Services Standard IdentityNow BeforeProvisioning Rule README

PS IdentityNow Setup Extension

Overview

This project is meant to utilize the following to enable quick setup of IdentityNow complex requirements:

- APIs to update source configurations to store configs
- Services Standard BeforeProvisioning Rule to read and use the configs to handle complex requirements

Future Plans

- Offline Script to read the source configs to automatically build standard requirements
- UI Based tooling to configure tenants
- Import/Export of configurations

Version History

Version	Date	Comments
1.0	05/01/2021	Initial Release
1.1	06/24/2021	Updated to include ThrowError action
1.2	08/06/2021	Updated to include StopProcessing action
1.3	10/05/2021	Update to include account attribute in dynamic value calculation Add AddArgument action
1.4	11/15/2021	Updated context calls to utilize IDN Rule Util
1.5	12/22/2021	Updated Value Matching to include wildcards * and ?
1.6	2/7/2022	Add “Entitlement Name Update Triggers” Added comments

Available Source Configurations

Before Provisioning Rule configurations

The Before Provisioning Rule configurations allows for the generic BeforeProvisioningRule to process common events based on standard best practices. The configuration itself is a list of JSON Maps that include 2 basic concepts

1. Trigger: This configuration determines if this event should be applied based on
   a. Operation of the account request
   b. Identity Attribute values for the user being modified
   c. Attribute updates in the account request
   d. Entitlement updates in the account request
   e. Entitlement Name updates in the account request
2. Action: This configuration determines what to do if this event should be applied. Available options are
   a. Move AD Account: Adds AC_NewParent to the specified OU

These are more like filters than triggers

b. Change Operation: changes the AccountRequest Operation to the specified operation
c. Remove Entitlements: removes all entitlements based on the account in IdentityNow
d. Remove AD Entitlements: sets the memberOf attribute to a list containing the one group
e. Scramble Password: sets the specified attribute to a string of characters
f. Update Attribute: Adds the specified attribute with the specified value
g. Add Argument: Adds the specified argument with the specified value
h. Throw Error: Throws an error specified in the value attribute or “Unspecified Exception”
i. Stop Processing: Stops processing the BeforeProvisioningRule and proceeds to provisioning.

Trigger Conditions

Each event will have a definition for how to trigger the event. These include

- Operation*: This is required and is a single value. It refers to which Account operation this event should trigger for.
- Identity Attribute Triggers: If configured, this will limit the triggering of this event to Identities matching the specified criteria.
- Account Attribute Update Triggers: If configured, this will limit the triggering of this event to requests modifying attributes in the specified way.
- Entitlement Update Triggers: If configured, this will limit the triggering of this event to requests modifying an account by adding/removing/setting entitlements in the specified way. This trigger checks directly against the value of the entitlement being modified.
- Entitlement Name Update Triggers: If configured, this will limit the triggering of this event to requests modifying an account by adding/removing/setting entitlements in the specified way. This trigger checks against the displayName attribute found in the entitlement attributes. Note: at this time, this is not necessarily the Display Name of the entitlement, it must be found in aggregation. Also, this requires a query to the database so performance may be a concern.

The following items list the basic options for Triggers

- Operation: This checks the operation of the Account Request.
    - Configured with the key “Operation” and the value for the specified Account Operation.

```
"Operation":"Disable"
```

- Options
    - Create
    - Enable
    - Disable
    - Modify
- Identity Attribute Triggers: This checks if the user being modified matches a specific value for an Identity attribute
    - Configured with a separate key "Identity Attribute Triggers" which is a list of attribute conditions to match
    - Each match will have an attribute, an operation, and a value.

These are more like FILTERS than TRIGGERS

```
"Identity Attribute Triggers":[
    {
        "Attribute":"cloudLifecycleState",
        "Operation":"eq",
        "Value":"inactive"
    },
    {
        "Attribute":"employeeType",
        "Operation":"ne",
        "Value":"Employee"
    }
]
```

- Supported attributes are any attributes available on the Identity
- Supported operations
    - eq: signifies the attribute for the user matches the value
    - ne: signifies the attribute for the user does not match the specified value

  - Supported value is Java String based matches. This will support static values or wild card values using the * for any character or ? for a single character (e.g. test_equals matches “test_equals” or “test_*” or “test?equals”). The following will be treated as a null value
    - the key word #{null} : “Value”:”#{null}”
    - an empty string : “Value”:””
    - null entry : “Value”:null
- Account Attribute Update Triggers: This checks if the account request contains updates on a specific attribute to a specific value
  - Configured with a separate key "Account Attribute Update Triggers" which is a list of attribute conditions to match
  - Each match will have an attribute, an operation, and a value

```
"Account Attribute Update Triggers":[
    {
        "Attribute":"department",
        "Operation":"eq",
        "Value":"*"
    }
]
```

  - Supported attributes are any account attributes that may be modified during Attribute Sync
  - Supported operations
    - eq: signifies the attribute for the user matches the value
    - ne: signifies the attribute for the user does not match the specified value
  - Supported value is Java String based matches. This will support static values or wild card values using the * for any character or ? for a single character (e.g. test_equals matches “test_equals” or “test_*” or “test?equals”). The following will be treated as a null value
    - the key word #{null} : “Value”:”#{null}”
    - an empty string : “Value”:””
    - null entry : “Value”:null
- Entitlement Update Triggers: This checks if the account request contains updates on an entitlement to add/remove/set to a specific value
  - Configured with a separate key "Entitlement Update Triggers" which is a list of attribute conditions to match
  - Each match will have an attribute, an operation, and a value

```
"Entitlement Update Triggers":[
    {
        "Attribute":"memberOf",
        "Operation":"Add",
        "Value":"CN=Group1,OU=Groups,DC=example,DC=com"
    }
]
```

  - Supported attributes are any account attributes that may be modified during Automated Role Refreshes or Access Requests
  - Supported operations
    - Add: trigger checks if the entitlement is being added
    - Remove: trigger checks if the entitlement is being removed
    - Set: trigger checks if the entitlement is being set to a value (should only be the case for single value attributes)
  - Supported value is Java String based matches. This will support static values or wild card values using the * for any character or ? for a single character (e.g. test_equals matches “test_equals” or “test_*” or “test?equals”). The following will be treated as a null value
    - the key word #{null} : “Value”:”#{null}”
    - an empty string : “Value”:””
    - null entry : “Value”:null
- Entitlement Name Update Triggers: This checks if the account request contains updates on an entitlement to add/remove/set to a specific value
  - Configured with a separate key "Entitlement Name Update Triggers" which is a list of attribute conditions to match
  - Each match will have an attribute, an operation, and a value

```
"Entitlement Name Update Triggers":[
    {
        "Attribute":"memberOf",
        "Operation":"Add",
        "Value":"Group1"
    }
]
```

- Supported attributes are any account attributes that may be modified during Automated Role Refreshes or Access Requests. Note: this is expected to be an entitlement as the rule will search for it
- Supported operations
    - Add: trigger checks if the entitlement is being added
    - Remove: trigger checks if the entitlement is being removed
    - Set: trigger checks if the entitlement is being set to a value (should only be the case for single value attributes)
- Supported value is Java String based matches. This will support static values or wild card values using the * for any character or ? for a single character (e.g. test_equals matches “test_equals” or “test_*” or “test?equals”). The following will be treated as a null value
    - the key word #{null} : “Value”:”#{null}”
    - an empty string : “Value”:””
    - null entry : “Value”:null

Event Actions

Each event will have a key "eventActions" that contain the actions to perform for the account once the event is triggered. Each action has an action key, attribute, and value.

The following items list the basic options for Triggers

- Scramble Password: This will add an attribute request to modify the password for the account
    - Configuration
        - action: "ScramblePassword"
        - attribute: password attribute for the source
        - value: NA. This will be auto generated by the rule

```
{
    "Action":"ScramblePassword",
    "Attribute":"password",
    "Value":null
}
```

- Update Attribute: This will add an attribute request to modify an attribute to a specified value
    - Configuration
        - action: "UpdateAttribute"
        - attribute: attribute to update
        - value: value to update to. This supports the following dynamic keys that will be replaced with either the corresponding value or a blank string
            - #{now}: Java new Date().toString() ... e.g. Thu Jan 28 15:30:03 CST 2021
            - #{now.format}: new Date() ... formatted using Java SimpleDateFormat. Format can also be specified as
                - #{now.EPOCH_TIME_WIN32}: creates the long timestamp for windows 32 bit date
                - #{now.EPOCH_TIME_JAVA}: creates the long timestamp for JAVA based dates in milliseconds
                - #{now.EPOCH_TIME}: creates the long timestamp for JAVA based dates in seconds
                - #{now.ISO8601}: Prints the date using standard ISO8601 format “yyyy-MM-dd'T'HH:mm:ss.SSSX”
            - #{identity.attribute}: replaces with the identity attribute value
            - #{manager.attribute}: replaces with the identity attribute value for the user's manager
            - #{account.attribute}: replaces with the account attribute value for the user's link for this account.

```
{
    "Action":"UpdateAttribute",
    "Attribute":"description",
    "Value":"Disabled by IdentityNow Automation on #{now.MM/dd/yyyy}"
}
```

- Add Argument: This will add an argument to the AccountRequest with a specified value
    - Configuration
        - action: "AddArgument"
        - attribute: argument to add
        - value: value to update to. This supports the following dynamic keys that will be replaced with either the corresponding value or a blank string
            - #{now}: Java new Date().toString() ... e.g. Thu Jan 28 15:30:03 CST 2021
            - #{now.format}: new Date() ... formatted using Java SimpleDateFormat. Format can also be specified as
                - #{now.EPOCH_TIME_WIN32}: creates the long timestamp for windows 32 bit date
                - #{now.EPOCH_TIME_JAVA}: creates the long timestamp for JAVA based dates in milliseconds
                - #{now.EPOCH_TIME}: creates the long timestamp for JAVA based dates in seconds
                - #{now.ISO8601}: Prints the date using standard ISO8601 format “yyyy-MM-dd'T'HH:mm:ss.SSSX”
            - #{identity.attribute}: replaces with the identity attribute value
            - #{manager.attribute}: replaces with the identity attribute value for the user's manager
            - #{account.attribute}: replaces with the account attribute value for the user's link for this account.

```
{
    "Action":"AddArgument",
    "Attribute":"identityEmail",
    "Value":"{identity.email}"
}
```

- Remove Entitlements: This will add attribute requests to remove the user's existing entitlements for the account. This removes only existing data in IdentityNow.
    - Configuration
        - action: "RemoveEntitlements"
        - attribute: entitlement attribute for the source
        - value: NA. This is calculated per user.

```
{
    "Action":"RemoveEntitlements",
    "Attribute":"memberOf",
    "Value":null
}
```

- Remove AD Entitlements: This will add an attribute request to set the memberOf attribute to a list containing the group specified by Value.
    - Configuration
        - action: "RemoveADEntitlements"
        - attribute: “memberOf” (though this is not used)
        - value: specified group for final group. Normally this is the Domain Users group, but can be a separate group which will remain.

```
{
    "Action":"RemoveADEntitlements",
    "Attribute":"memberOf",
    "Value":"CN=Domain Users,CN=Builtin,DC=example,DC=com"
}
```

- AD Move Account: This will add the AC_NewParent attribute to the designated value. Prior to adding, it will determine if the resulting DN is unique based on searching accounts in IdentityNow. If not, it will serialize by adding a number at the end of the CN starting with 2 ... such as CN=John Smith 2.
  - Configuration
    - action: "ADMoveAccount"
    - attribute: "AC_NewParent" (though this is not used)
    - value: value to update to. This supports the following dynamic keys that will be replaced with either the corresponding value or a blank string
      - #{identity.attribute}: replaces with the identity attribute value

```

{
    "Action":"ADMoveAccount",
    "Attribute":"AC_NewParent",
    "Value":"OU=Disabled Users,OU=Users,DC=example,DC=com"
}

or

{
    "Action":"ADMoveAccount",
    "Attribute":"AC_NewParent",
    "Value":"#{identity.disabledOU}"
}
```

- Change Operation: This will change the account operation to the specified operation.
  - Configuration:
    - action: "ChangeOperation"
    - attribute: NA
    - value: Operation to change to. Available options are
      - Modify
      - Enable
      - Disable
      - Delete

```

{
    "Action": "ChangeOperation",
    "Attribute": null,
    "Value": "Delete"
}
```

- Throw Error: This will throw an error so provisioning will not be processed
  - Configuration:
    - action: "ThrowError"
    - attribute: NA
    - value: String to include in the error. defaults to “Unspecified Exception” if null.

```

{
    "Action": "ThrowError",
    "Attribute": null,
    "Value": "Identity is in invalid state. Stopping provisioning."
}
```

- Stop Processing: This will end processing of the BeforeProvisioningRule to allow for provisioning to start immediately after
  - Configuration:
    - action: "StopProcessing"
    - attribute: NA
    - value: NA

```
{
    "Action": "StopProcessing",
    "Attribute": null,
    "Value": null
}
```

Example Use Cases

Simple Termination

Requirements

- On Termination, we want to
  - Disable the AD account
  - Move the AD account
  - Scramble the AD password
  - Remove the AD groups

Design

- Update the “inactive” LCS to disable AD
- Add the “Services Standard IdentityNow BeforeProvisioning Rule” to AD as the beforeProvisioningRule
- Add the following configuration to the AD source using V3 PATCH on the source.
  - **NOTE**: This will only perform these actions if the user is being disabled AND the lcs is inactive

```
[
  {
          "op": "add",
          "path": "/connectorAttributes/cloudServicesIDNSetup",
        "value": {
            "eventConfigurations": [
                {
                    "eventActions": [
                        {
                            "Action": "ADMoveAccount",
                            "Attribute": "AC_NewParent",
                            "Value": "OU=Disabled,OU=Users,OU=pa-
rshwarts,OU=training,DC=testing,DC=com"
                        },
                        {
                            "Action": "ScramblePassword",
                            "Attribute": "password",
                            "Value": null
                        },                          {
                            "Action": "RemoveADEntitlements",
                            "Attribute": "memberOf",
                            "Value": "CN=Domain Users,CN=Users,
DC=testing,DC=com"
                        }
                    ],
                    "Identity Attribute Triggers": [
                        {
                            "Attribute": "cloudLifecycleState",
                            "Value": "inactive",
                            "Operation": "eq"
                        }
                    ],
                    "Operation": "Disable"
                }
            ]
        }
    }
]
```

Complex Termination

Requirements

- On Termination, we want to
    - AD
        - Disable the AD account
        - Move the AD account
            - For Contractors: Move to Contractor Disabled OU
            - For Employees: Move to Employee Disabled OU dependent on department
        - Scramble the AD password
        - Update description to “Disabled by SailPoint on MM/dd/yyyy”
        - For Contractors only, Set AccountExpires to now
        - Remove the AD groups
    - Okta
        - Disable the Okta account
        - Remove Okta groups and roles
    - **Note**: We need to handle the use case where we term a user with a disabled AD account (example LOA to Term)
- 30 Days after term, we want to delete Okta and AD

Design

- Update the “inactive” LCS to disable AD and Okta
- Add extensionAttribute1 to sync the cloudLifecycleState (this will occur regardless if the user is already disabled so it will account for the move to inactive when user is already disabled.
- Add a “delete” LCS to be calculated at term + 30 and on. These should “Enable” AD and Okta
- Create an Identity attribute “Disabled OU” which calculates the appropriate OU for employees based on department. You could also set this for contractors and use that in the configuration.
- Add the “Services Standard IdentityNow BeforeProvisioning Rule” to AD and Okta as the beforeProvisioningRule
- Add the following configuration to the AD source using V3 PATCH on the source.
    - Map on Line 7: Term actions for all users
    - Map on Line 34: Term actions for employees only
    - Map on Line 56: Term actions for contractors only
    - Map on Line 83: Term actions for all users for users already disabled
    - Map on Line 117: Term actions for employees only for users already disabled
    - Map on Line 146: Term actions for contractors only for users already disabled
    - Map on Line 180: Deletion action

```
[
  {
          "op": "add",
          "path": "/connectorAttributes/cloudServicesIDNSetup",
        "value": {
            "eventConfigurations": [
                {
                    "eventActions": [
                        {
                            "Action": "ScramblePassword",
                            "Attribute": "password",
                            "Value": null
                        },
                        {
                            "Action": "UpdateAttribute",
                            "Attribute": "description",
                            "Value": "Disabled by IdentityNow
Automation on #{now.MM/dd/yyyy}"
                        },
                        {
                            "Action": "RemoveADEntitlements",
                            "Attribute": "memberOf",
                            "Value": "CN=Domain Users,CN=Users,
DC=testing,DC=com"
                        }
                    ],
                    "Identity Attribute Triggers": [
                        {
                            "Attribute": "cloudLifecycleState",
                            "Value": "inactive",
                            "Operation": "eq"
                        }
                    ],
                    "Operation": "Disable"
                },
                {
                    "eventActions": [
                        {
                            "Action": "ADMoveAccount",
                            "Attribute": "AC_NewParent",
                            "Value": "#{identity.disabledOU}"
                        }
                    ],
                    "Identity Attribute Triggers": [
                        {
                            "Attribute": "cloudLifecycleState",
                            "Value": "inactive",
                            "Operation": "eq"
                        },
                        {
                            "Attribute": "employeeType",
                            "Value": "Employee",
                            "Operation": "eq"
                        }
                    ],
                    "Operation": "Disable"
                },
                {
                    "eventActions": [
                        {
                            "Action": "ADMoveAccount",
                            "Attribute": "AC_NewParent",
                            "Value": "OU=Disabled Contractors,
DC=example,DC=com"
                        },
                        {
                                "Action": "UpdateAttribute",
                                "Attribute": "accountExpires",
                                "Value": "#{now.EPOCH_TIME_WIN32}"
                            }
                    ],
                    "Identity Attribute Triggers": [
                        {
                            "Attribute": "cloudLifecycleState",
                            "Value": "inactive",
                            "Operation": "eq"
                        },
                        {
                            "Attribute": "employeeType",
                            "Value": "Contractor",
                            "Operation": "eq"
                        }
                    ],
                    "Operation": "Disable"
                },
                {
                    "eventActions": [
                        {
                            "Action": "ScramblePassword",
                            "Attribute": "password",
                            "Value": null
                        },
                        {
                            "Action": "UpdateAttribute",
                            "Attribute": "description",
                            "Value": "Disabled by IdentityNow
Automation on #{now.MM/dd/yyyy}"
                        },
                        {
                            "Action": "RemoveADEntitlements",
                            "Attribute": "memberOf",
                            "Value": "CN=Domain Users,CN=Users,
DC=testing,DC=com"
                        }
                    ],
                    "Identity Attribute Triggers": [
                        {
                            "Attribute": "cloudLifecycleState",
                            "Value": "inactive",
                            "Operation": "eq"
                        }
                    ],
                    "Account Attribute Update Triggers": [
                        {
                            "Attribute": "extensionAttribute1",
                            "Value": "inactive",
                            "Operation": "eq"
                        }
                    ],
                    "Operation": "Modify"
                },
                {
                    "eventActions": [
                        {
                            "Action": "ADMoveAccount",
                            "Attribute": "AC_NewParent",
                            "Value": "OU=Disabled Users,DC=example,
DC=com"
                        }
                    ],
                    "Identity Attribute Triggers": [
                        {
                            "Attribute": "cloudLifecycleState",
                            "Value": "inactive",
                            "Operation": "eq"
                        },
                        {
                            "Attribute": "employeeType",
                            "Value": "Employee",
                            "Operation": "eq"
                        }
                    ],
                    "Account Attribute Update Triggers": [
                        {
                            "Attribute": "extensionAttribute1",
                            "Value": "inactive",
                            "Operation": "eq"
                        }
                    ],
                    "Operation": "Modify"
                },
                {
                    "eventActions": [
                        {
                            "Action": "ADMoveAccount",
                            "Attribute": "AC_NewParent",
                            "Value": "OU=Disabled Contractors,
DC=example,DC=com"
                        },
                        {
                                "Action": "UpdateAttribute",
                                "Attribute": "accountExpires",
                                "Value": "#{now.EPOCH_TIME_WIN32}"
                            }
                    ],
                    "Identity Attribute Triggers": [
                        {
                            "Attribute": "cloudLifecycleState",
                            "Value": "inactive",
                            "Operation": "eq"
                        },
                        {
                            "Attribute": "employeeType",
                            "Value": "Contractor",
                            "Operation": "eq"
                        }
                    ],
                    "Account Attribute Update Triggers": [
                        {
                            "Attribute": "extensionAttribute1",
                            "Value": "inactive",
                            "Operation": "eq"
                        }
                    ],
                    "Operation": "Modify"
                },
                {
                    "eventActions": [
                        {
                            "Action": "ChangeOperation",
                            "Attribute": null,
                            "Value": "Delete"
                        }
                    ],
                    "Identity Attribute Triggers": [
                        {
                            "Attribute": "cloudLifecycleState",
                            "Value": "delete",
                            "Operation": "eq"
                        }
                    ],
                    "Operation": "Enable"
                }
            ]
        }
    }
]
```

- Add the following configuration to the Okta source using V3 PATCH on the source.
    - Map on Line 7: Term actions for all users
    - Map on Line 29: Term actions for all users for users already disabled
    - Map on Line 58: Deletion action

```
[
  {
          "op": "add",
          "path": "/connectorAttributes/cloudServicesIDNSetup",
        "value": {
            "eventConfigurations": [
                {
                    "eventActions": [
                        {
                            "Action": "RemoveEntitlements",
                            "Attribute": "roles",
                            "Value": null
                        },
                        {
                            "Action": "RemoveEntitlements",
                            "Attribute": "groups",
                            "Value": null
                        }
                    ],
                    "Identity Attribute Triggers": [
                        {
                            "Attribute": "cloudLifecycleState",
                            "Value": "inactive",
                            "Operation": "eq"
                        }
                    ],
                    "Operation": "Disable"
                },
                {
                    "eventActions": [
                        {
                            "Action": "RemoveEntitlements",
                            "Attribute": "roles",
                            "Value": null
                        },
                        {
                            "Action": "RemoveEntitlements",
                            "Attribute": "groups",
                            "Value": null
                        }
                    ],
                    "Identity Attribute Triggers": [
                        {
                            "Attribute": "cloudLifecycleState",
                            "Value": "inactive",
                            "Operation": "eq"
                        }
                    ],
                    "Account Attribute Update Triggers": [
                        {
                            "Attribute": "extensionAttribute1",
                            "Value": "inactive",
                            "Operation": "eq"
                        }
                    ],
                    "Operation": "Modify"
                },
                {
                    "eventActions": [
                        {
                            "Action": "ChangeOperation",
                            "Attribute": null,
                            "Value": "Delete"
                        }
                    ],
                    "Identity Attribute Triggers": [
                        {
                            "Attribute": "cloudLifecycleState",
                            "Value": "delete",
                            "Operation": "eq"
                        }
                    ],
                    "Operation": "Enable"
                }
            ]
        }
    }
]
```

Update Description

Requirements

- Whenever IdentityNow makes a modification to an AD account, we want to set the description to update
- Password changes should NOT modify description

Design

- Normal Lifecycle processing according to best practices
- Add the “Services Standard IdentityNow BeforeProvisioning Rule” to AD as the beforeProvisioningRule

- Add the following configuration to the AD source using V3 PATCH on the source.
  - Map on Line 7: Stop processing for password updates
  - Map on Line 24: Update description for create
  - Map on Line 34: Update description for modify
  - Map on Line 44: Update description for enable
  - Map on Line 54: Update description for disable

NOTE: The description updates can be done via Account Profiles. If description is an account schema attribute, this should be done via the profile rather than BeforeProvisioningRule. This is just an example of usage.

```
[
  {
          "op": "add",
          "path": "/connectorAttributes/cloudServicesIDNSetup",
        "value": {
            "eventConfigurations": [
                {
                    "eventActions": [
                        {
                            "Action": "StopProcessing",
                            "Attribute": null,
                            "Value": null
                        }
                    ],
                    "Account Attribute Update Triggers":[
                                            {
                                                "Attribute":"password",
                                              "Operation":"ne",
                                              "Value":"*"
                                            }
                                        ],
                    "Operation": "Modify"
                },
                {
                    "eventActions": [
                        {
                            "Action": "UpdateAttribute",
                            "Attribute": "description",
                            "Value": "Created by IdentityNow Automation
on #{now.MM/dd/yyyy}"
                        }
                    ],
                    "Operation": "Create"
                },
                {
                    "eventActions": [
                        {
                            "Action": "UpdateAttribute",
                            "Attribute": "description",
                            "Value": "Updated by IdentityNow Automation
on #{now.MM/dd/yyyy}"
                        }
                    ],
                    "Operation": "Modify"
                },
                {
                    "eventActions": [
                        {
                            "Action": "UpdateAttribute",
                            "Attribute": "description",
                            "Value": "Enabled by IdentityNow Automation
on #{now.MM/dd/yyyy}"
                        }
                    ],
                    "Operation": "Enable"
                },
                {
                    "eventActions": [
                        {
                            "Action": "UpdateAttribute",
                            "Attribute": "description",
                            "Value": "Disabled by IdentityNow
Automation on #{now.MM/dd/yyyy}"
                        }
                    ],
                    "Operation": "Disable"
                }
            ]
        }
    }
]
```