications, Digital Boardrooms, Templates, Insights and Folders. It will still identify if the folder has any other types of artefacts, including models, datasets, etc. When false : the test is ignored.
	<code>file_action_users_that_created_public_content</code>	Boolean, optional (false)	When true : users are positively identified if they created content that is stored in the public folders and that content remains in the public folders. When false : the test is ignored.
	<code>file_action_users_that_did_not_create_public_content</code>	Boolean, optional (false)	When true : users are positively identified if they did not create content that is stored in a public folder or any content they created no longer remains in the public folder. When false : the test is ignored.
Environment	Single Service (SACserviceFQDN, SACplatform, SACTokenFQDN, Username, Password, TimeZoneHours, TimeZoneMinutes, TimeZoneDescription, SAMLSSO)		
How the script works	For each entry in the file, the team is read from SAP Analytics Cloud to determine the current membership of users. If the team does not exist, it will be created without a team folder, and the display name will match its name; otherwise, the existing team's display name is respected.		

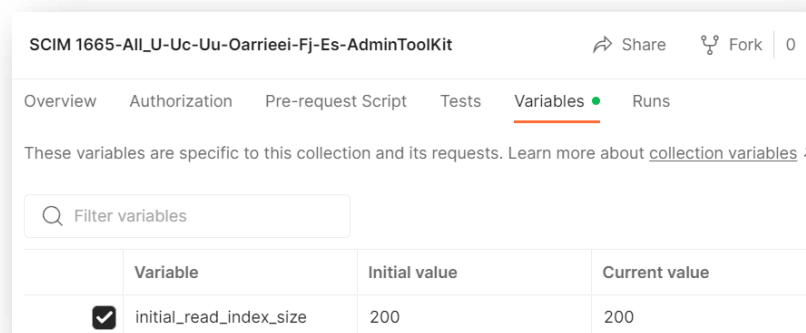
(When SAP Analytics Cloud is hosted on an SAP Data Centre (NEO), then teams are created **with** a team folder, and the display name for the team will have the text “with Team Folder”. E.g. “Team1 with Team Folder”. If you’d like the team folder description to share the same team display name, use sample script 1501 before running this sample script. Alternatively, update the team folder description manually.)

If any of the tests include a check related to the last login or the number of logins, then a request is made to the Activities log API to find the oldest log entry. If the `file_users_with_logins_within_last_days` is applicable, the script will abort if the oldest entry isn’t old enough. Otherwise, the script will continue and download all the Activities Logs without any limit on the time to ensure the ‘last’ login date is found. The download, though, is limited to only ‘Login’ events, and multiple pages of activities log will be made if necessary as there is a limit of 100,000. The logs are downloaded in a time ascending order, so the logic within the script is more straightforward to determine the number of logins since a given date, for example. For more details about downloading the activities log and best practices, please visit <https://blogs.sap.com/2023/01/18/sap-analytics-cloud-activities-log-command-line-interface-cli-to-automate-downloads-associated-best-practices/>

If any tests require a private or public folder content check, a single API request to the repositories API is made. `/api/v1/Repositories`.

This API endpoint is not documented, and its official support is therefore unclear. However, the official API is `/v1/Resources` ([API](#), [doc](#)), which is almost identical but cannot return folders as a resource. It means the `/api/v1/Repositories` can list folders, which is why it is used. If in doubt, please use the official API to ensure proper support from SAP.

Next, all the users registered in SAP Analytics Cloud are read page by page, each with an ‘index size’ number of users. The initial index size is 200, and this can be adjusted via the Collection Variables, but it’s doubtful this is necessary. (it would only be required if the page request of 200 users took longer than 5 minutes to complete!)



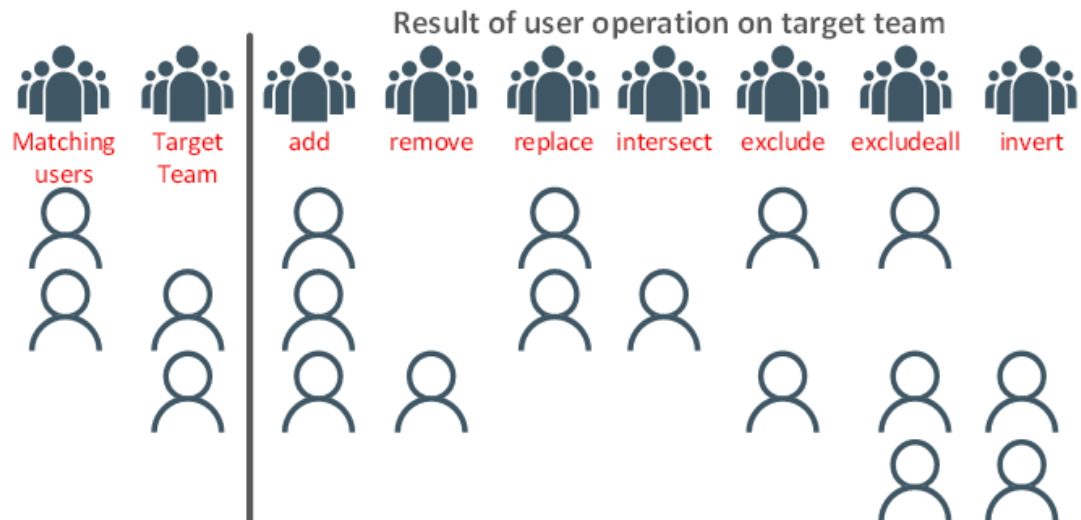
Variable	Initial value	Current value
<input checked="" type="checkbox"/> initial_read_index_size	200	200

As each page of users is returned, the index size is adjusted dynamically and programmatically depending on the response time. If more than 1 page (i.e., over 200 users) is returned, the index size will almost certainly be increased. However, the maximum index size set by SAP Analytics Cloud (currently 200) is identified by the script and subsequently set not to exceed this.

As each page of users is read, every user is processed and tested against all the 'tests' that are enabled (set to 'true'). The user is then either positively identified or not.

Operations (Oarrieei)

If the user is positively identified as meeting the required test(s), then the user is a 'matching' user, as shown in the diagram below:



Depending upon the `file_users_action` and the existing team membership (shown above as the 'target team'), the user is either added or removed to or from the team as appropriate.

Because all the users of the entire service are read, additional actions are available: 'exclude all' and 'invert'. However, unlike some other sample scripts, this sample script has no 'keep' action.

It's worth pointing out that this sample script can add a user to a team, even when they don't match any tests. For example, 'invert' is the opposite of the matching users. Here, a user will be added when they don't match any tests, and those that do match the tests will be removed. This invert option can be beneficial if you want the opposite of any test.

All users are processed except for users listed in `file_JSON_users_to_exclude`. This is to allow special users to be ignored, and typically, the `SAP_SUPPORT` user will fall into this category, which is why this is provided. However, any users listed here may still be 'matched' with the test `file_action_users_that_are_managers`, but only if they are managers, of course!

If more than one 'test' is enabled (set to true), then a user is considered 'matched' depending upon the `file_multiple_action_users_operator_is_AND` setting:

If `file_multiple_action_users_operator_is_AND` is **false**, then the user must pass any one of the enabled tests (effective 'OR' operation between the tests).

If `file_multiple_action_users_operator_is_AND` is **true**, then the user must pass ALL the enabled tests (effective 'AND' operation between the tests). However, an 'OR' operation is always with the `file_action_users_that_are_managers` test (if enabled).

For example, if `file_action_users_with_bi_concurrent_license` is **true** and `file_action_users_with_a_manager` is **true**:

Then, if `file_multiple_action_users_operator_is_AND` is **true** the user must have both a BI Concurrent License AND the same user must also have a manager. The user is then considered positively identified and 'matched'.

Then, if `file_multiple_action_users_operator_is_AND` is **false** the user must have either a BI Concurrent License OR the same user must also have a manager. The user is then considered positively identified and 'matched'.

For example, if `file_action_users_with_bi_concurrent_license` is **true** and `file_action_users_with_a_manager` is **true** and `file_action_users_that_are_managers` is **true**:

Then, if `file_multiple_action_users_operator_is_AND` is **true** the user must have (both a BI Concurrent License AND the same user must also have a manager) OR they must be a manager (regardless of their license type or if they have a manager). The user is then considered positively identified and 'matched'.

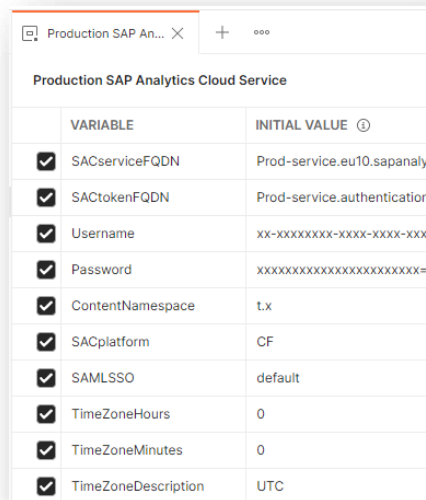
To combine `file_action_users_that_are_managers` as an 'AND' with other tests, please see sample data file "x665 sample - 40 - complex 3-5 - a - AdminToolKit_Users_That_Are_Managers_With_BI_Concurrent_Setting", which "intersect" two entries.

With the `file_action_users_with_a_manager` enabled, managers are processed at the end of reading all the users, and the console log will report the number of users added/removed due to this post-processing.

At the end of reading all the users and after any post-processing, the team may need to be updated.

The team is updated if the `display_name` contains the text 'TIMESTAMP', even if no users need to be added or removed. This is to make it more apparent when the team was last updated, even though SAP Analytics Cloud provides a 'last changed' column in the interface.

If the `display_name` contains the text 'TIMESTAMP', the text is replaced by the current date and time with a timezone description. The timezone is designed to be easily updated by changing the environment variables: `TimeZoneHours`, `TimeZoneMinutes` and `TimeZoneDescription`:



	VARIABLE	INITIAL VALUE ⓘ
<input checked="" type="checkbox"/>	SACserviceFQDN	Prod-service.eu10.sapanalytic
<input checked="" type="checkbox"/>	SACtokenFQDN	Prod-service.authentication.e
<input checked="" type="checkbox"/>	Username	xx-xxxxxxxx-xxxx-xxxx-xxxx-
<input checked="" type="checkbox"/>	Password	xxxxxxxxxxxxxxxxxxxxxxxxxx=
<input checked="" type="checkbox"/>	ContentNamespace	t.x
<input checked="" type="checkbox"/>	SACplatform	CF
<input checked="" type="checkbox"/>	SAMLSSO	default
<input checked="" type="checkbox"/>	TimeZoneHours	0
<input checked="" type="checkbox"/>	TimeZoneMinutes	0
<input checked="" type="checkbox"/>	TimeZoneDescription	UTC

TimeZoneHours should be an integer from -23 to 23.

TimeZoneHours should be -30, 0, or 30

And TimeZoneDescription can be any text that best describes your timezone.

Assuming the team needs to be updated, and this will always be the case if users need to be added or removed, then the team update behaves similarly to sample script 1653 and is described below. Any roles the team is a member of are always respected.

If the team is to be updated, the team displayname is always updated with the value of file_team_displayname (with TIMESTAMP being appropriately substituted). Any team folder description is not updated.

A comparison is made of the existing user's assignment. The net difference is calculated to determine which users and roles should be added or removed.

Only the 'net' difference to the team is made. This means only the users/roles that need to be added or removed are added or removed. Users are not just added or removed without intelligently inspecting the current assignment. This could mean no updates are performed against a team. In such cases, the 'UPDATE team' test will neither pass nor fail.

Users are updated in 'chunks', meaning multiple PUT calls are made one after the other until all changes have been completed. The 'chunk' size is automatically determined by the existing number of users in the team. For an empty team, the chunk size will be 4590 users; otherwise, the chunk size is determined by this formula (rounding when necessary):

$$\text{User chunk size} = (4590 - (\text{number_of_users_in_the_team} \times 0.14))$$

After the initial PUT request has been made, subsequent chunk sizes are dynamically and automatically adjusted as users are added and removed from the team and are no longer determined by the above formula. Instead, the chunk sizes are based on the response time of the API itself. Changes to the chunk sizes are only performed when the last PUT used the 'full' chunk size, and it wasn't a repeated call attempting to recover due to a previous error or failure since such calls tend to respond very quickly. This ensures the most significant throughput possible and allows for any performance optimisation of the API to be realised. The console output reports the throughput and any chunk size changes. Chunk sizes are always at least 1.

	<p>The maximum number of users in a team is 32,767, and this sample script will ignore any users over this limit to avoid an error from the API.</p>
Known script volume limitations	<p>None yet identified.</p> <p>The most significant influence of stability is the number of users in the Service. The larger the number of users, the more likely instability issues.</p> <p>The next most likely instability would occur when processing 'file_action_users_that_are_managers' is true, and with the action is either 'invert' or 'excludeall'. This is because of the extra memory required for post-processing. This is then followed by when 'file_action_users_that_are_managers' is true, regardless of the action setting.</p>
Script throughput	<p>Not calculated.</p> <p>However, a direct linear correlation should be expected as the total number of users increases.</p>
Example data files	<p>The sample data files provide many possible use cases or scenarios and are likely to meet your requirements with only minor tweaks needed.</p> <p>The name of each sample script describes its use-case.</p> <p>x665 sample - 1- Master_template.json x665 sample - 2 - AdminToolKit_All_Users.json x665 sample - 3 - AdminToolKit_Users_With_BI_Concurrent_Setting.json x665 sample - 4 - AdminToolKit_Users_With_Named_User_License.json x665 sample - 5 - AdminToolKit_Users_With_A_Manager.json x665 sample - 6 - AdminToolKit_Users_With_Manager_MANAGERID.json x665 sample - 7 - AdminToolKit_Users_Without_A_Manager.json x665 sample - 8 - AdminToolKit_Users_That_Are_Managers.json x665 sample - 9 - AdminToolKit_Users_That_Are_Not_Managers.json x665 sample - 10 - AdminToolKit_Users_That_Are_Activated.json x665 sample - 11 - AdminToolKit_Users_That_Are_Deactivated.json x665 sample - 12 - AdminToolKit_Users_With_UserID_Ending__1.json x665 sample - 13 - AdminToolKit_Users_With_SAP_email.json x665 sample - 14 - AdminToolKit_Users_Without_SAP_email.json x665 sample - 15 - AdminToolKit_Users_Without_a_Role_and_Without_a_Team.json x665 sample - 16 - AdminToolKit_Users_Without_a_Team.json x665 sample - 17 - AdminToolKit_Users_Without_a_Role.json x665 sample - 18 - AdminToolKit_Users_With_Directly_Assigned_BIAdmin_Role.json x665 sample - 19 - AdminToolKit_Users_With_Default_Settings.json x665 sample - 20 - AdminToolKit_Users_Created_Over_A_Year_Ago.json x665 sample - 21 - AdminToolKit_Users_Created_Recently.json x665 sample - 22 - AdminToolKit_Users_With_A_Recent_Login.json x665 sample - 23 - AdminToolKit_Users_Without_A_Recent_Login.json x665 sample - 24 - AdminToolKit_Users_With_Few_Logins_Since_Oldest_Log_Entry.json x665 sample - 25 - AdminToolKit_Users_With_Few_Logins_This_Last_Month.json x665 sample - 26 - AdminToolKit_Users_Without_A_Recent_Login_And_Fewer_Than_3_Logins_Last_3_Months.json x665 sample - 27 - AdminToolKit_Users_With_100_Or_More_Logins_Since_Oldest_Log_Entry.json x665 sample - 28 - AdminToolKit_Users_With_100_Or_More_Logins_Within_Last_3_Months.json x665 sample - 29 - AdminToolKit_Users_With_Private_Folder_Content.json x665 sample - 30 - AdminToolKit_Users_Without_Private_Folder_Content.json x665 sample - 31 - AdminToolKit_Users_That_Created_Public_Folder_Content.json x665 sample - 32 - AdminToolKit_Users_That_Did_Not_Create_Public_Folder_Content.json x665 sample - 33 - AdminToolKit_Users_Dormant_A_Created_3_Months_No_Recent_Login_Fewer_Than_3_Logins_Last_3_Months.json x665 sample - 34 - AdminToolKit_Users_Dormant_B_Dormant_A_Plus_No_Private_Folder_Content.json x665 sample - 35 - AdminToolKit_Users_Dormant_C_Dormant_B_Plus_No_Public_Folder_Content.json x665 sample - 36 - AdminToolKit_Users_Dormant_D_Dormant_C_Plus_Are_Not_Managers.json x665 sample - 37 - AdminToolKit_Users_Dormant_E_Dormant_D_Plus_Are_Activated.json x665 sample - 38 - AdminToolKit_Users_Dormant_F_Dormant_E_Plus_Are_Named_Licensed.json x665 sample - 39 - All samples 2 to 38.json x665 sample - 40 - complex 3-5 - a - AdminToolKit_Users_That_Are_Managers_With_BI_Concurrent_Setting.json x665 sample - 41 - complex 5-8 - a - AdminToolKit_Users_In_Managment_Structure.json</p>

x665 sample - 42 - complex 5-8 - b - AdminToolKit_Users_Not_In_Managment_Structure.json
 x665 sample - 43 - complex 19 - a - AdminToolKit_Users_With_Default_Settings and AdminToolKit_Users_Without_Default_Settings.json
 x665 sample - 44 - complex 19 - b - AdminToolKit_Users_With_American_Settings and AdminToolKit_Users_Without_American_Settings.json
 x665 sample - 45 - complex 19 - c - AdminToolKit_Users_With_British_Settings and AdminToolKit_Users_Without_British_Settings.json
 x665 sample - 46 - complex 19 - d - AdminToolKit_Users_With_French_Settings and AdminToolKit_Users_Without_French_Settings.json
 x665 sample - 47 - complex 19 - e - AdminToolKit_Users_With_German_Settings and AdminToolKit_Users_Without_German_Settings.json

Your initial changes will be to search and replace the following in all the data files:

- Replace **SAP_SUPPORTXXXXXXXXXX** with the SAP SUPPORT User in your service
- Replace **sap.com** with your organisation's e-mail domain name

Additionally:

- Change the `file_team` name for sample data files 13 and 14 to match your organisation name unless you work for SAP!

Introducing 'scenarios' that support additional use-cases

As mentioned in the notes above, many of these sample data files are designed to be used with other sample scripts. The 'Scenarios' section of this document provides preconfigured example data files to match a variety of use cases or scenarios that these scripts don't fulfil on their own. Please refer to the Scenarios section for more details.

Dormant users

Six scenarios (A to F) explain the sample data files for Dormant Users (33 to 38). Please refer to the scenarios section for an in-depth explanation.

1665 sample data files – a little clarification:

A few of the sample data files require a bit further clarification:

12 - Users with userid ending _1

Users with a userid ending _1 (or _2) can sometimes be problematic as they are not always the intended userid but rather a user that was created via dynamic user creation or via the SCIM API when the user's e-mail address has changed, but the request to CREATE a user is made. A new (duplicate) user is then created with _1. Users with a _1 often result from an incorrectly managed user update.

Such users are worth identifying so they can be appropriately managed (stay tuned for a new sample script that transfers the new e-mail to the original user and then deletes this duplicate user!). Users with a _1 aren't necessarily problematic. Still, they can introduce issues of inconsistency in the landscape. Row-level data security for acquired models can be controlled based on this ID, which might make this task more complicated than needed).

16 - Users without a team

A very common use case is to identify users who aren't a member of any team and then add them to a team. The sample data file:

x665 sample - 16 - AdminToolKit_Users_Without_a_Team.json

Provides the settings preconfigured. All you need to do is change the entry:


`"file_team": "AdminToolKit_Users_Without_a_Team",`

to be the team you'd like to add the user into rather than the team called


`"AdminToolKit_Users_Without_a_Team"`

	<p>41 - Users in Management Structure x665 sample - 41 - complex 5-8 - a - AdminToolKit_Users_In_Managment_Structure.json Provides a team of all users that are either managers or have a manager.</p> <p>24 - Users not in Management Structure x665 sample - 42 - complex 5-8 - b - AdminToolKit_Users_Not_In_Managment_Structure.json Provides a team of all users that are neither managers nor have a manager.</p> <p>The sample data files for script 1801/1802, which deletes teams, also contain samples to delete all the teams created by the sample data files for this sample script.</p>

1801-T-D-Dt-Fcj-Es-Delete teams (ok to timeout)


Sample	1801-T-D-Dt-Fcj-Es-Delete teams (ok to timeout)		
Script Basis	T: Team		
OAuth Access	'User Provisioning'		
Basic description	Dt: Deletes teams		
Ideal for	Deleting teams without deleting any users of the team and regardless of the team size.		
Not suitable when	Deleting users of the team. Use sample script 1851 to delete users of the team.		
Notes	<p>If deleting a team times-out, it will still be deleted.</p> <p>Script 1802 could be used to empty the team first; however, it could take longer to complete. The only benefit of script 1802 is a 'clean' delete; the result is the same as this script.</p>		
Data file syntax	F: .csv and .json		
	Field	Type	Description
	File_delete_team	string	Team name
Environment	Single Service (SACserviceFQDN, SACplatform, SAcTokenFQDN, Username, Password)		
How the script works	<p>Reads the team, and if the team exists, it will delete it.</p> <p>If the team doesn't exist, the 'READ team' test will fail.</p> <p>The duration of the deletion operation is dependent on the number of users in the team. If the operation times-out, which will typically only occur with a team size of more than 12,000 users, the script will check the team has been deleted, and the test 'TEAM DELETE was successful' will pass.</p>		
Known script volume limitations	None		
Script throughput		Service with 500 users	Service with 80,000 users
	Deleting empty team	0.45 teams / sec 2.22 secs / team (7.1%)	0.27 teams / sec 3.64 secs / team (4.3%)
Video tutorial	 SAP HANA Academy provide a step-by-step tutorial for sample script 1801 and 1802 https://youtu.be/ptDDM_W5eNc (Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)		

1802-T-D-Dt-Fcj-Es-Delete teams (empty first)

Sample	1802-T-D-Dt-Fcj-Es-Delete teams (empty first)								
Script Basis	T: Team								
OAuth Access	'User Provisioning'								
Basic description	Dt: Deletes teams								
Ideal for	<ul style="list-style-type: none">Deleting teams without deleting any users of the teamWhen the team size is over 10,000 usersAnd a 'clean' delete is desired rather than allowing a time-out								
Not suitable when	Deleting users of the team. Use sample script 1851 to delete users of the team.								
Notes	It's generally better to use script 1801, as emptying the team first could take longer. The only benefit of script 1802 is a 'clean' delete; the result is the same as this script.								
Data file syntax	F: .csv and .json <table><tr><td>Field</td><td>Type</td><td>Description</td></tr><tr><td>File_delete_team</td><td>string</td><td>Team name</td></tr></table>			Field	Type	Description	File_delete_team	string	Team name
Field	Type	Description							
File_delete_team	string	Team name							
Environment	Single Service (SACserviceFQDN, SACplatform, SAcTokenFQDN, Username, Password)								
How the script works	<p>Reads the team, and if the team exists, it will remove its users and its membership of roles before deleting the team.</p> <p>The 'READ team' test will fail if the team doesn't exist.</p> <p>The script to empty the team behaves the same way as script 1612 with a 'replace' action for users and roles with an empty array [] for users and roles. See script 1612 for details on how the 'empty' action is performed.</p> <p>Contents of any team folder remain untouched.</p>								
Known script volume limitations	None								
Script throughput		Service with 500 users	Service with 80,000 users						
	Deleting empty team	0.45 teams / sec 2.22 secs / team (7.1%)	0.27 teams / sec 3.64 secs / team (4.3%)						
	Deleting team of 500 users	0.09 teams / sec 10.56 secs / team 9.47 users / sec 0.11 secs / user (7.1%)	0.06 teams / sec 15.9 secs / team 6.30 users / sec 0.16 secs / user (1.1%)						
Video tutorial		SAP HANA Academy provide a step-by-step tutorial for sample script 1801 and 1802 https://youtu.be/ptDDM_W5eNc (Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)							

1851-T-D-Dtu-Fcj-Es-Delete teams & users in the team

Sample	1851-T-D-Dtu-Fcj-Es-Delete teams & users in the team													
Script Basis	T: Team													
OAuth Access	'User Provisioning'													
Basic description	Dt: Deletes teams and/or users of the team													
Ideal for	<ul style="list-style-type: none"> Deleting teams on their own without deleting any users of the team. Deleting all users of a team, without deleting the team itself. Deleting both all the users of the team and deleting the team itself. 													
Not suitable when														
Notes	<p>This provides the same functions as scripts 1301 and 1801 combined. i.e. compared to script 1801, this enables the team users to be deleted (1301) and the team itself (1801).</p> <p>Contents of any team folder remain untouched.</p> <p>Consider "Scenario D01 - Delete users then delete managers" in the scenarios section to perform a 'clean' delete of users if you delete users with managers.</p>													
Data file syntax	F: .csv and .json <table border="1"> <thead> <tr> <th>Field</th><th>Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>File_delete_team</td><td>string</td><td>Team name</td></tr> <tr> <td>file_delete_users_in_team</td><td>Boolean</td><td>Deletes the users of the team if true</td></tr> <tr> <td>file_delete</td><td>Boolean</td><td>Deletes the team if true</td></tr> </tbody> </table>		Field	Type	Description	File_delete_team	string	Team name	file_delete_users_in_team	Boolean	Deletes the users of the team if true	file_delete	Boolean	Deletes the team if true
Field	Type	Description												
File_delete_team	string	Team name												
file_delete_users_in_team	Boolean	Deletes the users of the team if true												
file_delete	Boolean	Deletes the team if true												
Environment	Single Service (SACserviceFQDN, SACplatform, SACtokenFQDN, Username, Password)													
How the script works	<p>Reads the team.</p> <p>If the team exists, and:</p> <ul style="list-style-type: none"> If the file_delete_users_in_team is true, each user of the team will be deleted. If file_delete is true, then the team will be deleted. <p>The 'READ team' test will fail if the team doesn't exist.</p> <p>Any combination of true/false for file_delete_users_in_team and file_delete can be used. If both file_delete_users_in_team and file_delete are false, then no actions are performed.</p> <p>Users are deleted before the team is deleted, assuming file_delete_users_in_team and file_delete are both true.</p> <p>A false setting for file_delete_users_in_team and true for file_delete will perform the same function as script 1801.</p> <p>A setting of true for file_delete_users_in_team and false for file_delete will perform the same function as script 1301 without specifying the list of users.</p> <p>Deleting users</p> <p>Deleting a user that is a manager of an existing user will fail the 'DELETE user' test, and the next team in the data file will be read. Under this condition, the team will not be deleted even if file_delete is true. You can delete the manager(s) using script 1301 and then re-run this script to complete the deletion of the users in the team.</p>													

	<p>When deleting a user, any and all of their personal content stored in their personal folder, will be transferred to the System Owners personal folder and the System Owner will become the owner of that content.</p> <p>There is a rare condition where the script might report a failure because the user doesn't exist, but the user did exist, and it was successfully deleted. This infrequent event occurs when the session expires mid-request or when there's a 'wobble' on the SAP Analytics Cloud API.</p> <p>Deleting team The duration of deletion team operation is dependent on the number of users in the team. If the operation times-out, which will typically only occur with a team size more than 12,000 users, the script will check the team has been deleted and the test 'TEAM DELETE was successful' will pass.</p>	
Known script volume limitations	The maximum team size is around 1000 users when deleting team users. It means stability issues should be expected around this team size, and the alternative script of 1301 would provide more excellent stability.	
Script throughput	The performance will be a combination of the performance of 1801 and 1301.	
Video tutorial	 <p>SAP HANA Academy provide a step-by-step tutorial for this sample script 1851 https://youtu.be/BK8jlfCalQY</p> <p>(Since the recording, the script numbers are now preceded with a '1', 001 becomes 1001, etc.)</p>	
Related Scenarios	D01 - Delete users then delete managers	Given a team of users to be deleted, it will delete the users first before deleting the managers to avoid the problem of not being able to delete a user who is a manager of another user.

1903-U-UC-Oarrk-Fcj-Em-Transport users

Sample	1903-U-UC-Oarrk-Fcj-Em-Transport users		
Script Basis	U: User		
OAuth Access	'User Provisioning'		
Basic description	UC: Transports a user from one SAP Analytics Cloud Service to another. It will update or create the user if it doesn't exist. The users' roles and team membership are also optionally transported, creating teams in the target if needed.		
Ideal for	<ul style="list-style-type: none"> Transporting individual users, or when the number of users to be transported is more significant than about 200 You are using the default Identity Provider that comes with SAP Analytics Cloud, OR SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'USER ID' 		
Not suitable when	<ul style="list-style-type: none"> You'd like to transport a team of users rather than specifying the users individually. When the number of team users is fewer than 200, then use sample script 1953. You need to ensure the 'user id' is determined by you and not derived from the e-mail because SAP Analytics Cloud Authentication is configured to use your own Identity Provider with SAML SSO mapping on the 'e-mail' or on 'custom'. 		
Notes	<ul style="list-style-type: none"> The user is only partially transported since the user's personal content is not transported. However, all the user properties that are exposed by the API are transported. Transporting users that have managers requires special attention, as the manager must be transported first or at least already be present in the target; otherwise, these users will not be created or updated. In such cases, the user is ignored. For role membership to be transported, the roles must already exist in the target with the same namespace as the source. The contents of any team folder and any security associated with the teams aren't transported. 		
Data file syntax	F: .csv and .json		
	Field	Type	Description
	file_userid	String	UserID
	file_roles_action	String	Defines the action to perform on the users' assignment to roles. Possible values are: "add", "remove", "replace" and "keep".
	file_teams_action	String	Defines the action to perform on the users' assignment to teams. Possible values are: "add", "remove", "replace" and "keep".
Environment	Two Services, source, and target: (Source-SACserviceFQDN, Source-SACtokenFQDN, Source-Username, Source-Password, Source-ContentNamespace, Source-SACplatform, Source-SAMLSSO, Target-SACserviceFQDN, Target, SACTokenFQDN, Target-Username, Target-Password, Target-ContentNamespace, Target-SACplatform, Target-SAMLSSO)		

How the script works

For each file entry, the source user and target user are read.

If the target user doesn't exist, it will be created with the same properties as the source.

If the target users' properties do not match that of the source, the target user will be updated accordingly to match that of the source.

For both creating and updating target users, the following properties form the creation/update, which will be copied from the source to the target user:

- preferredLanguage
- active
- email address
- photos
- dataAccessLanguage
- dateFormatting
- timeFormatting
- numberFormatting
- cleanUpNotificationsNumberOfDays
- systemNotificationsEmailOptIn
- marketingEmailOptIn
- isConcurrent
- manager

Additionally, the

- roles
- teams (groups)

of the target user are also updated depending upon the related action.

The `file_teams_action` and `file_roles_action`, can be "add", "remove", "replace" and "keep" (arrk).

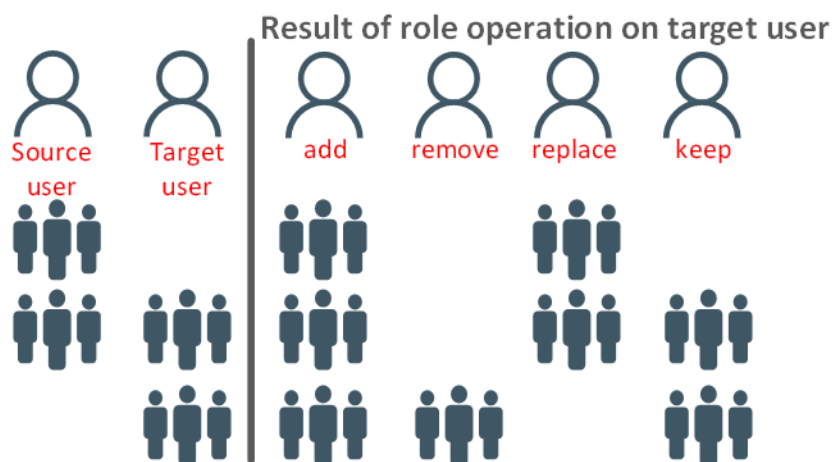
For `file_teams_action`:

add: will add the users to the team.

remove: will remove the users from the team if the user is in the team.

replace: will replace (set) the users in the team (i.e. remove and add accordingly).

keep: will respect any existing users in the team, and no changes will be made to the user membership.



For `file_roles_action`:

add: will add the team to the role.

remove: will remove the team from the role if the team is a member of any of the roles listed.

replace: will replace (set) the team assignment to roles (i.e. remove and add accordingly).
keep: will respect any roles the team is already a member of; no changes will be made.

For any user that doesn't require an update, the test 'UPDATE user' will neither pass nor fail. This is also the case, even when the users' team membership requires a change since the team membership can't be changed on the user endpoint. Team changes are only made on the groups' endpoint.

Creating Users

If the e-mail address already exists, but the username does not, a 400 error will be returned, and the script will record a 'CREATE user' failure.

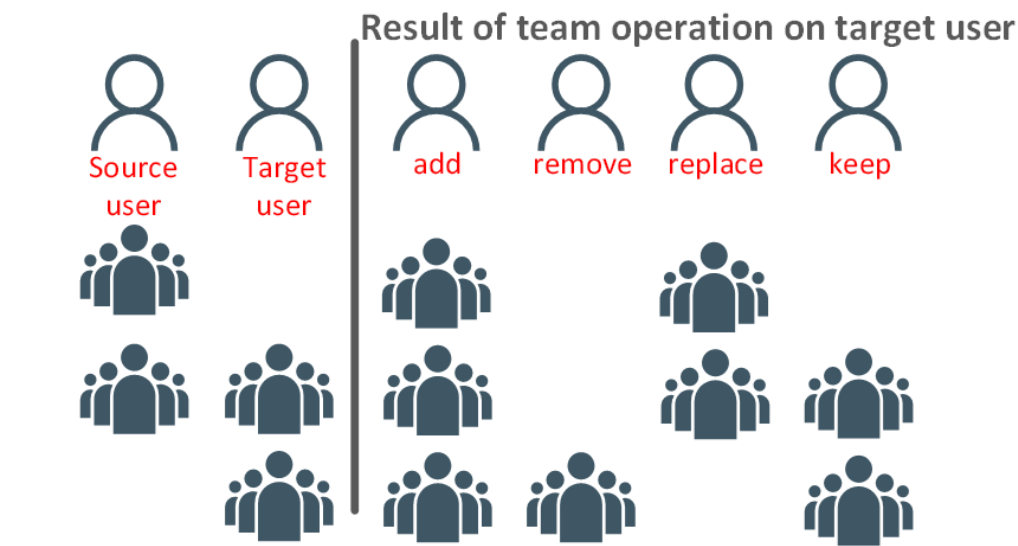
A new user will not be created if the e-mail address provided belongs to a user that already exists.

Additionally, for any user with a manager, the manager must be transported first or at least already be present in the target; otherwise, these users will not be created or updated, and the corresponding test 'CREATE user' or 'UPDATE user' will fail.

Roles

For any relationship the users have to roles, the roles must already exist in the target with the same content namespace as the source; otherwise, the user will not be created or updated (assuming the action is "add" or "replace").

Roles are only added to new users if the roles_action is "add" or "replace", otherwise, the roles are ignored, and the user will not be assigned to any roles.



Teams

User-to-team assignment cannot currently be performed using the version 1 API endpoint that is used to create or update a user; instead, a different endpoint must be used. This endpoint is for the Teams (Groups) themselves. It means more than one user can be specified per team, but if a single user requires membership changes to multiple teams, it will require multiple GET/PUT pair requests, one GET/PUT pair per team.

User membership to teams is performed every 500 entries (users), and at the end of the file, i.e. they are 'batched' together to achieve the most significant throughput. For each team,

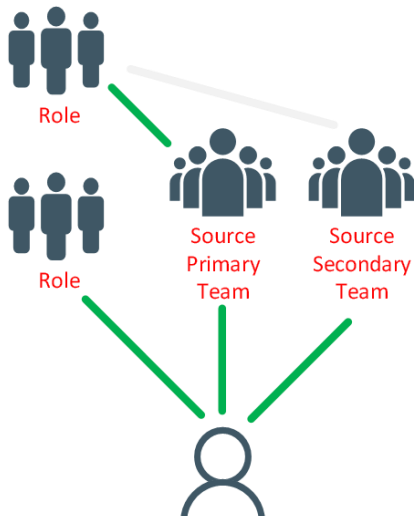
	<p>the users are batched together to form a single GET/PUT pair per team rather than a GET/PUT pair per user, per team.</p> <p>If the team does not exist, it will be created without a team folder, and the display name will match its name; otherwise, the existing team's display name is respected.</p> <p>(When SAP Analytics Cloud is hosted on an SAP Data Centre (NEO), then teams are created with a team folder, and the display name for the team will have the text “with Team Folder”. E.g. “Team1 with Team Folder”. If you’d like the team folder description to share the same team display name, use sample script 1501 before running this sample script. Alternatively, update the team folder description manually.)</p> <p>Users added into teams are ‘chunked’ into multiple PUT calls. However, given the batch size is 500 users, it’s unlikely any chunking will be applied, and it would only occur when the team has 29225 users!</p> <p>Teams are only added to new users if the teams_action is “add” or “replace”, otherwise, the teams are ignored, and the user will not be assigned to any teams.</p>		
Known script volume limitations	None		
Script throughput	Transport user from Service with 500 users to a Service with 50 users	0.24 users / sec 4.17 secs / user	
	Transport user from Service with 80,000 users to a Service with 50 users	0.11 users / sec 8.96 secs / user (3.7%)	
	Transport user from Service with 80,000 users to a Service with 50 users	0.12 users / sec 8.04 secs / user (4.3%)	
Related Scenarios	T01 - Transport Managers then Users	Given a team of users to be transported, it transports the managers first and then all the users, avoiding the problem that you can’t create a user with a manager if the manager doesn’t exist.	

1933-U-UC-Oarrk-Fcj-Em-SAML Transport users

Sample	1933-U-UC-Oarrk-Fcj-Em-SAML Transport users
Script Basis	U: User
OAuth Access	'User Provisioning'
Basic description	UC: Transports a user from one SAP Analytics Cloud Service to another. It will update or create the user if it doesn't exist. The users' roles and team membership are also optionally transported, creating teams in the target if needed.
Ideal for	<ol style="list-style-type: none"> 1. Ensuring the SAP Analytic Cloud user id is the same and consistent across the landscape (across multiple SAC services) 2. Transporting individual users, or when the number of users to be transported is greater than about 200
Not suitable when	You'd like to transport a team of users rather than specifying the users individually. When the number of users in the team is fewer than about 200, then use sample script 1983.
Notes	<p>This is identical to sample 1903 except that when users are created:</p> <p>A user will initially be created with the wrong e-mail address! This is to ensure the SAC user ID is the correct one. Once the has been created, the user is updated with the correct e-mail.</p> <p>Please refer to sample 1903 for details.</p>
Data file syntax	Please refer to sample 1903 for details.
Environment	Please refer to sample 1903 for details.
How the script works	Please refer to sample 1903 for details.
Known script volume limitations	None
Script throughput	No details recorded

1953-TU-UC-TOarrk-UOarrkie-Fcj-Em-Transport teams and users

Sample	1953-TU-UC-TOarrk-UOarrkie-Fcj-Em-Transport teams and users		
Script Basis	TU: Team and optional, then each user of the team		
OAuth Access	'User Provisioning'		
Basic description	<p>UC: Transports a team from one SAP Analytics Cloud Service to another.</p> <p>A versatile script that enables the transport of the team:</p> <ul style="list-style-type: none"> • with or without the membership of its users • with or without updating the users of the team or creating them if they don't already exist • with or without the teams' assignment to roles • with or without the users' assignment to both roles and/or teams <p>or most combinations of the above.</p>		
Ideal for	<ol style="list-style-type: none"> 1. Transporting teams and the team membership assignment without the need to transport the users 2. Transporting teams and the roles assignment of the team (though this can also be achieved with the SAP Content Network) 3. Transporting all the users of a team when the number of users in any given team is fewer than about 200. 4. Transporting users without transporting any team itself or assigning them to teams or roles in the target. 5. SAP Analytics Cloud Authentication is set to use your own Identity Provider with SAML SSO mapping on the 'USER ID.' 		
Not suitable when	<ol style="list-style-type: none"> 1. Transporting users of a team (rather than the user assignment to a team), and the team size is more significant than about 200. 2. Any of those users in a single team have a manager in the team that doesn't already exist in the target. Use sample script 1903 to transport the manager user first, then run this script to transport the remaining users. 3. You need to ensure the 'user id' is determined by you and not derived from the e-mail because SAP Analytics Cloud Authentication is configured to use your own Identity Provider with SAML SSO mapping on the 'e-mail' or on 'custom' 		
Notes	<p>The display name of the primary team is the only display name that is transported, even for when new teams are created in the target. To transport the display name of each team, an entry per team in the file is required, but this doesn't mean that the users of each team need to be transported.</p> <p>As with script 1903:</p> <ul style="list-style-type: none"> • The user is only partially transported since the user's personal content is not transported. However, all the user properties that are exposed by the API are transported. • Transporting users that have managers requires special attention, as the manager must be transported first or at least already be present in the target; otherwise, these users will not be created or updated. In such cases, the user is ignored. • For role membership to be transported, the roles must already exist in the target with the same namespace as the source. • The contents of any team folder and any security associated with the teams aren't transported. 		
Data file syntax	F: .csv and .json		
	Field	Type	Description

	file_primary_team	String	Team to be transported
	file_transport_primary_team_displayname	boolean	Defines if the team's display name of the primary team should be transported
	file_transport_primary_team_roles_action	String	Defines the action to perform on the primary team's assignment to roles. Possible values are: "add", "remove", "replace", and "keep".
	file_transport_primary_team_user_membership	boolean	Defines if the user's membership of the primary team should be applied (may conflict with users_team_action)
	file_transport_users_of_primary_team	boolean	Defines if users of the primary team should be transported
	file_users_roles_action	String	Defines the action to perform on the users' assignment to roles. Possible values are: "add", "remove", "replace", and "keep".. Applicable only if users are transported.
	file_users_teams_action	String	Defines the action to perform on the users' assignment to teams. Possible values are: "add", "remove", "replace", and "keep". Applicable only if users are transported.
Environment	Two Services, source, and target: (Source-SACserviceFQDN, Source-SACtokenFQDN, Source-Username, Source-Password, Source-ContentNamespace, Source-SACplatform, Source-SAMLSSO, Target-SACserviceFQDN, Target, SACTokenFQDN, Target-Username, Target-Password, Target-ContentNamespace, Target-SACplatform, Target-SAMLSSO)		
How the script works	<p>For each file entry, the primary team in the source and the target are read. Only a single team can be specified as 'primary', which means it is only this team that is considered for:</p> <ul style="list-style-type: none"> • Updating the team's display name • Updating the team's role membership • The list of users to be transported and their optionally subsequent relationship with other teams and roles. <p>If the primary team doesn't exist in the target, it will be created.</p> 		

file_transport_primary_team_displayname

If `file_transport_primary_team_displayname` is true, the displayname of the primary team is updated or set if the primary team needs to be created.

file_transport_primary_team_roles_action

The primary team may be members of roles in the target. Its membership to roles is updated according to `file_transport_primary_team_roles_action` setting:

add: will add the primary team to all the roles in the target that the primary team is currently a member of in the source.

remove: will remove the primary team, in the target, from roles the target primary team is a member of which are also roles the source primary team is a member of.

replace: will replace (set) the team assignment to roles (i.e. remove and add accordingly) of the target to match that of the source.

keep: will respect any roles the team is already a member of, no changes will be made.

file_transport_primary_team_user_membership

The primary team may have users as members. If this setting is true, the primary team's user membership of the target is updated to match that of the source.

A 'true' setting here will update the team membership, even if the users themselves are not transported (`file_transport_users_of_primary_team`). This is particularly helpful when all the users of the team have already been transported, and you only wish to re-synchronise the team membership across multiple SAP Analytics Cloud services. A 'true' setting works in the same way as an 'add' for the team's action.

A 'false' setting here will be ignored when both the users are transported (`file_transport_users_of_primary_team`), AND the user to team membership (`file_users_teams_action`) is 'add' or 'replace'. For any user (of the primary team) that is transported with its team membership, that team membership must include the primary team.

file_transport_users_of_primary_team

As with script 1903, users are transported, but only if `file_transport_users_of_primary_team` is true. If it is true, then:

If the target user doesn't exist, it is created with the same properties as the source.

If the target users' properties do not match that of the source, the target user will be updated accordingly to match that of the source.

For both creating and updating target users, the following properties form the creation/update, which will be copied from the source to the target user:

- preferredLanguage
- active
- email address
- photos
- dataAccessLanguage
- dateFormatting
- timeFormatting
- numberFormatting
- cleanUpNotificationsNumberOfDays

- systemNotificationsEmailOptIn
- marketingEmailOptIn
- isConcurrent
- manager

Additionally, the

- roles
- teams (groups)

of the target user are also updated depending upon the related action.

The `users_file_teams_action` and `users_file_roles_action`, can be “add”, “remove”, “replace” and “keep” (arrk).

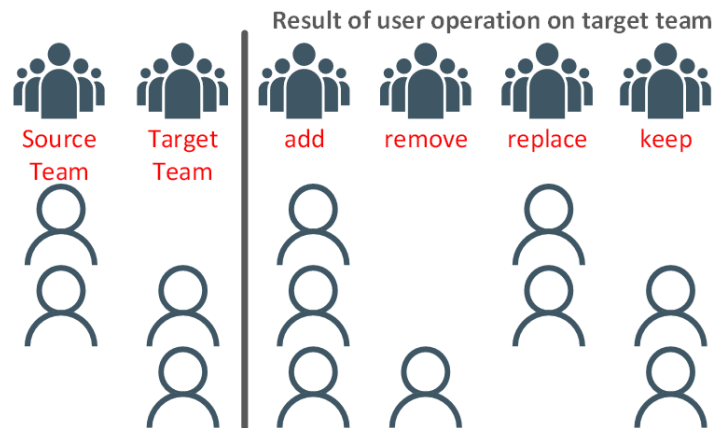
For `users_file_teams_action`:

add: will add the users to the team.

remove: will remove the users from the team if the user is in the team.

replace: will replace (set) the users in the team (i.e. remove and add accordingly).

keep: will respect any existing users in the team, and no changes will be made to the user membership.



For `users_file_roles_action`:

add: will add the team to the role.

remove: will remove the team from the role, if the team is a member of any of the roles listed.

replace: will replace (set) the team assignment to roles (i.e. remove and add accordingly).

keep: will respect any roles the team is already a member of, no changes will be made.

For any user that doesn't require an update, the test 'UPDATE user' will neither pass nor fail. This is also the case, even when the users' team membership requires a change since the team membership can't be changed on the user endpoint. Team changes are only made on the group's endpoint.

Creating Users

If the e-mail address already exists, but the username does not, a 400 error will be returned, and the script will record a 'CREATE user' failure.

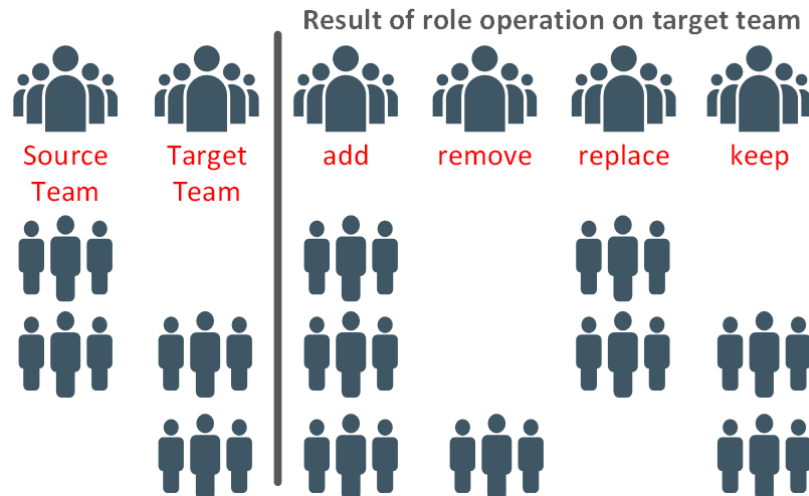
A new user will not be created if the email address provided belongs to a user that already exists.

Additionally, for any user with a manager, the manager must be transported first or at least already be present in the target; otherwise, these users will not be created or updated, and the corresponding test 'CREATE user' or 'UPDATE user' will fail.

Roles

For any relationship the users have to roles, the roles must already exist in the target with the same content namespace as the source; otherwise, the user will not be created or updated (assuming the action is "add" or "replace").

Roles are only added to new users if the `users_roles_action` is "add" or "replace", otherwise the roles are ignored and the user will not be assigned to any roles.



Teams

User-to-team assignment cannot currently be performed using the API endpoint that is used to create or update a user; instead, a different endpoint must be used. This endpoint is for the Teams (Groups) themselves. It means more than one user can be specified per team, but if a single user requires membership changes to multiple teams, it will require multiple GET/PUT pair calls one GET/PUT pair per team.

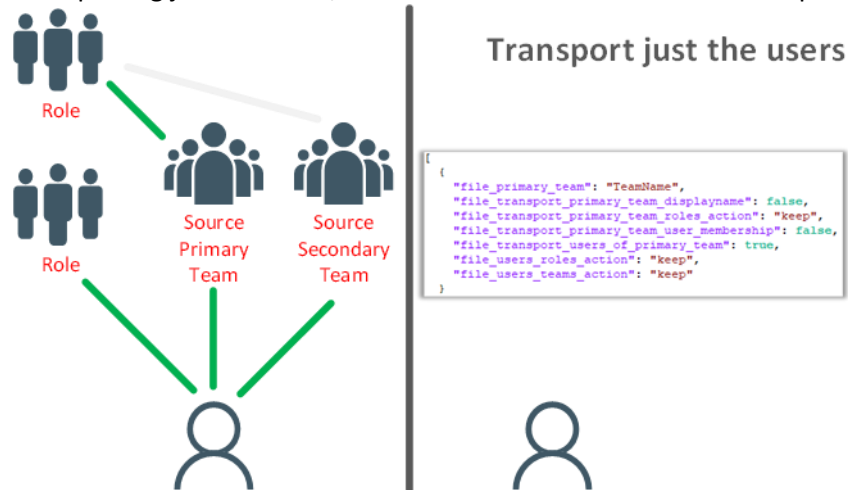
User membership to teams is performed every 100 entries (users), and at the end of the file, i.e. they are 'batched' together to achieve the most significant throughput. The users for each team are batched together to form a single GET/PUT pair per team rather than a GET/PUT pair per user, per team.

If the team does not exist, it will be created **without** a team folder, and the display name will match its name; otherwise, the existing team's display name is respected.

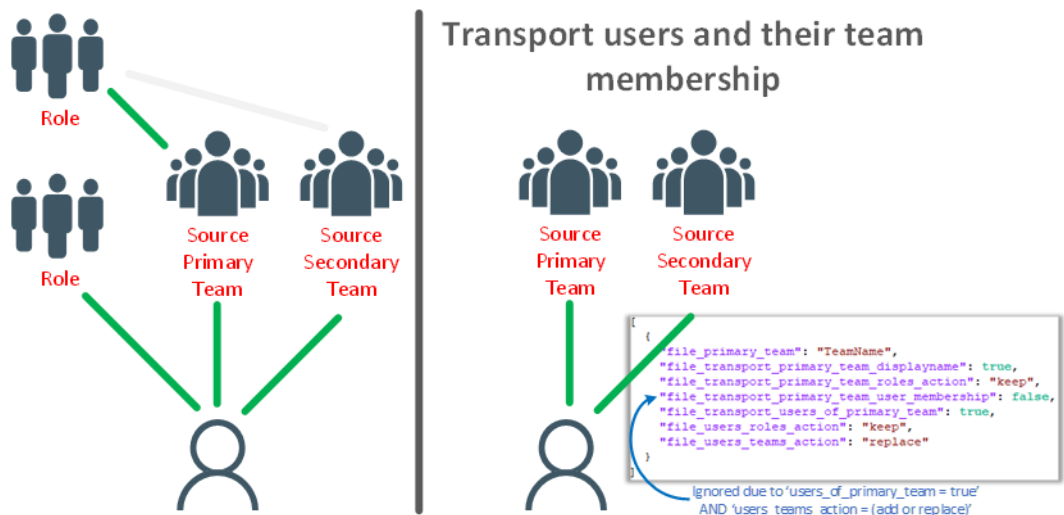
(When SAP Analytics Cloud is hosted on an SAP Data Centre (NEO), then teams are created **with** a team folder, and the display name for the team will have the text "with Team Folder". E.g. "Team1 with Team Folder". If you'd like the team folder description to share the same team display name, use sample script 1501 before running this sample script. Alternatively, update the team folder description manually.)

Teams are only added to new users if the `users_teams_action` is "add" or "replace"; otherwise, the teams are ignored, and the user will not be assigned to any teams.

Example 1: Transporting just the users, without their role or team membership

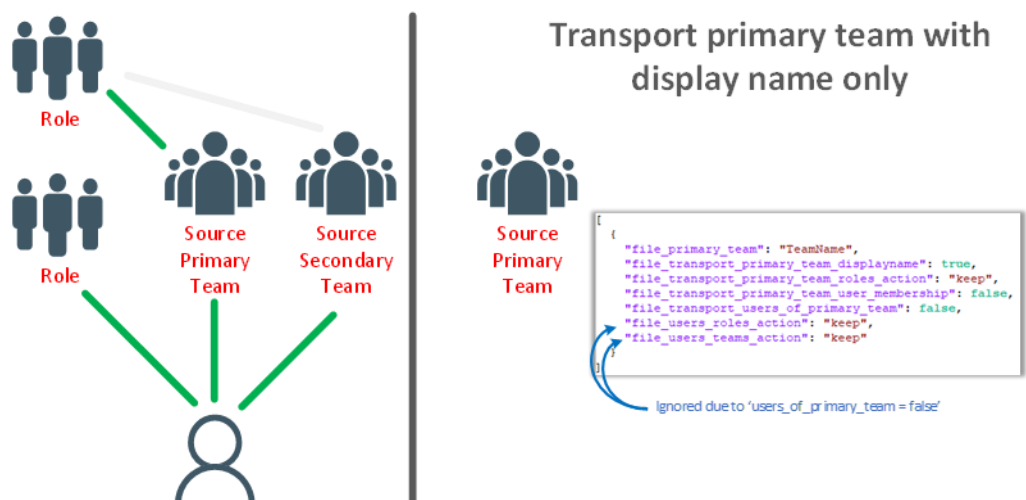


Example 2: Transporting the users and their team membership.

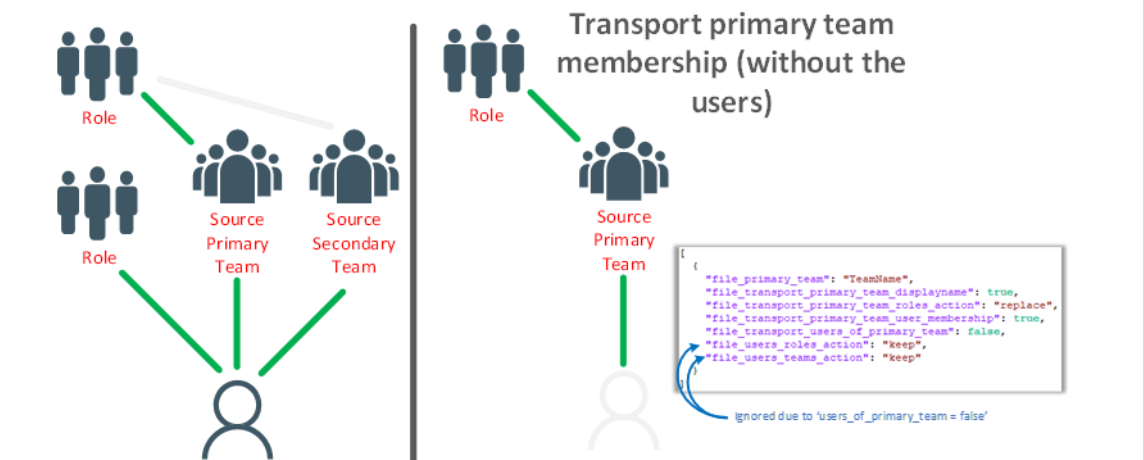


The secondary teams' display names will not be updated. In order to update the display names for each of the teams, example 3 is applicable for each team:

Example 3: Transporting primary team with display name only



Example 4: Transporting the primary team, the team's membership, but without transporting the actual users.



In this example, the users should already exist in the target; otherwise, the console will log an error for each missing user.

Known script volume limitations	The maximum team size is around 200 users. It means stability issues should be expected around this size of the team, and the alternative script of 1903 would provide more excellent stability.	
Script throughput	The performance will be similar to scripts 1903 and 1612, depending upon which options are used.	
Related Scenarios	T01 - Transport Managers then Users	Given a team of users to be transported, it transports the managers first and then all the users, avoiding the problem that you can't create a user with a manager if the manager doesn't exist.

1983-TU-UC-TOarrk-UOarrkie-Fcj-Em-SAML Transport teams and users

Sample	1983-TU-UC-TOarrk-UOarrkie-Fcj-Em-SAML Transport teams and users
Script Basis	TU: Team and optional, then each user of the team
OAuth Access	'User Provisioning'
Basic description	<p>UC: Transports a team from one SAP Analytics Cloud Service to another.</p> <p>A versatile script that enables the transport of the team:</p> <ul style="list-style-type: none"> • with or without the membership of its users • with or without updating the users of the team or creating them if they don't already exist • with or without the teams' assignment to roles • with or without the users' assignment to both roles and/or teams <p>or most combinations of the above.</p>
Ideal for	<ol style="list-style-type: none"> 1. Transporting teams and the team membership assignment without the need to transport the users 2. Transporting teams and the roles assignment of the team (though this can also be achieved with the SAP Content Network) 3. Transporting all the users of a team when the number of users in any given team is fewer than about 200. 4. Transporting users without transporting any team itself or assigning the users to any teams or roles in the target. 5. Ensuring the SAP Analytic Cloud user ID is the same and consistent across the landscape (across multiple SAC services)
Not suitable when	<ol style="list-style-type: none"> 1. Transporting users of a team (rather than the user assignment to a team), and the team size is greater than about 200. 2. If any of those users in a single team have a manager in the team that doesn't already exist in the target. Use sample script 1933 to transport the manager user first, then run this script to transport the remaining users.
Notes	<p>This is identical to sample 1953 except that when users are created:</p> <p>A user will initially be created with the wrong e-mail address! This is to ensure the SAC user ID is the correct one. Once the has been created, the user is updated with the correct e-mail.</p> <p>Please refer to sample 1953 for details.</p>
Data file syntax	Please refer to sample 1953 for details.
Environment	Please refer to sample 1953 for details.
How the script works	Please refer to sample 1953 for details.
Known script volume limitations	Please refer to sample 1953 for details.
Script throughput	No details recorded

2601-All_T-List all teams

Sample	2601-All_T-List all teams
Script Basis	All Teams, one or multiple requests to return all the Teams
OAuth Access	'User Provisioning'
Basic description	List: Lists all the teams found, and for each team, the number of members of the team and the number of roles the team is a member of
Ideal for	Listing all the teams, understanding how many teams there are, and how many of them have users. Understanding the team UUID, which is needed when using API version 2.
Not suitable when	
Notes	<p>This is a read-only script and outputs to the console. No harm can be done by running this script!</p> <p>The same sample script is available using API version 1; please see script 1601.</p> <p>Version 2 identifies teams by the UUID, unlike version 1, which uses the team name. Two teams with the same name may exist, which is problematic for the version 1 API, unlike with version 2.</p>
Data file syntax	There are no data file requirements.
Environment	Single Service (SACserviceFQDN, SACplatform, SAcTokenFQDN, Username, Password)
How the script works	<p>Sends a request using the version 2 API endpoint <code>/api/v1/scim2/Groups</code> to retrieve all the teams. Multiple requests will be made if pages of teams need to be returned due to the number of them being limited per page.</p> <p>As each page of teams is returned, the UUID, the team's display name, and the number of users and roles are output to the console.</p> <pre> ... 15:22:20.474 *Team id 9e9812cd-153e-477a-9d24-3c016effea33 with displayName AdminToolKit_Users_Dormant_B__Dormant_A_Plus_No_Private_Folder_Content has 418 users and is a member of 0 roles" 15:22:20.476 *Team id 1d8e48a5-988b-4dae-9c78-0671bb6b63c8 with displayName AdminToolKit_Users_Dormant_C__Dormant_B_Plus_No_Public_Folder_Content has 279 users and is a member of 0 roles" 15:22:20.477 *Team id 966829dc-fe09-4b29-ae0f-bbdf235cb74d with displayName AdminToolKit_Users_Dormant_D__Dormant_C_Plus_Are_Not_Managers has 279 users and is a member of 0 roles" 15:22:20.477 *Team id 848c3681-701a-482f-972f-62cbfd776efa with displayName AdminToolKit_Users_Dormant_E__Dormant_D_Plus_Are_Activated has 268 users and is a member of 0 roles" 15:22:20.478 *Team id 8f714a81-06ef-436a-9f9b-422fab835d16 with displayName AdminToolKit_Users_Dormant_F__Dormant_E_Plus_Are_Named_Licensed has 265 users and is a member of 0 roles" 15:22:20.483 *>>>>> Total: 130 teams using the v2 API <<<<<< "</pre>
Known script volume limitations	None
Script throughput	

Scenarios

A scenario is a straightforward concept comprising only sets of pre-configured sample data files. Each scenario addresses a single use case by combining different sample scripts (Postman collections) in a particular order.

It means most of the thinking has been done for you. All you need to do is tweak the data files for your needs, such as the team names, role names, manager IDs, etc.

This section describes each scenario and the purpose of each script and data file. Please use this documentation in conjunction with the scenario sample data files provided.

For an overview on managing dormant users please visit this [blog](#).

Scenario D01 - Delete users then delete managers

Purpose	Delete multiple users when some of those users are managers.
Description	<p>A user that is a manager of other users can't be deleted until they are no longer a manager of others.</p> <p>This scenario deletes all the users first, then deletes the managers</p> <p>It assumes all the users to be deleted, have managers in the same team, and this team are all the users to be deleted.</p> <p>There is no need to use this scenario if you're happy to simply re-run sample scripts 1301 or 18xx twice, allowing the first run to fail on some users. This scenario just enables a smoother deletion.</p> <p>When deleting a user, any of their personal content stored in their personal folder, will be transferred to the System Owners personal folder, and the System Owner will become the owner of that content.</p> <p>You may need to re-assign users with a new manager if you plan to delete a manager but keep some of their 'reports'. Use scenario "M01 Reassign users of a given manager to another" to achieve this.</p>

Sample data file: **Step 01 - 1653 Copy DeleteTeam to temp teams.json**

- Use collection "1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams" with this data file
- It assumes all the users you wish to delete are in a team called Delete_Users_In_This_Team.
- This team is then copied to two other teams:
 - AdminToolKit_D01_Users_That_Are_Not_Managers
 - and AdminToolKit_D01_Users_That_Are_Managers

Sample data file: **Step 02 - 1665 Remove managers.json**

- Use collection "1665-All_U-Uc-Uu-Oarrieei-Fj-Es-AdminToolKit" with this data file
- This scans all the users of SAP Analytics Cloud and removes any managers from the team AdminToolKit_D01_Users_That_Are_Not_Managers
- The data file includes the following settings:
 - "file_users_action": "remove"
 - "file_action_users_that_are_managers": true

Sample data file: **Step 03 - 1653 Swap managers for non-managers.json**

- Use collection "1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams" with this data file
- This removes the users that are not managers (from AdminToolKit_D01_Users_That_Are_Not_Managers), leaving only managers in the team AdminToolKit_D01_Users_That_Are_Managers.
- It uses "file_users_action": "exclude" to do this.

Sample data file: **Step 04 - 1851 Delete users managers and teams.json**

- Use collection "1851-T-D-Dtu-Fcj-Es-Delete teams & users in the team" with this data file
- This deletes all the users in the correct order.
- It first deletes the users (and the team itself) of AdminToolKit_D01_Users_That_Are_Not_Managers
- Then it deletes the managers (and the team) of AdminToolKit_D01_Users_That_Are_Managers

Scenario D02 - Delete dormant users A

Purpose

- Created over 3 months ago,
- have not logged in within the last 30 days,
- and have 2 or fewer logins within the last 90 days

Description

Deletes dormant users

User	Created	Logins (90 days)	Logins (30 days)	Dormant?
User 1	90 days ago	0	0	Yes
User 2	30 days ago	0	0	No
User 3	90 days ago	2	0	Yes
User 4	90 days ago	3	1	No
User 5	90 days ago	2	1	No
User 6	90 days ago	4	1	No

User 1	User-created over 3 months ago, without any logins since creation.	The user is dormant.
User 2	User-created under 3 months, without any logins since creation.	The user is not dormant.
User 3	User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.	The user is dormant.
User 4	User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.	The user is not dormant.
User 5	User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but there was a login within the last 30 days.	The user is not dormant.
User 6	User-created over 3 months ago, but has 4 logins within the last 90 days, although no login within the last 30 days.	The user is not dormant.

to share private content directly with others, if this has been done, the system owner could start to own a lot of private content unexpectedly when the user is deleted.

- They may have created content now stored in public folders. By default, such content grants special access rights to the creator, known as owner rights. Whilst ownership can be transferred to others, this is only possible via the user interface and not via any API. It means when the user is deleted, the ownership is lost. Content without an owner isn't necessarily a problem if others have been given rights; however, some artefacts, like models, require an owner to perform specific tasks on them. Overall, it could present unexpected and unnecessary surprises for others later. Ownership can be restored by recreating the user with the same user ID as the previously deleted user.
- The user could be a manager. Users who are managers can't be deleted. Please see 'Scenario M01 - Reassign users of given manager to another' for transferring users to another manager.
- The user could be deactivated, which means they don't consume a license, and in turn, this means the need to delete a user for license compliance may be unnecessary. Unnecessarily deleting users could cause surprises later, and this can be avoided simply by not deleting them in the first place.
- The user could be consuming a Business Intelligence concurrent session license. These users only consume a license at the time of login; they don't consume a license when they have no active session. This means deleting a concurrent session-based user for license compliance may be unnecessary.

Sample data file: **Step 01 - 1665 Identify Dormant Users A.json**

- Use collection "1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams" with this data file.
- This scans all the users of SAP Analytics Cloud and identifies users that were:
 - Created over 3 months ago,
 - have not logged in within the last 30 days,
 - and have 2 or fewer logins within the last 90 days.
- The identified users are added to a team called AdminToolKit_D02_Dormant_Users_A with any users not identified removed from the team if they were previously members. This ensures that any user previously identified as dormant, is no longer in the team which is used in the next step to delete them.
- The data file includes the following settings:
 - "file_users_action": "replace",
 - "file_JSON_users_to_exclude": [{"value": "SAP_SUPPORTXXXXXXXXX"}],
 - "file_multiple_action_users_operator_is_AND": true,
 - "file_action_users_created_more_than_days_ago": true,
 - "file_users_created_more_than_days": 90,
 - "file_action_users_with_most_recent_login_at_least_days_ago": true,
 - "file_users_with_with_most_recent_login_at_least_days": 30,
 - "file_action_users_with_fewer_logins_than": true,
 - "file_users_with_fewer_logins_than": 2,
 - "file_action_users_with_fewer_logins_within_last_days": true,
 - "file_users_with_logins_within_last_days": 90

Add any users you would like to exclude from being identified. i.e., even if such users are dormant, they are not added to the team to be deleted. Such users could be SAP Support users or a System Owner. Add multiple by separating them with a comma (,), for example
[{"value": "SAP_SUPPORT12345"}, {"value": "MATTHEW"}]

Adjust these settings as you see fit. With file_action_users_with_fewer_logins_within_last_days set as true and file_users_with_logins_within_last_days set as 90, it means that any login 'count tests' both require 90 days'

worth of logs to exist, and the related test(s) will only consider logins within those last 90 days. In this case, there is only 1 login count test; the total logins (users_with_fewer_logins_than) need to be fewer than or equal to 2, so only logins within those 90 days are counted for that test, unlike others including most recent login (most_recent_login_at_least_days) for example which is not limited by the 90 days setting.

The value for file_users_created_more_than_days, 90, doesn't need to be the same as file_users_with_logins_within_last_days.

Sample data file: **Step 02 - 1653 Preserve users who are members of teams A B.json**

- This is an optional step and is only needed if you have users, you do not want deleted. It could be a more favourable option to exclude users from being deleted than listing individual users in step 1 by file_JSON_users_to_exclude.
- Use collection "1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams" with this data file.
- This preserves any users that are members of Teams TeamA or TeamB from being deleted. It does this by removing users of these teams from the team updated in step 1 that identified dormant users. It means that even if any users of TeamA or TeamB are dormant, they are not deleted in the subsequent step.
- The data file includes the following settings:

```
{
  "file_source_team": "TeamA",
  "file_target_team": "AdminToolKit_D02_Dormant_Users_A",
  "file_users_action": "remove",
  "file_roles_action": "keep"
},{
  "file_source_team": "TeamB",
  "file_target_team": "AdminToolKit_D02_Dormant_Users_A",
  "file_users_action": "remove",
  "file_roles_action": "keep"
}
```


Adjust the team names accordingly and add or remove other teams for your needs.

- The teams TeamA and TeamB are not updated; they are only read.

Sample data file: **Step 03 - 1851 Delete dormant users and team.json**

- It is an optional step since dormant users do not need to be deleted for license compliance reasons since a deactivated user does not consume any license. There is no API or automated means to set users as deactivated programmatically, which means the user interface is the only means to achieve this. Currently, it is not possible to de-activate a whole team of users or easily select users of a given team for deactivation, though this may change in the future.
- Use collection "1851-T-D-Dtu-Fcj-Es-Delete teams & users in the team" with this data file.
- This deletes all the dormant users, including the team AdminToolKit_D02_Dormant_Users_A

Scenario D03 - Delete dormant users B

Purpose	Delete dormant users, which are those: <ul style="list-style-type: none">Created over 3 months ago,have not logged in within the last 30 days,have 2 or fewer logins within the last 90 days,and have no private folder content																			
Description	<p>Same as ‘Scenario D02 - Delete dormant users A’, only users with certain types of content in their private folders are excluded.</p> <p>Deletes dormant users</p> <div><table><tr><td>User 1</td><td>User-created over 3 months ago, without any logins since creation.</td><td>The user is dormant.</td></tr><tr><td>User 2</td><td>User-created under 3 months, without any logins since creation.</td><td>The user is not dormant.</td></tr><tr><td>User 3</td><td>User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.</td><td>The user is dormant.</td></tr><tr><td>User 4</td><td>User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.</td><td>The user is not dormant.</td></tr><tr><td>User 5</td><td>User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but there was a login within the last 30 days.</td><td>The user is not dormant.</td></tr><tr><td>User 6</td><td>User-created over 3 months ago, but has 4 logins within the last 90 days, although no login within the last 30 days.</td><td>The user is not dormant.</td></tr></table></div>		User 1	User-created over 3 months ago, without any logins since creation.	The user is dormant.	User 2	User-created under 3 months, without any logins since creation.	The user is not dormant.	User 3	User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.	The user is dormant.	User 4	User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.	The user is not dormant.	User 5	User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but there was a login within the last 30 days.	The user is not dormant.	User 6	User-created over 3 months ago, but has 4 logins within the last 90 days, although no login within the last 30 days.	The user is not dormant.
User 1	User-created over 3 months ago, without any logins since creation.	The user is dormant.																		
User 2	User-created under 3 months, without any logins since creation.	The user is not dormant.																		
User 3	User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.	The user is dormant.																		
User 4	User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.	The user is not dormant.																		
User 5	User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but there was a login within the last 30 days.	The user is not dormant.																		
User 6	User-created over 3 months ago, but has 4 logins within the last 90 days, although no login within the last 30 days.	The user is not dormant.																		

Deleting User 1 and User 3 could be problematic:

- They may have created content now stored in public folders. By default, such content grants special access rights to the creator, known as owner rights. Whilst ownership can be transferred to others, this is only possible via the user interface and not via any API. It means when the user is deleted, the ownership is lost. Content without an owner isn't necessarily a problem if others have been given rights; however, some artefacts, like models, require an owner to perform specific tasks on them. Overall, it could present unexpected and unnecessary surprises for others later. Ownership can be restored by recreating the user with the same user ID as the previously deleted user.
- The user could be a manager. Users who are managers can't be deleted. Please see 'Scenario M01 - Reassign users of given manager to another' for transferring users to another manager.
- The user could be deactivated, which means they don't consume a license, and in turn, this means the need to delete a user for license compliance may be unnecessary. Unnecessarily deleting users could cause surprises later, and this can be avoided simply by not deleting them in the first place.
- The user could be consuming a Business Intelligence concurrent session license. These users only consume a license at the time of login; they don't consume a license when they have no active session. This means deleting a concurrent session-based user for license compliance may be unnecessary.

Private folder content

This scenario partly resolves a problem when users have private content:

When the user is deleted via the API, any content in their private folder is moved to the System Owner's private folder. Even though it isn't best practice to share private content directly with others, if this has been done, the system owner could start to own a lot of private content unexpectedly when the user is deleted.

Users that have the following content types in their private folder are excluded:

- Stories,
- Applications,
- Digital Boardrooms,
- Templates,
- Insights and
- Folders.

This issue is only partly resolved because of many other types of content that aren't identified.

However, it's likely that given these types of content are the most popular, it's unlikely that a user with different content types would not have one of these identifiable types.

Sample data file: Step 01 - 1665 Identify Dormant Users B.json

- Use collection "1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams" with this data file.
- This scans all the users of SAP Analytics Cloud and identifies users that were:
 - Created over 3 months ago,
 - have not logged in within the last 30 days,
 - have 2 or fewer logins within the last 90 days,
 - and have no private folder content.

- The identified users are added to a team called AdminToolKit_D03_Dormant_Users_B, with any users not identified removed from the team if they were previously members. This ensures that any user previously identified as dormant is no longer in the team, which is used in the next step to delete them.
- The data file includes the following settings:
 - "file_users_action": "replace",
 - "file_JSON_users_to_exclude": [{"value": "SAP_SUPPORTXXXXXXXX"}],
 - "file_multiple_action_users_operator_is_AND": true,
 - "file_action_users_created_more_than_days_ago": true,
 - "file_users_created_more_than_days": 90,
 - "file_action_users_with_most_recent_login_at_least_days_ago": true,
 - "file_users_with_with_most_recent_login_at_least_days": 30,
 - "file_action_users_with_fewer_logins_than": true,
 - "file_users_with_fewer_logins_than": 2,
 - "file_action_users_with_fewer_logins_within_last_days": true,
 - "file_users_with_logins_within_last_days": 90,
 - "file_action_users_without_private_folder_content": true

Add any users you would like to exclude from being identified. i.e., even if such users are dormant, they are not added to the team to be deleted. Such users could be SAP Support users or a System Owner. Add multiple by separating them with a comma (,), for example
 [{"value": "SAP_SUPPORT12345"}, {"value": "MATTHEW"}]

Adjust these settings as you see fit. With file_action_users_with_fewer_logins_within_last_days set as true and file_users_with_logins_within_last_days set as 90, it means that any login 'count tests' both require 90 days' worth of logs to exist, and the related test(s) will only consider logins within those last 90 days. In this case, there is only 1 login count test; the total logins (users_with_fewer_logins_than) need to be fewer than or equal to 2, so only logins within those 90 days are counted for that test, unlike others including most recent login (most_recent_login_at_least_days) for example which is not limited by the 90 days setting.

The value for file_users_created_more_than_days, 90, doesn't need to be the same as file_users_with_logins_within_last_days.

Sample data file: **Step 02 - 1653 Preserve users who are members of teams A B.json**

- This is an optional step and is only needed if you have users, you do not want deleted. It could be a more favourable option to exclude users from being deleted than listing individual users in step 1 by file_JSON_users_to_exclude.
- Use collection “1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams” with this data file.
- This preserves any users that are members of Teams TeamA or TeamB from being deleted. It does this by removing users of these teams from the team updated in step 1 that identified dormant users. It means that even if any users of TeamA or TeamB are dormant, they are not deleted in the subsequent step.
- The data file includes the following settings:

```
{
  "file_source_team": "TeamA",
  "file_target_team": "AdminToolKit_D03_Dormant_Users_B",
  "file_users_action": "remove",
  "file_roles_action": "keep"
},{
  "file_source_team": "TeamB",
  "file_target_team": "AdminToolKit_D03_Dormant_Users_B",
  "file_users_action": "remove",
  "file_roles_action": "keep"
}
```


Adjust the team names accordingly and add or remove other teams for your needs.

- The teams TeamA and TeamB are not updated; they are only read.

Sample data file: **Step 03 - 1851 Delete dormant users and team.json**

- It is an optional step since dormant users do not need to be deleted for license compliance reasons since a deactivated user does not consume any license. There is no API or automated means to set users as deactivated programmatically, which means the user interface is the only means to achieve this. Currently, it is not possible to de-activate a whole team of users or easily select users of a given team for deactivation, though this may change in the future.
- Use collection “1851-T-D-Dtu-Fcj-Es-Delete teams & users in the team” with this data file.
- This deletes all the dormant users, including the team AdminToolKit_D03_Dormant_Users_B.

Scenario D04 - Delete dormant users C

Purpose	<div>Delete dormant users, which are those:</div> <ul style="list-style-type: none">Created over 3 months ago,have not logged in within the last 30 days,have 2 or fewer logins within the last 90 days,have no private folder content,and did not create public folder content																		
Description	<div>Same as ‘Scenario D03 - Delete dormant users B’, only users with that created public folder content are excluded.</div> <div>Deletes dormant users</div> <div></div> <table><tr><td>User 1</td><td>User-created over 3 months ago, without any logins since creation.</td><td>The user is dormant.</td></tr><tr><td>User 2</td><td>User-created under 3 months, without any logins since creation.</td><td>The user is not dormant.</td></tr><tr><td>User 3</td><td>User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.</td><td>The user is dormant.</td></tr><tr><td>User 4</td><td>User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.</td><td>The user is not dormant.</td></tr><tr><td>User 5</td><td>User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but there was a login within the last 30 days.</td><td>The user is not dormant.</td></tr><tr><td>User 6</td><td>User-created over 3 months ago, but has 4 logins within the last 90 days. although</td><td>The user is not dormant.</td></tr></table>	User 1	User-created over 3 months ago, without any logins since creation.	The user is dormant.	User 2	User-created under 3 months, without any logins since creation.	The user is not dormant.	User 3	User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.	The user is dormant.	User 4	User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.	The user is not dormant.	User 5	User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but there was a login within the last 30 days.	The user is not dormant.	User 6	User-created over 3 months ago, but has 4 logins within the last 90 days. although	The user is not dormant.
User 1	User-created over 3 months ago, without any logins since creation.	The user is dormant.																	
User 2	User-created under 3 months, without any logins since creation.	The user is not dormant.																	
User 3	User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.	The user is dormant.																	
User 4	User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.	The user is not dormant.																	
User 5	User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but there was a login within the last 30 days.	The user is not dormant.																	
User 6	User-created over 3 months ago, but has 4 logins within the last 90 days. although	The user is not dormant.																	

no login within the last 30 days.

Deleting User 1 and User 3 could be problematic:

- The user could be a manager. Users who are managers can't be deleted. Please see 'Scenario M01 - Reassign users of given manager to another' for transferring users to another manager.
- The user could be deactivated, which means they don't consume a license, and in turn, this means the need to delete a user for license compliance may be unnecessary. Unnecessarily deleting users could cause surprises later, and this can be avoided simply by not deleting them in the first place.
- The user could be consuming a Business Intelligence concurrent session license. These users only consume a license at the time of login; they don't consume a license when they have no active session. This means deleting a concurrent session-based user for license compliance may be unnecessary.

Private folder content

This scenario partly resolves a problem when users have private content:

When the user is deleted via the API, any content in their private folder is moved to the System Owner's private folder. Even though it isn't best practice to share private content directly with others, if this has been done, the system owner could start to own a lot of private content unexpectedly when the user is deleted.

Users that have the following content types in their private folder are excluded:

- Stories,
- Applications,
- Digital Boardrooms,
- Templates,
- Insights and
- Folders.

This issue is only partly resolved because of many other types of content that aren't identified.

However, it's likely that given these types of content are the most popular, it's unlikely that a user with different content types would not have one of these identifiable types.

Public folder content

This scenario partly resolves a problem when users create content stored in public folders:

By default, any content created is granted special access rights to the creator, known as owner rights. Whilst ownership can be transferred to others, this is only possible via the user interface and not via any API. It means when the user is deleted, the ownership is lost. Content without an owner isn't necessarily a problem if others have been given rights; however, some artefacts, like models, require an owner to perform specific tasks on them. Overall, it could present unexpected and unnecessary surprises for others later. Ownership can be restored by recreating the user with the same user ID as the previously deleted user.

This issue is only partly resolved because there is no check to see if the user owns the content.

Ownership rights may have changed since creation. The check is if the user created the content, not if they are still the owner or have any particular rights to the content. However, any users that created public folder content will be excluded and thus help reduce future surprises with a lack of access to any content. It is more unlikely that such users would be owners of other content, which could be problematic if deleted.

Sample data file: **Step 01 - 1665 Identify Dormant Users C.json**

- Use collection “1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams” with this data file.
- This scans all the users of SAP Analytics Cloud and identifies users that were:
 - Created over 3 months ago,
 - have not logged in within the last 30 days,
 - have 2 or fewer logins within the last 90 days,
 - have no private folder content,
 - and did not create public folder content.
- The identified users are added to a team called AdminToolKit_D04_Dormant_Users_C, with any users not identified removed from the team if they were previously members. This ensures that any user previously identified as dormant is no longer in the team, which is used in the next step to delete them.
- The data file includes the following settings:
 - "file_users_action": "replace",
 - "file_JSON_users_to_exclude": [{"value": "SAP_SUPPORTXXXXXXXX"}],
 - "file_multiple_action_users_operator_is_AND": true,
 - "file_action_users_created_more_than_days_ago": true,
 - "file_users_created_more_than_days": **90**,
 - "file_action_users_with_most_recent_login_at_least_days_ago": true,
 - "file_users_with_with_most_recent_login_at_least_days": **30**,
 - "file_action_users_with_fewer_logins_than": true,
 - "file_users_with_fewer_logins_than": **2**,
 - "file_action_users_with_fewer_logins_within_last_days": true,
 - "file_users_with_logins_within_last_days": **90**,
 - "file_action_users_without_private_folder_content": true,
 - "file_action_users_that_did_not_create_public_content": true

Add any users you would like to exclude from being identified. i.e., even if such users are dormant, they are not added to the team to be deleted. Such users could be SAP Support users or a System Owner. Add multiple by separating them with a comma (,), for example

```
[{"value": "SAP_SUPPORT12345"}, {"value": "MATTHEW"}]
```

Adjust these settings as you see fit. With file_action_users_with_fewer_logins_within_last_days set as true and file_users_with_logins_within_last_days set as 90, it means that any login ‘count tests’ both require 90 days’ worth of logs to exist, and the related test(s) will only consider logins within those last 90 days. In this case, there is only 1 login count test; the total logins (users_with_fewer_logins_than) need to be fewer than or equal to 2, so only logins within those 90 days are counted for that test, unlike others including most recent login (most_recent_login_at_least_days) for example which is not limited by the 90 days setting.

The value for file_users_created_more_than_days, 90, doesn’t need to be the same as file_users_with_logins_within_last_days.

Sample data file: **Step 02 - 1653 Preserve users who are members of teams A B.json**

- This is an optional step and is only needed if you have users, you do not want deleted. It could be a more favourable option to exclude users from being deleted than listing individual users in step 1 by file_JSON_users_to_exclude.
- Use collection “1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams” with this data file.
- This preserves any users that are members of Teams TeamA or TeamB from being deleted. It does this by removing users of these teams from the team updated in step 1 that identified dormant users. It means that even if any users of TeamA or TeamB are dormant, they are not deleted in the subsequent step.
- The data file includes the following settings:

```
{
  "file_source_team": "TeamA",
  "file_target_team": "AdminToolKit_D04_Dormant_Users_C",
  "file_users_action": "remove",
  "file_roles_action": "keep"
},{
  "file_source_team": "TeamB",
  "file_target_team": "AdminToolKit_D04_Dormant_Users_C",
  "file_users_action": "remove",
  "file_roles_action": "keep"
}
```

Adjust the team names accordingly and add or remove other teams for your needs.

- The teams TeamA and TeamB are not updated; they are only read.

Sample data file: **Step 03 - 1851 Delete dormant users and team.json**

- It is an optional step since dormant users do not need to be deleted for license compliance reasons since a deactivated user does not consume any license. There is no API or automated means to set users as deactivated programmatically, which means the user interface is the only means to achieve this. Currently, it is not possible to de-activate a whole team of users or easily select users of a given team for deactivation, though this may change in the future.
- Use collection “1851-T-D-Dtu-Fcj-Es-Delete teams & users in the team” with this data file.
- This deletes all the dormant users, including the team AdminToolKit_D04_Dormant_Users_C.

Scenario D05 - Delete dormant users D

Purpose	<div>Delete dormant users, which are those:</div> <ul style="list-style-type: none">Created over 3 months ago,have not logged in within the last 30 days,have 2 or fewer logins within the last 90 days,have no private folder content,did not create public folder content,and are not managers																			
Description	<div>Same as ‘Scenario D04 - Delete dormant users C’, only users who are managers are excluded.</div> <div>Deletes dormant users</div> <div></div> <table><tr><td>User 1</td><td>User-created over 3 months ago, without any logins since creation.</td><td>The user is dormant.</td></tr><tr><td>User 2</td><td>User-created under 3 months, without any logins since creation.</td><td>The user is not dormant.</td></tr><tr><td>User 3</td><td>User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.</td><td>The user is dormant.</td></tr><tr><td>User 4</td><td>User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.</td><td>The user is not dormant.</td></tr><tr><td>User 5</td><td>User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but there was a login within the last 30 days.</td><td>The user is not dormant.</td></tr><tr><td>User 6</td><td>User-created over 3 months ago, but has 4 logins within the last 90 days, although</td><td>The user is not dormant.</td></tr></table>		User 1	User-created over 3 months ago, without any logins since creation.	The user is dormant.	User 2	User-created under 3 months, without any logins since creation.	The user is not dormant.	User 3	User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.	The user is dormant.	User 4	User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.	The user is not dormant.	User 5	User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but there was a login within the last 30 days.	The user is not dormant.	User 6	User-created over 3 months ago, but has 4 logins within the last 90 days, although	The user is not dormant.
User 1	User-created over 3 months ago, without any logins since creation.	The user is dormant.																		
User 2	User-created under 3 months, without any logins since creation.	The user is not dormant.																		
User 3	User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.	The user is dormant.																		
User 4	User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.	The user is not dormant.																		
User 5	User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but there was a login within the last 30 days.	The user is not dormant.																		
User 6	User-created over 3 months ago, but has 4 logins within the last 90 days, although	The user is not dormant.																		

no login within the last 30 days.	
-----------------------------------	--

Deleting User 1 and User 3 could be problematic:

- The user could be deactivated, which means they don't consume a license, and in turn, this means the need to delete a user for license compliance may be unnecessary. Unnecessarily deleting users could cause surprises later, and this can be avoided simply by not deleting them in the first place.
- The user could be consuming a Business Intelligence concurrent session license. These users only consume a license at the time of login; they don't consume a license when they have no active session. This means deleting a concurrent session-based user for license compliance may be unnecessary.

Private folder content

This scenario partly resolves a problem when users have private content:

When the user is deleted via the API, any content in their private folder is moved to the System Owner's private folder. Even though it isn't best practice to share private content directly with others, if this has been done, the system owner could start to own a lot of private content unexpectedly when the user is deleted.

Users that have the following content types in their private folder are excluded:

- Stories,
- Applications,
- Digital Boardrooms,
- Templates,
- Insights and
- Folders.

This issue is only partly resolved because of many other types of content that aren't identified.

However, it's likely that given these types of content are the most popular, it's unlikely that a user with different content types would not have one of these identifiable types.

Public folder content

This scenario partly resolves a problem when users create content stored in public folders:

By default, any content created is granted special access rights to the creator, known as owner rights. Whilst ownership can be transferred to others, this is only possible via the user interface and not via any API. It means when the user is deleted, the ownership is lost. Content without an owner isn't necessarily a problem if others have been given rights; however, some artefacts, like models, require an owner to perform specific tasks on them. Overall, it could present unexpected and unnecessary surprises for others later. Ownership can be restored by recreating the user with the same user ID as the previously deleted user.

This issue is only partly resolved because there is no check to see if the user owns the content.

Ownership rights may have changed since creation. The check is if the user created the content, not if they are still the owner or have any particular rights to the content. However, any users that created public folder content will be excluded and thus help reduce future surprises with a lack of access to any content. It is more unlikely that such users would be owners of other content, which could be problematic if deleted.

Users who are managers

Users who are managers can't be deleted. Any attempts to delete the manager will fail, which could be unexpected or require some exception to manage. Excluding managers resolves this problem and

would mean that the sample script that deletes users should not fail for this reason. Managers are thus excluded from being identified as dormant.

Sample data file: **Step 01 - 1665 Identify Dormant Users D.json**

- Use collection “1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams” with this data file.
- This scans all the users of SAP Analytics Cloud and identifies users that were:
 - Created over 3 months ago,
 - have not logged in within the last 30 days,
 - have 2 or fewer logins within the last 90 days,
 - have no private folder content,
 - did not create public folder content,
 - and are not managers
- The identified users are added to a team called AdminToolKit_D05_Dormant_Users_D, with any users not identified removed from the team if they were previously members. This ensures that any user previously identified as dormant is no longer in the team, which is used in the next step to delete them.
- The data file includes the following settings:
 - "file_users_action": "invert",
 - "file_JSON_users_to_exclude": [{"value": "SAP_SUPPORTXXXXXXXX"}],
 - "file_multiple_action_users_operator_is_AND": false,
 - "file_action_users_that_are_managers": true,
 - "file_action_users_created_recently": true,
 - "file_users_created_recently_in_days": 90,
 - "file_action_users_with_most_recent_login_within_last_days": true,
 - "file_users_with_with_most_recent_login_within_last_days": 30,
 - "file_action_users_with_greater_logins_than": true,
 - "file_users_with_greater_logins_than": 2,
 - "file_action_users_with_greater_logins_within_last_days": true,
 - "file_users_with_logins_within_last_days": 90,
 - "file_action_users_with_private_folder_content": true,
 - "file_action_users_that_created_public_content": true

The test to check if a user is a manager can not be ‘ANDed’ with other tests due to the design of the sample script. It means that to ‘AND’ conditions require a change to the logic to test the opposite of the tests and then ‘invert’ the users. To explain, the logic of

(A and B)

is the same as

NOT (NOT A or NOT B).

Thus, the users_action of ‘invert’ is the NOT, and all the other tests are the opposite tests as the users_operator_is_AND is false, meaning an ‘OR’ operator between the tests is applied.

Add any users you would like to exclude from being identified. i.e., even if such users are dormant, they are not added to the team to be deleted. Such users could be SAP Support users or a System Owner. Add multiple by separating them with a comma (,), for example

```
[{"value": "SAP_SUPPORT12345"}, {"value": "MATTHEW"}]
```

Adjust these settings as you see fit, but you are setting tests for the opposite of a dormant user, with a logical ‘OR’ between the tests. Thus, the earlier scenarios tested for ‘fewer than’, ‘created at least’, or ‘last login was at least’. These are now replaced with the opposite tests of ‘greater than’, ‘created within last’ and ‘last login was recent’. There are others, including the private and public content tests.

With `file_action_users_with_greater_logins_than` set as `true` and `file_users_with_logins_within_last_days` set as `90`, it means that any login 'count tests' both require 90 days' worth of logs to exist, and the related test(s) will only consider logins within those last 90 days. In this case, there is only 1 login count test; the total logins (`users_with_greater_logins_than`) need to be greater than 2, so only logins within those 90 days are counted for that test, unlike others, including most recent login (`most_recent_login_within_last_days`) for example which is not limited by the 90 days setting.

The value for `file_users_created_recently_in_days`, `90`, doesn't need to be the same as `file_users_with_logins_within_last_days`.

Sample data file: **Step 02 - 1653 Preserve users who are members of teams A B.json**

- This is an optional step and is only needed if you have users, you do not want deleted. It could be a more favourable option to exclude users from being deleted than listing individual users in step 1 by `file_JSON_users_to_exclude`.
- Use collection "1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams" with this data file.
- This preserves any users that are members of Teams TeamA or TeamB from being deleted. It does this by removing users of these teams from the team updated in step 1 that identified dormant users. It means that even if any users of TeamA or TeamB are dormant, they are not deleted in the subsequent step.
- The data file includes the following settings:

```
{
  "file_source_team": "TeamA",
  "file_target_team": "AdminToolKit_D05_Dormant_Users_D",
  "file_users_action": "remove",
  "file_roles_action": "keep"
},{
  "file_source_team": "TeamB",
  "file_target_team": "AdminToolKit_D05_Dormant_Users_D",
  "file_users_action": "remove",
  "file_roles_action": "keep"
}
```

Adjust the team names accordingly and add or remove other teams for your needs.

- The teams TeamA and TeamB are not updated; they are only read.

Sample data file: **Step 03 - 1851 Delete dormant users and team.json**

- It is an optional step since dormant users do not need to be deleted for license compliance reasons since a deactivated user does not consume any license. There is no API or automated means to set users as deactivated programmatically, which means the user interface is the only means to achieve this. Currently, it is not possible to de-activate a whole team of users or easily select users of a given team for deactivation, though this may change in the future.
- Use collection "1851-T-D-Dtu-Fcj-Es-Delete teams & users in the team" with this data file.
- This deletes all the dormant users, including the team `AdminToolKit_D05_Dormant_Users_D`.

Scenario D06 - Delete dormant users E

Purpose	<div>Delete dormant users, which are those:</div> <ul style="list-style-type: none">Created over 3 months ago,have not logged in within the last 30 days,have 2 or fewer logins within the last 90 days,have no private folder content,did not create public folder content,are not managers,and are activated																		
Description	<div>Same as ‘Scenario D05 - Delete dormant users D’, only deactivated users are excluded.</div> <div>Deletes dormant users</div> <div></div> <table><tr><td>User 1</td><td>User-created over 3 months ago, without any logins since creation.</td><td>The user is dormant.</td></tr><tr><td>User 2</td><td>User-created under 3 months, without any logins since creation.</td><td>The user is not dormant.</td></tr><tr><td>User 3</td><td>User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.</td><td>The user is dormant.</td></tr><tr><td>User 4</td><td>User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.</td><td>The user is not dormant.</td></tr><tr><td>User 5</td><td>User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but there was a login within the last 30 days.</td><td>The user is not dormant.</td></tr><tr><td>User 6</td><td>User-created over 3 months ago, but</td><td>The user is not dormant.</td></tr></table>	User 1	User-created over 3 months ago, without any logins since creation.	The user is dormant.	User 2	User-created under 3 months, without any logins since creation.	The user is not dormant.	User 3	User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.	The user is dormant.	User 4	User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.	The user is not dormant.	User 5	User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but there was a login within the last 30 days.	The user is not dormant.	User 6	User-created over 3 months ago, but	The user is not dormant.
User 1	User-created over 3 months ago, without any logins since creation.	The user is dormant.																	
User 2	User-created under 3 months, without any logins since creation.	The user is not dormant.																	
User 3	User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.	The user is dormant.																	
User 4	User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.	The user is not dormant.																	
User 5	User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but there was a login within the last 30 days.	The user is not dormant.																	
User 6	User-created over 3 months ago, but	The user is not dormant.																	

has 4 logins within the last 90 days, although no login within the last 30 days.
--

Deleting User 1 and User 3 could be problematic:

- The user could be consuming a Business Intelligence concurrent session license. These users only consume a license at the time of login; they don't consume a license when they have no active session. This means deleting a concurrent session-based user for license compliance may be unnecessary.

Private folder content

This scenario partly resolves a problem when users have private content:

When the user is deleted via the API, any content in their private folder is moved to the System Owner's private folder. Even though it isn't best practice to share private content directly with others, if this has been done, the system owner could start to own a lot of private content unexpectedly when the user is deleted.

Users that have the following content types in their private folder are excluded:

- Stories,
- Applications,
- Digital Boardrooms,
- Templates,
- Insights and
- Folders.

This issue is only partly resolved because of many other types of content that aren't identified.

However, it's likely that given these types of content are the most popular, it's unlikely that a user with different content types would not have one of these identifiable types.

Public folder content

This scenario partly resolves a problem when users create content stored in public folders:

By default, any content created is granted special access rights to the creator, known as owner rights. Whilst ownership can be transferred to others, this is only possible via the user interface and not via any API. It means when the user is deleted, the ownership is lost. Content without an owner isn't necessarily a problem if others have been given rights; however, some artefacts, like models, require an owner to perform specific tasks on them. Overall, it could present unexpected and unnecessary surprises for others later. Ownership can be restored by recreating the user with the same user ID as the previously deleted user.

This issue is only partly resolved because there is no check to see if the user owns the content.

Ownership rights may have changed since creation. The check is if the user created the content, not if they are still the owner or have any particular rights to the content. However, any users that created public folder content will be excluded and thus help reduce future surprises with a lack of access to any content. It is more unlikely that such users would be owners of other content, which could be problematic if deleted.

Users who are managers

Users who are managers can't be deleted. Any attempts to delete the manager will fail, which could be unexpected or require some exception to manage. Excluding managers resolves this problem and would mean that the sample script that deletes users should not fail for this reason. Managers are thus excluded from being identified as dormant.

De-activated users

Deactivated users don't consume a license and are excluded from being identified as dormant. There's no need to delete users unnecessarily.

Sample data file: **Step 01 - 1665 Identify Dormant Users E.json**

- Use collection "1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams" with this data file.
- This scans all the users of SAP Analytics Cloud and identifies users that were:
 - Created over 3 months ago,
 - have not logged in within the last 30 days,
 - have 2 or fewer logins within the last 90 days,
 - have no private folder content,
 - did not create public folder content,
 - are not managers,
 - and are activated
- The identified users are added to a team called AdminToolKit_D06_Dormant_Users_E, with any users not identified removed from the team if they were previously members. This ensures that any user previously identified as dormant is no longer in the team, which is used in the next step to delete them.
- The data file includes the following settings:
 - "file_users_action": "invert",
 - "file_JSON_users_to_exclude": [{"value": "SAP_SUPPORTXXXXXXXX"}],
 - "file_multiple_action_users_operator_is_AND": false,
 - "file_action_users_not_active": true,
 - "file_action_users_that_are_managers": true,
 - "file_action_users_created_recently": true,
 - "file_users_created_recently_in_days": 90,
 - "file_action_users_with_most_recent_login_within_last_days": true,
 - "file_users_with_with_most_recent_login_within_last_days": 30,
 - "file_action_users_with_greater_logins_than": true,
 - "file_users_with_greater_logins_than": 2,
 - "file_action_users_with_greater_logins_within_last_days": true,
 - "file_users_with_logins_within_last_days": 90,
 - "file_action_users_with_private_folder_content": true,
 - "file_action_users_that_created_public_content": true

The test to check if a user is a manager can not be 'ANDed' with other tests due to the design of the sample script. It means that to 'AND' conditions require a change to the logic to test the opposite of the tests and then 'invert' the users. To explain, the logic of

(A and B)

is the same as

NOT (NOT A or NOT B).

Thus, the users_action of 'invert' is the NOT, and all the other tests are the opposite tests as the users_operator_is_AND is false, meaning an 'OR' operator between the tests is applied.

Add any users you would like to exclude from being identified. i.e., even if such users are dormant, they are not added to the team to be deleted. Such users could be SAP Support users or a System Owner. Add multiple by separating them with a comma (,), for example

```
[{"value": "SAP_SUPPORT12345"}, {"value": "MATTHEW"}]
```

Adjust these settings as you see fit, but you are setting tests for the opposite of a dormant user, with a logical 'OR' between the tests. Thus, the earlier scenarios tested for 'fewer than', 'created at least', or 'last login was at least'. These are now replaced with the opposite tests of 'greater than', 'created within last' and 'last login was recent'. There are others, including the private and public content tests.

With `file_action_users_with_greater_logins_than` set as `true` and `file_users_with_logins_within_last_days` set as `90`, it means that any login 'count tests' both require 90 days' worth of logs to exist, and the related test(s) will only consider logins within those last 90 days. In this case, there is only 1 login count test; the total logins (`users_with_greater_logins_than`) need to be greater than 2, so only logins within those 90 days are counted for that test, unlike others, including most recent login (`most_recent_login_within_last_days`) for example which is not limited by the 90 days setting.

The value for `file_users_created_recently_in_days`, `90`, doesn't need to be the same as `file_users_with_logins_within_last_days`.

Compared to the previous scenario, the only difference is `file_action_users_not_active`, which is the opposite of a dormant user because we are 'inverting' the action. 'Not active', is a deactivated user.

Sample data file: **Step 02 - 1653 Preserve users who are members of teams A B.json**

- This is an optional step and is only needed if you have users, you do not want deleted. It could be a more favourable option to exclude users from being deleted than listing individual users in step 1 by file_JSON_users_to_exclude.
- Use collection “1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams” with this data file.
- This preserves any users that are members of Teams TeamA or TeamB from being deleted. It does this by removing users of these teams from the team updated in step 1 that identified dormant users. It means that even if any users of TeamA or TeamB are dormant, they are not deleted in the subsequent step.
- The data file includes the following settings:

```
{
  "file_source_team": "TeamA",
  "file_target_team": "AdminToolKit_D06_Dormant_Users_E",
  "file_users_action": "remove",
  "file_roles_action": "keep"
},{
  "file_source_team": "TeamB",
  "file_target_team": "AdminToolKit_D06_Dormant_Users_E",
  "file_users_action": "remove",
  "file_roles_action": "keep"
}
```

Adjust the team names accordingly and add or remove other teams for your needs.

- The teams TeamA and TeamB are not updated; they are only read.

Sample data file: **Step 03 - 1851 Delete dormant users and team.json**

- It is an optional step since dormant users do not need to be deleted for license compliance reasons since a deactivated user does not consume any license. There is no API or automated means to set users as deactivated programmatically, which means the user interface is the only means to achieve this. Currently, it is not possible to de-activate a whole team of users or easily select users of a given team for deactivation, though this may change in the future.
- Use collection “1851-T-D-Dtu-Fcj-Es-Delete teams & users in the team” with this data file.
- This deletes all the dormant users, including the team AdminToolKit_D06_Dormant_Users_E.

Scenario D07 - Delete dormant users F

Purpose	<div>Delete dormant users, which are those:</div> <ul style="list-style-type: none">Created over 3 months ago,have not logged in within the last 30 days,have 2 or fewer logins within the last 90 days,have no private folder content,did not create public folder content,are not managers,are activated,and have a named-user license.																
Description	<div>Same as ‘Scenario D06 - Delete dormant users E’, only users with a Business Intelligence concurrent-session license are excluded.</div> <div>Deletes dormant users</div> <div></div> <table><tr><td>User 1</td><td>User-created over 3 months ago, without any logins since creation.</td><td>The user is dormant.</td></tr><tr><td>User 2</td><td>User-created under 3 months, without any logins since creation.</td><td>The user is not dormant.</td></tr><tr><td>User 3</td><td>User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.</td><td>The user is dormant.</td></tr><tr><td>User 4</td><td>User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.</td><td>The user is not dormant.</td></tr><tr><td>User 5</td><td>User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but</td><td>The user is not dormant.</td></tr></table>		User 1	User-created over 3 months ago, without any logins since creation.	The user is dormant.	User 2	User-created under 3 months, without any logins since creation.	The user is not dormant.	User 3	User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.	The user is dormant.	User 4	User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.	The user is not dormant.	User 5	User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but	The user is not dormant.
User 1	User-created over 3 months ago, without any logins since creation.	The user is dormant.															
User 2	User-created under 3 months, without any logins since creation.	The user is not dormant.															
User 3	User-created over 3 months ago and has 2 or fewer logins within the last 90 days. The user has no login within the last 30 days.	The user is dormant.															
User 4	User-created over 3 months ago, but has 3 logins within the last 90 days and a login within the last 30 days.	The user is not dormant.															
User 5	User-created over 3 months ago and has 2 or fewer logins within the last 90 days, but	The user is not dormant.															

	there was a login within the last 30 days.	
User 6	User-created over 3 months ago, but has 4 logins within the last 90 days, although no login within the last 30 days.	The user is not dormant.

Private folder content

This scenario partly resolves a problem when users have private content:

When the user is deleted via the API, any content in their private folder is moved to the System Owner's private folder. Even though it isn't best practice to share private content directly with others, if this has been done, the system owner could start to own a lot of private content unexpectedly when the user is deleted.

Users that have the following content types in their private folder are excluded:

- Stories,
- Applications,
- Digital Boardrooms,
- Templates,
- Insights and
- Folders.

This issue is only partly resolved because of many other types of content that aren't identified.

However, it's likely that given these types of content are the most popular, it's unlikely that a user with different content types would not have one of these identifiable types.

Public folder content

This scenario partly resolves a problem when users create content stored in public folders:

By default, any content created is granted special access rights to the creator, known as owner rights. Whilst ownership can be transferred to others, this is only possible via the user interface and not via any API. It means when the user is deleted, the ownership is lost. Content without an owner isn't necessarily a problem if others have been given rights; however, some artefacts, like models, require an owner to perform specific tasks on them. Overall, it could present unexpected and unnecessary surprises for others later. Ownership can be restored by recreating the user with the same user ID as the previously deleted user.

This issue is only partly resolved because there is no check to see if the user owns the content.

Ownership rights may have changed since creation. The check is if the user created the content, not if they are still the owner or have any particular rights to the content. However, any users that created public folder content will be excluded and thus help reduce future surprises with a lack of access to any content. It is more unlikely that such users would be owners of other content, which could be problematic if deleted.

Users who are managers

Users who are managers can't be deleted. Any attempts to delete the manager will fail, which could be unexpected or require some exception to manage. Excluding managers resolves this problem and would mean that the sample script that deletes users should not fail for this reason. Managers are thus excluded from being identified as dormant.

De-activated users

Deactivated users don't consume a license and are excluded from being identified as dormant. There's no need to delete users unnecessarily.

Business Intelligence concurrent-session users

Users with a Business Intelligence concurrent-session-based license only consume a licence while having an active session in SAP Analytics Cloud. It means you can have as many registered users as you please that are configured this way and remain compliant; there is no limit to the number of Business Intelligence concurrent-session-based users in SAP Analytics Cloud. For more details about managing licenses, please visit <https://blogs.sap.com/2020/03/10/sap-analytics-cloud-managing-licenses-with-roles-and-teams/>

In this scenario, users with the 'concurrent-session' setting are excluded from being identified as dormant; however, there is a limitation. The 'concurrent-session' setting is only a 'request' for a concurrent-session license, and it does not mean the user will consume a concurrent-session license. The user could consume a named-user license because they were assigned a Planning Role (or an Analytics Hub Role). For more details on this, please visit the blog just mentioned. It means users could incorrectly be removed from the 'dormant list' of users because their setting is 'requesting' a concurrent-session license when they are consuming a named-user license. In turn, this means these users may not get deleted when you might want them to.

If you want to understand which dormant users might not get deleted when you might want them to, make a change to the data file in Step 1:

Replace: "file_action_users_with_bi_concurrent_license": true

With: "file_action_users_with_bi_named_user_license": true

The team will then contain those dormant users because their license type is concurrent session-based. The action is 'invert', so the test needs to be the opposite of what we want; hence, testing for a 'bi_named_user_license' as true results in only users 'requesting' a concurrent-session license. However, because of possible Planning Role assignment, that concurrent-session license may not be consumed; thus, the user will consume a named-user license, and this type of user is one you may wish to delete. The result is a team with dormant users requests a concurrent-session license, ignoring all dormant named users. Some of these may be assigned a Planning Role and consume a named-user license, not a concurrent session, despite the concurrent-session setting. You will need to check their actual assignment to know either way. Refer to the blog mentioned above for more details.

Sample data file: **Step 01 - 1665 Identify Dormant Users F.json**

- Use collection "1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams" with this data file.
- This scans all the users of SAP Analytics Cloud and identifies users that were:
 - Created over 3 months ago,
 - have not logged in within the last 30 days,
 - have 2 or fewer logins within the last 90 days,
 - have no private folder content,
 - did not create public folder content,
 - are not managers,
 - are activated,
 - and consume a named-user license.
- The identified users are added to a team called AdminToolKit_D07_Dormant_Users_F, with any users not identified removed from the team if they were previously members. This ensures that any user previously identified as dormant is no longer in the team, which is used in the next step to delete them.
- The data file includes the following settings:
 - "file_users_action": "invert",

- "file_JSON_users_to_exclude": [{"value": "SAP_SUPPORTXXXXXXXX"}],
- "file_multiple_action_users_operator_is_AND": **false**,
- "file_action_users_with_bi_concurrent_license": **true**,
- "file_action_users_not_active": **true**,
- "file_action_users_that_are_managers": **true**,
- "file_action_users_created_recently": true,
- "file_users_created_recently_in_days": **90**,
- "file_action_users_with_most_recent_login_within_last_days": true,
- "file_users_with_with_most_recent_login_within_last_days": **30**,
- "file_action_users_with_greater_logins_than": true,
- "file_users_with_greater_logins_than": **2**,
- "file_action_users_with_greater_logins_within_last_days": true,
- "file_users_with_logins_within_last_days": **90**,
- "file_action_users_with_private_folder_content": true,
- "file_action_users_that_created_public_content": true

The test to check if a user is a manager can not be 'ANDed' with other tests due to the design of the sample script. It means that to 'AND' conditions require a change to the logic to test the opposite of the tests and then 'invert' the users. To explain, the logic of

(A and B)

is the same as

NOT (NOT A or NOT B).

Thus, the users_action of 'invert' is the NOT, and all the other tests are the opposite tests as the users_operator_is_AND is false, meaning an 'OR' operator between the tests is applied.

Add any users you would like to exclude from being identified. i.e., even if such users are dormant, they are not added to the team to be deleted. Such users could be SAP Support users or a System Owner. Add multiple by separating them with a comma (,), for example
[{"value": "SAP_SUPPORT12345"}, {"value": "MATTHEW"}]

Adjust these settings as you see fit, but you are setting tests for the opposite of a dormant user, with a logical 'OR' between the tests. Thus, the earlier scenarios tested for 'fewer than', 'created at least', or 'last login was at least'. These are now replaced with the opposite tests of 'greater than', 'created within last' and 'last login was recent'. There are others, including the private and public content tests.

With file_action_users_with_greater_logins_than set as true and file_users_with_logins_within_last_days set as 90, it means that any login 'count tests' both require 90 days' worth of logs to exist, and the related test(s) will only consider logins within those last 90 days. In this case, there is only one login count test; the total logins (users_with_greater_logins_than) need to be greater than 2, so only logins within those 90 days are counted for that test, unlike others, including most recent login (most_recent_login_within_last_days) for example which is not limited by the 90 days setting.

The value for file_users_created_recently_in_days, 90, doesn't need to be the same as file_users_with_logins_within_last_days.

Compared to the previous scenario, the only difference is file_action_users_with_bi_concurrent_license, which is the opposite of a dormant user because we are 'inverting' the action.

Sample data file: **Step 02 - 1653 Preserve users who are members of teams A B.json**

- This is an optional step and is only needed if you have users, you do not want deleted. It could be a more favourable option to exclude users from being deleted than listing individual users in step 1 by file_JSON_users_to_exclude.
- Use collection “1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams” with this data file.
- This preserves any users that are members of Teams TeamA or TeamB from being deleted. It does this by removing users of these teams from the team updated in step 1 that identified dormant users. It means that even if any users of TeamA or TeamB are dormant, they are not deleted in the subsequent step.
- The data file includes the following settings:

```
{
  "file_source_team": "TeamA",
  "file_target_team": "AdminToolKit_D07_Dormant_Users_F",
  "file_users_action": "remove",
  "file_roles_action": "keep"
},{
  "file_source_team": "TeamB",
  "file_target_team": "AdminToolKit_D07_Dormant_Users_F",
  "file_users_action": "remove",
  "file_roles_action": "keep"
}
```

Adjust the team names accordingly and add or remove other teams for your needs.

- The teams TeamA and TeamB are not updated; they are only read.

Sample data file: **Step 03 - 1851 Delete dormant users and team.json**

- It is an optional step since dormant users do not need to be deleted for license compliance reasons since a deactivated user does not consume any license. There is no API or automated means to set users as deactivated programmatically, which means the user interface is the only means to achieve this. Currently, it is not possible to de-activate a whole team of users or easily select users of a given team for deactivation, though this may change in the future.
- Use collection “1851-T-D-Dtu-Fcj-Es-Delete teams & users in the team” with this data file.
- This deletes all the dormant users, including the team AdminToolKit_D07_Dormant_Users_F.

Scenario L01 - Managers with Blconcurrent to Blnamed license

Purpose	Ensures all managers have a BI named user license
Description	Assign any managers with a BI named user license in case any manager currently has a concurrent session license.
Options	If you'd like to check only certain managers rather than all managers, then edit step 1 as described below

Sample data file: **Step 01 - 1665 Create team.json**

- Use collection "1665-All_U-Uc-Uu-Oarrieei-Fj-Es-AdminToolKit" with this data file.
- This scans all the users of SAP Analytics Cloud, though it needs to do this twice.
- The first scan is to identify all the managers, and it adds them to a team called AdminToolKit_L01_Users_That_Are_Managers_With_BI_Concurrent_License
- If you want to filter the managers to only certain managers, then edit this file and
 - set file_action_users_with_named_managerids to true
 - set file_multiple_action_users_operator_is_AND to true
 - and enter the correct managerid for file_JSON_named_managerids. But you must ONLY do this for the first entry in the file, not the second.
- The second scan identifies users that have a BI concurrent session license and removes those that have a BI named user license from the team. It uses "file_users_action": "intersect" to do this.

Sample data file: **Step 02 - 1451 Update license.json**

- Use collection "1451-TU-U-UI-Fcj-Es-Update Team License" with this data file
- This updates all the managers in the team AdminToolKit_L01_Users_That_Are_Managers_With_BI_Concurrent_License with a BI named user license

Sample data file: **Step 03 - 1801 Delete team.json**

- Use collection "1801-T-D-Dt-Fcj-Es-Delete teams (ok to timeout)" with this data file
- This deletes just the team AdminToolKit_L01_Users_That_Are_Managers_With_BI_Concurrent_License, without deleting the users.

Scenario L02 - Disabled users to Blconcurrent license (now redundant)

This scenario is now redundant as it is no longer relevant because disabled users no longer consume a license. Hence, there is no need to change the license type of users based on the 'isActive' property, which identifies users as 'disabled' or 'deactivated'.

The data files that used to support this scenario are no longer available.

Scenario L03 - Convert all Blconcurrent to Blnamed license

Purpose	Updates all users with a Business Intelligence 'concurrent session' license to a Business Intelligence 'named user' license.
Description	<p>It is ideal if you have enough 'named user' licenses for all your users, and there's no need for any of them to consume a 'concurrent session' license.</p> <p>(This scenario was previously more suitable for a different use case no longer relevant.)</p>
Limitations	<p>Step 02 - 1451 Update license is expected to work perfectly when the number of users in the team AdminToolKit_L03_Users_With_BI_Concurrent_Session_License is less than around 1,000 users. If you have over 1,000 users that need to be updated, use sample script 1401 instead of 1451. Sample 1401 will read a CSV file of users (1451 reads a team). To create the required data file for sample 1401, use the sample provided as a basis. Generate the list of users by using the user interface to export a file. Simply select menu-security-users and then export. Filter the list of users with IS_CONCURRENT flag equal to '1'. Then copy/paste the userids into the sample data file, ensuring that 'file_isconcurrent' is false. Then, run sample script 1401, which has no limit on the number of users it can update.</p>

Sample data file: **Step 01 - 1665 Create team.json**

- Use collection "1665-All_U-Uc-Uu-Oarrieei-Fj-Es-AdminToolKit" with this data file.
- This scans all the users of SAP Analytics Cloud and adds all users that have a BI concurrent session license into a team AdminToolKit_L03_Users_With_BI_Concurrent_Session_License
- The data file includes the following setting:
 - "file_action_users_with_bi_concurrent_license": true

Sample data file: **Step 02 - 1451 Update license.json**

- Use collection "1451-TU-U-UI-Fcj-Es-Update Team License" with this data file
- This updates all the users in the team AdminToolKit_L03_Users_With_BI_Concurrent_Session_License with a BI named user license

Sample data file: **Step 03 - 1801 Delete team.json**

- Use collection "1801-T-D-Dt-Fcj-Es-Delete teams (ok to timeout)" with this data file
- This deletes just the team AdminToolKit_L03_Users_With_BI_Concurrent_Session_License, without deleting the users.

Scenario M01 - Reassign users of given manager to another

Purpose	Reassigned users of a manager to another manager
Description	<p>This is useful when a user needs to be assigned to a different manager. This will also be the case if the manager needs to be deleted, since deleting a manager is not permitted until the user (manager) is no longer a manager as specified by any user.</p> <p>You must change the MANAGERID as described below</p>
Options	<p>The sample data file assumes a single manager, but it is easily adapted to re-assign users to a new manager, given a list of current managers.</p> <p>Feel free to change the temporary team name, but you must then also do this for the all the data files so the same name is used consistently throughout.</p> <p>In step 2, instead of assigning users to a new manager, you could simply remove the manager altogether. Just use an empty value for the manager: [{"value":""}]</p>

Sample data file: **Step 01 - 1665 Create temp team.json**

- Use collection “1665-All_U-Uc-Uu-Oarrieei-Fj-Es-AdminToolKit” with this data file.
- This scans all the users of SAP Analytics Cloud and adds all users having a manager with a managerID of ‘MANAGERID’. The users will be added into a team AdminToolKit_M01_Users_With_Manager_MANAGERID
- Change the MANAGERID as appropriate. This is the id of the manager you’d like to replace.
- The data file includes the following settings:
 - "file_action_users_with_named_managerids": true
 - "file_JSON_named_managerids": [{"value":"MANAGERID"}]
- If you need to re-assign multiple managers to a new manager simply add a new value to the array. The value would be like [{"value":"MANAGERID1"}, {"value":"MANAGERID2"}]

Sample data file: **Step 02 - 1454 Update Users with New Manager.json**

- Use collection “1454-TU-U-Um-Fcj-Es-Update Team Manager” with this data file
- This updates each user found in the team AdminToolKit_M01_Users_With_Manager_MANAGERID with the new manager
- The data file includes the following settings:
 - "file_managerid": "NEWMANAGERID"
- You must change the NEWMANAGERID to be the id of the new manager

Sample data file: **Step 03 - 1801 Delete temp team.json**

- Use collection “1801-T-D-Dt-Fcj-Es-Delete teams (ok to timeout)” with this data file
- This deletes just the team AdminToolKit_M01_Users_With_Manager_MANAGERID, without deleting the users.

Scenario R01 - Swap directly assigned role for a team role

Purpose	Removes a directly assigned role(s) from a user and instead adds the user to a team and adds the team to be a member of the role(s)
Description	<p>The scenario assumes you'd like to re-assign the 'BI_Admin' role by removing this role from all users, adding these users into a new team called R01_BIAdmin_Team and then adding the team into the role BI_Admin.</p> <p>In general, it is considered best practice to assign users to teams and then assign the teams to roles rather than assign a role directly to a user. One of the reasons for this is that updating multiple users to a different role can be as simple as updating a single team rather than updating every user.</p>
Options	<p>Simple modifications can be made to use a different role and team.</p> <p>The scenario assumes the team R01_BIAdmin_Team already has users in it.</p> <p>If the team (or an equivalent) does not already exist, step 2 should be changed to add users to this team rather than the temporary team R01_BIAdmin_Team_for_new_users_found. Step 3 would also need to use the team R01_BIAdmin_Team, and there'd be no need for steps 4 or 5.</p> <p>R01_BIAdmin_Team_for_new_users_found is a temporary holding place, so step 3 only processes the users that need to have the role removed from them.</p>

Sample data file: **Step 01 - 1612 Add BIAdmin role to team.json**

- Use collection "1612-T-Uc-Uur-Oarrk-Fcj-Es users/roles actions on Teams" with this data file
- This adds the team called R01_BIAdmin_Team into the role BI_Admin. If the team already exists, existing users and roles are respected. If the team doesn't exist a new team will be created (with a team folder).

Sample data file: **Step 02 - 665 Create temp team.json**

- Use collection "1665-All_U-Uc-Uu-Oarrieei-Fj-Es-AdminToolKit" with this data file.
- This adds all users with the role BI_Admin into a temporary team called R01_BIAdmin_Team_for_new_users_found
- This team is then used by the next script to remove the roles
- The data file includes the following settings:
 - "file_users_action": "replace"
 - "file_action_users_with_all_named_roles": true
 - "file_JSON_named_roles": [{"value": "PROFILE:sap.epm:BI_Admin"}]

Sample data file: **Step 03 - 1458 Remove role from users.json**

- Use collection "1458-TU-U-Ur-Oarrk-Fj-Es-Update Role for Each User of Team" with this data file.
- This removes the role BI_Admin of all users in the team R01_BIAdmin_Team_for_new_users_found

Sample data file: **Step 04 - 1653 Copy users from temp to BIAdmin team.json**

- Use collection “1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams” with this data file.
- This copies all users from R01_BIAdmin_Team_for_new_users_found to R01_BIAdmin_Team
- The data file includes the following settings:
 - "file_source_team": "R01_BIAdmin_Team_for_new_users_found"
 - "file_target_team": "R01_BIAdmin_Team"
 - "file_users_action": "add"
 - "file_roles_action": "keep"

Sample data file: **Step 05 - 1801 Delete temp teams.json**

- Use collection “1801-T-D-Dt-Fcj-Es-Delete teams (ok to timeout)” with this data file
- This deletes the temporary team R01_BIAdmin_Team_for_new_users_found without deleting the users of the team.

Scenario S01 - Assign settings for recently created with default settings

Purpose	Assigns the correct settings (date/time/number formats and language) for those users created in the last week
Description	If users have been created manually (via the user interface), by a csv upload, then they will have all the default settings. This can be a little annoying as each user will need to update their own settings. This scenario identifies these users and updates their settings for them saving the users having to do it for themselves
Options	Script assuming you're only interested in users created in the last week and their settings should be set to British settings, though obviously you'll need to update the data file for step 3 to reflect the settings for your organisation.

Sample data file: **Step 01 - 1665 Create temp team.json**

- Use collection "1665-All_U-Uc-Uu-Oarrieei-Fj-Es-AdminToolKit" with this data file.
- This adds users, created in the last 8 days, with default settings into a team called AdminToolKit_S01_Users_Created_Recently_With_Default_Settings
- The data file includes the following settings:
 - "file_multiple_action_users_operator_is_AND": true
 - "file_action_users_created_recently": true
 - "file_users_created_recently_in_days": 8
 - "file_actions_users_with_LangDateTimeNumberFormats_different_from": true

Sample data file: **Step 02 - 1456 Assign settings.json**

- Use collection "1456-TU-U-Upddtn-Fj-Es-Update Team DateTimeNumFormat DataAccessLang" with this data file.
- This updates all users that are members of the team AdminToolKit_S01_Users_Created_Recently_With_Default_Settings with new settings
- The data file includes the following settings:
 - "file_preferredlanguage": "en-GB"
 - "file_dataaccesslanguage": "zz"
 - "file_dateformatting": "dd/MM/yyyy"
 - "file_timeformatting": "H:mm:ss"
 - "file_numberformatting": "1,234.56"
- See this user guide documentation for collection 1456 for other settings and update accordingly

Sample data file: **Step 03 - 1801 Delete temp team.json**

- Use collection "1801-T-D-Dt-Fcj-Es-Delete teams (ok to timeout)" with this data file
- This deletes the temporary team AdminToolKit_S01_Users_Created_Recently_With_Default_Settings without deleting the users of the team.

Scenario S02 - Assign settings concurrent lic for recently created w default settings

Purpose	Assigns a BI concurrent session license and the correct settings (date/time/number formats and language) for those users created in the last week
Description	If users have been created manually (via the user interface), by a csv upload, then they will have all the default settings. This can be a little annoying as each user will need to update their own settings. This scenario identifies these users and updates their settings for them saving the users having to do it for themselves
Options	Script assuming you're only interested in users created in the last week and their settings should be set to British settings, though obviously you'll need to update the data file for step 3 to reflect the settings for your organisation.

Sample data file: **Step 01 - 1665 Create temp team.json**

- Use collection "1665-All_U-Uc-Uu-Oarrieei-Fj-Es-AdminToolKit" with this data file.
- This adds users, created in the last 8 days, with default settings into a team called AdminToolKit_S02_Users_Created_Recently_With_Default_Settings
- The data file includes the following settings:
 - "file_multiple_action_users_operator_is_AND": true
 - "file_action_users_created_recently": true
 - "file_users_created_recently_in_days": 8
 - "file_actions_users_with_LangDateTimeNumberFormats_different_from": true

Sample data file: **Step 02 - 1457 Assign settings.json**

- Use collection "1456-TU-U-Upddtn-Fj-Es-Update Team DateTimeNumFormat DataAccessLang" with this data file.
- This updates all users that are members of the team AdminToolKit_S02_Users_Created_Recently_With_Default_Settings with new settings
- The data file includes the following settings:
 - "file_isconcurrent": true
 - "file_preferredlanguage": "en-GB"
 - "file_dataaccesslanguage": "zz"
 - "file_dateformatting": "dd/MM/yyyy"
 - "file_timeformatting": "H:mm:ss"
 - "file_numberformatting": "1,234.56"
- See this user guide documentation for collection 1456 for other settings and update accordingly.

Sample data file: **Step 03 - 1801 Delete temp team.json**

- Use collection "1801-T-D-Dt-Fcj-Es-Delete teams (ok to timeout)" with this data file
- This deletes the temporary team AdminToolKit_S02_Users_Created_Recently_With_Default_Settings without deleting the users of the team.

Scenario T01 - Transport Managers then Users

Purpose	Ensure that, when transporting users from one SAP Analytics Cloud Service to another, they are transported in the correct order.
Description	<p>When creating users that have a manager assigned, that manager must already exist; otherwise, the user is not created. So, this scenario transports all the managers first, then transports the users.</p> <p>It assumes the users that are to be transported exist in a team called TransportThisTeam</p>
Options	<p>If you want to transport all the users, then add a step before step 1, by using sample script 1665 to add all users into a team TransportThisTeam</p> <p>Consider sample collection 1983 in step 4 (it reads the same data file), as 1983 will ensure the user id is consistent across the landscape in all cases, unlike collections 1903 and 1953.</p>

Sample data file: **Step 01 - 1653 Copy TransportTeam to temp teams.json**

- Use collection “1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams” with this data file
- It assumes all the users you wish to transport are in a team called TransportThisTeam.
- This team is then copied to two other teams:
 - AdminToolKit_T01_Users_That_Are_Not_Managers
 - and AdminToolKit_T01_Users_That_Are_Managers

Sample data file: **Step 02 - 1665 Remove managers.json**

- Use collection “1665-All_U-Uc-Uu-Oarrieei-Fj-Es-AdminToolKit” with this data file
- This scans all the users of SAP Analytics Cloud and removes any managers from the team AdminToolKit_T01_Users_That_Are_Not_Managers
- The data file includes the following settings:
 - "file_users_action": "remove"
 - "file_action_users_that_are_managers": true

Sample data file: **Step 03 - 1653 Swap managers for non-managers.json**

- Use collection “1653-T-Uc-Utr-Oarrkie-Fcj-Es-Teams on Teams” with this data file
- This removes the users that are not managers (from AdminToolKit_D01_Users_That_Are_Not_Managers), leaving only managers in the team AdminToolKit_D01_Users_That_Are_Managers.
- It uses "file_users_action": "exclude" to do this.

Sample data file: **Step 04 - 1953 Transport Managers then users.json**

- Use collection “1953-TU-UC-TOarrk-UOarrkie-Fcj-Em-Transport teams and users” with this data file
- This transports all the managers first (users in the team AdminToolKit_D01_Users_That_Are_Managers) and then transports all the non-managers (uses in the team AdminToolKit_D01_Users_That_Are_Not_Managers)

Sample data file: **Step 05 - 1801 Delete temp team.json**

- Use collection “1851-T-D-Dtu-Fcj-Es-Delete teams & users in the team” with this data file
- This deletes the temporary teams without deleting the users. AdminToolKit_D01_Users_That_Are_Managers and AdminToolKit_D01_Users_That_Are_Not_Managers