COLUMN
Information Security

ARAMCO

IAM IMPLEMENTATION - WORKFLOWS

COLUMN INFORMATION SECURITY
03/27/2018

CORPORATE HEADQUARTERS
10 E. 22nd Street, Suite 300
Lombard, IL 60148
Phone: 630.515.6660
Fax: 630.271.1508
https://www.columnit.com

NORTHEAST HEADQUARTERS
Wall Street Plaza
88 Pine Street
Suite 501, Water Street Entrance
New York, NY 10005
Phone: 212.227.3232

Table of Contents

Version Control

Date	Author	Version	Change Reference
03/27/2018	Shrijit Chandran	1.0	Initial Document
05/10/2018	Shrijit Chandran	1.1	Updated Lifecycle Event Workflows
05/16/2018	Shrijit Chandran	1.2	Updated On Demand Workflows

Project Title

Identity and Access Management for Aramco Services Corporation (Aramco).

Purpose of this Document

This document describes the major workflows configured during development of IAM system using Sailpoint Identity IQ 7.0 p8 for Aramco Services Corporation (“Aramco”) . For Classification the workflows have been categorized into two broad categories: 1) Life Cycle Event Based Workflows and 2) Non-Life Cycle Event Based Workflows or On-Demand Workflows.

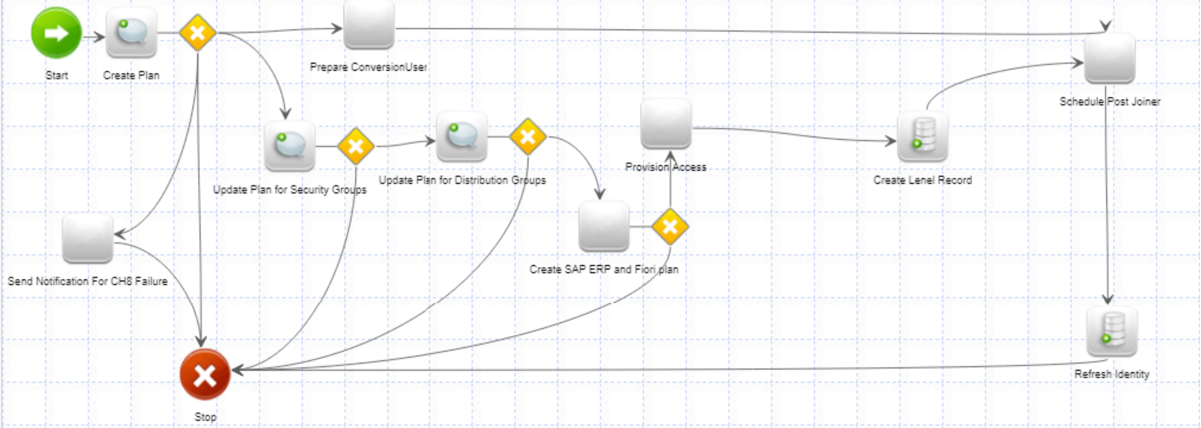
Life Cycle Events:

1.1 Joiner

- Trigger: ASC-Joiner
  - Type: Create
- Population: *SAP HR Accounts*
  - *Definition: All Identities with SAP HR account.*
- *Workflow: ASC-Lifecycle Event - Joiner v2*

1.1.1 ASC-Lifecycle Event - Joiner v2

Triggered by Identity trigger of Identity creation. Primary workflow responsible for creating accounts.


Start
Create Plan
Prepare ConversionUser
Schedule Post Joiner
Update Plan for Security Groups
Update Plan for Distribution Groups
Provision Access
Create Lenel Record
Create SAP ERP and Fiori plan
Send Notification For CH8 Failure
Refresh Identity
Stop

Custom defined Workflow Variables:

Variable Name	Comment
Identityname	Input variable. Holds identity name for whom workflow triggered
Project	Variable to hold project prepared, passed across steps
isConversionUser	Variable to define conversion. Initialized to false.
Plan	Variable to hold plan prepared , passed across steps
tempPassword	Variable to hold password generated, passed across steps
calEmpEndDate	Variable to hold user end date, passed across steps
identityDisplayName	Variable to hold displayname for identity. Calculated from identity names.
Department	Variable to hold department for identity. Calculated from departmentCode of identity.
Division	Variable to hold Division for identity. Calculated from divisionCode identity attribute
employeeType	Variable to hold Division for identity. Calculated from employeeType identity attribute.
empEndDate	Variable to hold Division for identity. Calculated from employeeEndDate identity attribute.
userManager	Variable to hold Division for identity. Calculated from displayName attribute of identity manager. If manager not present then set as spadmin

Referenced Rules:

- ASC-Rule-SP Util Rule Library
- ASC-Rule-Constants

- Workflow Library

Workflow steps

- **Start:**
    - Start Joiner process
    - Next Step -> Create Plan
- **Create Plan**
    - Reads attributes from SAP HR account of the identity and creates AD plan.
    - If employeeType z1, w5 or w7 (ch8 employees), uses SY-UNAME from SAP HR to use in sAMAccountName and related AD fields.
    - If network ID populated as a part of SAP HR (SY-UNAME); checks if any other link already exist with same network ID. If links are found, isConversionUser flag is marked as true. A variable *conversionAccId* is saved with the value of the user's network ID whi holds the matching links. This will be the converted from Identity. Also save an Audit entry for the same.
    - If user is of type WG , W5 or w7 and does not have an employeeEndDate or set as 0000-00-00 then employee end date is set as 3 months from current end date.
    - Set attributes from SAP HR for AD Plan
    - Transitions:
        - Step to Add Security groups to plan
        - To prepare conversion users: if isConversionUser is set as true
        - To Send chapter 8 failure: if user of ch8 type and SY-UNAME from SAP HR not available.

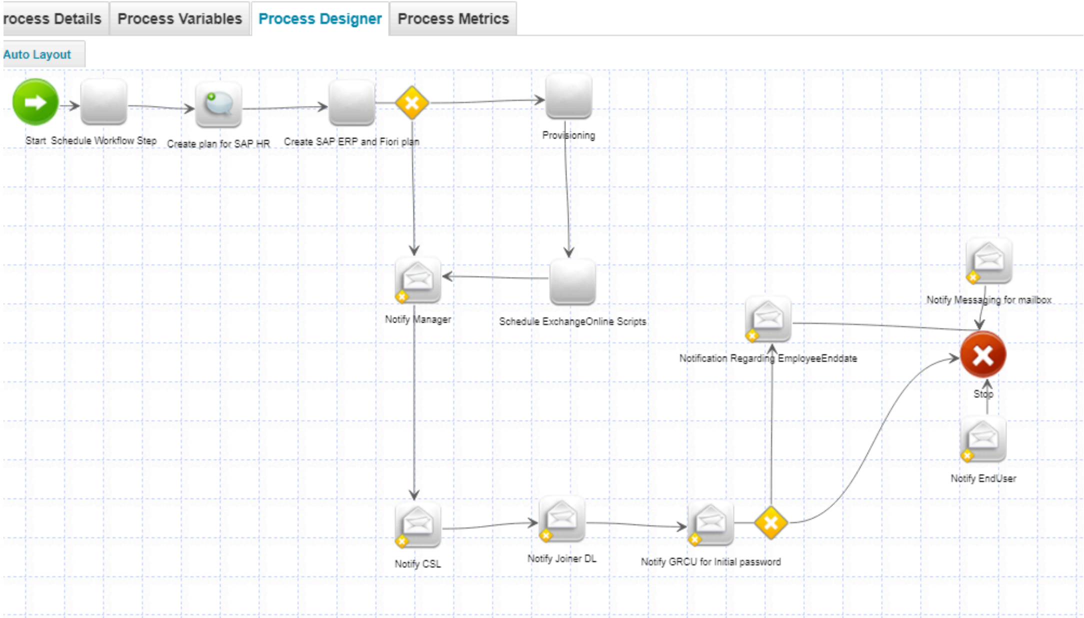
- **Update Plan for Security Groups:**
    - Get list of Security groups based on orgKey of the identity from custom object" *ASC-Security Groups*"
    - Transition: Update Plan for Distribution Groups

- **Update Plan for Distribution Groups:**
    - Calls sp Util library method: build_DistributionGroup_List() to get list of distribution groups based on the identity departmental info. Methods checks for identity's company, department, division ,unit and orgKey to define distribution groups.
    - Transition:
        - Create SAP ERP and Fiori Plan if plan is not null
        - Stop
- **Create SAP ERP and Fiori Plan:**
    - Create Account request for SAP ERP and SAP Fiori based on identity attributes and add to the plan from previous step.
    - Transition :
        - Provision Access when plan is not null
        - Stop
- **Provision Access:**
    - Calls for OOB LCM provisioning workflow.
    - Transition: Create Lenel Record
- **Create Lenel Record:**
    - Gather identityAttributes : firstName, lastName, middleName, employeeId, employeeType.
    - Create an SQL insert query with connection details from LENEL application. Updates EMP table and UFEMP table.
    - Transition: Schedule Post Joiner.
- **Schedule Post joiner:**
    - Initialize the workflow : *ASC-PostJoiner-Workflow* to execute after 60 minutes. Arguments send to workflow are: identityName, event, tempPassword, UserName, empEndDate
    - Transition: Refresh Identity
- **Refresh Identity:**
    - Calls for workflow library method: refreshIdentity , with provisionAssignment, promote attribute and correlate entitlements to true. This step will result in assigning the IT roles assigned to identity as a result of birthright business roles.

  - Transition : Stop
- **Send Notification for ch8 failure:**
  - Calls for powershell which sends emails. Powershell folder: D:\Install\Scripts\Email_Notification\NewHireNotification
- **Prepare Conversion User:**
  - Get the conversionIdenName (converted from identity) and captures its Fiori, AD and ERP link.
  - Move over the links from that coverted from identity to current identity by calling an SP util library method: moveAccountsToConversionUser
  - Transition: Schedule Post joiner.
- **Stop :** End of Workflow

1.1.2 ASC-PostJoiner-Worfklow

This workflow is called from Joiner workflow and performs following functions : send notifications, update SAP HR, schedule Exchange scripts.



Custom defined Workflow Variables:

Variable Name	Comment
Identityname	Input variable
employeeEndDate	Input variable
calEmpEndDate	Input variable
Username	Input variable
tempPassword	Input variable
calEmpEndDate	Input variable
identityDisplayName	Variable to hold displayname for identity. Calculated from identity names.
Department	Variable to hold department for identity. Calculated from departmentCode of identity.
Division	Variable to hold Division for identity. Calculated from divisionCode identity attribute
employeeType	Variable to hold Division for identity. Calculated from employeeType identity attribute.
empEndDate	Variable to hold Division for identity. Calculated from employeeEndDate identity attribute.

userManager	Variable to hold Division for identity. Calculated from displayName attribute of identity manager. If manager not present then set as spadmin
sendMails	Flag variable to turn off/on sending notifications
employee_Email_ID	Variable to hold mail attribute of identity.
accList	List type variable containing name of all accounts provisioned to identity except SAP HR
CurrentDate	Variable holds current date
CSL_Recipient	Variable holds email id of CSL for the user. Calculated using custom objects ASC-Custom-CSL-Joiner , ASC-Custom-CompanyCode-ADGroup-Mapping and referes to SP UTIl Linrary method: getCodeFromIdAttribute
userManager_mailID	Variable to hold Division for identity. Calculated from email attribute of identity manager. If manager not present then set as “tuan.le@aramcoservices.com”.
GRCU_MailID	Gets GRCU email id from workgroup: GRCU
HelpDesk_mailID	Gets Helpdesk email id from workgroup: Helpdesk
LenelAdmin_mailID	Gets Lenel Admin email id from workgroup: LenelAdmin
Messaging_mailID	Gets Messaging Group email id from workgroup: Messaging Group
adMailID	Gets mail attribute from identity’s ad link
joinerDL	Gets All Static Joiner Notifications: Set to: ascitdgrcuiam3@aramcoservices.com
joinerPasswordDL	Gets joiner’s password: ascitdgrcuiam@aramcoservices.com

Referenced Rules:

- ASC-Rule-SP Util Rule Library
- ASC-Rule-Constants
- Workflow Library

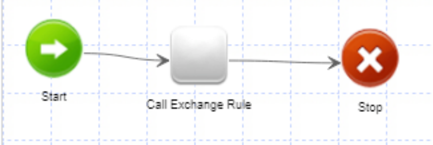
Workflow Steps:

- **Start**
    - Starts the Post joiner process.
    - Transition : Create Plan for SAP HR
- **Create Plan for SAP HR**
    - Create an account request to update identity’s SAP HR with attributes : *System user name (SY-UNAME)* and *Email* with network ID and mail attribute from ad link.
    - Transition: Create SAP ERP and Fiori plan
- **Create SAP ERP And Fiori Plan**
    - Check if identity has SAP ERP and Fiori links. If present create an account request to create SAP ERP and Fiori accounts. If accounts already preset create an account request to update SAP EERP and Fiori with “Email” attribute with value from AD “mail”.
    - Transition:
        - Provisioning: If plan is not null
        - Notify Manager
- **Provisioning:**
    - Call OOB LCM provisioning Workflow to compile and provision the plan prepared.
    - Transition:
        - Schedule Exchange Online Scrips
- **Schedule Exchange Online Scripts:**
    - Calls Worklow “ASC-CallScripts-ForExchangeOnline” to execute Exchange online scripts via powershell, this enabled the azure mailbox for the user.

- Transition:
    - Notify Manager
- **Notify Manager:**
    - Send Notification to user manager using powershell script for sending email. Powershell script located at: D:\Install\Scripts\Email_Notification\NewHire_Notification
    - If Manager not available. Notification sent to: tuan.le@aramcoservices.com
    - Transition: Notify CSL
- **Notify CSL:**
    - Send email to CSL email ID identified for identity using PowerShell script for sending email. PowerShell script located at: D:\Install\Scripts\Email_Notification\NewHire_Notification.
    - If CSL mail id not available. Notification sent to: tuan.le@aramcoservices.com.
    - Transition: Notify Joiner DL
- **Notify JoinerDL:**
    - Send email to joinerDL email ID configured using PowerShell script for sending email. PowerShell script located at: D:\Install\Scripts\Email_Notification\NewHire_Notification.
    - If joinerDL mail id not available. Notification sent to: tuan.le@aramcoservices.com.
    - Transition:
        - Notify GRCU for Initial Password
- **Notify GRCU for Initial password**
    - Send email to LenelAdmin email ID configured using PowerShell script for sending email. PowerShell script located at: D:\Install\Scripts\Email_Notification\NewHire_Notification.
    - If LenelAdmin mail id not available. Notification sent to: tuan.le@aramcoservices.com.
    - Transition:
        - Notification Regarding Employee End Date: when empEnDate = 0000-00-00
        - Stop
- **Notification Regarding Employee End Date**
    - Send Email to tuan.le@ aramcoservices.com when an employee end date is set incorrectly in SAP.
    - Transition:
        - Stop

1.1.3 ASC-CallScripts-ForExchangeOnline

This workflow is used to call Exchange scripts which triggers joiners Online mailbox.



Custom defined Workflow Variables:

Variable Name	Comment
Identityname	Input variable
networkID	Input variable
Trace	Set to True

Reference Library: ASC-Rule-SP Util Rule Library

Workflow Steps:

- Call Exchange Rule:
    - Gets Active Directory Application object and get the password used to connect with AD
    - Prepare a data map with networkID, mail ID and Rule containing exchange powershell scripts.
    - Call IQ Service executeRPCService method to execute the powershell scripts on the IQ Server.

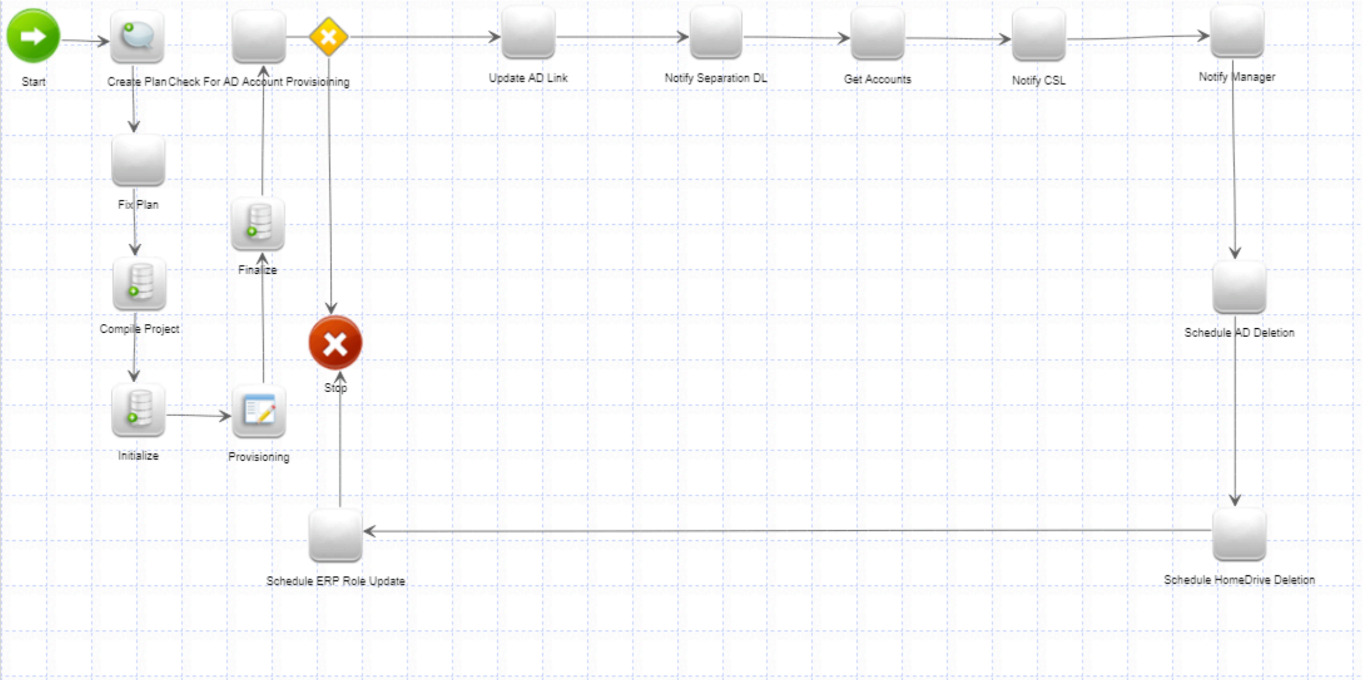
- Rule containing powershell script: ASC-Rule-ActiveDirectory_ExchangeOnline
- Powershell Scripts for each identity:

  $enableMailBox = Enable-RemoteMailbox $sAMAccountName -RemoteRoutingAddress $cloudMail
  Start-Sleep -s 5
  $setMailbox= Set-RemoteMailbox $sAMAccountName -EmailAddresses $SMTPEmail,$localMail,$cloudMail -EmailAddressPolicyEnabled $false
- Transition: Stop

1.2 Leaver

1.2.1 ASC-Lifecycle Event – Leaver_V2

This workflow is the primary workflow which handles terminations. It is triggered via Identity trigger and on Identity attribute Inactive.



Custom workflow variables

Variable Name	Identity Attribute / Comment
Identityname	Input variable
acctList	Input variable
Project	Holds the provisioning project created in the steps
separationDL	Email to notify on sepration. Set to: ascitdgrcuiam2@aramcoservices.com
Plan	
CurrentDate	Todays date
terminationDate	sapActionStDt identity Attribute (Date when the termination action is started)
identityDisplayName	Identity's display name
identityEmail	Identity Email
identityEmployeeId	employeeId
Identity Full Name	firstName + LastName
identityNetworkID	networkID
Department	departmentCode
Division	divisionCode
employeeType	employeeType
userManager	displayName of identity's manager
userManager_mailID	Mail Id of user's manager
CSL_recipient	CSL email id calculated for identity

Referenced Libraries:

- ASC-Rule-SP Util Rule Library
- ASC-Rule-Constants

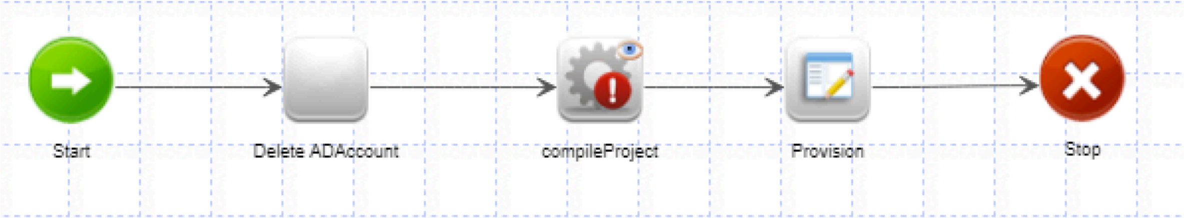
- Workflow Library

Workflow Steps:

- Create Plan:
  - Calls OOB buildEventPlan method with disable operation as argument to get disablement plan for all accounts of the identity
- Fix Plan:
  - Update the AD Plan to set OU of the account to disabled OU
  - Update the SAP ERP and SAP Fiori Plan to Valid to Date as termination date.
- Compile Project, Initialize, Provisioning, Finalize:
  - All these steps call the OOB provisioning methods to complete the provisioning of the plan prepared earlier.
- Check for AD Account Provisioning:
  - Check AD Account provisioning error (if any)
  - Transition:
    - Update AD Link
    - Stop
- Update AD Link:
  - Update the identity's AD link with disable OU distinguished name
  - Transition: Notify Separation DL
- Notify Separation DL:
  - Send Email notification to separation DL using the PowerShell module. Located: D:\Install\scripts\Email_Notifications\Termination_Notification\Generic_Newhire.ps1
  - Transition: Get Accounts
- Get Accounts:
  - Prepare a list of all accounts user have except SAP HR
  - Transition: Notify CSL
- Notify CSL:
  - Send Notification to CSL calculated. If not available send to : tuan.le@aramcoservices.com
  - Transition: Notify Manager
- Notify Manager:
  - Send Notification to employee manager. If not available : send to tuan.le@aramcoservices.com
  - Transition to: Schedule AD Deletion
- Schedule AD Deletion:
  - Schedule a workflow to delete AD account. Workflow is schedule at 180 days from termination. Workflow name "ASC-DeleteADAccount"
  - Transition: Schedule Home Drive Deletion
- Schedule Home Drive Deletion:
  - Schedule a workflow to run after 60 days and delete users home Drive.
  - Workflow name: "ASC-DeleteHomeDrive"
  - Trasition: Schedule ERP Role Update
- Schedule ERP Role Update:
  - Schedule a workflow to update ERP roles to delmit them to current date. Workflow name: "ASC_Update_ERP_Roles_PostLeaver". Schedule to run after 15 mins.
  - Transtion : Stop

1.2.2 ASC-DeleteADAccount

Workflow called by Leaver workflow to delete AD account. Leaver workflow schedule this ti run after 180 days of processing leaver.



Variable Name	Identity Attribute / Comment
Identityname	Input variable

Referenced Libraries:

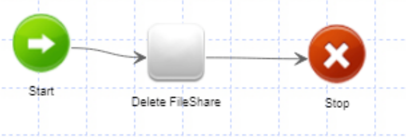
- ASC-Rule-SP Util Rule Library

Workflow Steps:

- Delete AD Account:
    - Check if account I still disabled and is in the disabled OU, then prepare a plan ro delete AD account.
- Compile Project, Provision Project:
    - Calls OOB library method to compile and provision the project plan.

1.2.3 ASC-DeleteHomeDrive

Workflow responsible for deleting Home Drive for a user. Called from AS_Leaver_V2 workflow.



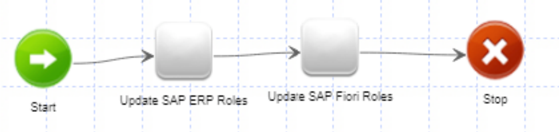
Variable Name	Identity Attribute / Comment
Identityname	Input variable
networkID	Input- networkID of the identity

Workflow Steps:

- Delete FileShare:
    - Delete theHome Drive for the user under \\asc-llocal\users\ using native powershell scripts.
    - Call powershell script using native rule: "ASC-Rule-ActiveDirectory_DeleteHomeDrive"
    - Before deleting validate the account is still in disabled accounts OU

1.2.4 ASC_Update_ERP_Roles_PostLeaver

Workflow called by ASC_Leaver_v2 to delimit SAP Role after termination.



Variable Name	Identity Attribute / Comment
Identityname	Input variable
Username	Input- networkID of the identity
noFiltering	set as true. Causes plan to not filter entitlements when compiling
CurrentDate	Today's Date

Workflow Steps:

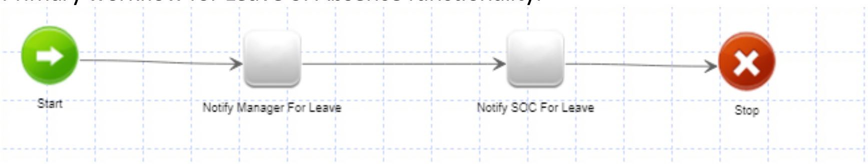
- Update SAP ERP roles :
  - Calls BAPI_USER_GET_DETAIL and ACTIVITYGROUPS to get all SAP ERP roles of the users.
  - Update every role starting with z (primary roles) and set the FROM Date and TO_DATE . TO_DATE is set to currentDate
  - Updates ACTIVITYGROUPS under BAPI_USER_ACTGROUPS_ASSIGN function with update role information
  - Transition : Update SAP Fiori Role
- Update SAP Fiori Roles:
  - Calls BAPI_USER_GET_DETAIL and ACTIVITYGROUPS to get all SAP Fiori roles of the users.
  - Update every role starting with z (primary roles) and set the FROM Date and TO_DATE . TO_DATE is set to currentDate
  - Updates ACTIVITYGROUPS under BAPI_USER_ACTGROUPS_ASSIGN function with update role information
  - Transition : Stop

1.3 Leave of Absence:

Leave of Absence (L.O.A) is an activity when an employee goes for a temporary leave. Leave of Absense workflow is triggered using. a trigger rule "ASC-Rule-LOAStart". The rule validates for action type. Current defined action types are AE and AG.

1.3.1 ASC-LifeCycleEvent-Schedule-LeaveOfAbsence-Workflow

Primary workflow for Leave of Absence functionality.



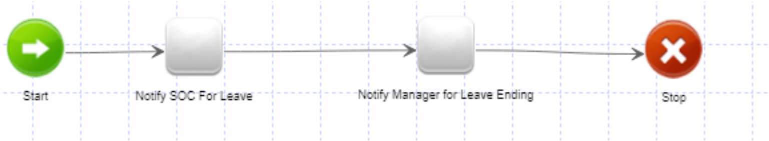
Variable Name	Identity Attribute / Comment
Identityname	Input variable
Username	Input- networkID of the identity
identityDisplayName	Display name calculated using firstname and lastname
CurrentDate	Today's Date
LOAStartDate	sapActionStartDt attribute
userManager_mailID	Mail ID attribute of the employee's manager. If not available selects tuan.le@aramcoservices.com
userManager	Manager name of the identity If not available set as spadmin
Department	departmentCode
Divison	Division
LOA Monitoring	Workgroup LOA_Monitoring's mail id. If not available then tuan.le@aramcoservices.com

Workflow Steps:

- Notify Manager for Leave
  - Calls PowerShell code to notify manager about the leave for the employee.
  - PowerShell used: D:\\Install\Scrips\\Email_Notification\Leave_Notification_Send_Leave_Notification.ps1
- Notify SOC For leave:
  - Notify SOC team for leave , in case they need to put the identity on monitoring list.
  - PowerShell used: D:\\Install\Scrips\\Email_Notification\Leave_Notification_Send_Leave_Notification.ps1

1.3.2 ASC-LifeCycleEvent-Schedule-LOAReturn-Workflow

This workflow is primary responsible to take action on the end of leave for an employee. This is triggered via a rule: "ASC-Rule-LOAReturn". The rule refer to the action type to initiate the LOA Return. Currently configured Action type for LOAreturn is : AH



Variable Name	Identity Attribute / Comment
Identityname	Input variable
Username	Input- networkID of the identity
identityDisplayName	Display name calculated using firstname and lastname

CurrentDate	Today's Date
LOAStartDate	sapActionStartDt attribute
userManager_mailID	Mail ID attribute of the employee's manager. If not available selects tuan.le@aramcoservices.com
userManager	Manager name of the identity If not available set as spadmin
Department	departmentCode
Divison	Division
LOA Monitoring	Workgroup LOA_Monitoring's mail id. If not available then tuan.le@aramcoservices.com

Workflow Steps:

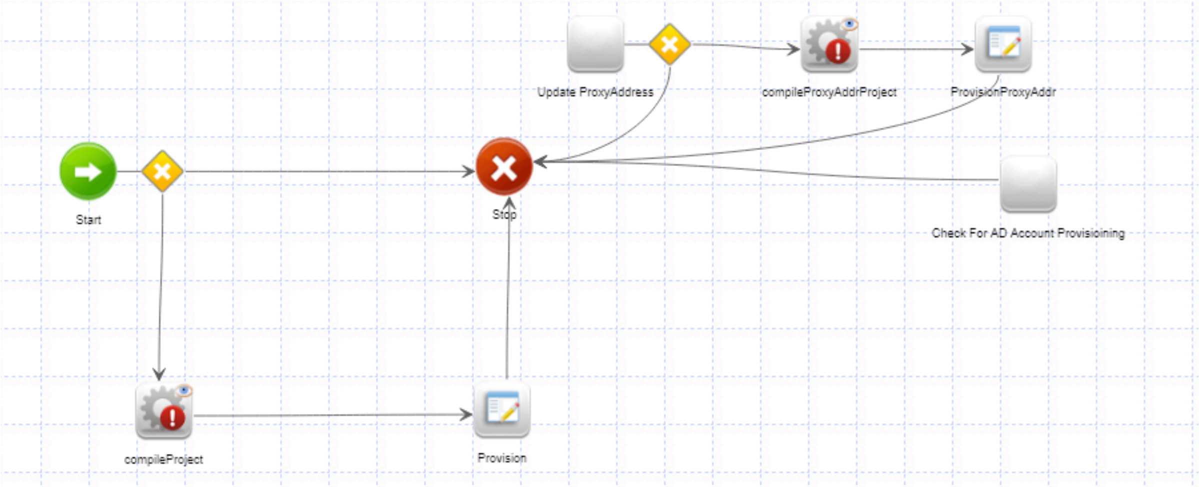
- Notify SOC For leave:
    - Notify SOC team for leave , in case they need to put the identity out of
    - monitoring list.
    - PowerShell used: D:\\Install\Scrips\\Email_Notification\Leave_Notification_Send_Leave_Notification.ps1
- Notify Manager for Leave
    - Calls PowerShell code to notify manager about the leave ending for the employee.
    - PowerShell used: D:\\Install\Scrips\\Email_Notification\Leave_Notification_Send_Leave_Notification.ps1

1.4 Name Change

Name Change Workflow handle scenarios when a person changes names mostly last name due to various reasons.
There are two name change workflows ie. Firstname change workflow and lastname change workflow

1.4.1 ASC-LifeCycle Event -NameChange

This workflow is called when the lastname or firstname of an identity changes. The workflow trigger is attribute : lastname and is restricted to SAP HR population.



Custom Variables:

Variable Name	Identity Attribute / Comment
Identityname	Input variable
newEmployeeLastName	New version of Identity's last name
oldEmployeeLastName	Old version of identity's lastname (previous lastname)
oldEmailID	Mail attribute from from old version of identity

Referrenced library:

- ASC-Rule-SP Util Rule Library

Workflow Steps:

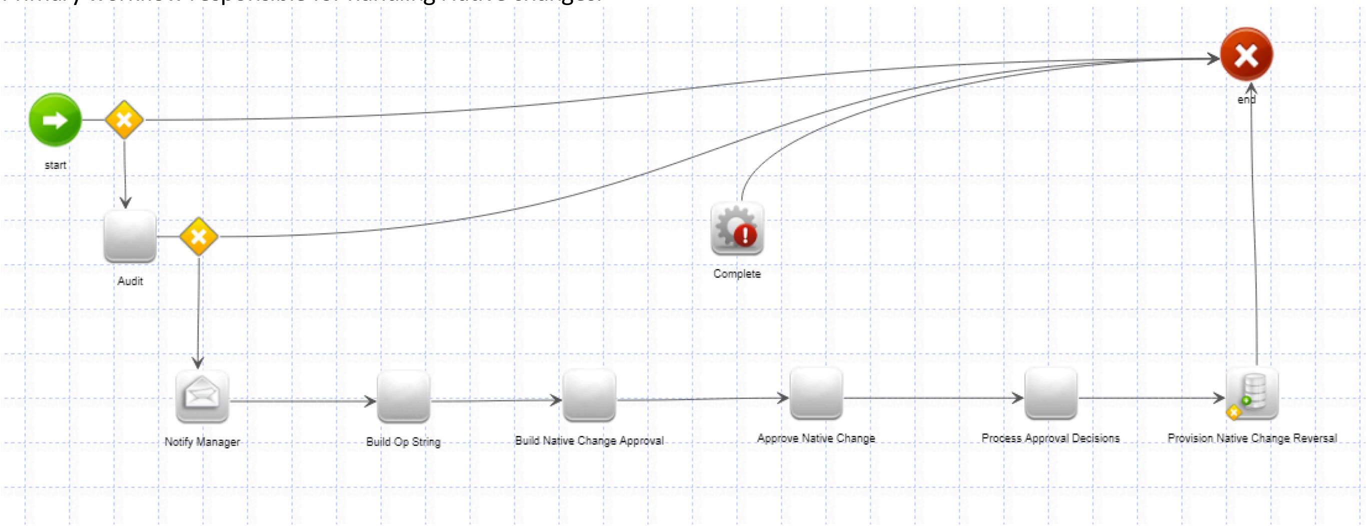
- Start
  - Generate new email address since lastname has changed by calling library method "GenerateEmailAddress()".
  - Get the proxyAddress list from the link and add oldEmail address to proxy address list
  - Prepare and account request to modify the AD link.
  - Add the account request to plan
- Compile Project, Provision Project:
  - Calls OOB library methods to compile and provision project.

1.5 Native Change

Native Change detection allows to capture any changes to Identity made directly in the target system. In Aramco implementation we monitor for native changes on Account Entitlements for identity.
Native Change is triggered based workflow configured to detect native changes.

1.5.1 ASC-Lifecycle Event - Manager Approval for all native changes

Primary workflow responsible for handling Native changes.


start
Audit
Complete
end
Notify Manager
Build Op String
Build Native Change Approval
Approve Native Change
Process Approval Decisions
Provision Native Change Reversal

Custom Variables

Variable Name	Identity Attribute / Comment
Identityname	Input variable
notifyNativeChanges	Set as true, defines if native changes needs to be notified or not
identityDisplayName	Display name calculated based on firstname and lastname
provisionRejectedItems	Set as True. Setting to false will not de-provision rejected native changes from target.

Referenced Libraries:

- LCM Workflow Library
- ASC-SP-Util Rule Library

Workflow Steps:

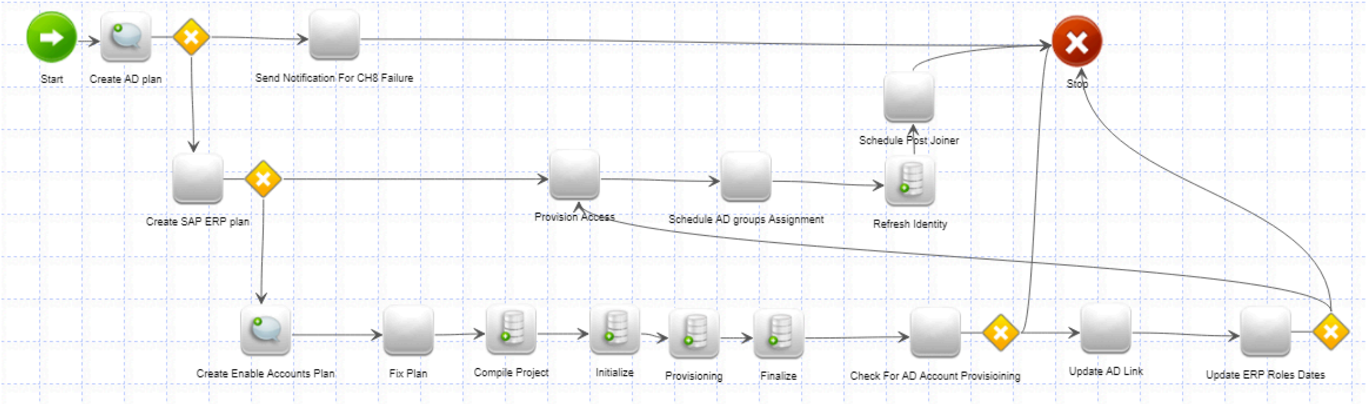
- Start:
  - Check from the native change object of the user if all the native changes are part of Birth right access or not, if all birth right then don't notify for native changes.
  - Refers to a custom object: ASC-NativeChange-Birthright and entry "All Apps"
  - Transition: Audit
- Audit:
  - Create an Audit Entry in the Audit Objects with "NativeChange" as the object name and details of change.
  - Transition:
    - Notify Manager
    - End (When no native Change to notify)
- Notify Manager:
  - Uses IIQ Email notification to notify manager about the native change.
  - Utilize EmailTemplate: Native Account Change Manager Notification.
  - Note: Notification is temporarily bypassed to send to Tuan.le@aramcoservices.com

  - Transition: Build Operation String
- Build Operation String
  - Check Each native change detected and prepare a list of operations on account. Ex: Modify, Add, remove etc.
  - Transition: Build Native Change Approval
- Build Native Change Approval
  - Calls library method, buildApprovalSetFromNativeChanges() to build an approval set to be approved by manager
  - Transition : Approve native changes
- Approve Native Changes:
  - Prepares an approval form with native changes to be approved by manager.
  - Uses renderer: nativeChangeApprovalRenderer.xhtml
  - Transition: Process Approval Decision
- Process Approval Decision:
  - Calls Library method processNativeChangesApprovalDecisions() to build approval or rejection decision(removal) plans.
  - Transition: Process Native Change Reversal
- Process Native Change Reversal:
  - Calls LCM Provisioning Workflow to provision reversal of native changes when native changes are rejected.
  - Note: This step only execute even with reject if variable provisionRejectedItems is declared as true.
  - Transition: End
- Complete:
  - A step with catch action which is used to clean up and related identityRequest.

1.6 Rehire

1.6.1 ASC -LifeCycle Event – Rehire

Primary workflow responsible for managing rehire users.



Workflow Variables:

Variable Name	Identity Attribute / Comment
Identityname	Input variable
identityDisplayName	Display name calculated based on firstname and lastname
tempPassword	Password generated using the library method for various accounts
calEmpEndDate	End Date to be set for different accounts. Based on employeeEndDate attribute.
Username	Variable to hold networkID

Dn	Variable to hold AD distinguish name
ADNativeIdentity	Variable to hold native identity for AD link
Department	departmentCode
Division	DivisionCode
employeeType	employeeType
empEndDate	employeeEndDate
userManager	Manager name for identity

Referenced Library:

- ASC-SP-Util Rule Library
- ASC-Rule-Constants
- Workflow Library

Workflow Steps:

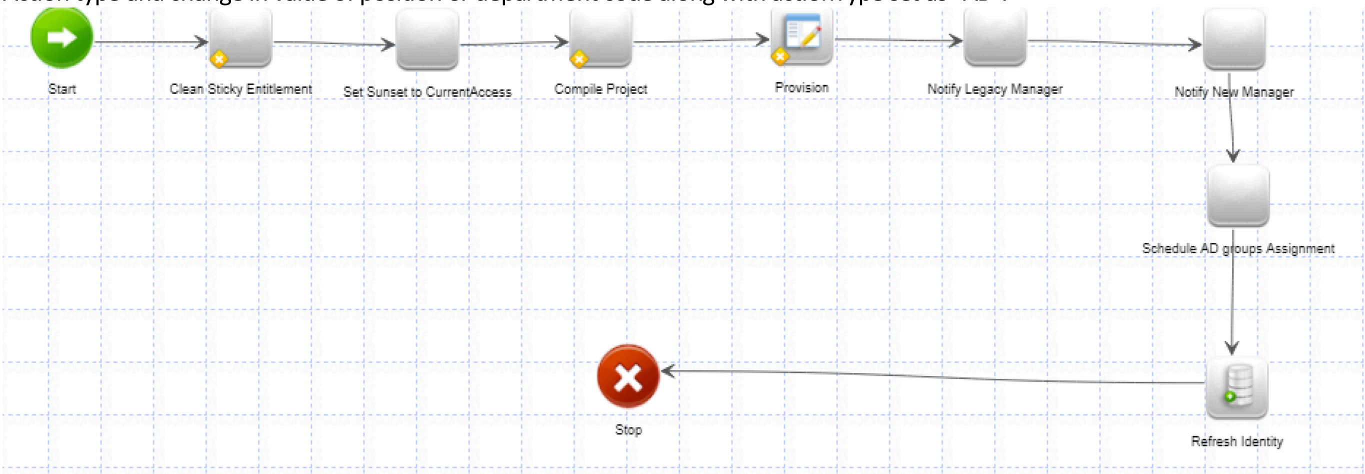
- Create AD Plan:
  - If the account don't have an AD link: Creates an account request for Active Directory account creation.
  - If identity already have and AD link, update the network ID to variable called Username
  - Transition:
    - Send Notification for Chapter 8 failure if network ID is not available
    - Create SAP ERP plan
- Create SAP ERP Plan
  - Check for Provisioning plan passed from previous step
  - If the plan from is not empty uses the username from the same step and further checks for SAP ERP link.
  - If ERP links already exists and disabled prepare plan to enable it.
  - If ERP link don't exist, update the provisioning plan to add ERP creation account request to it.
  - Transition:
    - Provision Access: when provisioning plan is not empty
    - Create Enable account Plan
- Create Enable Account Plan
  - Calls buildEventPlan library method with operation enable to create and enablement plan for all accounts held by identity
  - Transition: Fix Plan
- Fix Plan:
  - Update the Active Directory Account Request in the plan to modify the distinguished name attribute to regular account OU by modifying attribute "AC_newParent".
  - Update SAP ERP valid to and Valid From Dates in the ERP account Request
  - Update IdentityIQ attribute "inactive" as false for the user.
  - Transition: Compile Project
- Compile project, Initialize, Provision Project, Finalize:
  - Calls LCM library methods to complete provisioning of accounts
  - Transition: Check for AD Account Provisioning
- Check for AD Account Provisioning
  - Check for AD account provisioning for any errors,
  - Transition: Update AD Link
- Update AD Link:
  - Update the AD Link for the identity to set distinguish name with correct value for an account in enabled account OU.
  - Transition: Update ERP Role Dates
- Update ERP Role Dates
  - Check for all ERP roles and update the assigned roles end dates with 9999-12-31
  - Transition: Provision Access
- Provision Access:
  - Calls LCM provisioning workflow to provision the plan for ERP role update
  - Transition: Schedule AD Group Assignment

- Schedule AD Group Assignment
  - Schedule a workflow to run after 10 minutes to assign AD groups based on department and org structure for the user. Workflow called" ASC-UpdateADGroups-Workflow".
  - Transition: Refresh Identity
- Refresh Identity:
  - Calls refreshIdentity method with options selected to provision birthright roles for the identity.
  - Transition: Schedule Post Joiner
- Schedule Post Joiner:
  - Post joiner workflow scheduled to run after 30 minutes.
  - This will take care sending notifications and exchange scripts and other tasks described earlier in the workflow section for post joiner.
  - Transition: Stop

1.7 Transfer

1.7.1 ASC-Lifecycle Event - TransferV2

Transfer workflow take care of identity when an identity is transferred from one department to another. This is identified using Action type and change in value of position or department code along with actionType set as "AB".



Custom Variables:

Variable Name	Identity Attribute / Comment
Identityname	Input variable
identityDisplayName	Display name calculated based on firstname and lastname
tempPassword	Password generated using the library method for various accounts
Employee_email_id	Mail attribute
Username	Variable to hold networkID
Dn	Variable to hold AD distinguish name
ADNativeIdentity	Variable to hold native identity for AD link
Department	departmentCode
Division	DivisionCode
employeeType	employeeType
legacyManager	Manager id from the old Identity Object. Represent the previous manager
newManager	Manager name from the updated Identity. Represent the new manager
retainedList	Temporary variables
identityRoleList	Temporary Variables
Nonbr_roleList	Temporary variables
legacyManagerEmail	Email ID of the manager from old IdentityObject
NewManager_mailID	Email ID of the manager from the changed identity Object

Referenced Library:

- ASC-SP-Util Rule Library

Workflow Steps:

- Start
    - Prepares a list of all non-birthright entitlements Identity have and saves in temporary variable nonbr_roleList
    - Transition: Clean Sticky Entitlements
- Clean Sticky Entitlement
    - Looks for any sticky entitlements assigned to Identity, checks Identity's AttributeAssignment section and check for entitlements. (Sticky entitlements are entitlements which are assigned from IIQ UI directly to Identity, IIQ treat these entitlements as mandatory and not to be removed.)
    - Check if the sticky entitlement is part of non birthright role and remove them if they are.
    - Transition: Set Sunset to Current Access
- Set Sunset to Current Access:
    - Prepare Plan to remove non birthright roles from accounts
    - Transition: Compile Project
- Compile Project
    - Call Library method compileProject for the plan compilation. Argument NoFiltering is passed to not filter any entitlement from the removal plan.
    - Transition: Provision Project
- Provision Project
    - Call library method provisionProject to provision the compile project from the previous step.
    - Transition: Notify legacy Manager
- Notify Legacy Manager:
    - Send an Email Notification using the powershell script to previous manager that the user has been transferred from his team
    - Powershell script used: D:\Installs\scripts\Email_Notification\Transfer_Notification\Generic_Transfer.ps1
    - Transition: Notify New Manager
- Notify New Manager:
    - Send an Email Notification using the powershell script to new manager that the user has been transferred to his team
    - Powershell script used: D:\Installs\scripts\Email_Notification\Transfer_Notification\Generic_Transfer.ps1
    - Transition: Schedule AD group Assignment
- Schedule AD Group Assignment
    - Schedule the workflow"ASC-UpdateADGroups-Workflow" to update users AD groups based on his departmental information.
    - Transition: refresh Identity
- Refresh Identity
    - Refresh the identity object with provision argument set to true to check for any new birthright role assignment and provision any IT role entitlement assigned.

Non-Lifecycle Workflows

There are several workflow configured which are based on user request. These workflows are triggered from IIQ UI using quiklinks and Forms.

1.8 Request Access

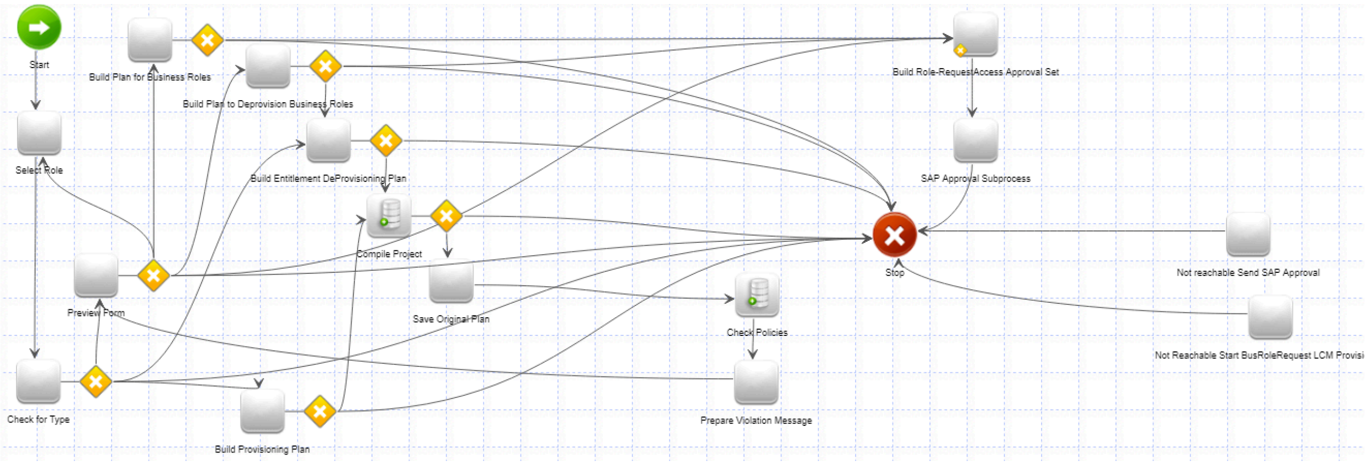
Two Request Access Quicklinks are configured.

1. Request Access for Me
2. Request Access for Others

Both of the Quicklinks calls for same workflow, however the second Quicklink will pass the user object selected to the workflow while the first one pass the identity object launching the quiklink.

1.8.1 ASC-Role-RequestAccess-Provisioning

Primary workflow to present the request access form and UI to the end user.


Start
Build Plan for Business Roles
Build Plan to Deprovision Business Roles
Build Role-RequestAccess Approval Set
Select Role
Build Entitlement DeProvisioning Plan
SAP Approval Subprocess
Compile Project
Stop
Not reachable Send SAP Approval
Preview Form
Save Original Plan
Check Policies
Not Reachable Start BusRoleRequest LCM Provisi
Check for Type
Prepare Violation Message
Build Provisioning Plan

Custom Variables

Variable Name	Identity Attribute / Comment
quickLinkIdentityId	I/P: Identity id for which workflow was launched
applicationName	Temporary Variable
Entitlements	Temporary Variable
Transient	Variable set to true. Does not save the workflow unless there is an access request.
Roles	Temporary Variable
approvalSet	Temporary Variable
Plan	Temporary Variable
violationMessage	Temporary Variable
State	Variable of Map type to hold entitlements. Reduce the back and forth to database in order to fetch requestable entitlements and roles
identityName	Identity name
Identity	Identity Object from quiklinkIdentityID
Custom	Hold custom object “ASC-RoleRequest-Custom”
Manager	Identity Manager name

Referenced Library:

- ASC-RuleLibrary-AccessRequest
- LCM Workflow Library

Workflow Steps

- Start
  - Prepare the state variable (Map object) with all requestable Entitlement and roles. This variable will be referenced by later workflow steps and form objects to query application specific entitlement/roles. This was added to reduce the back and forth database in order to query desired objects and thus improve performance.
  - Results: State variable
  - Transition: Select Role
- Select Role
  - Step contains the Form for requesting Access. Form is divided into multiple sections for request types.
  - Each section has its own criteria to display /hide in the UI.
  - Form has following sections:
    - "Select provisioning type for the selected Identity": Section displaying user information and requestable Items. Contains various fields with each filed having creiteria to show/hide in the UI based on user inputs. This section contains following fields:
      - Selected user
      - Type Of Request (Add/ Remove)
      - Request Access Type (Application /Business)
      - AD Groups
      - SAP ERP Roles
      - Fiori roles
      - User Role (Business Roles) to remove
    - "Business Roles" : Section displaying Business Roles for assignment. Contains Field:
      - Business Roles
    - "Application Details". Section with details for applications to assign. Have fields:
      - Application Name
    - "Active Directory Request Items". Have fields:
      - adEntitlements
    - "CDD-STS Request Items". Have Fields:
      - CDDSTSRoles
    - "SPIIVFS Request Items". Field:
      - SPIIVFS Roles
    - "TMS Request Items". Field:
      - TMSRoles
    - "epeas Request items". Field:
      - EpeasRoles
    - eRewards Request Items Field:
      - eRewardRoles
    - "SAP Request Items": Fields:
      - erpFromDate
      - erpToDate
      - sapRoles
    - "SAP Fiori Request Items": Fields:
      - fioriFromDate
      - fioriToDate
      - sapFioriRoles
    - "VFS Request Items".Fields:
      - VFS Roles
    - "COF Request Items". Fields:
      - COF Roles
    - TOS Logical Request Items. Fields:

        - TOS_Logical_Roles
    - Policies Site Logical Request items: Fields:
        - PoliciesSite_Logical_Role
    - Business Justification. Field:
        - Business Justification
- Check For Type:
    - Steps check the type of Access Request made by the user if it is for Adding(as application entitlements / Business Roles) or Removing access. Based on that this steps redirects the flow to different steps.
    - Transition:
        - Build Provisioning Plan: When Add request for application entitlement
        - Build Entitlement Deprovisioning plan: When request is for removing application entitlement
        - Preview Form: When request is for Business roles
- Build Provisioning Plan:
    - Calls library method buildProvisioningPlan() to prepare entitlement assignment provisioning plan
    - Transition: Compile Project
- Compile Project:
    - Calls library method: compileProvisionignProject()
    - Transition: Save Original Plan
- Save Original Plan:
    - Save the master provisioning plan from compiled project in a temporary variable for later use.
    - Transition: Check Policies
- Check Policies:
    - Calls library methods checkPolicyViolations to check for any policy violations which could result by provisioning this request. Returns policy violations to a variable “policyViolations”.
    - Transition: Prepare Violation message
- Prepare Violation Message:
    - Format the policy violation received from previous step into a presentable text format by pulling the policy violation description. (Note: policy violation description object refer to the policy violation entry name defined in the policy violations. It’s a good idea to name the policy violation in such a way that it is useful for the approver to read and understand).
    - Transition: Preview Form
- Preview Form:
    - This steps present a form with selected values from the previous Form to be verified by the user.
    - Transitions:
        - Build Plan to deprovision Business roles: When request is to remove Business roles
        - Build Plan for Business Roles: When request is to add business roles
        - Build role-requestAccess Approval Set: When request is for Entitlements and plan prepared for them by earlier step is not empty.
- Build Entitlement Deprovisioning Plan:
    - Call library method: buildEntitlementDeProvisioningPlan() to create a provisioning plan to remove entitlements selected by user.
    - Transition: Compile Project
- Build Plan to deprovision Business Roles:
    - Calls library method getBusinessRoleDeProvisionPlan() to build business role removal plan.
    - Transition:
        - Build Role-RequestAccess Approval Set
        - Build Entitlement Deprovision Plan: When request type is to remove an application entitlement only
- Build Role-Request Approval Set:
    - Calls library method buildApprovalSet() to prepare an approval set for the plan prepared. Store approval set received in variable “Approval Set”
    - Transition: SAP Approval Subprocess
- SAP Approval Subprocess:

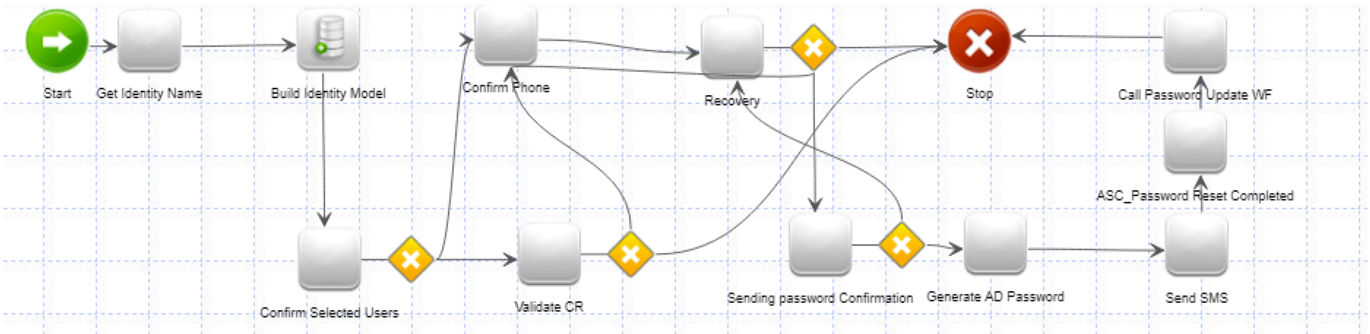
- Call a sub workflow “*ASC_Send_Approval_TO_SAP”* with approvalSet and other arguments from the form. This workflow will in turn result in sending approval to SAP.
- Transition: Stop

1.9 Helpdesk Password Reset

Helpdesk staff performs password reset on behalf of end users when they call in and need assistance with password reset. A quiklink, form and workflow has been created to achieve this.

Quicklink: ASC-Reset User Account
Workflow: ASC_Workflow_HelpDesk_Password_Management_V2
Form: ASC_Form_Password_Reset_User_Confirmation , ASC_Form_Password_Reset_Validate_Have_Phone, ASC_Form_Password_Reset_User_Recovery

1.9.1 ASC_Workflow_Helpdesk_password_Management_V2



Variable Name	Identity Attribute / Comment
identityModel	A Map type variable to hold the identityModel
Phone	Temporary Variable
emailID	Temporary Variable
Transient	Variable set to true. Does not save the workflow unless there is an access request.
networkID	Temporary Variable
tempPassword	Temporary Variable
Plan	Temporary Variable
haveCompanyCell	Temporary Variable
OtherCellPhone	Temporary Variable
haveOtherCellPhone	Temporary Variable
AuthenticationNotSetForuser	Temporary Variable
Requester	Temporary Variable
quickLinkIdentityID	Temporary Variable
identityName	Temporary Variable
ChallengeResponseNeeded	Temporary variable. Set as False to disable Challenge/Response.

Referenced Libraries:

- ASC-Rule-SP Util Rule Library

Workflow Steps:

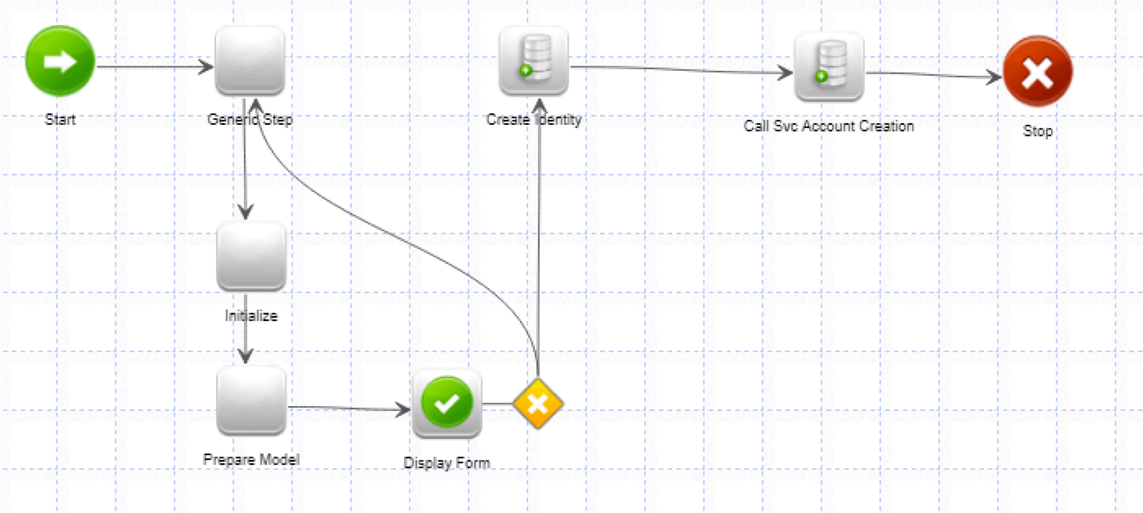
- Get Identity Name:
  - Save the name of the identity to identityName variable from quickLinkIdentityID object
  - Transition: Build Identity Model
- Build Identity Model:
  - Calls library method getIdentityModel() to get a map of identity attributes.
  - Transition: Confirm Selected User
- Confirm Selected user:

  - Present the details of the user selected to the helpdesk personnel in UI form. Form “ASC_Form_Password_Reset_User_Confirmation”
  - Transition: Confirm Phone : When ChallengeResponseNeeded variable is set as False
  - Transition : Validate CR
- Confirm Phone:
  - Present a Form with user’s phone detail to verify. Form” ASC_Form_Password_Reset_Validate_Have_Phone”.
  - Transition: Recovery
- Recovery:
  - Present a confirmation Form to the user to validate the details. Form: “ASC_Form_Password_Reset_User_Recovery”
  - Transition: Stop : When user don’t have a phone on his profile.
  - Transition: Sending password confirmation
- Sending Password Confirmation:
  - Present a Form about confirmation of the process. Form “ASC_Form_Password_Reset_Validation_Confirmation”.
  - Transition: Generate AD Password.
- Generate AD Password:
  - Calls library method: getTempPasswordLong() to generate a password for the user calling. This will be later used as the password for his account.
  - Transition: Send SMS
- Send SMS:
  - Send SMS to the user phone using Twilio SDK. Twilio configuration is saved in system config. To Modify Twilio config: Global Settings – Login Configuration – User Reset – SMS verification configuration.
  - Transition: ASC_Password_Reset_Completed
- ASC_Password_Reset_Completed:
  - Display a form to show password reset process is initiated for the user. Form: ASC_Password Reset Completed
  - Transition: Call Password Update WF
- Call Password Update Workflow
  - Call the custom subprocess to perform the password reset on the user AD account. Argument temporaryPassword passed into the workflow defines if the password has to be set permanently or need to be set to change at next login. SubProcess Called: ASC-Password-Reset
  - Transition: Stop

1.10 Create Service Account

A Quicklink has been created to provide functionality on Creating Service Account. Quicklink presents a Form to input details for the account and submits the approval to SAP Approval Engine.
Quicklink: ASC-Service Account : Display Value: Create Service Account
Workflow: ASC-Service Account

1.10.1 ASC-Service Account



Variable Name	Identity Attribute / Comment
sessionOwner	Identity launching the workflow
Trace	Variable defines logging to stdout
Transient	Variable set to true. Does not save the workflow unless there is an access request.

Workflow Steps:

- Generic Step:
  - Present the form to capture input from the user to create service account. Form contains following fields:
    - Requestor: Auto populated with the identity Launching the workflow
    - Name of the account: User Input with validation. Validation:
      - Must be less than 20 characters.
      - Name should be in the format: svc-myapp-environment and should start with svc.
    - Business Reason for creation
  - Transition: Initialize
- Initialize:
  - Call for library method getIdentityModel() to get map of identity attributes.
  - Transition: Prepare Model
- Prepare Model:
  - Update the identityModel with business Justification, accountName and requestor info.
  - Transition: Display Form
- Display Form:
  - Display the confirmation form to the user
  - Transition: Create Identity
- Create Identity
  - Create an Identity in IIQ to hold the service account created.

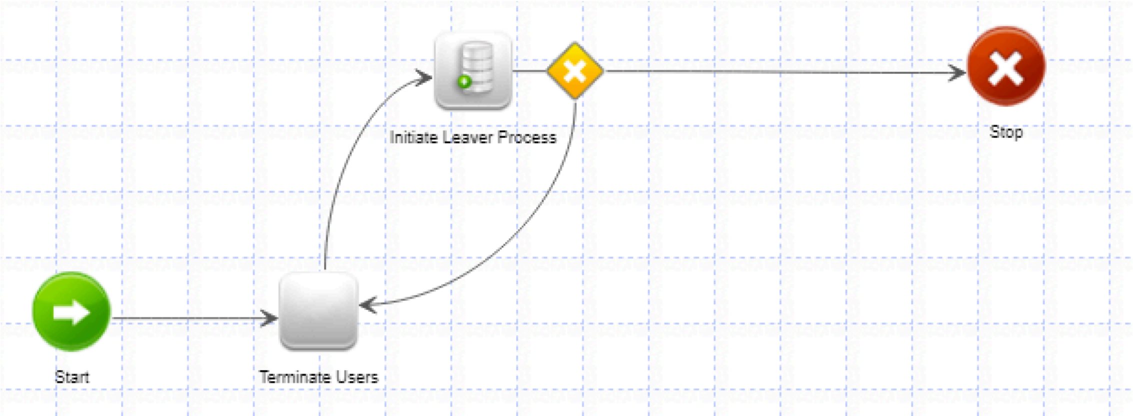
  - Transition: Call SVC Account Creation
- Call SVC Account Creation:
  - Calls the subprocess ASC-Service Account Creation to create prepare approval Set to be sent to SAP for approval of account.
  - Transition: Stop

1.11 Terminate Identity

A Quicklink is created for enablement of manual termination of an identity. The Quicklink allows users to be selected and Leaver workflow is initiated for those users.

Quicklink : Name- ASC-Terminate Identity ; Display Value :Terminate Identity
Workflow:ASC-LifceCycle Event – Leaver OnDemand

1.11.1 ASC-LifeCycle Event – Leaver OnDemand



Custom Variables:

Variable Name	Identity Attribute / Comment
quickLinkIdentityIds	List of identity selected for termination. I/P variable
Trace	Variable defines logging to stdout
Transient	Variable set to true. Does not save the workflow unless there is an access request.
Incrementor	Variable to hold count of identities for termination

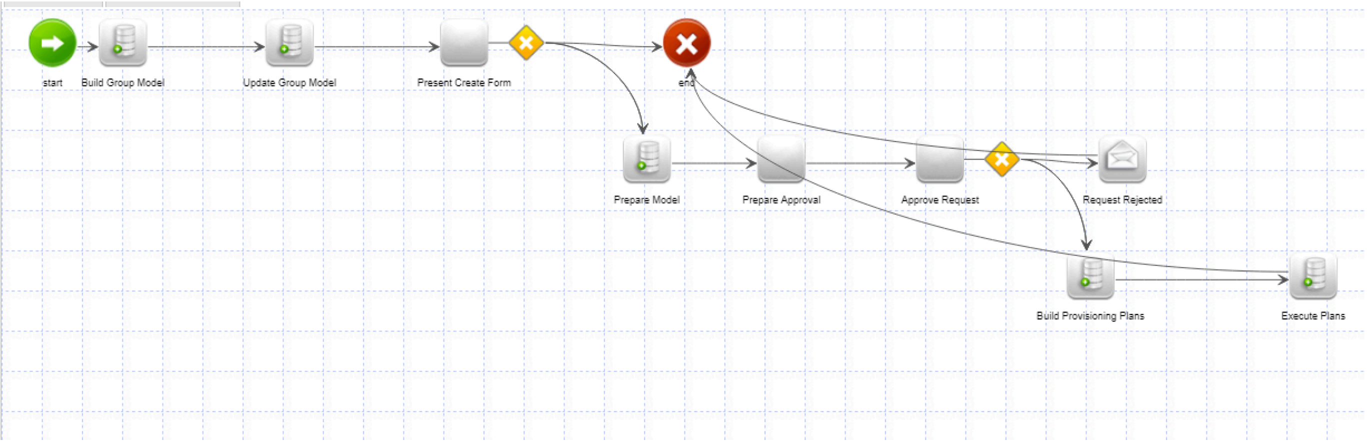
Workflow Steps:

- Terminate Users:
  - Capture the identity from the list of identities provided by quickLinkIdentityIds object.
  - Set each identity as inactive and initiate leaver process for each of them one by one.
  - Transition: Initiate Leaver Process
- Initiate leaver Process:
  - Calls Sub-process "*ASC-Lifecycle_Event_ImmediateTermination*" for termination of identity.
  - Transition : Stop

1.13 Create Groups

A functionality has been created to allow users to create groups in AD from IIQ. A Quicklink, Form and workflow has been created to enable this functionality.
Quicklink : Create Group ; Display name: Request Group
Workflow: AD Group Management – Create

1.13.1 AD Group Management – Create



Custom Variables:

Variable Name	Identity Attribute / Comment
requesterDisplayName	Temporary variable
Trace	Variable defines logging to stdout
Transient	Variable set to true. Does not save the workflow unless there is an access request.
groupMembership	Temporary variable
appId	Temporary variable
basePath	Temporary variable
dnTemplate	Temporary variable holding the OU for group creation. Modify this value to change group OU.
groupModel	Temporary variable

Workflow Steps:

- Start
  - Prepare appID for Active Directory application and store in a temporary variable for later use.
  - Transition: Build Group Model
- Build Group Model:
  - Call library method: getManagedAttributeModel() prepare a map for group provisioning. Save the map to groupModel variable
  - Transition : Update Group Model
- Update Group Model:
  - Update the Group Model map to add: ATTR_TRANSFORMER_CLASS ad sailpoint.transformer.ManagedAttributeTransformer and ATTR_TRANSFORMER_OPTIONS as empty string. (Note this is needed only in 7.0 version of IIQ)
  - Transition: Present Create Form
- Present Create Form
  - Display the UI form to get input from the user for group creation. Form" Group Management - Request AD Group "
- Prepare Model:

  - Update the group model map with user provided group name and other details.
  - Transition: Prepare Approval
- Prepare Approval:
  - Send the approval for creating group to owner of AD application.
  - Transition: Request Rejected
  - Transition: Build Provisioning Plans
- Build Provisioning Plans:
  - Calls library method buildPlanFromMamanagedAttributeModel() to prepare an Object provision plan.
  - Transition: Execute Plans
- Execute Plans:
  - Calls library method executeManagedAttributePlans() to perform the provision of the plan prepared.
  - Transition: End