ACS UAC V4
Specification Dossier

VALIDATION				
Release Name and date	**Features**	**Updates**	**Paragraphs**	**Document version Validated by**
Opal 23/10/2021				UAC V1 S. DUFOUR
Panama 21/01/2022				UAC V2 C. SANTAMANS
Queen Bee 22/05/2022				UAC V2 L. PELLERAY
Red Lotus 05/07/2022				UAC V3 L. PELLERAY

Airbus Industrie 2001. All rights reserved. This document contains Airbus Industrie **proprietary information** and shall at all times remain the property of Airbus Industrie; no intellectual property right or licence is granted by Airbus Industrie in connection with any information contained in it. It is supplied on the express condition that said information is treated as confidential, shall not be used for any purpose other than that for which it is supplied, shall not be disclosed in whole or in part, to third parties other than the Airbus Members and Associated Partners, their subcontractors and suppliers (to the extent of their involvement in Airbus projects), without Airbus Industrie's prior written consent.

History table

MODIFICATIONS					
Release	Iteration	Date	US Ref	Modified by	US title (comment)
1.0	21.3	24/08/2021		S.DUFOUR	Initialisation of document
Queen Bee	23.2	23/03/2022	S-527561	L.PELLERAY	Check on loader for mandatory fields
Queen Bee	23.2	23/03/2022	S-477011	L.PELLERAY	Implement rules on loader
Queen Bee	23.2	23/03/2022	S-527492	L.PELLERAY	Add roles and modules
Queen Bee	23.3	19/04/2022	S-527728	L.PELLERAY	Check available attributes per Role AND Module
Queen Bee	23.4	27/04/2022	S-529551	C.TARROUX	LDAP Sync. & Update tables
Queen Bee	23.4	27/04/2022	S-535239	C.TARROUX	Apply restrictions LDAP subgroup - module
Queen Bee	23.4	27/04/2022	S-535238	C.TARROUX	Historization
REDLOTUS	24.1	27/04/2022	S-527492	C.TARROUX	[UAC] Add roles and module
REDLOTUS	24.2	04/04/2022	S-477011	C.TARROUX	[UAC] Implement rules on loader (PART 1)
REDLOTUS	24.2	01/04/2022	S-527561	C.TARROUX	[UAC] Check on loader for mandatory fields
REDLOTUS	24.3	19/04/2022	S-527728	C.TARROUX	[UAC] Check available attributes per Role AND Module (in GUI)
REDLOTUS	24.4	29/04/2022	S-529551	C.TARROUX	[UAC] LDAP Sync. & Update tables
REDLOTUS	24.5	16/05/2022	S-491556	C.TARROUX	[UAC] Display rules for fields - Part 2
REDLOTUS	24.5	16/05/2022	S-535239	C.TARROUX	[UAC] Apply restrictions LDAP subgroup - module
REDLOTUS	25.2	13/06/2022	S--580535	C.TARROUX	[UAC] Add module in UAC (dépendance)
REDLOTUS	25.3	17/06/2022	D-53057	C.TARROUX	[UAC] Enum creation UAC competence for replace Participant_function

Table of content

Introduction **5**
- Purpose of the Document 5
- Scope of the document 5
- Terminology 5

Web Interface **6**
- Environments 6
- User Interface 6
  - UAC user access 6
  - Search Page - Search users 7
  - Search page: extract user policies 8
    - Extract policies with import mode 9
  - Extract policies with full mode 9
  - Consult Page 10
  - Edit Policies 10
    - Edit by table 10
    - Edit by text 21

Loaders **23**
- Loader 'Roles' 23
- Loader 'Policies' 23
  - Loader checks 24
  - Loader particularities 26

UAC Technical Architecture **27**
- System Level Requirements 27
  - Data Requirements 27
  - Data Model 27
  - Global Architecture 27
  - Database management 28
  - View and Process 28
- Development Environment 29
- Launching the Loaders 29
  - Loader 'Roles' 30
  - Loader 'Policies' 30
- LDAP Synchronisations 31
  - Global synchronization process 31
  - Airbus LDAP to UAC 31
  - UAC to ACS LDAP (Batch) 31

UAC to ACS LDAP (Services) 33
UAC-external 33
Description 33
Connection between any ACS project and UAC data 34
Error handling 35
Front-End 35
Back-End 36

1. INTRODUCTION

1.1 Purpose of the Document

The Specification Dossier:

- Generates a comprehensive description of the operational scope,
- Defines both functional and technical constraints, e.g. functionalities, user interface design…

This document is written during the Solution Definition phase and specifies in detail the solution:

- Responding to the Business Requirements, described in the Business Requirements Dossier,
- According to the architecture and solution described by the Architecture Dossier,
- For the whole system, or for a subsystem defined in the Architecture Dossier,
- It refines the System Requirements from the Architecture Dossier into Component Requirements for a specific subsystem or set of subsystems.

The Specification Dossier is compulsory and it is dedicated to milestone M7, Development launching.

Requirements and traceability matrices incorporated in this document are extracted from **<repository>**, **<repository baseline>**.

The intent of this document is to be submitted for approval. **Requirements must not be directly added, deleted or modified in this document**. They shall be modified in the repository instead, and new versions of this document will be issued as necessary if approval must be obtained on the new version.

1.2 Scope of the document

This Specification Dossier documents the detailed specification for the UAC ACS solution to be deployed.

1.3 Terminology

Acronym	
UAC	User Access Control

2. Web Interface

2.1 Environments

UAC is available on ACS UAC module (int/e2e/val/prod).

2.2 User Interface

The goal of User Access Control Interface is to search users, visualize their profile and update their policies for the UAC Module.

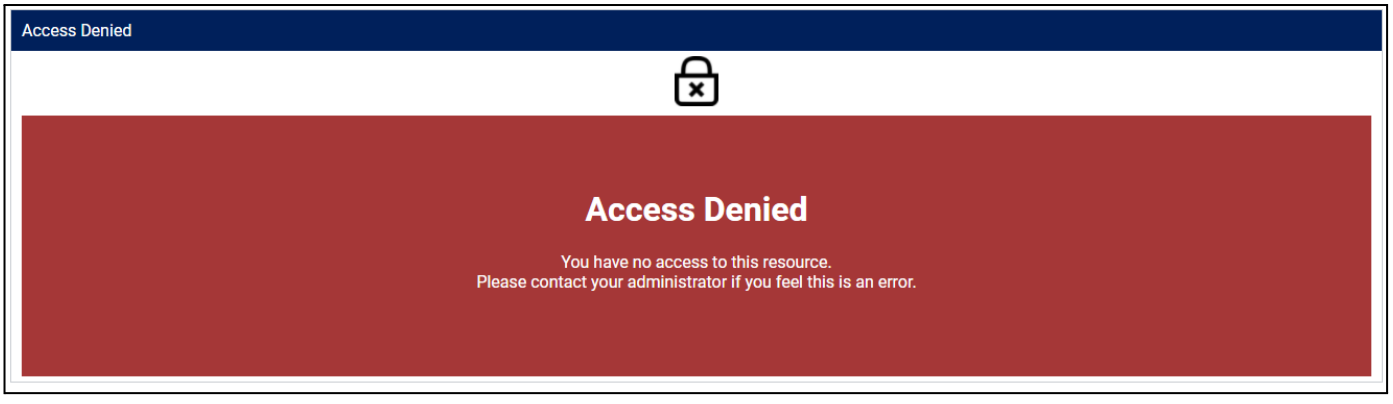
UAC is used for the access control and right identification for:

- UAC
- MODMGT:
  - Nomenclature
  - HPP Load Validation
  - MOD Management (In preparation)
- ECP
- MCT
- AC Conf (In preparation)
- OCP (In preparation)
- ACP (In preparation)
- SA/LR BaseLine (In preparation)
- Tom-Tom (In preparation)
- OFFERIT
- CONTRACT_MGT

2.2.1 UAC user access

UAC access is granted to any person having an active access to UAC or any Module whose access is managed through UAC.

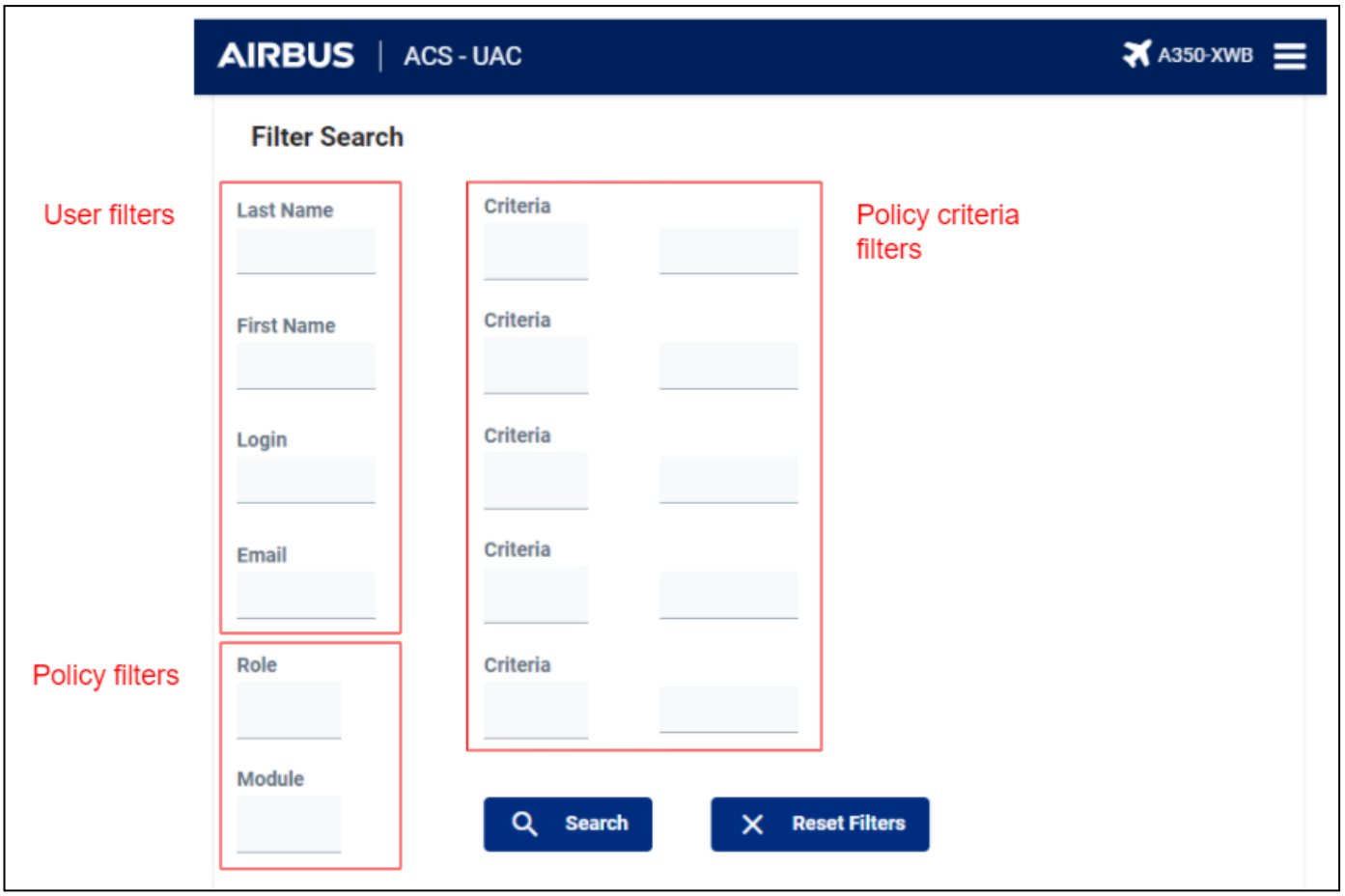
If the user has no access, an 'Access Denied' page will be displayed.



REACT request shall be done to grant access.

2.2.2 Search Page - Search users

First Screen of the application, the search page allows the user to get a list of users based on specific filters.



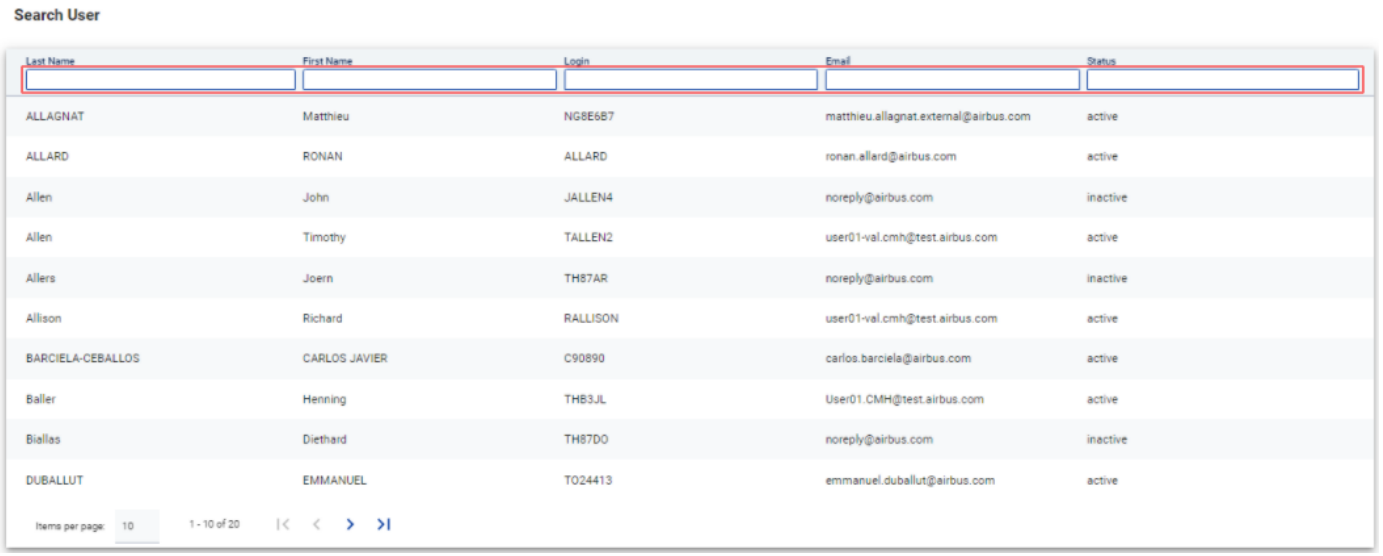
Clicking the search button without any filter will return a full list of all ACS users in the UAC Database.

Results can be refined with three types of filters:

- **User Filters**:
  - Search on user Data.
    Return one/several user(s) if the specific field contains the entered characters.
- **Policy Filters**:
  - Search on user Policies.
    Policies values are defined in UAC tables UAC_MODULE and UAC_ROLE
- **Policy criteria filters**:
  - The couple Type/Value has to be entered to apply the filter
    Policy filters possible values are contained in the table UAC_ENUM_CRITERION_VALUE

Retrieved user(s) match all filters entered.

Search Results are displayed in a table on the search Page above Filters.



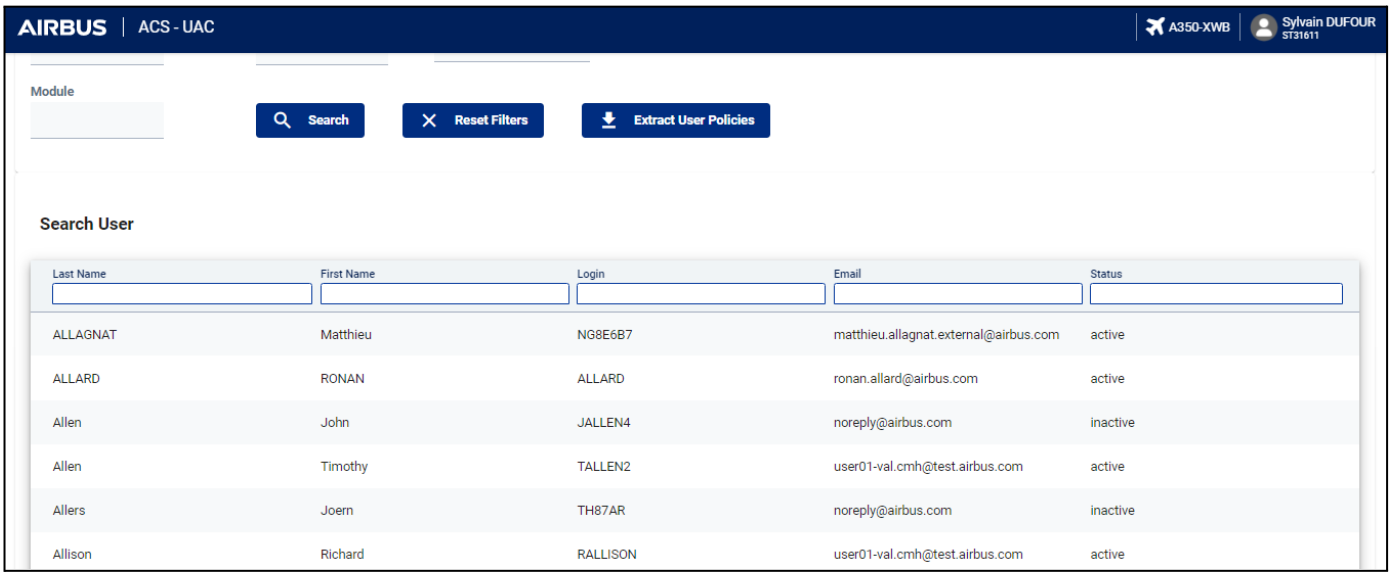
If the number or users retrieved is important, additional dynamic filtering fields on top of the table can be used to reduce the number of results.

Any result containing entered characters will be filtered in.

2.2.3 Search page: extract user policies

The current connected user can search for UAC users then extract all the user policies related to the retrieved results.

- Connected user launches a search and retrieve some UAC users.
- A button named 'Extract User Policies' appears to the right of the 'Reset Filters' button. If no users are retrieved from the search request, the 'Extract User Policies' won't appear.

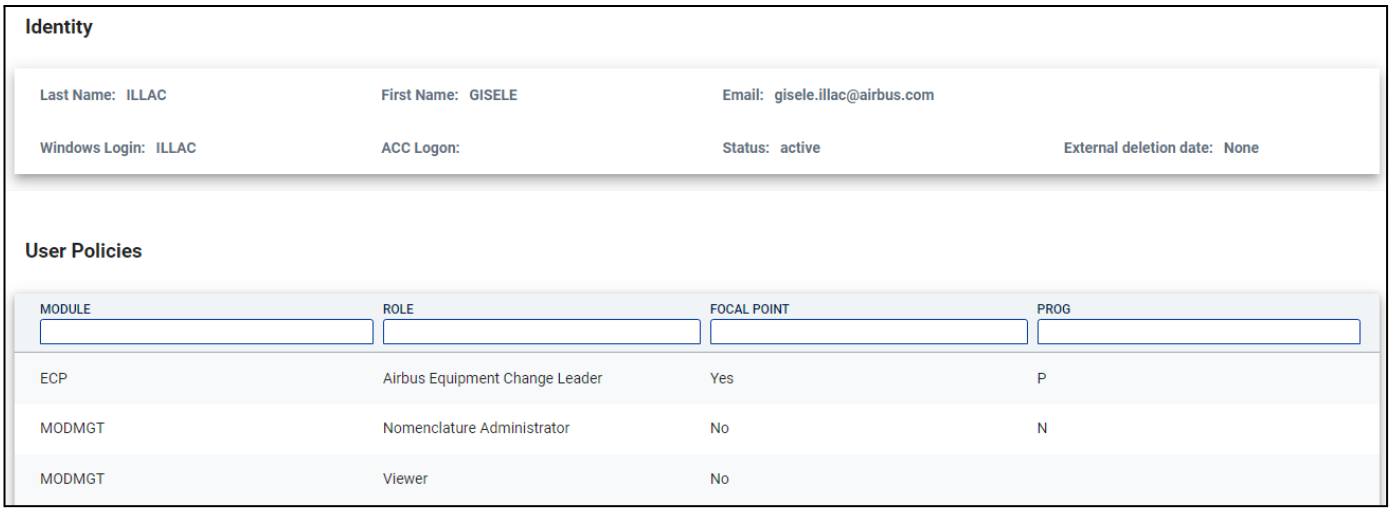


When user clicks on 'Extract User Policies' button, an extract of the database for the retrieved users is downloaded automatically by the browser and available for opening as a .csv file.

The extract only contains the lines corresponding to the search criteria. The extract is generated such as it can be directly imported in the loader.

2.2.4 Consult Page

The user have access to the Consult page by double clicking a user in the result table on Search page.
The Consult Page displays all information concerning the user: the global information and the Policies.



Identity fields are fix based on LDAP information of selected user.

Exception:

- ACC Logon is managed manually.
  Any update shall be requested through an Incident.

User Policies can be filtered through fields available on top of the result table.
Filters are dynamic, any result containing entered characters will be filtered in.

2.2.5 Edit Policies

2.2.5.1 Edit by table

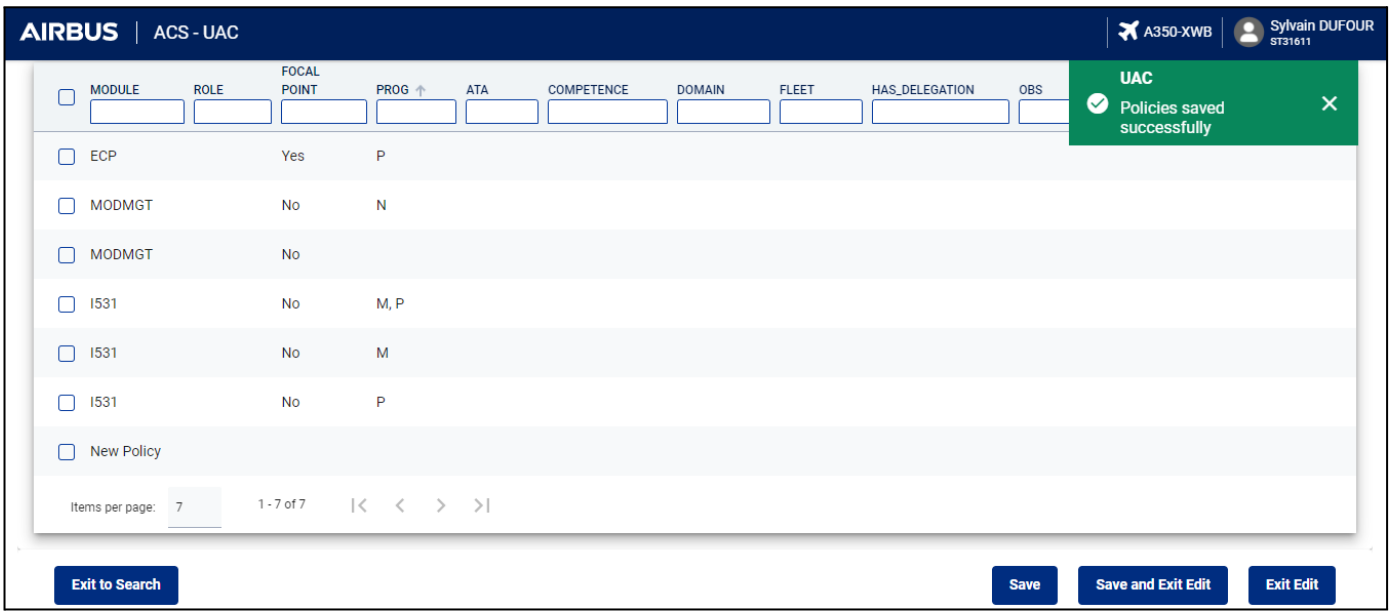
By clicking on "Edit Policies By Table" the policy table on consult page become editable:

- On click in a cell, an autocomplete will be displayed:
  - For fields having less than 20 values possibles (exemple: MODULE, ROLE, FOCAL POINT, PROG, HAS DELEGATION), all the values available will be displayed without any further user interaction.
  - For fields having more than 20 values (exemple: ATA, DOMAIN, OBS, SUPPLIER, WORKPACKAGE), due lot of values available, user must enter at least 3 characters to have a list of prefiltered values displayed.
- User can also enter manually its values in multi values fields (exemple: PROG, ATA, DOMAIN, OBS, SUPPLIER, WORKPACKAGE).
- At the End of the table, an empty line is generated to allow the user to create a new Policy.

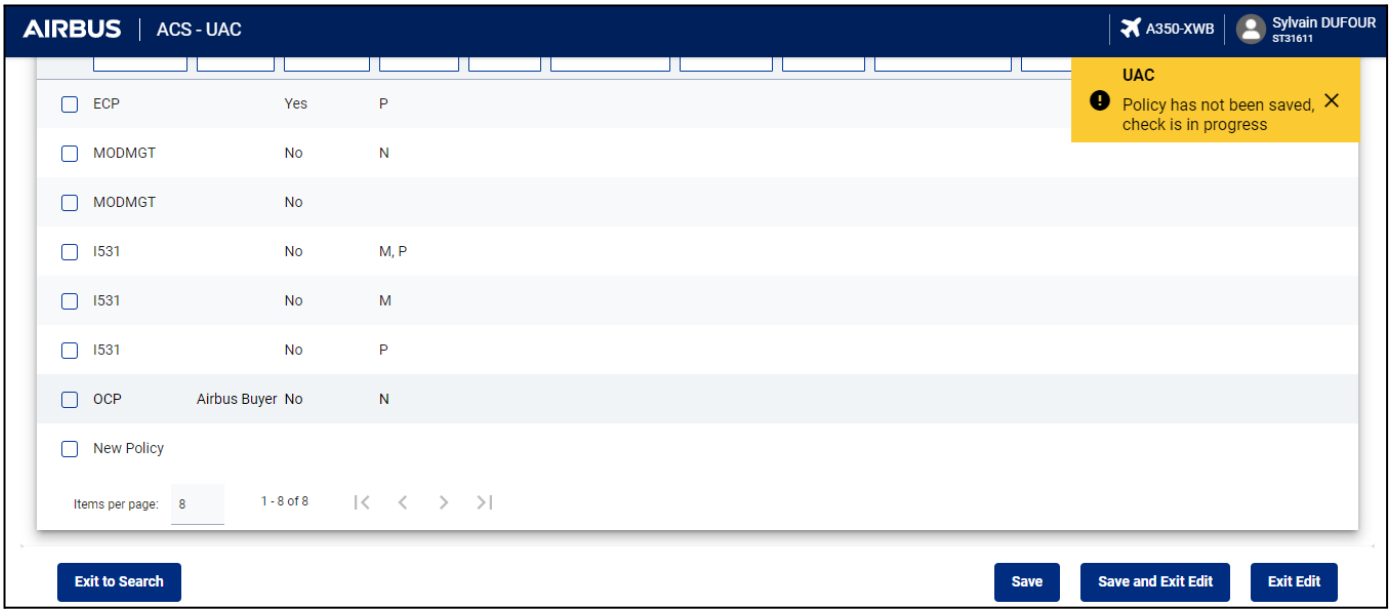
When a cell is modified, the program checks the integrity of the data in the database: these checks are described below in the Policy rules section.

Use Cases:

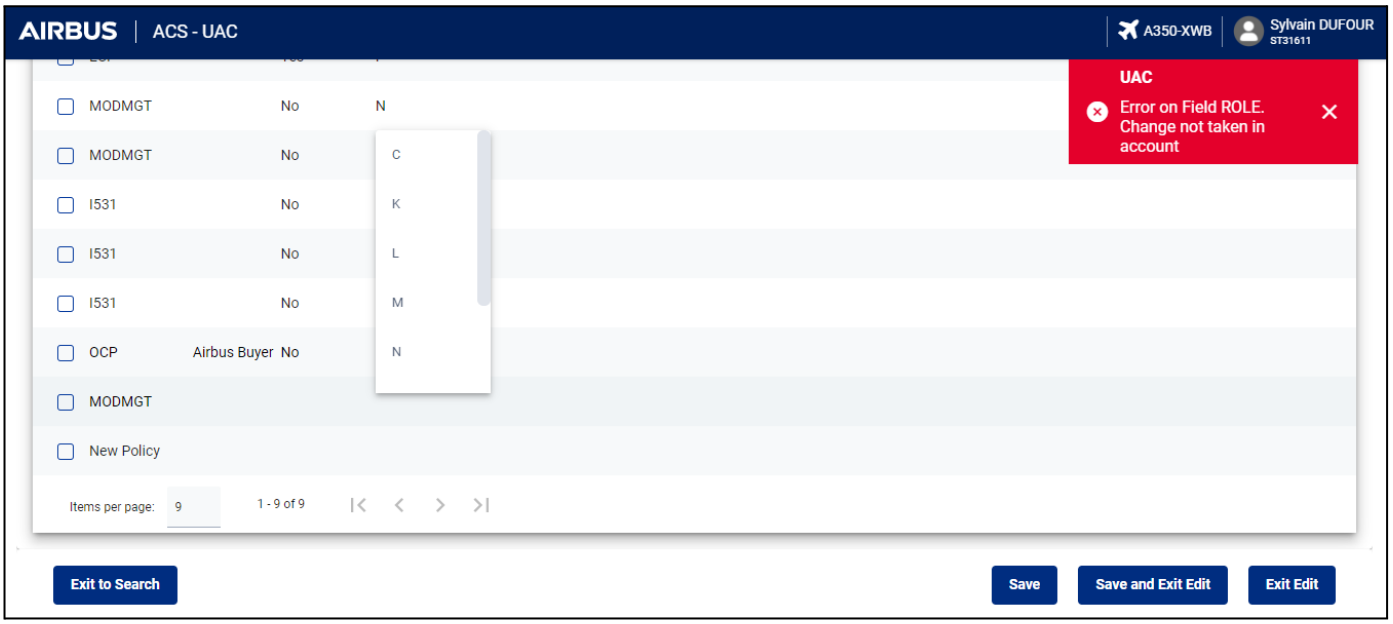
- Case 1: User try to save a policy (click on <Save> or <Save and Exit Edit> buttons) with correct values successfully checked, a ***success*** toast will be displayed to inform user.
  - **Policy is saved.**



- Case 2: User try to save a policy (click on <Save> or <Save and Exit Edit> buttons) while policy checking is in progress, a ***warning*** toast will be displayed to inform the user.
    - **Policy won't be saved.**



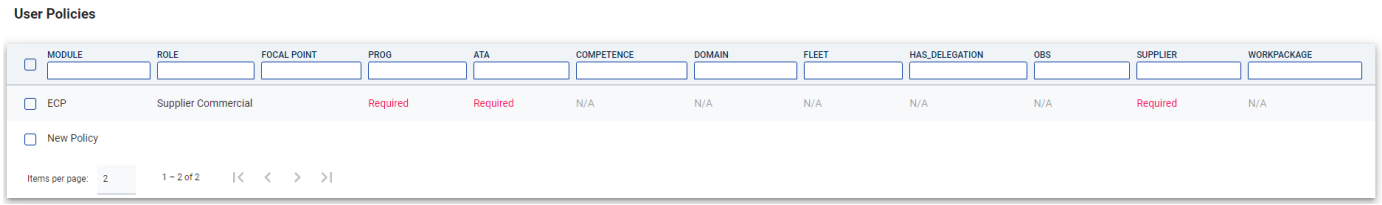
- Case 3: User try to save a policy (click on <Save> or <Save and Exit Edit> buttons) with incorrect values (here a new policy with missing role value), an ***error*** toast will be displayed to inform user.
    - **Policy won't be saved.**



POLICY RULES

Required *red value is set in fields for those criterias are missing depending on the role chosen. Saving is not available if required fields are not filled.*

N/A *grey value is set in fields for those criterias are forbidden depending on the role chosen. Users can't fill these fields.*



- Rule 1:
    - When a cell is modified, the program checks the integrity of the data in the database. If the value doesn't exist in the database, the application returns an error, the value text becomes red, user cannot save the policy until the error is corrected.
- Rule 2 :
    - When the 'PROG' cell is modified, the program checks if there is more than one program in the cell. If it is the case, the application returns an error (More than one program per line), the value text becomes red, user cannot s/ave the policy until the error is corrected.
- Rule 3 :

- If a new Policy is created and it's not the last one created, the fields **MODULE** and **ROLE** are mandatory, an error will be displayed for empty/erroneous values, user won't be able to save policies without fixing the errors.

- Rule 4 :
  - The last policy is alway a new Policy. All fields can be left empty, no error will be displayed, user will be able to save policies but the last one (which is incomplete) won't be taken in account.

- Rule 5 : *This check is disabled since April of 2021. It can be enabled again on demand by dev team on deployment ("is_domain_obsfic_check_enabled": "true" on Ansible).*
  - Specific check is applied on DOMAIN/OBS FIC. These specific check is done only if the value changed exists in database (rule 1):

- GIVEN a new policy or an existing policy
- WHEN DOMAIN is updated first without OBS FIC values
- THEN Legacy and/or ACP DOMAIN are allowed.

  This rule is deactivated. 09/03/2022

- GIVEN a new policy or an existing policy
- WHEN OBS FIC is updated first without DOMAIN values
- THEN OBS FIC TYPE PROD and/or TEAM DOMAIN are allowed.

- GIVEN a new policy or an existing policy
- WHEN ACP DOMAIN is updated first and OBS FIC values are existing
- THEN only FIC TYPE 'TEAM' are allowed. If FIC TYPE is 'PROD' a notification error is displayed, user cannot save policy

---

- GIVEN a new policy or an existing policy
- WHEN Legacy DOMAIN is updated first and OBS FIC values are existing
- THEN only FIC TYPE 'PROD' are allowed. If FIC TYPE is 'TEAM' a notification error is displayed, user cannot save policy

---

- GIVEN a new policy or an existing policy
- WHEN FIC TYPE 'TEAM' is updated first and DOMAIN values are existing
- THEN only ACP DOMAIN are allowed. If DOMAIN is Legacy a notification error is displayed, user cannot save policy

---

- GIVEN a new policy or an existing policy
- WHEN any field is updated on this policy
- THEN a notification error on DOMAIN/OBS incorrect values is displayed, user cannot save policy

---

- GIVEN an existing policy where DOMAIN/FIC TYPE couple is not consistent (Legacy DOMAIN with FIC TYPE 'TEAM' or ACP DOMAIN with FIC TYPE 'PROD')
- WHEN any field is updated on this policy
- THEN a notification error on DOMAIN/OBS incorrect values is displayed, user cannot save policy

---

- GIVEN an existing policy where DOMAIN/FIC TYPE couple is not consistent (Legacy DOMAIN with FIC TYPE 'TEAM' or ACP DOMAIN with FIC TYPE 'PROD')
- WHEN an other policy is updated
- THEN save is allowed (provided that updated policy is not the one where inconsistence exists)

- GIVEN a new policy or an existing policy
- WHEN OBS (or PROG) field is updated on this policy
- THEN PROG (or OBS) field suggestions are updated corresponding to OBS (or PROG) value chosen.

- GIVEN a new policy or an existing policy
- WHEN ATA (or PROG) field is updated on this policy
- THEN PROG (or ATA) field suggestions are updated corresponding to ATA (or PROG) value chosen.

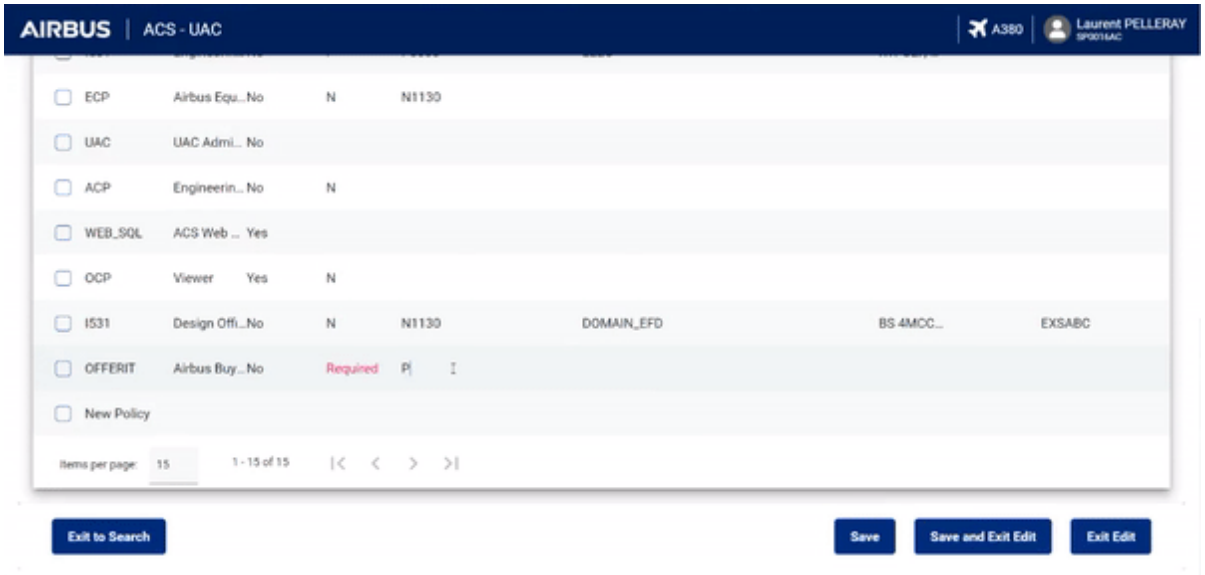
- GIVEN a new policy or an existing policy
- WHEN MODULE (or ROLE) field is updated on this policy
- THEN ROLE (or MODULE) field suggestions are updated corresponding to MODULE (or ROLE) value chosen.

- GIVEN a new policy or an existing policy with the specific UAC user (<=> USER_ID)
- WHEN you click on MODULE field without ROLE selected
- THEN the MODULE list proposed is according to the LDAP-SubGroup declaration, T_MODULE_AUTHORIZED

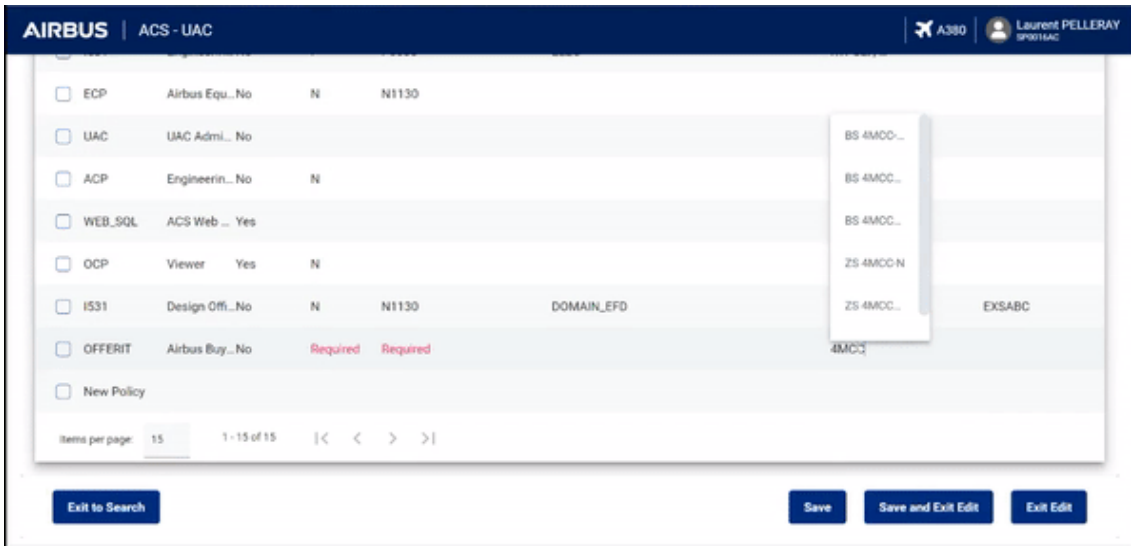
- GIVEN a new policy or an existing policy with the specific UAC user (<=> USER_ID)
- WHEN you click on ROLE field without MODULE selected
- THEN the ROLE list proposed is according to the LDAP-SubGroup declaration, T_MODULE_AUTHORIZED

Autocomplete values

- GIVEN a new policy or an existing policy
- WHEN ATA field is updated on this policy
- THEN PROG field is set corresponding to the ATA value chosen. (if ATA and OBS corresponds to the same program)



- GIVEN a new policy or an existing policy
- WHEN OBS field is updated on this policy
- THEN PROG field is set corresponding to the OBS value chosen. (if ATA and OBS corresponds to the same program)

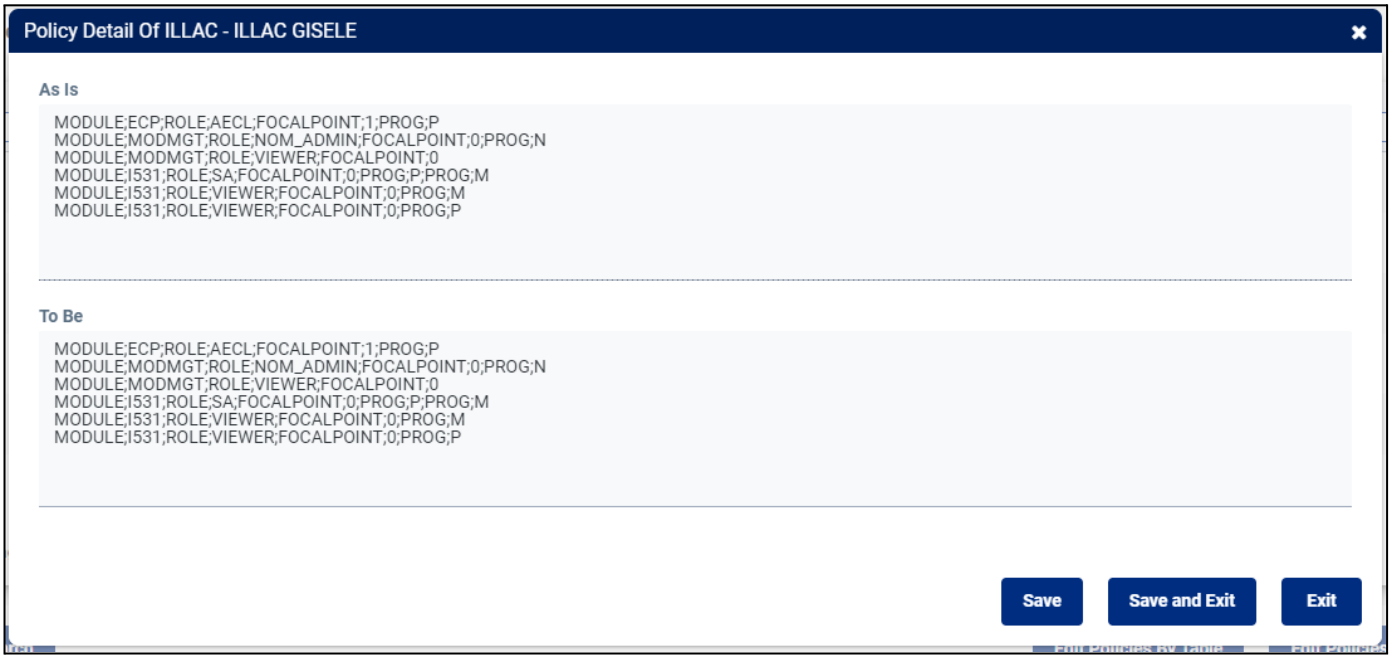


- GIVEN a new policy or an existing policy
- WHEN MODULE field is updated on this policy
- THEN all criterias fields are set corresponding to the following table, refer below :

UAC ROLE/MODULE/CRITERIA

2.2.5.2 Edit by text

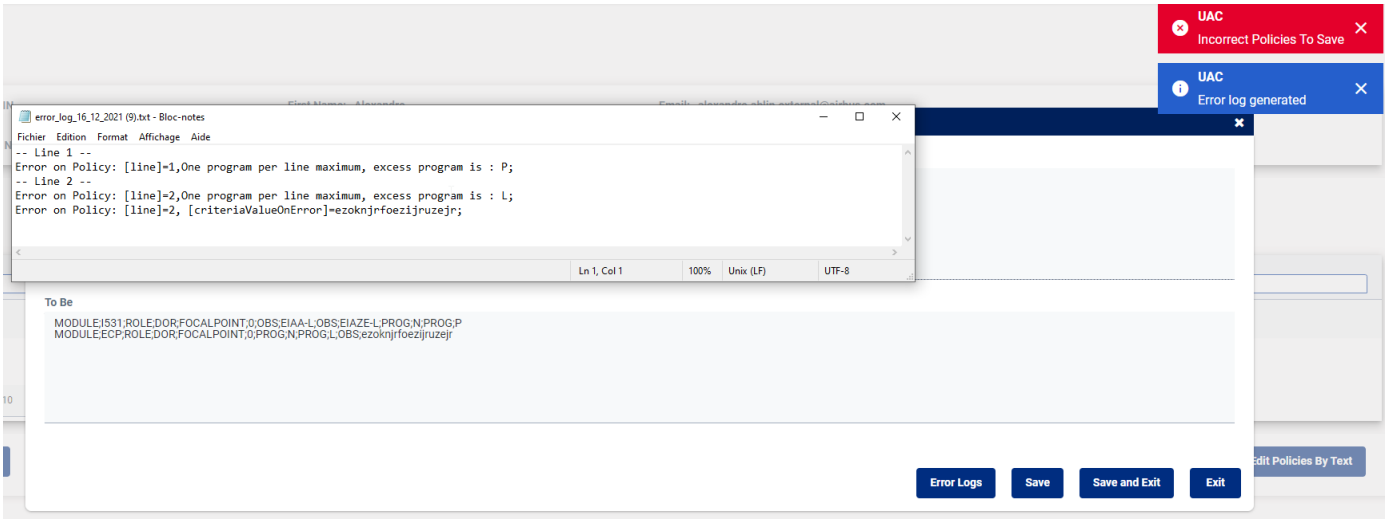
The user can modify policies with a text based interface by clicking on "Edit Policies by Text".



- The **As Is** text area represents the actual state of the policies, it is not editable but it is useful for the user to visualize easily the changes on the policies.
- The **To Be** text area represents the new state of the policies, the user can edit it to enter the new values of the Policies. Each Line is a policy and it is formatted into this format => type; value; type; value;.....

During saving the integrity of all policies for each couple Type/Value is checked in the database, if a data is missing or incorrect, an error log file is generated and downloadable from the interface.

Also checks if there is more than one PROG value per line.



There is also a check being made to make sure that the DOMAIN and FIC TYPE are compatibles.

This check is disabled since April of 2021. It can be enabled again on demand by dev team on deployment ("is_domain_obsfic_check_enabled": "true" on Ansible).

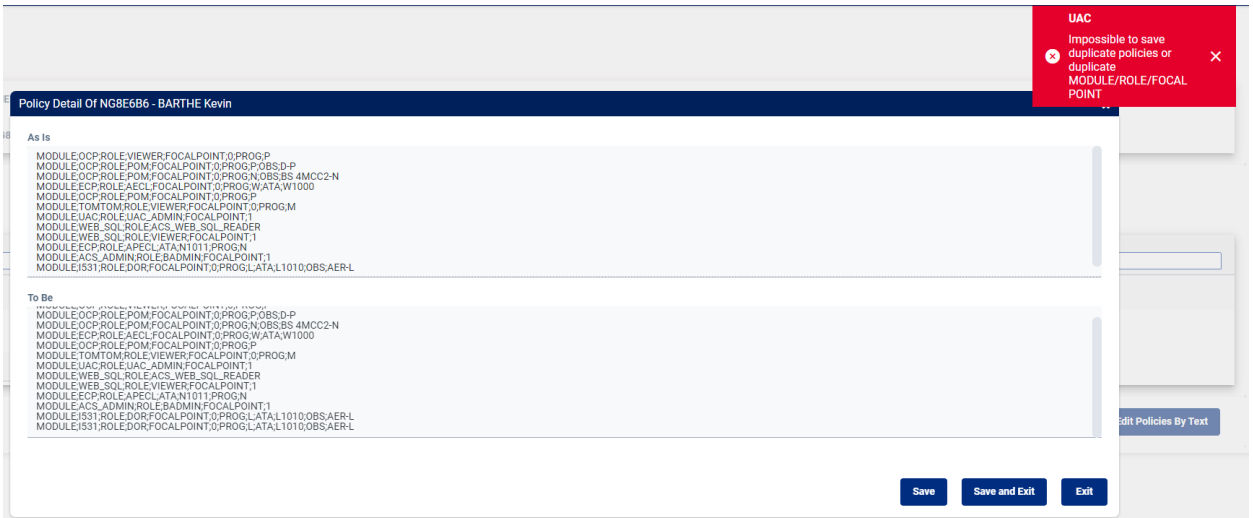
Policy rules are the ones used for table edition. The rules are as follows :

- With one or several Legacy DOMAIN, associated OBS FIC TYPE should be PROD
- With one or several ICC ACP DOMAIN, associated OBS FIC TYPE should be TEAM
- Legacy and ICC ACP DOMAIN can be shared if there is no OBS FIC type/value
- OBS FIC Type PROD and TEAM can be shared if there is no DOMAIN type/value

Otherwise a detailed error appears and the policies are not saved.

NB : As you could save inconsistent policies in edit by table if they're not modified, in Edit by Text all policies must be consistent else the save will be impossible.

In this section, the user is not authorized to duplicate one or several policies. For example if you want to copy/paste one or several policies, we can see an error when the user tries to save the policies or save and exit the edition mode like in the screenshot below :



3. LOADERS

There are 2 types of loader:

- Loader 'Roles' which is used to create/update existing roles
- Loader 'Policies' which is used to load user's policies

3.1 Loader 'Roles'

Loader ‘Roles’ allows to create new roles and associated criteria values in UAC database.

Once created, these new roles will be available for policy creation or ‘Policy’ Loader.

Loader 'Roles' has following format (see attached file)

Column	Explication	Example
C_ROLE_ID		AECL
C_ROLE_Label		Airbus Equipment Change Leader
C_STATUS		active
C_CRITERION_TYPE_CODE	existing code for criterion of the role	PROGRAM, ATA, SUPPLIER
C_CRITERION_TYPE_CODE		
...		

Exemple:

https://drive.google.com/file/d/1Tz-JD-m4asSD8WLq1B130OZ6ZP0zpNJr/view?usp=sharing

3.2 Loader 'Policies'

Loader 'Policies' allow to remove existing policies and create new policies from a flat file for given mandatory tuple <action,user,module,role> and existing criteria types/values. This kind of loader should follow the Role/Module/Criteria scheme. That means that we have a role/module dependency for the criterias definition.

It works in DELTA mode:

- A policy from a CSV file is considered the same as one existing in UAC if the mandatory tuple <user,role,focal point,module> and criteria types and values are exactly the same.
- If the first field is declared as ‘C’ in file, the policy will be loaded if it does not exist yet in UAC.
- If the first field is declared as ‘D’ in file, the policy will be deleted if it exists in UAC.

- Incase of duplicated policies in CSV file (ex: row 1 and 5 have the same tuple and same criteria types and values, not necessarily in the same order), row 1 will be processed (created or deleted) into UAC but row 5 will be ignored.

It shall have following format (see attached file)

Column	Mandatory	Explication	Example
ACTION	Yes		C(reate) or D(elete)
N_USER_ID	Yes		NG73A14
C_ROLE_CODE	Yes		DOR
N_FOCAL_POINT	No		0/1(not empty)
MODULE	Yes		I531
C_CRITERION_TYPE_CODE	N		OBS
C_CRITERION_VALUE_CODE	N		XWNOFSTR, XW AFT
C_CRITERION_TYPE_CODE	N		...
C_CRITERION_VALUE_CODE	N		...
...			

Exemple:

https://drive.google.com/file/d/1n3ick80cNX8KfmxwQSWGQnG7mCJsOvtd/view?usp=sharing

3.2.1 Loader checks

In order to strengthen policies loading and to help business to quickly fix any discrepancies present in the CSV file, a 2-step CHECK mode is available in the policy loader:

- First step will verify if any error related to the CSV parsing is present. If a policy row has an error it will be displayed in the log with the reason:
  - Format of the row not compliant (must contain at least the 4 mandatory fields and having an even number of fields).
  - Empty fields (like NG73A14;DOR;0;I531;PROG;;OBS;XWNOFSTR, XW AFT).
  - Focal point not equal to 0 or 1.
- Second step will be executed ONLY If the first step is OK. This second step will verify that ALL the data in policy rows exists in UAC database.
  - User LDAP uid, role, module will be checked and if value does not exist in the database, an ERROR will be added next to the value. Ex: module=[TOTO[ERROR]]
  - Each criteria type AND value will be checked also.
    - If a criteria **type** does not exist, it will be flagged as PROGRAM[ERROR]=[]
    - If a criteria **value** does not exist, it will be flagged as PROG=[Z[ERROR]]

- If a required criteria value for a role and module is not specified, it will display an error message as following : “The criteria(s) 'XXX;' are required for the role 'ROLE_CODE' and the module ‘MODULE_CODE’” with the specific line number.
- If a criteria value is defined in a policy but cannot be specified for a role/module, it will display an error as following : “The criteria 'XXX' cannot be specified with the role 'ROLE_CODE' and the module 'MODULE_CODE’” with the specific line number.
- If the module are not allowed for the UAC user according to his LDAP sub-Group (declare through **T_MODULE_AUTHORIZED** in UAC_USER**)** then it will display an error message as following : "The GROUP_OCP (LDAP Subgroup) with OCP (module) is not allowed for the UAC user (N_USER_ID)”
  - If Airbus declared in **T_MODULE_AUTHORIZED** no check will be done
  - Check **ONLY** if Module declaration in **T_MODULE_AUTHORIZED** is declared into properties file
- If a role is not allowed for the specified module, it will display an error message as following : "The role 'ROLE_CODE’ is not allowed for the module 'MODULE_CODE'. Only allowed for modules 'XXX;YYY;ZZZ;'"; Allowed roles per modules are in UAC_ROLE_MODULE
- Only one Program per policy is allowed. Error message :"Multiple PROG not authorized"
- The policy program (PROG), ATA program in and OBS program(Defined by column C_PROG in VW_UAC_ATA and VW_UAC_FIC) must all be the same program. Error message : "Policy program (PROG_CODE) and ATA program (PROG_CODE) are different" or "OBS and ATA are present but not on the same program" if Prog is absent
- If several ATA/OBS are present, they must all have the same program . Error message : "Multiple ATA but not on the same program";"Multiple OBS but not on the same program";

In CHECK mode, loading will never be executed. If all policies are OK, the batch will just inform users that CSV file is OK

Column	Mandatory	Explication	Example
ACTION	Yes		C(reate) or D(elete)
N_USER_ID	Yes		NG73A14
C_ROLE_CODE	Yes		POM
N_FOCAL_POINT	No	Focalpoint can now be skipped entirely, value will be set to 0 by default.	0/1(not empty)
MODULE	Yes		I531
C_CRITERION_TYPE_CODE	Yes		PROG
C_CRITERION_VALUE_CODE	Yes		P

C_CRITERION_TYPE_CODE	No		OBS ..
C_CRITERION_VALUE_CODE	No		AAESKD-P
...			

In this example, if the program is not specified for the Role Code “POM” in one or several lines of the loader (.csv) file, in that case, it will display an error message with the line number, and the rejection reason. So we will see which criteria type required for the role is missing.

3.2.2 Loader particularities

- Even if the loader is in LOAD mode, the 2-step checks are always executed to ensure proper loading and data integrity into UAC. If any error is present, the batch won't load anything (even policies that are correct).
- Policy won't be loaded if it has criteria not declared for role (this control isn't implemented in GUI)
- Loader doesn't historize existing policies (historization is done if update is done by GUI)

4. UAC Technical Architecture

4.1. System Level Requirements

4.1.1. Data Requirements

No specific data requirements for the moment.

4.1.2. Data Model

O470_UAC TABLES	*Use*
UAC_CRITERION_TYPE	Policies Criterion types
UAC_ENUM_CRITERION_VALUE	Possible couple (Type/Value) of policy criterion
UAC_POLICY_HIST_CRIT_VALUES	Historized Policy Criterion Values
UAC_POLICY_HISTORY	Historized Policies
UAC_ROLE	Policy Roles
UAC_ROLE_CRITERIA_TYPES	Authorized Criterion values on Roles
UAC_USER	UAC Users Data
UAC_USER_POLICY	Users Policies
UAC_USER_POLICY_CRIT_VALUES	Policies Criterion Values
UAC_MODULE	ACS Modules

UAC_POLICY_HISTORY Table - Column description	Data Type	Value
ID	NUMBER(19,0)	Unique ID technique for the history log
C_UPDATER	VARCHAR2(256 CHAR)	LOGIN ID of the UAC User how update the Policy
D_CREATION_DATE	TIMESTAMP(6)	Creation date of the POLICY
D_HISTORY_DATE	TIMESTAMP(6)	Update date of the POLICY update by C_UPDATER
C_ROLE_CODE	VARCHAR2(64 CHAR)	CODE associate to a specific ROLE
N_USER_ID	NUMBER(19,0)	Unique USER ID number associate to the LOGIN ID
N_VERSION	NUMBER(10,0)	[0]:Created , [1…N] Updated , [0…N] Deleted
N_FOCAL_POINT	NUMBER(1,0)	0, 1 value

POLICY_ID	NUMBER(19,0)	Unique Policy ID
MODULE	VARCHAR2(64 CHAR)	ACS module Name
T_ACTION	VARCHAR2(64 CHAR)	Created, Updated or Deleted

4.1.3. Global Architecture

UAC environments schema is mapped as following:

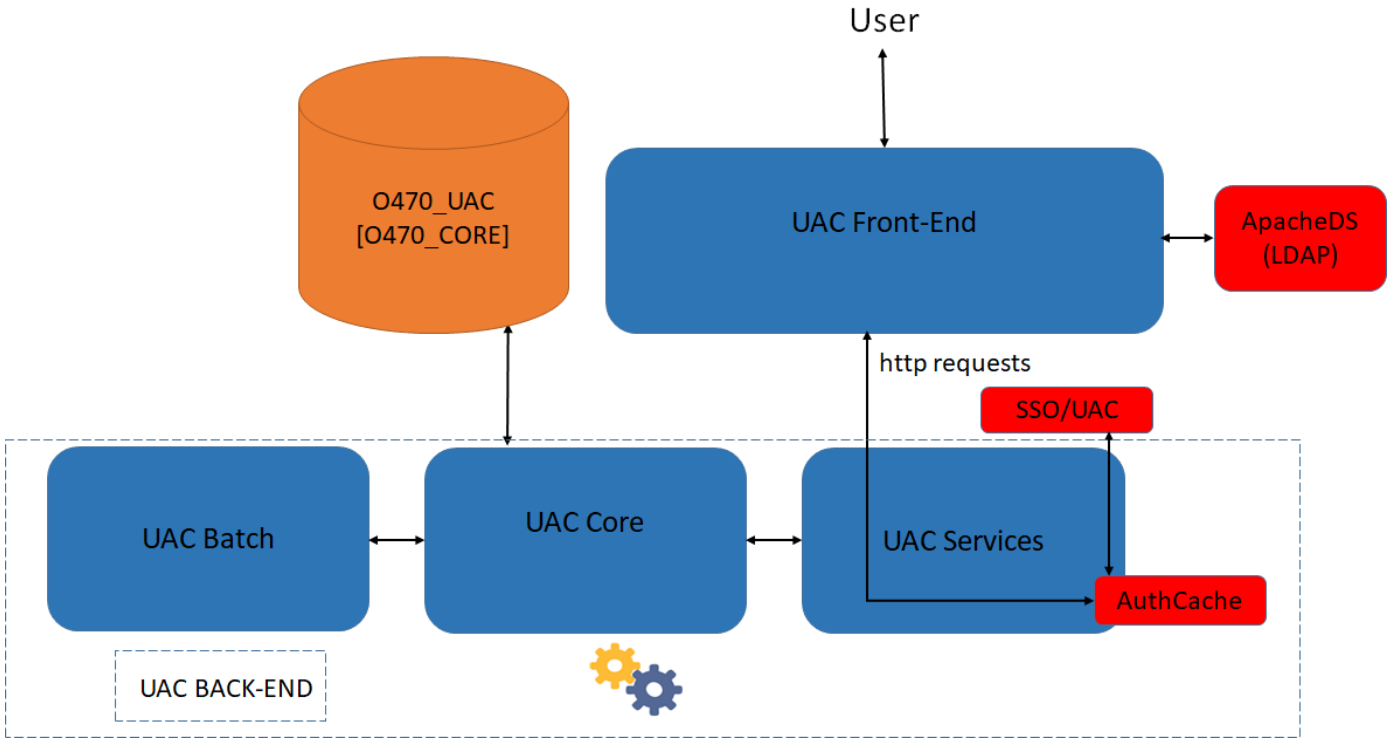
ACS	*Oracle Database*
FFA Project	INT
FFA E2E	INT
VAL	VAL
PROD	PROD

4.1.3.1. Database management

Detailed process:

- When UAC back-end server is deployed in a given environment (FFA Project/FFA E2E/VAL/PROD), Liquibase apply all changes on Oracle O470_UAC:
    - It creates all required tables and their associated constraints (foreign keys, etc...).
    - It creates a sequence for each created table, which will auto-increment the PK id when a new entry will be added, starting from 1.
    - It creates or replaces triggers.
    - It creates or replaces views and synonyms.
    - It gives grants on some tables and views to O470_CORE with grantable option. Grants must be only given to O470_CORE (some old forbidden grants are revoked).
- When UAC Back-End server is deployed in a given environment (int/e2e/val/prod), Liquibase checks all new changes on Oracle O470_UAC and apply them.

4.1.3.2. View and Process



4.2. Development Environment

The following table summarizes the versions of the tools involved in UAC web site development:

Tool name	Version	Description
Eclipse	4.7.1	IDE for Java
IntelliJ	2019.1	IDE for Java
Visual Studio Code	1.32	IDE for Typescript
WebStorm	2019.1 (eval)	IDE for Typescript
Java Development Kit	1.8	Java Language
Spring Boot (web/jpa/security)	2.1.5	Back-End framework
Angular	>7.2	Front-End framework
Slf4j	2.10.0	Logs
Log4j	2.11.2	Logs
Oracle	11g	Database

4.3. Launching the Loaders

WARNING:

- **If the batch has been already launched for the day by another user (FFA Project/FFA E2E/VAL/PROD) or by Control-M (VAL/PROD), you won't be able to launch it as the owner of the log will be the previous account (user or control-M).**
- Go to tmp folder and delete the log file for your day. In VAL/PROD don't forget to believe your log file when you have finished to allow control-M to launch the batch after you.
  - cd /tmp and ls -l to list all the files
  - Log in as root => sudo su then enter your XPLOIT password for the target environment
  - Delete file with rm -rf uac_batch_synchro.<today's date>.log

4.3.1. Loader 'Roles'

In FFA platform, follow steps:

1- Connect to serveur fr0-vsiaas-8039.eu-v.airbus-v.corp (use your own login and password)

2- Put the roles.csv (format describes above) in /tmp folder

3- Launch the script: /local/opt/O470/UAC/UAC-batch/sh/batch.sh loadRoles -path /tmp/your_roles_filename.csv

4.3.2. Loader 'Policies'

In FFA platform, follow steps:

1- Connect to serveur fr0-vsiaas-8039.eu-v.airbus-v.corp (use your own login and password)

2- Put the policy.csv (format describes above) in /tmp folder

3- Launch the script either in CHECK or LOAD mode:

=> CHECK: /local/opt/O470/UAC/UAC-batch/sh/batch.sh loadUserPolicies -check true -path /tmp/your_policy_filename.csv

=> LOAD: /local/opt/O470/UAC/UAC-batch/sh/batch.sh loadUserPolicies -path /tmp/your_policy_filename.csv

Rules implemented for loader are identical to rules implemented for GUI edition.
Therefore some exceptions may be exist for the moment (work in progress).

4.4. LDAP Synchronisations

Synchronisation is done twice a day : 1 pm and 11 pm. Each of them will be done as follows:

- First, Control-M will launch the batch that will synchronise the Airbus LDAP with UAC database.

UAC database is master for authentication (fine grain), therefore it drives also the ACS LDAP used for authorizations (coarse grain).

4.4.1. Global synchronization process

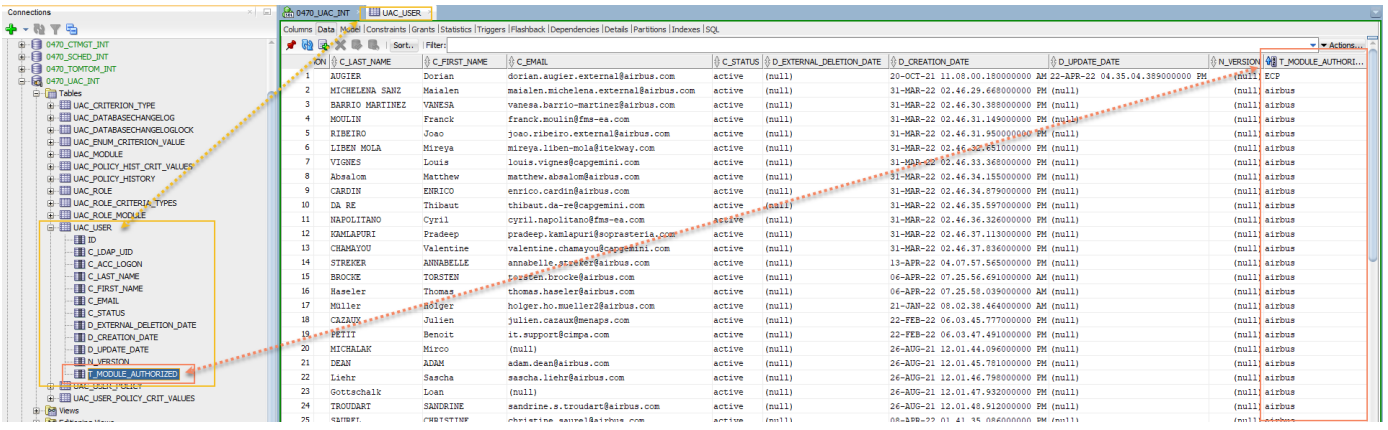
- Control-M drives all the batch synchronization between Airbus LDAP, UAC and ACS LDAP. These synchronizationsThen Control-M will launch the batch that will synchronize UAC with ACS LDAP. Through Control-M, the UAC to ACS LDAP batch will always be launched in delta mode.

4.4.2. Airbus LDAP to UAC

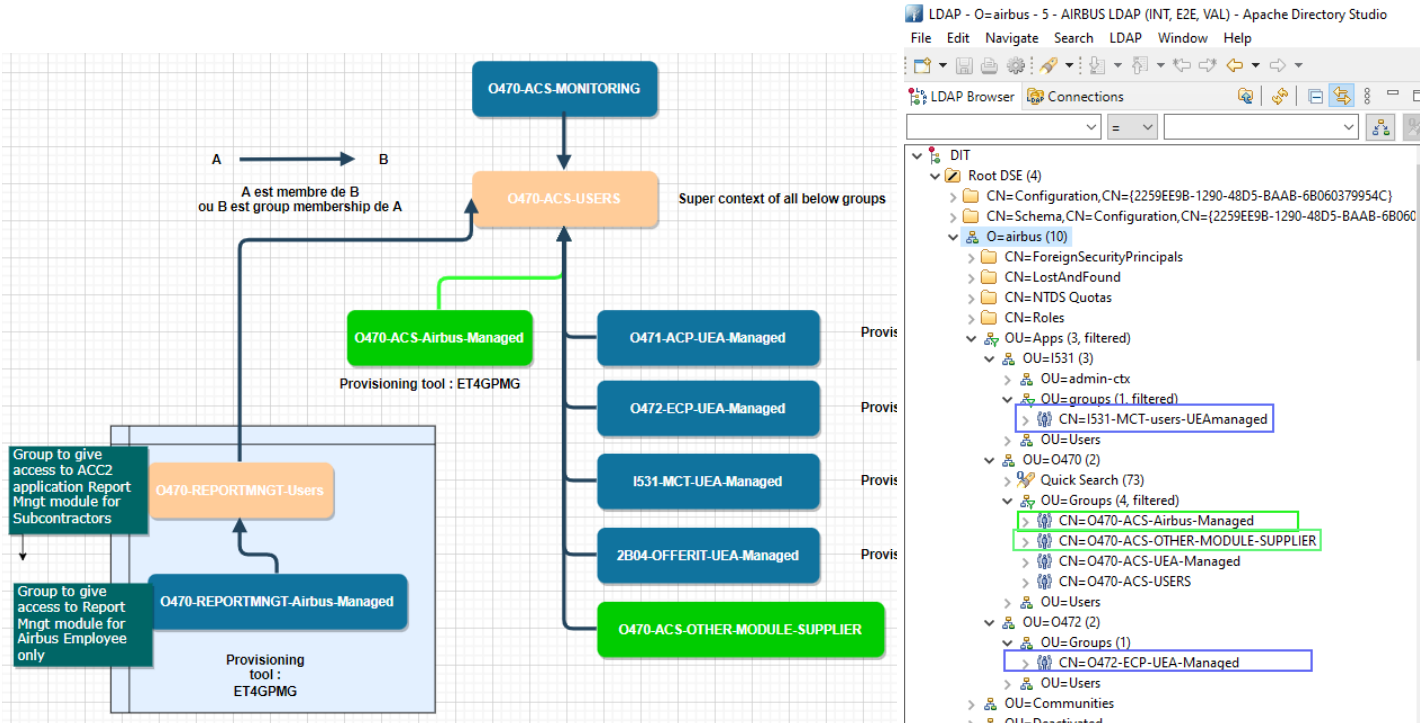
The Airbus LDAP to UAC batch is a full batch so all users in UAC will be checked against Airbus LDAP.

All Users' status and Module authorization in UAC database will be updated accordingly from Airbus LDAP information:

- Users not present in Airbus LDAP but in UAC will be disabled on UAC (status will be set to 'inactive').
- Users present in Airbus LDAP but not in UAC will be added in UAC with a status 'active' and all according to the information, and all the LDAP Sub-Group rule list below:
  - **Rule1**: If the User in UAC is declared in ***AIRBUS-Managed LDAP Sub-Group*** then the User is authorised to access to all the module and the **T_MODULE_AUTHORIZED** attribute is set with Airbus value in the **UAC_USER** Table in O470_UAC database will be set with airbus value (refer to the **Picture.a** below)
  - **Rule2:** If the user is declared only to a **specific LDAP Sub-Group** then the User can access ONLY to the module associated to the Sub-Group(Refer to the **Picture.b** below) and the **T_MODULE_AUTHORIZED** attribute is set with value associate to the specific ACS module (list of the module authorised, separated by ";"). All the right associated to an another module than the LDAP Sub-Group declaration will be deleted according to **T_MODULE_AUTHORIZED** declaration.
- Users present in Airbus LDAP and UAC will have their information updated if any required field has changed (status, external deletion date, first name, last name or email) and according to the LDAP Sub-Group rule above.



Picture.a: UAC_USER table in O470_UAC database, module authorization attribute



Picture.b: Airbus LDAP Sub-Group declaration

4.4.3. UAC to ACS LDAP (Batch)

The UAC to ACS LDAP batch can be launched in full or delta mode. The purpose of this batch is to synchronize ou=users and ou=apps on ApacheDS from users currently in UAC database.

In **Full mode** (generally used to initialize the ApacheDS LDAP):

- All users AND policies in UAC database will be retrieved.
- All users present in ACS LDAP will be deleted.
- *Full load of users in ACS LDAP will begin.*
- UAC users with an 'active' status will be loaded into ApacheDS with the following mapping.

UAC Field	UAC Field Value	ApacheDS ou=users, o=airbus
C_STATUS	active	User will be added in directory
C_STATUS	inactive	no action
C_LDAP_UID	any <value>	dn = cn=<value>, ou=users, o=airbus user.cn=<value> and user.uid=<value>
C_LAST_NAME	any <value>	user.sn=<value>

C_EMAIL	any <value>	user.mail=<value>

- When all users are loaded, full load of modules in ACS LDAP will begin.
- All modules present in ACS LDAP will be deleted
- For each policy related to a user, if the user is active, the user dn will be built (ex: cn=NG7BB0F, ou=users, o=airbus) and added to the members of the corresponding module into ApacheDS if the user is not already a member of this module.
    - For example an active user NG7BB0F on UAC has two policies : one on MODMGT module and one for UAC module. In this case we will have:
        - In cn=MODMGT,ou=apps,o=airbus we will have a new value in the member attribute of this module: cn=NG7BB0F, ou=users,o=airbus
        - In cn=UAC,ou=apps,o=airbus we will have a new value in the member attribute of this module: cn=NG7BB0F, ou=users,o=airbus
- if a policy is related to a module which is not in the ACS LDAP, the module is created and the user added as a member.

In **Delta mode** (used for ACS LDAP update without service outage):

- All users AND policies in UAC database will be retrieved.
- *Delta load of users in ACS LDAP will begin.*
- All users in ACS LDAP will be retrieved and mapped by their ldap Uid. Then the delta load of users will be processed in 3 steps:
    - 1 - All inactive users in UAC will be **removed** from ou=users, o=airbus if they are present.
    - 2 - All active users in UAC will be **added** in ou=users, o=airbus if they are not present.
    - 3 - All active users in UAC which are present in ou=users, o=airbus will be **updated** if some specific fields has changed:

UAC Field	UAC Field Value	ApacheDS ou=users, o=airbus
C_LAST_NAME	any <value>	user.sn=<value>
C_EMAIL	any <value>	user.mail=<value>

- *When all users are loaded, delta load of modules in ACS LDAP will begin.*
- All modules in ACS LDAP will be retrieved and mapped by their module code.
- All policies will be parsed to create a map of <Module Code, Set<SimpleUser>>. At this step, users in each module could be active or inactive.
- For each ACS LDAP module:
    - For the corresponding module in UAC, the active users in the list of users are retrieved and added to the members of that module.
- All ACS LDAP modules are saved.
- Then the delta load of users will be processed in 3 steps:
- For each policy related to a user, if the user is active, the user dn will be built (ex: cn=NG7BB0F, ou=users, o=airbus) and added to the members of the corresponding module into ApacheDS.
    - For example an active user NG7BB0F on UAC has two policies : one on MODMGT module and one for UAC module. In this case we will have:

- In cn=MODMGT,ou=apps,o=airbus we will have a new value in the member attribute of this module: cn=NG7BB0F, ou=users,o=airbus
- In cn=UAC,ou=apps,o=airbus we will have a new value in the member attribute of this module: cn=NG7BB0F, ou=users,o=airbus

4.4.4. UAC to ACS LDAP (Services)

UAC also updates the ACS LDAP on the fly when a policy is changed on an active user.

On adding a new policy to a user:

- The user will be added to the members of the module if they are not already present

On modifying an existing policy:

- Modifying the module of a policy is autorized in UAC which can have an impact on ACS LDAP
- A delta is done on all the policy linked to the user to update the ACS LDAP
- if the user is linked to a module but not considered as a member, they will be added
- if the user is a member of a module but has no more policies linked to it, they will be deleted from the members

On deleting a policy:

- if the user has no more policies linked to the concerned module, they will be deleted from the members list else they will remain

4.5. UAC-external

UAC database is master for authentication (fine grain), therefore it drives also the ACS LDAP used for authorizations (coarse grain).

4.5.1. Description

Any ACS application can view UAC users and their policies on their own schema through 2 synonyms from O470_CORE:

- O470_UAC_USER
- O470_UAC_USER_POLICY

To access and read these synonyms, a specific module named UAC-external is available on Artifactory and can be added easily as a dependency to any ACS project with Maven or Gradle.

4.5.2. Connection between any ACS project and UAC data

Follow these steps to fully configure your project for UAC authentication.

- Add UAC-external dependency in the pom.xml of your project. Remove any other dependency related to UAC-Core if present. Current version of UAC-external is 1.0.2.

```
<dependency>
<groupId>com.airbus.idcc.iccservices.uac</groupId>
<artifactId>UAC-external</artifactId>
<version>1.0.2</version>
</dependency>
```

- Fix build errors:
  - Update the classpath in your datasource configuration class

```
com.airbus.idcc.iccservices.uac.external
```

- Initialize UAC-external repositories with your project datasource
  - Init DAO beans

```
@Bean(name = "uacUserDao")
public IUacUserDao uacUserDao(@Qualifier("OracleDataSourceNOMENCLATURE") DataSource
dataSource) throws SQLException {
IUacUserDao uacUserDao = new UacUserDao();
uacUserDao.setDatasource(dataSource);
return uacUserDao;
}
```

```
@Bean(name = "uacUserPolicyDao")
public IUacUserPolicyDao uacUserPolicyDao(@Qualifier("OracleDataSourceNOMENCLATURE")
DataSource dataSource) throws SQLException {
IUacUserPolicyDao uacUserPolicyDao = new UacUserPolicyDao();
uacUserPolicyDao.setDatasource(dataSource);
return uacUserPolicyDao;
}
```

- Update the autowired for the beans in CustomUserDetailsService class and eventually other classes (spring security, unit tests...)

```
@Autowired
private IUacUserDao userDao;

@Autowired
private IUacUserPolicyDao policyDao;
```

- Fix remaining imports and the getEmail() method as its name has been changed from geteMail() to getEmail().
- Update any test classes (Unit/Integration)

4.6. Error handling

UAC monitors any error occurring on the Front-End or Back-End.
These errors are saved in log files that are accessible on VM or ELK GUI.

4 files contains the logs:

- **uac.log** stores all the applicative logs.
- **monitoring.log** stores all the errors detected by Error Monitoring Service (from Front or Back).
- **diagnostic.log** stores all the diagnostic log from server, database, LDAP server and properties (if readable or not).
- **springboot.log** stores all the related Spring logs (server start/stop, etc...).

4.6.1. Front-End

Any error detected in running UAC module on browser is sent to UAC Back-End to be stored in the monitoring log.
For example:

```
2021-06-07 12:17:45.709 [ERROR] [http-nio-8080-exec-10] ErrorMonitoringService:31 -
User NG7BB0F had the error UAC Front TypeError: TypeError: Cannot read property 'get'
of null
```

The error will also be displayed on the GUI to inform the user.

4.6.2. Back-End

Any exception detected will be caught and written in the monitoring log.
For example:

```
2021-06-07 12:22:04.075 [ERROR] [http-nio-8080-exec-3] ErrorMonitoringService:31 -
User NG7BB0F had the error java.lang.NullPointerException:
java.lang.NullPointerException
```

The error will also be displayed on the GUI to inform the user.