ELRR Statement Processing Rules

Rev. 07/25/2025 - Cliff

1. Learning Record Statements

Statements that update learning record state and represent learning resource interactions

1.1. Completed (no `success` value)

This rule covers the case of a completed statement with no success value.

Verb: https://adlnet.gov/expapi/verbs/completed
`Conditions:`
- `$.result.success = null`
- `$.result.complete`

Result(s):
- Create Actor if does not exist
- Create Learning Resource if does not exist
- Create Learning Record if does not exist
- LearningRecord.status = COMPLETED

1.2. Completed (`success=true`)

This rule covers the case of a completed statement with success = true.

Verb: https://adlnet.gov/expapi/verbs/completed
`Conditions:`
- `$.result.success = true`

Result(s):
- Create Actor if does not exist
- Create Learning Resource if does not exist
- Create Learning Record if does not exist
- LearningRecord.status = PASSED

1.3. Completed (`success=false`)

This rule covers the case of a completed statement with success = true.

Verb: `https://adlnet.gov/expapi/verbs/completed`
`Conditions:`
```
- $.result.success = false
```

Result(s):
- Create Actor if does not exist
- Create Learning Resource if does not exist
- Create Learning Record if does not exist
- LearningRecord.status = FAILED

1.4. Passed

This rule covers the case of a passed statement. Possibly combine with 1.2

Verb: `https://adlnet.gov/expapi/verbs/passed`

Result(s):
- Create Actor if does not exist
- Create Learning Resource if does not exist
- Create Learning Record if does not exist
- LearningRecord.status = PASSED

1.5. Failed

This rule covers the case of a passed statement. Possibly combine with 1.3

Verb: `https://adlnet.gov/expapi/verbs/failed`

Result(s):
- Create Actor if does not exist
- Create Learning Resource if does not exist
- Create Learning Record if does not exist
- LearningRecord.status = FAILED

1.6 Satisfied

This rule covers the case of a satisfied statement. The difference is usually in the LMS as a course might have many sub-components and automatically get marked satisfied by the completion of them. Functionally in ELRR they have the same effect as completed. Possibly combine with 1.1, 1.2, and 1.3.

Verb: https://w3id.org/xapi/adl/verbs/satisfied

Result(s):
- Create Actor if does not exist
- Create Learning Resource if does not exist
- Create Learning Record if does not exist
- LearningRecord.status = COMPLETED (or PASSED/FAILED depending on $.result.success)

1.7 Attempted

This rule covers the launch/attempt of an activity. Launched verb is not used because it may not actually start the content.

Verb: https://adlnet.gov/expapi/verbs/initialized
Result(s):
- Create Actor if does not exist
- Create Learning Resource if does not exist
- Create Learning Record if does not exist
- LearningRecord.status = ATTEMPTED

1.8 Registered

This rule covers the registration of a learner for an activity.

Verb: https://w3id.org/xapi/tla/verbs/registered
Result(s):
- Create Actor if does not exist
- Create Learning Resource if does not exist
- Create Learning Record if does not exist
- LearningRecord.status = REGISTERED (*requires adding a new enum to schema/entities* in code)

1.9 Scheduled

This rule covers the scheduling of an activity for a learner. It has the same record outcome as registration.

Verb: https://w3id.org/xapi/tla/verbs/scheduled
Result(s):
- Create Actor if does not exist
- Create Learning Resource if does not exist
- Create Learning Record if does not exist

- LearningRecord.status = REGISTERED (*requires adding a new enum to schema/entities* in code)

2. Competency and Credential (Qualifications)

Statements that update qualification (comp & cred) state and represent data from assertion engines

2.1. Achieved (Competency)

This rule covers the case of an achieved statement for a competency. Optionally sets expiry.

Verb: `http://adlnet.gov/expapi/verbs/achieved`
`Conditions:`
- `$.object.definition.type` **`!=`** `https://w3id.org/xapi/cred/activities/credential`
- `Competency ActivityType can be numerous things`

Result(s):
- Create Actor if does not exist
- Create Competency if does not exist
- Create Personal Competency if does not exist
- Set PersonalCompetency.HasRecord to true
- If `$.context.extensions['https://w3id.org/xapi/comp/contextextensions/expires'] is not null`
    - Set PersonalCompetency.expires to extension value (should be ISO8601 date field, requires addition of java LocalDate field to schema/entities)
    - Else do not set expires

2.2. Achieved (Credential)

This rule covers the case of an achieved statement for a credential. Optionally sets expiry.

Verb: `http://adlnet.gov/expapi/verbs/achieved`
`Conditions:`
- `$.object.definition.type` **`==`** `https://w3id.org/xapi/cred/activities/credential`

Result(s):
- Create Actor if does not exist
- Create Credential if does not exist
- Create Personal Credential if does not exist
- Set PersonalCredential.HasRecord to true

- If `$.context.extensions['https://w3id.org/xapi/comp/contextextensions/expires'] is not null`
    - Set PersonalCredential.expires to extension value (should be ISO8601 date field, requires addition of java LocalDate field to schema/entities)
    - Else do not set expires

3. Goal Statements

Statements which update learner goals, either by assignment, recommendation, or self-assignment.

3.1. Assigned (Not by learner)

This rule covers the case of an assigned statement for a goal, coming from an assigner.

Verb: `https://w3id.org/xapi/tla/verbs/assigned`
`$.object.defintion.type` `== https://w3id.org/xapi/activities/goal`
`$.object.definition.extensions[‘http://xapi.edlm/goals/activity-extensions/goal-type’]` `== ASSIGNED, RECOMMENDED`

Result(s):

- Actor is ‘assigner’ (you may need to change Goals entity and DDL to accommodate this)
- Extract learner from context extension: `https://yetanalytics.com/profiles/prepositions/concepts/context-extensions/to`
- Take Goal Type from goal-type extension above. Default is assigned.
- Set expiry from context extension http://xapi.edlm/goals/activity-extensions/expires
- Set achieved by from context extension http://xapi.edlm/goals/activity-extensions/achieved-by
- Set Start from timestamp
- Populate goals as new (or linked) Learner Resources, Competencies, and Credentials sourced from $.context.contextActivities.other. They are already xAPI Objects, no need to marshall

3.2. Was-Assigned (By learner or assigner)

This rule covers the case of an inverted assigned statement for a goal.

Verb: `https://w3id.org/xapi/tla/verbs/was-assigned`
`$.object.defintion.type` `== https://w3id.org/xapi/activities/goal`
`$.object.definition.extensions[‘http://xapi.edlm/goals/activity-extensions/goal-type’]` `== ASSIGNED, RECOMMENDED, SELF-ASSIGNED`

Result(s):

- Actor is learner
- Extract assigner (if present) from context extension: https://yetanalytics.com/profiles/prepositions/concepts/context-extensions/by
    - If not present it's default self-assigned
    - If present its default assigned
- Take Goal Type from goal-type extension above
- Set expiry from context extension http://xapi.edlm/goals/activity-extensions/expires if present
- Set achieved by from context extension http://xapi.edlm/goals/activity-extensions/achieved-by if present
- Set Start from timestamp
- Populate goals as new (or linked) Learner Resources, Competencies, and Credentials sourced from $.context.contextActivities.other. They are already xAPI Objects, no need to marshall

3.3. Removed (By learner)

This rule covers the case of a learner removing a self-signed goal

Verb: `http://activitystrea.ms/removed`
`$.object.defintion.type` `==` `https://w3id.org/xapi/activities/goal`
`$.object.definition.extensions['http://xapi.edlm/goals/activity-extensions/goal-type']` `== SELF-ASSIGNED`

Result(s):

- Actor is learner
- Delete goal from learner if exists

3.4. Removed (By assigner)

This rule covers the case of an assigner removing a goal from a learner

Verb: `http://activitystrea.ms/removed`
`$.object.defintion.type` `==` `https://w3id.org/xapi/activities/goal`
`$.object.definition.extensions['http://xapi.edlm/goals/activity-extensions/goal-type']` `== SELF-ASSIGNED`

Result(s):

- Assigner is actor
- Learner can be found at https://yetanalytics.com/profiles/prepositions/concepts/context-extensions/from
- Delete goal from learner if exists