

Life cycle rules management

Life cycle rules can be used to automate the large number of manual tasks that administrators must perform due to common reoccurring events, such as account inactivity, password expiration, or contract expiration, which are driven by business policies. Life cycle rules can also eliminate the potential of some policies to go unenforced.

Overview

Establishing life cycle rules allows administrators to define events that can be triggered based on a time interval or based on time and matching criteria evaluated against an entity. The administrator can then associate life cycle operations to run as a result of that event. All life cycle rules consist of two parts:

- The definition of an event that triggers the rule
- The identification of the life cycle operation that runs the actions specified in the rule

Each rule can be defined in one of these ways:

- Global
- Associated with an entity type
- Associated with an entity

For global rules, an event is defined by a time interval. For example, once a month, or on every Monday at 8:00 a.m. Global life cycle rules are independent of any particular system entity. The life cycle operations that can be invoked by a global rule must also be global in nature because there is no context available to call an entity- or entity type-based operation.

Entity and entity type rules also have an event with a time interval. However, the goal of these rules is to affect multiple entities at one time.

Matching criteria for events

A separate event is triggered for each life cycle object. To prevent events occurring for possibly thousands of objects that might not be related to the rule, a matching criteria is available for these events.

Without the matching criteria, every object of the given entity or entity type has the associated life cycle operation performed on it.

With the criteria, only objects that meet the criteria have the operations performed. The criteria is defined using LDAP filter syntax. The filter identifies any objects that meet the criteria and causes the event to be triggered for only those objects. If no object matches the filter, the event is not triggered. For example, the criteria might be for any accounts where (erAccountStatus=1), which means the accounts are suspended.

Note: The 'erstatus' replaced with 'erAccountStatus' and ' '(empty space) before and after the equal sign (=) are removed.

Life cycle rule filtering and scheduling

Because the filter is based on attributes, only the attributes associated with the schema of the entity or entity type are accepted.

There might also be the need to include environment data or external data into the filter, such as the current time or a value obtained from a customer database. The inclusion of this data is achieved by allowing macros to be placed in the filter. For example, a filter checking if a password has been changed within the last 30 days might read as follows:

`(erPswdLastChanged>=${system.date - 30}).`

Note: Leaving the filter blank will return all entities. Entity relationship macros can be used in life cycle rule filters.

The interval defined for an event can be constructed from the following options:

Daily

Triggers the life cycle event every day. After you select this option, click the clock icon to specify a time in the At this time field.

Weekly

Triggers the life cycle event once a week. After you select this option, select a day from the On this day of the week list, and then click the clock icon to specify a time in the At this time field.

Monthly

Triggers the life cycle event once a month. After you select this option, select a date from the On this day of the month list, and then click the clock icon to specify a time in the At this time field.

Hourly

Triggers the life cycle event once an hour. After you select this option, select a time from the At this minute list.

Annually

Triggers the life cycle event on a specific date and time of the year. After you select this option, select a month from the Month list. Then select a date from the On this day of the month list, and then click the clock icon to specify a time in the At this time field.

During a specific month

Triggers the life cycle event on a specific month, day, and time. After you select this option, select a month from the Month list. Then select a day from the On this day of the week list, and then click the clock icon to specify a time in the At this time field.

Quarterly

Trigger\s the life cycle event four times per year on a specific day and time of the quarter. The reconciliation will occur on the specified day past January 1, April 1, July 1, and October 1. After you select this option, select a day from the On this day list, and then click the clock icon to specify a time in the At this time field.

Semi-Annually

Triggers the life cycle event two times per year on a specific day and time of the half-year. The reconciliation will occur on the specified day past January 1 and July 1. After you select this option, select a day from the On this day list, and then click the clock icon to specify a time in the At this time field.

Note: More than one schedule can be specified.

A life cycle rule evaluation schedule contains only a reference to a corresponding rule definition. Therefore, if a life cycle rule definition changes before the scheduled evaluation starts, the evaluation uses the updated version of the definition, and not the rule definition that was originally scheduled.

In the following example, a life cycle rule is created to check once a day for accounts which have not had their password changed in 90 days. An e-mail notification will be sent to owners of accounts that meet the life cycle rule search criteria, informing them that they need to change their passwords.

First, a life cycle operation named `remindToChangePassword` is constructed for the Account entity type. It is defined as an instance-based (not static) operation, and so it has the account object itself as an input parameter. The business logic of the operation is defined simply with one work order activity that sends the reminder message to the owner of the account and includes the user ID of the account in the message.

A life cycle rule is then constructed for the Account Entity Type named `passwordExpiration` that references the `remindToChangePassword` operation and has an event with an evaluation interval of **daily** at **12:00 A.M.**. It also has the following filter: `(&(erAccountStatus=0)(erPswdLastChanged<= ${system.date - 90}))`.

Note 1:

- Filter `'(&(employeeType=active) (erPswdLastChanged >= ${system.date - 30}))'` is replaced with `'(erPswdLastChanged >= ${system.date - 30})'`.
- Extra brackets, extra ' ' (empty spaces) and `'&(employeeType=active)'` part of the filter are removed from the filter.

Note 2:

- Filter `'(&(employeeType=active) (erPswdLastChanged<= ${system.date - 90}))'` is replaced with `'(&(erAccountStatus=0)(erPswdLastChanged<= ${system.date - 90}))'`.
- This `'(employeeType=active)'` part of filter is replaced with `'(erAccountStatus=0)'` and extra ' ' (empty spaces) is removed from the filter.

Section: Configuring -> Life cycle rules management -> Adding life cycle rules for entities

URL: http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc_5.0/tsk/tsk_ic_lifecycle_add.htm

Adding life cycle rules for entities

Use these instructions to define life cycle rules for entities.

Before you begin

Only system administrators can define life cycle rules.

About this task

Life cycle rules trigger operations that are defined in the Manage Operations task. Depending on the type of life cycle rule, the corresponding operations defined at the level are available.

Life cycle rules are different from operations in that the life cycle rule that is defined at entity type or entity level does not override the life cycle rule defined at a higher level. Each are valid life cycle events that are capable of being run independently based on the schedule that is defined.

Procedure

To add a new life cycle rule for an entity type, complete these steps:

1. From the navigation tree, click Configure System > Manage Life Cycle Rules. The Manage Life Cycle Rules page is displayed.
2. On the Manage Life Cycle Rules page, select one of the following life cycle rule levels:
 - Select Global level to define a life cycle rule that has no entity context.
 - Select Entity type level to define a life cycle rule that is applicable to the entity type. Select an entity type from the Entity Type list.
 - Select Entity level to define a life cycle rule that is applicable to a specific entity instance type. Select an entity type from the Entity Type list, and then select an entity from the Entity list.
3. Click Add. The Manage Life Cycle Rules notebook is displayed.
4. On the General page of the Manage Life Cycle Rules notebook, complete these steps:
 - a. In the Name field, type a unique name for the life cycle rule that you want to define for the corresponding system entity.
 - b. Optional: In the Description field, type a description for the life cycle rule.

- c. From the Operation list, select an operation to be invoked when the event occurs. Only operations without input parameters are allowed to be run by the life cycle rule.
 - d. Click the Event tab.
2. On the Event page of the Manage Life Cycle Rules notebook, complete these steps:
- a. In the Search filter field, type an LDAP filter that identifies the objects that are affected by the event. For example, the following filter captures all active employees who have not changed their passwords in the past 90 days, calculated from the date that the life cycle event occurs: **(&(employeeType=active)(erPswdLastChanged<= \${system.date} - 90))**

Note: The Search filter is not applicable to global level life cycle rules because global level life cycle rules do not have entity context.

- b. Click Add to define a schedule for the life cycle rule. The Define Schedule page is displayed.
3. On the Define Schedule page, define a schedule for the life cycle rule to run, and then click OK. The fields displayed depend on the scheduling option that is selected. The new schedule is displayed on the Event page of the Manage Life Cycle Rules notebook.
4. Click OK to save the life cycle rule and close the notebook.

Results

A message is displayed, indicating that you successfully created a new life cycle rule for the entity. Click Close.

What to do next

When the Manage Life Cycle Rules page is displayed, click Refresh to refresh the Life Cycle Rules table and display the new life cycle rule.

Note: The spelling of 'employeeType' is corrected to 'employeeType' and extra ' '(empty space) is removed from the filter.

Section: Configuring->Life cycle rules management->LDAP filter expressions->System expressions

URL:http://publib.boulder.ibm.com/infocenter/tivihelp/v2r1/topic/com.ibm.itim.doc_5.0/ref/ref_ic_life_cycle_system_expressions.htm

System expressions

System expressions are used to target domain objects based on generalized time values relative to the current system date.

The system expression syntax has relatively few elements.

System expressions consist of an attribute name, a relational operator (`<=` or `>=`), a dollar sign (`$`) followed by a curly brace (`{`) immediately followed by the `system.date` keywords, then a plus or minus arithmetic operator (`+`/`-`) followed by a number in days, and then a right curly brace (`}`) to close the expression. For example:

```
(gmtattributename[<|=|>=]$ {system.date [ + | - ] days})
```

System expressions resolve to a concrete LDAP filter that is understood by an LDAP directory server or the built-in Tivoli Identity Manager filter interpreter. For example, below is a filter that targets accounts with passwords 90 days or older:

```
(erpswdlastchanged<= ${system.date - 90})
```

The above example can be used in an ACI for accounts that grants read and write access to the password attribute to allow users to update their passwords. The same filter can also be used in a life cycle rule that suspends accounts if the account's password has not been changed in the last 90 days. This particular filter expression resolves to the following concrete LDAP filter:

```
(erpswdlastchanged<=200912311200Z)
```

It is also possible and syntactically valid to express a range of dates as the criteria to match against domain objects by embedding more than one system expression in a composite filter as in the following example:

```
(&(erpswdlastchanged>= ${system.date - 90})(!(erpswdlastchanged>= ${system.date - 30})))
```

The filter matches accounts with passwords ranging from 90 to 30 days old. Other combinations and composite filters are useful, depending on how complex the filter needs to be and how many objects are targeted for a match.

Note:

- The filter (gmtattributename[<=|>=]\${system.date [+ | -] days}) is corrected by replacing ‘=>’ with ‘>=’ and removing ' '(empty space).
- The attribute name ‘erpswlastchanged’ is changed to correct attribute name ‘erpswdlastchanged’.
- The bracket '(' present before the character ‘!’ in ‘(&(erpswlastchanged>=\${system.date – 90})((!(erpswlastchanged>=\${system.date – 30})))’ filter is removed from the filter.