# xAPI Profile Primer

*This document uses the Human-Readable template is provided by Jason Haag as part of his presentation to xAPI Cohort Fall 2021's team-xAPI-Profile.*


**Authors:**

Brian Duck

Laurel Ryshpan,

Chetan Sachdev,

Peter Hawkins,

Judy Katz,

Henry Ryng

Mile Divovic

and

Julian Davis.



Thanks to Jason Haag and Duncan Welder.


**Document Version: 2-10**

# Table of Contents

# Revision History

| Version | Release Date | Notes |
|---|---|---|
| 1-00 | | Initial draft of the profile. Team-xAPI-Profile review only. |
| **1-01** | | Revised Overview |
| **1-02** | | Added section 2 content |
| **1-03** | | Added URLs |
| **1-04** | | Trimmed extra content, revised footers, update ToC |
| **1.05** | | Added authors, reviewed with team |
| **1.06** | | Updated Authors, rebuilt page numbers in Table of Contents. |
| **1.07** | | Review with xAPI Cohort Fall 2021's team-xAPI-Profile before team presentations. |
| **2.08** | | Update prior to review with IEEE xAPI Profiles Work Group |
| **2.09** | | Added content, renumbered sections, rebuilt TOC. |
| **2.10** | | Added Navy / NETG Graphics |

*Document Revision History*

# 1. Overview

This document is a product of the xAPI Cohort Fall 2021 team-xAPI-Profile.

Our goal was to build a reference document for those interested in *locating and using* xAPI Profiles as a guide to create statements for a project or organization.

The team discussion page, where xAPI Cohort weekly updates were posted, is available for reference:

**https://github.com/orgs/xapicohort/teams/team-xapi-profile**

Note: for direct access to this page, you will need to:

1. Establish a free GitHub account
2. Join the free xAPI Cohort: **http://xapicohort.com**
3. Join the xAPI Cohort Organization in GitHub: **https://github.com/orgs/xapicohort**
4. View the page in GitHub: **https://github.com/orgs/xapicohort/teams/team-xapi-profile**

## 1.1 How to Read this Document

This document is provided as a primer to those wishing to locate and use a set of xAPI statements needed to capture interactions with learning materials.

In general, this is done through a series of statements that are specific to the type of interaction being captured.

For example, a course interaction has typically been handled by an LMS and captured via SCORM communications back to an LMS.

The current recommendation is to consider CMI5 as a alternate to SCORM, with traffic being sent to the LRS via xAPI statements using the CMI5 profile.

## 1.2 References for Vocabulary, Profiles, and the TinCan Registry

There are a number of tools and websites supporting xAPI, some dating back to the TinCan beta. Here we list those focused on the use of xAPI Profiles, including vocabulary servers and repositories.

### 1.2.1 TinCan Registry

There has been a great deal of progress made since the publication of the original TinCan registry. The TinCan registry is considered out of date, and has been replaced by the ADL xAPI Vocabulary.

#### TinCan Registry (*Out of Date*, use xAPI Profile Server)

https://registry.tincanapi.com

### 1.2.2 ADL xAPI Vocabulary Server

Although the xAPI Vocabulary Server remains active, it is presumed that the ADL xAPI Profile server should be used for the latest statement data.

#### xAPI Vocabulary Server (*No longer recommended*, use xAPI Profile Server)

http://xapi.vocab.pub/verbs/index.html

### 1.2.3 ADL Profile Server (Now the recommended source)

The 'ADL xAPI Profile Server' serves as the current source of truth for xAPI vocabulary and profile information.

The ADL has published a User Guide and Authoring Guide for the profile server, but from our perspective, they both are focused on the xAPI Profile server's new profile authoring features.

#### ADL xAPI Profile Server (2022 Recommended)

https://profiles.ADLnet.gov

#### ADL xAPI Profile Server (Broken and outdated link)

https://profiles.usalearning.net/profiles

#### User Guide

https://adlnet.gov/guides/xapi-profile-server/user-guide/

#### Authoring Guide

https://adlnet.gov/guides/xapi-profile-server/authoring-guide/

### 1.2.4 Torrance Learning's Learning Profile Flattener

Based on data from the xAPI Profile Server, Torrance Learning has published a tool the call the 'Learning Profile Flattener.' As of June 2022, the Profile Flattener has been updated to use profiles.ADLnet.gov as the source.

This tool should also be considered, but pulls data from the xAPI Profile Server via API.

#### Torrance learning Profile Flattener (Alternate to ADL xAPI Profile Server)

**https://torrancelearning.com/xapi/profiles/tools/flattener/**

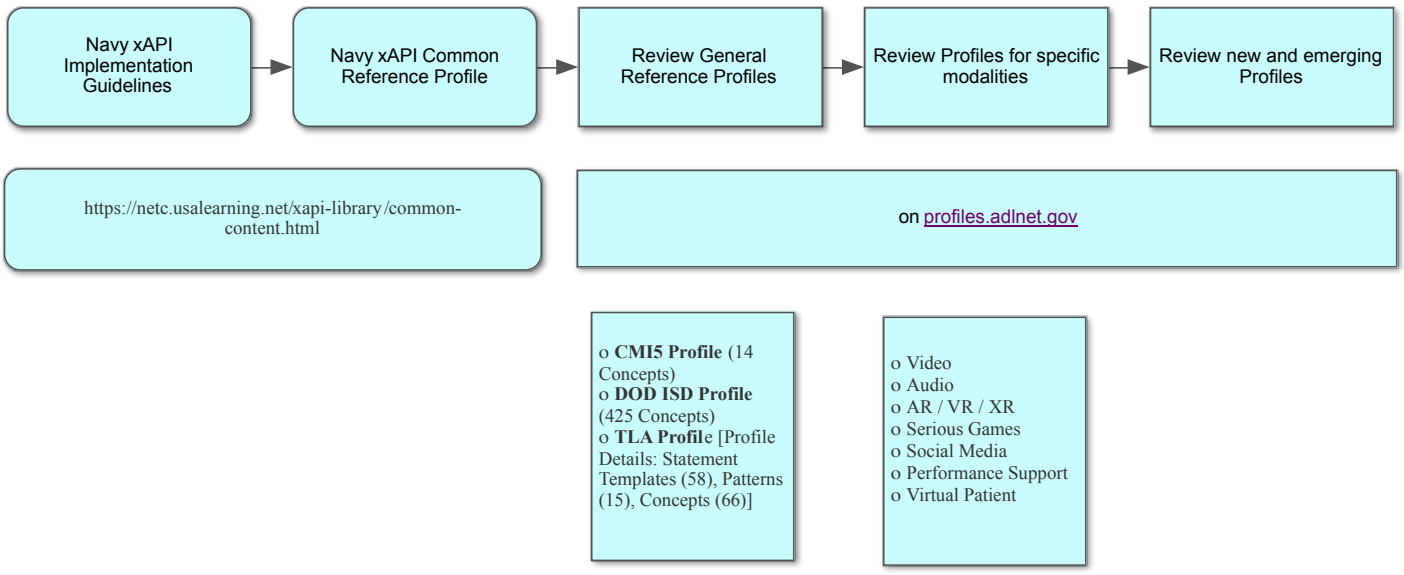
# 2. When to Use a Profile

The purpose of this document is to provide content developers with best practices, technical guidance, and standard requirements for implementing xAPI in specific content. This document is intended to serve as a starting reference, or primer, and provides implementation and conformance recommendations for locating resources to support xAPI.

When a project is initiated, it often has a specific audience associated with the sponsoring organization, using specific terminology from that organization is expected, but the fundamental purpose of xAPI is to store training data in a common format. Initiating a vocabulary specific to a specific audience should be avoided at the beginning of a project.

Generally, job, task, department, division, organization and industry specific terms should be confined to the 'context' segment of an xAPI statement.

Our recommendation is to approach a project with tiered structure in mind:



## 2.1 Review the "Navy xAPI Implementation Guidelines"

The Navy / NETC Implementation Guidelines may serve as the base of a foundation for the implementation of xAPI.

"The Navy xAPI Implementation Guidelines provides content developers with best practices, technical guidance, and standard requirements for implementing xAPI in Navy learning stack environments."

This approach should be customized to fit your project and organization.

This file is located at: **https://netc.usalearning.net/xapi-library/common-content.html**

The documentation have been produced in a human readable format, utilizing the template provided by Jason Haag, and covers a multitude of implementation considerations.

A summary overview of the xAPI Implementation Guidelines details the xAPI statement structure: Actor, Verb, Object, Result, Context, Timestamp. The document includes code samples, JSOD data structures, examples (best practices) and non-examples (to be avoided).

# NETC xAPI Implementation Guidelines

*Requirements and Guidelines for Implementing xAPI*

**Document Version: 1.0**

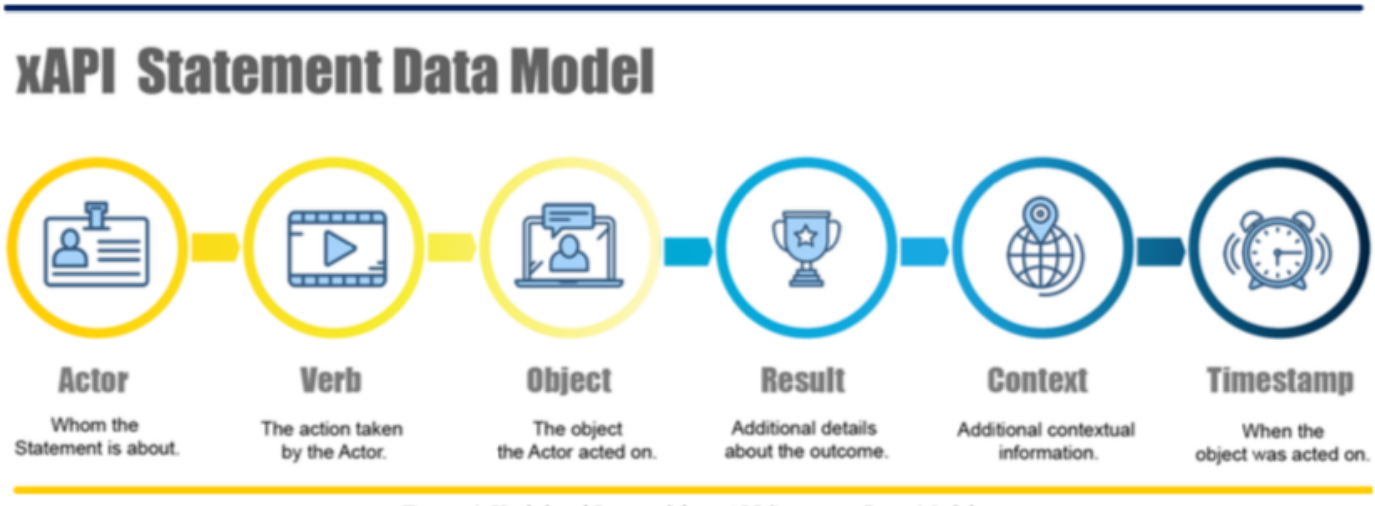
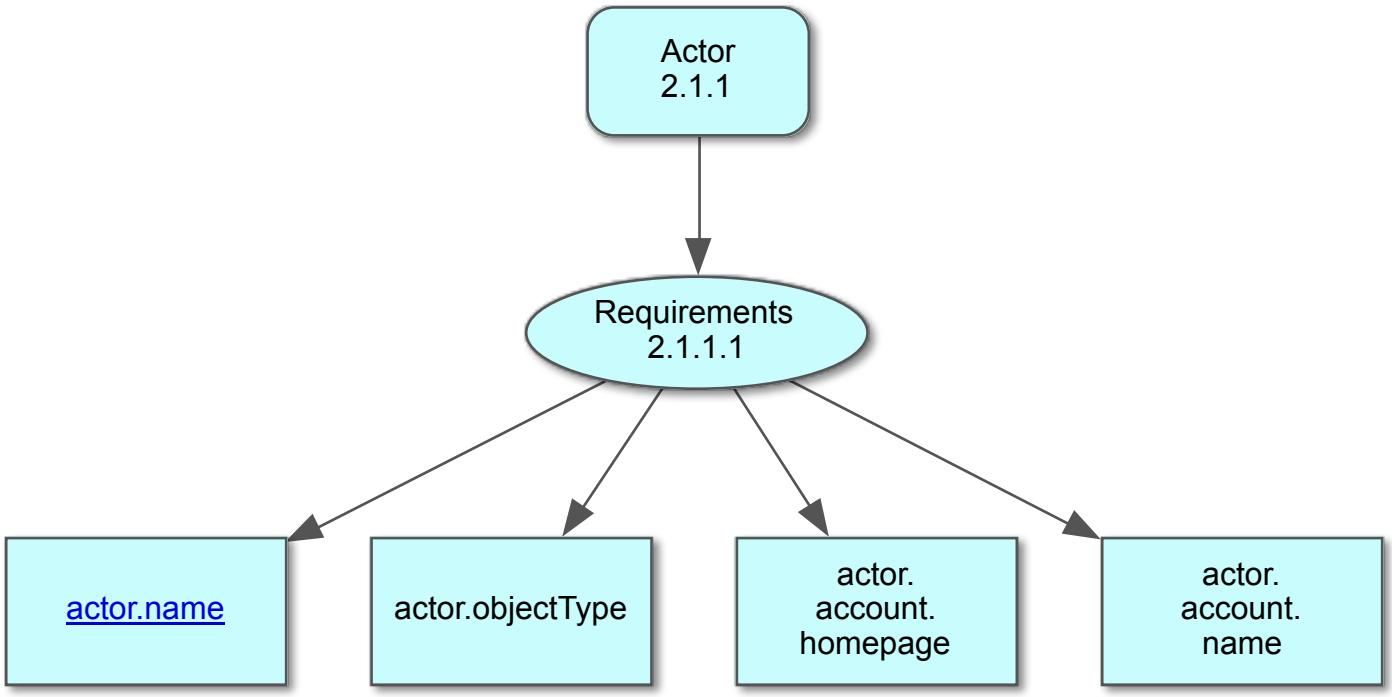
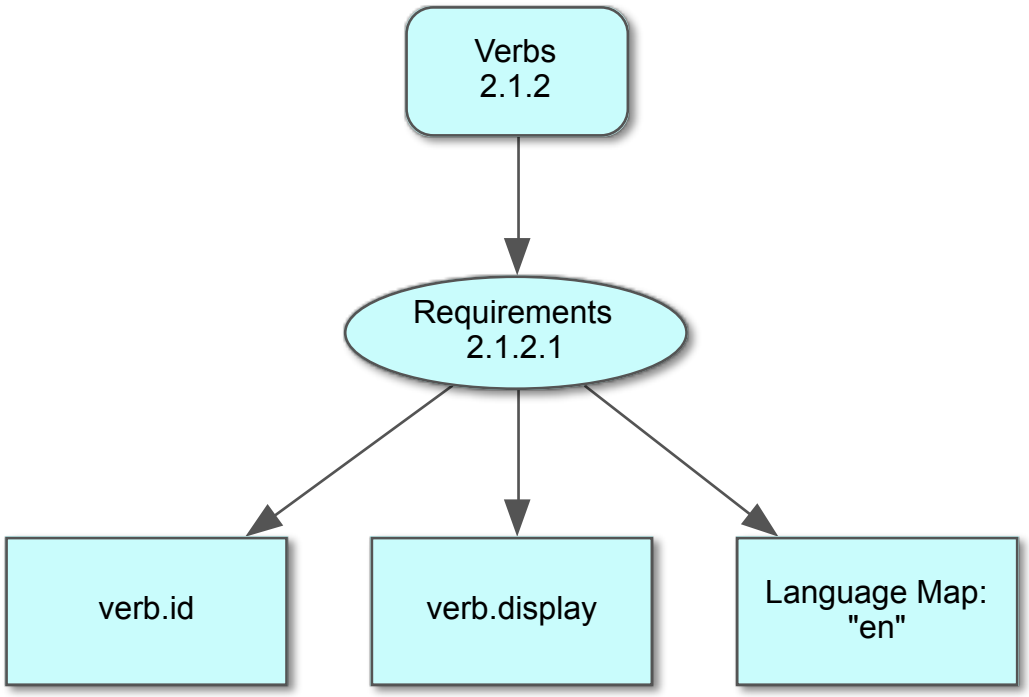
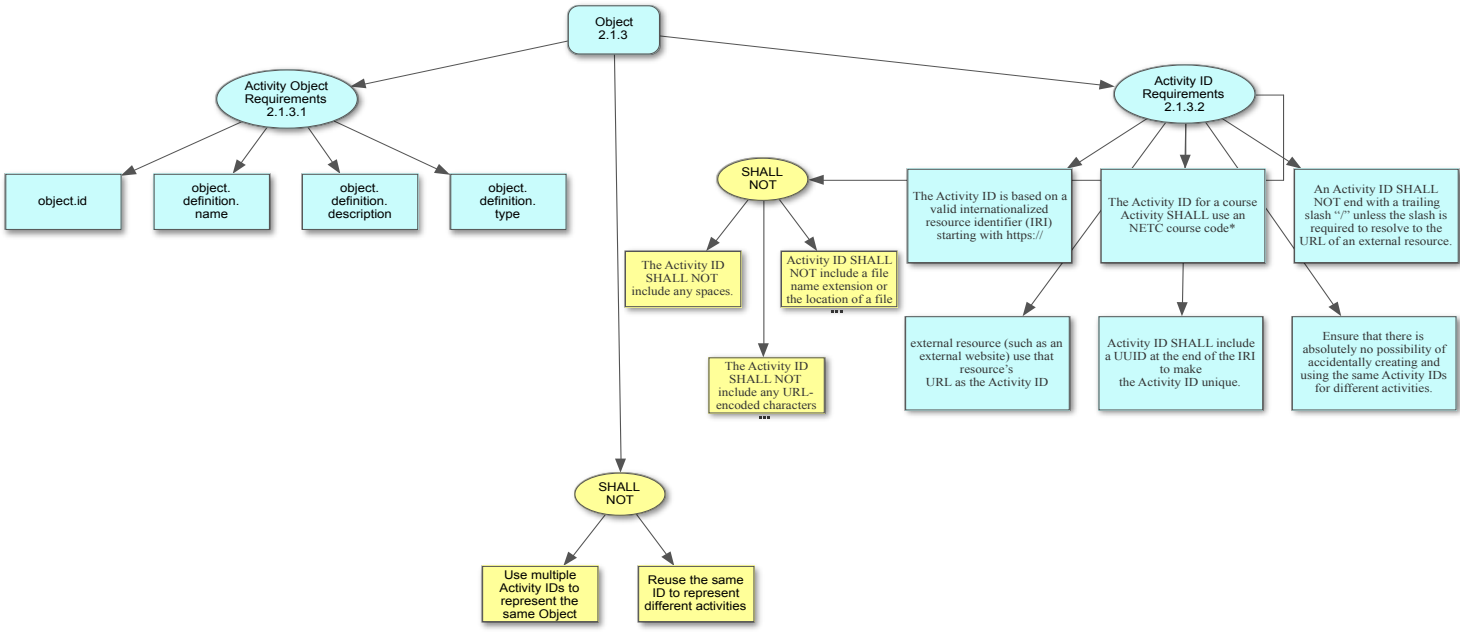
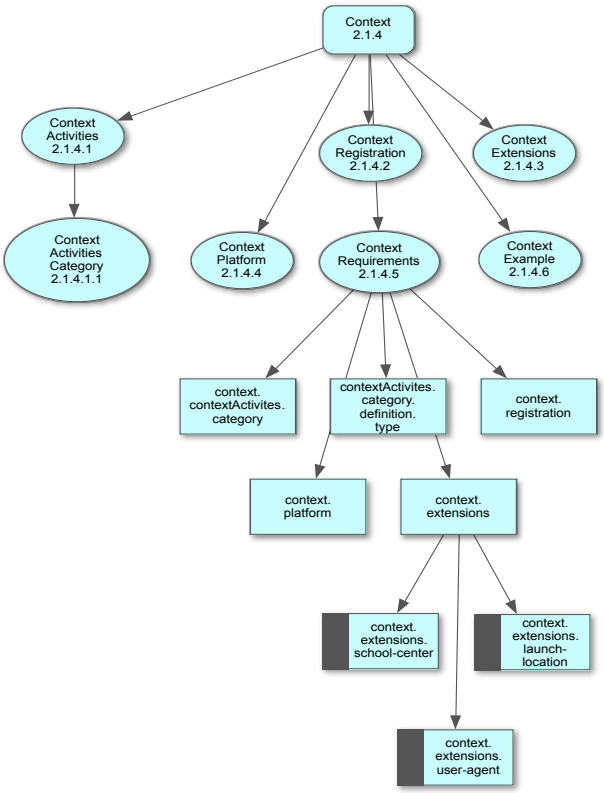


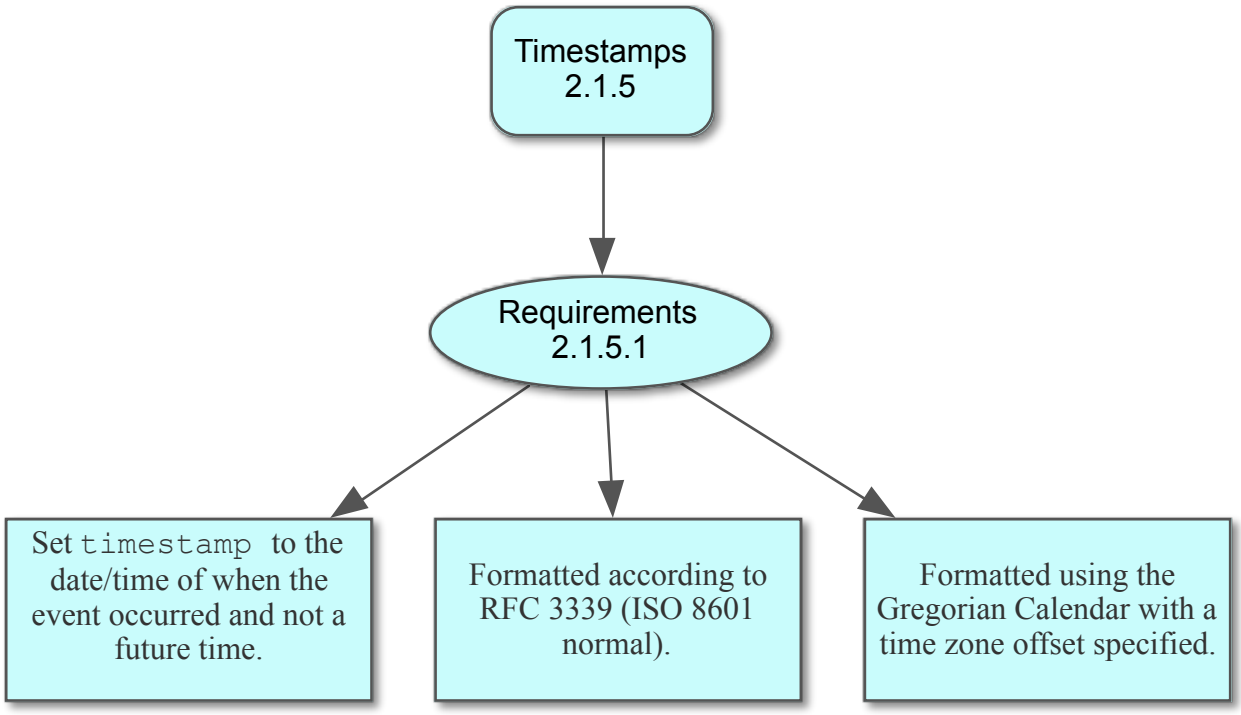

*Figure 1: High-level Parts of the xAPI Statement Data Model*


Actor
2.1.1
Requirements
2.1.1.1
actor.name
actor.objectType
actor.
account.
homepage
actor.
account.
name


Verbs
2.1.2
Requirements
2.1.2.1
verb.id
verb.display
Language Map:
"en"


Object
2.1.3
Activity Object Requirements
2.1.3.1
object.id
object.
definition.
name
object.
definition.
description
object.
definition.
type
Activity ID Requirements
2.1.3.2
SHALL NOT
The Activity ID SHALL NOT include any spaces.
Activity ID SHALL NOT include a file name extension or the location of a file
...
The Activity ID SHALL NOT include any URL-encoded characters
...
The Activity ID is based on a valid internationalized resource identifier (IRI) starting with https://
The Activity ID for a course Activity SHALL use an NETC course code*
An Activity ID SHALL NOT end with a trailing slash "/" unless the slash is required to resolve to the URL of an external resource.
external resource (such as an external website) use that resource's
URL as the Activity ID
Activity ID SHALL include a UUID at the end of the IRI to make
the Activity ID unique.
Ensure that there is absolutely no possibility of accidentally creating and using the same Activity IDs for different activities.
SHALL NOT
Use multiple Activity IDs to represent the same Object
Reuse the same ID to represent different activities


Context
2.1.4
Context Activities
2.1.4.1
Context Activities Category
2.1.4.1.1
Context Registration
2.1.4.2
Context Extensions
2.1.4.3
Context Platform
2.1.4.4
Context Requirements
2.1.4.5
Context Example
2.1.4.6
context.
contextActivites.
category
contextActivites.
category.
definition.
type
context.
registration
context.
platform
context.
extensions
context.
extensions.
school-center
context.
extensions.
launch-
location
context.
extensions.
user-agent




## 2.2 Review the "Navy Common Reference Profile"

The U.S. Navy / NETC common Reference Profile may serve as the next layer in a foundation for the implementation of xAPI.

"The U.S. Navy Common Reference Profile provides the rules, xAPI Statement requirements, and Statement examples for tracking general types of activities and content."

This file is located at:

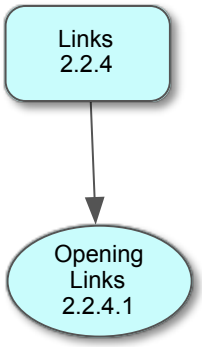
**https://netc.usalearning.net/xapi-library/common-content.html**

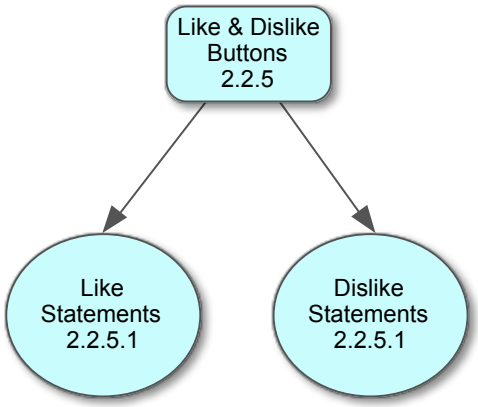
This approach should be customized to fit your project and organization.

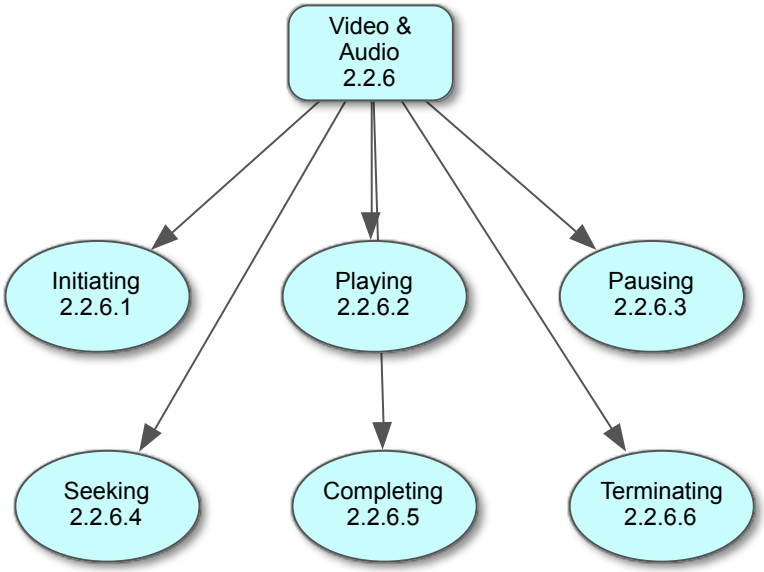
The documentation have been produced in a human readable format, utilizing the template provided by Jason Haag, and covers a multitude of implementation considerations.

A summary overview of the Common Reference Profile shows how files, pages, menus, links, like & dislike buttons, video and audio might be handled.

As these are common functions for enable standard features in applications, their familiarity is assured. Likewise, they are common features of eLearning, multimedia, and presentations.


Links
2.2.4
Opening
Links
2.2.4.1


Like & Dislike
Buttons
2.2.5
Like
Statements
2.2.5.1
Dislike
Statements
2.2.5.1


Video &
Audio
2.2.6
Initiating
2.2.6.1
Playing
2.2.6.2
Pausing
2.2.6.3
Seeking
2.2.6.4
Completing
2.2.6.5
Terminating
2.2.6.6

## 2.3 General xAPI Profiles:

A search of the xAPI Profile Server returns a number of broad profiles, which we have termed as 'General Profiles,' these are utilized when the vast majority of content to be published will utilize a common format, hosting profile, or be used to implement a learning ecosystem, as in the case of the Total Learning Architecture (TLA).

General xAPI Profiles include, but are not limited to:

- CMI5 Profile (14 Concepts)
- DOD ISD Profile (425 Concepts)
- TLA Profile [Profile Details: Statement Templates (58), Patterns (15), Concepts (66)]

**Search the ADL xAPI Profile Server:**

https://profiles.ADLnet.gov

See "How to select a profile" for next steps.

NOTE: Certain titles on the profile server have been identified as outdated, and *should be avoided*, these include: TinCan Vocabulary, Activity Streams Vocabulary and Actionable Data Book (ADB) Profile. If in doubt, copy and paste the URI from a few verbs into a browser to determine if they resolve correctly. If they do not resolve correctly, the profile may have become out of date.

## 2.4 Profiles for specific modailities:

Next, we would search the xAPI Profile Server for the modalities for existing profiles in use by course developers, or being considered by course designers.

These may include, but not be limited to:

- Video
- Audio
- AR / VR / XR
- Serious Games
- Social Media
- Performance Support
- Virtual Patient

**Search the ADL xAPI Profile Server:**

**https://profiles.ADLnet.gov** Also, see "How to select a profile" for next steps.

## 2.5 Review New and Emerging Vocabularies

After the recommended review of the Implementation Guidelines, Common Reference Profile, General xAPI Profiles, and profiles for specific Modalities, you should review the xAPI Profile Server for new and emerging vocabularies.

Although it may seem unlikely, there may be another effort to capture new and emerging technologies (currently including AR/VR/XR/MR) on the profile server. Prior to creating and publishing a new vocabulary, it would be best to review the xAPI Profile server for drafts, and newly published, profiles of interest.

**Search the ADL xAPI Profile Server:**

https://profiles.ADLnet.gov

# 3.0 When to Build a new profile

## 3.1 Define Custom Vocabulary

Finally, after the recommended review of the Implementation Guidelines, Common Reference Profile, General xAPI Profiles, and profiles for specific Modalities.

Even if this is identified as a need, it should be considered as a last resort and completed when other profiles have been considered.

As an example, Judy Katz the Product Manager for PeBL Pro, mentioned that their product was based on the profile for Actionable Data Book, however they have significantly enriched the vocabulary beyond the scope of the existing profile.

This is a solid indicator that they should consider the creation of a new profile, curated to the specifics of their project, to allow others to produce content that would be compatible with their efforts.

## 3.2 Publish a new profile to the profile server

If a custom vocabulary is developed, it should be considered for publication as new profile to the xAPI Profile Server.

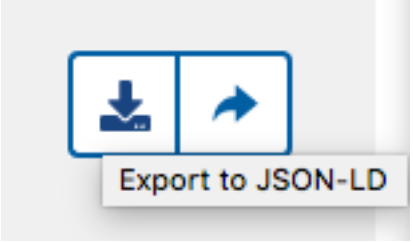
This ensures that materials will be produced in a consistent format, and be compatible with materials authored elsewhere.

In the PeBL Pro example, they will be creating a new profile on the xAPI Profile server. This ensures that future work with ePuBs will have the PeBL Profile as a starting point, and ensures some level of compatibility *may be enabled* between emerging tools.

## 3.3 How to select a Profile

Browse, or search, the profile server for terms of interest, modalities under consideration for development, or that are supported by existing capabilities of your learning ecosystem: Audio, Video, etc.

Once the profile has been identified, a PDF published from the profile server may serve as a valuable reference later in the development cycle. Care should be taken to review and document each populated tab in the profile: Profile Details, Statement Templates, Patterns, and Concepts make up the tabs available as of this publication date.




Additionally, the **JavaScript Object Notation – Linked Data** (JSON-LD) file should be downloaded for reference, but note that the current implementation on the xAPI Profile server means that this file will only contain concepts *specific to the profile*. Related concepts linked from other published profiles will appear on the concepts page of that profile, but will not be contained in the JSON-LD downloaded from the profile server or retrieved via the xAPI Profile Server's API.

# 4.0 How to Use a Profile

Once the JSON-LD file is downloaded, or retrieved via the xAPI Profile Server's API, it can be utilized to build the statements to be published via xAPI Statements from the published content.

This is typically handled by a JavaScript Library, such as TinCan JS, or JS Wrapper. Within the JS Library, there will be a file identified to have writable LRS credentials added.

Security of the LRS data is maintained by encryption of data, and these credentials. Note that creating credentials with write only capabilities for sending statements to the LRS is considered the norm. These credentials must be different from those used to access the LRS admin functions for creating or removing credentials, and defining and running reports.

Within an authoring tool, such as StoryLine or Captivate, the tool may have a JavaScript interface for publishing the course file with embedded JavaScript needed to send statements to the LRS.