ADL xAPI RMF Accreditation Project (xRAP)

Final Project Report / Test Results

29 July 2021

This work was supported by the U.S. Advanced Distributed Learning (ADL) Initiative (IDIQ# POM2615D0001 TO# 6010-01). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the ADL Initiative or the U.S. Government. The U.S.

2021

Experience API (xAPI)

Technical Report

A description of the changes and associated impacts from xAPI Version 1.0.3 to the draft IEEE Standard 9274.1.1

Prepared by the
USALearning Vendor Team
7/21/2021

EXECUTIVE SUMMARY

The Advanced Distributed Learning (ADL) Initiative, a government program reporting to the Defense Human Resources Activity (DHRA), established the Experience API (xAPI) technology in response to DoD stakeholder requirements. xAPI is an e-learning software specification that defines a common data format and interoperable communications protocol to track and access data about learning experiences. The xAPI data format is often thought of as a sentence like:

Mary Doe	**Completed**	**Cybersecurity 101**
{Actor}	{Verb}	{Object}

xAPI Version 1.0.3 is the current stable version. It was released in September 2016 and has since gained adoption in government, industry and academia. In 2018, members of the Institute of Electrical and Electronics Engineers (IEEE) formed a working group to standardize the xAPI specification. The IEEE standardization process includes additional rigor and Distinctive stakeholders that better ensure a generally applicable technology standard.

At the time this report was written, IEEE 9274.1.1 (xAPI 2.0) was in final draft form, nearing completion of the IEEE standardization process. Many of the changes in the IEEE draft standard are additive. While they add new features that do not change existing ones, there are several changes and clarifications to existing recommendations that affect xAPI developers.

This report describes each change from xAPI Version 1.0.3 to IEEE 9274.1.1 and its associated impact on developers of Learning Record Stores (LRS) and xAPI-enabled clients (e.g., e-learning content). The goal of this report is to ease the migration of xAPI-enabled products to conform to the new draft standard.

The remainder of this paper is focused on technical stakeholders that build LRSs, xAPI authoring tools, and xAPI content.

Experience API (xAPI)

FOUNDING ORGANIZATION

Advanced Distributed Learning (ADL) Initiative

xAPI REQUIREMENT

Provide an interoperable, vendor-neutral technology to store and access learning experience data

GOALS OF THE TECHNICAL REPORT

Ease the burden of migrating content, systems and tools from xAPI Version 1.0.3 to the new xAPI draft standard at IEEE (9274.1.1)

ADL Initiative
https://www.adlnet.gov

IEEE
https://www.ieee.org

1. OVERVIEW

When this paper was written, the Experience API (xAPI) was undergoing development at the Institute of Electrical and Electronics Engineers (IEEE). The Learning Technology Standards Committee (LTSC) of the IEEE Standards Association (SA) sponsored a Project Authorization Request (PAR) submitted on September 4, 2018 to move xAPI through the standardization process at IEEE. From September of 2018 to mid 2021, the xAPI Working Group created an update of the xAPI documentation known as IEEE Standard for JavaScript Object Notation (JSON) Data Model Format and Representational State Transfer (RESTful) Web Service for Learner Experience Data Tracking and Access (9274.1.1).

The IEEE 9274.1.1 Working Group[1] started with the xAPI Version 1.0.3 baseline hosted on the Advanced Distributed Learning (ADL) Initiative Github page[2] and began to make changes that were discussed and documented over the last five (5) years within the community. Many of these changes were additive and as a result do not affect prior implementations. However, some updates may affect certain xAPI implementations. The complete list of changes available in the official change log is also included in the sections below.

It is important to note that at the time this document was authored, the IEEE xAPI Standard was not yet an approved, accredited standard at IEEE. This could result in sections of this document being incorrect, invalid, or incomplete once the standard is complete.

2. PURPOSE

The purpose of this document is to provide a complete list of the changes from xAPI Version 1.0.3 to IEEE 9274.1.1. In addition, this document evaluates each change to determine the technical impact for implementers of xAPI. These impacts are divided into Learning Record Store (LRS) impacts and Client[3] impacts. Impacts include a potential severity as described in the Technical Impacts section of this document.

This document is primarily intended for a technical audience. It is written to inform LRS vendors, authoring tool developers, and other xAPI-enabled systems with information on technical updates required to conform to IEEE 9274.1.1. Although end users of these tools may find some utility in this information, most stakeholders should be able to continue using their xAPI environment with the understanding that LRSs, authoring tools and other systems will address the impacts herein.

[1] https://sagroups.ieee.org/9274-1-1/
[2] https://github.com/adlnet/xAPI-Spec
[3] Learning Record Provider (LRP) and Learning Record Consumer (LRC)

3. TECHNICAL IMPACTS

The changes and associated impacts described in the following subsections fall into several categories. These categories were used by the working group to divide the standard development work as the document was modified. The categories are:

1. General Updates
2. Should* Updates
3. Cybersecurity Updates
4. Context Agents and Context Groups Updates

Table 1 includes a description of the color-coded key used to indicate the impact severity at a glance.

<LRS | Client>	**NO IMPACT**
There is no technical impact to either LRS or Client (LRP/LRC) implementers. xAPI Version 1.0.3 implementations of the features require no change for IEEE 9274.1.1	
<LRS | Client>	**MINIMAL IMPACT**
There is a potential impact to either, or both, LRS and Client (LRP/LRC) implementers. Minimal technical impact changes either 1) require small development changes taking on average less than a week to update and test or 2) in surveys of the community, were already implemented as described in IEEE 9274.1.1 even though the requirement did not exist in xAPI Version 1.0.3.	
<LRS | Client>	**MAJOR IMPACT**
There is a potential impact to either, or both, LRS and content implementers. Major technical impact changes require significant development changes taking on average more than a week to update and test.	

Table 1-Color-coded impact severity key

Each section below includes a title including the change unique identifier (e.g., g1), a short summary of the change, the complete record(s) from the change log describing the update, and then the impact for LRSs and Clients, if any. In the event there is a minimal or major impact, a description of this impact and potential changes is provided.

References to the section of the two documentation sets are included as text only, not as hyperlinks. This is to account for the potential change in location of each specification. At the

time this document was created, the xAPI Version 1.0.3 is available at https://github.com/adlnet/xAPI-Spec and the IEEE 9274.1.1 open source resources are available at https://gitlab.com/IEEE-SA/xapi/9274.1.1/xapi-base-standard-documentation[4].

3.1. General Updates

The *General Updates* changes include mostly non-technical edits like organization updates, spelling, and grammar fixes. In these cases, there is generally no impact to xAPI implementers. However, there is a single general technical item describing the change to the xAPI version. This change does impact both LRS and Client developers. The sections below describe the *General Updates*.

g1 Specification Reorganization

The editors reorganized the xAPI specification when converting to the IEEE format. The documentation was broken down into separate documents for Learning Record Consumers/Providers and Learning Record Stores. Additionally, the structure and location of information was modified. However, the reorganization alone did not change the requirements within the specification.

The change log record for this item is:

ID	xAPI Version 1.0.3 Description	IEEE 9274.1.1 Updated Description
g1	The specification organized into several markdown files primarily compromising the following sections • **Part One: About the Experience API** Overall information about the xAPI • **Part Two: Experience API Data** xAPI JSON data formats and associated information • **Part Three: Data Processing, Validation, and Security** xAPI Resources/Endpoint description, validation and security considerations Requirements for LRS, LRPs, and LRCs are included in Part Two and Part Three.	The standard is organized into several markdown files • **9274.1.1 xAPI Base Standard Front Matter** IEEE required front matter materials and detailed standard table of contents • **9274.1.1 xAPI Base Standard Authors** IEEE xAPI working group xAPI authors • **9274.1.1 xAPI Base Standard Contributors** IEEE xAPI working group individual contributors

[4] Currently requires access to be granted by the IEEE xAPI Working Group Chair

		• **9274.1.1 xAPI Base Standard Acknowledgements** Additional acknowledgments beyond the current IEEE xAPI working group • **9274.1.1 xAPI Base Standard Overview** Overall information about the xAPI • **9274.1.1 xAPI Base Standard LRSs** Normative information for LRSs • **9274.1.1 xAPI Base Standard Content** Normative information for content (Learning Record Providers and Learning Record Consumers) implementing xAPI Requirements for LRSs and content (LRPs and LRCs) are organized into separate books targeted at a specific audience.

LRS	**NO IMPACT**
Client (LRP|LRC)	**NO IMPACT**

g2 MUST to SHALL

IEEE uses the term ‘MUST’ instead of ‘SHALL’ to express required behaviors or features. The editors updated the use of ‘SHALL’ to ‘MUST’ within xAPI specification to match the IEEE standard development guidelines. This does not change requirements so it does not impact software implementations.

The change log record for this item is:

ID	xAPI Version 1.0.3 Description	IEEE 9274.1.1 Updated Description
g2	There are three levels of obligation with	To align with the IEEE Standards

	regards to conformance to the xAPI specification identified by the terms MUST, SHOULD and MAY. A service or system that fails to implement a MUST (or a MUST NOT) requirement is non-conformant. Failing to meet a SHOULD requirement is not a violation of conformity, but goes against the recommendations of the specification. MAY indicates an option, to be decided by the developer with no consequences for conformity. Usage of these terms outside of requirement language does not designate a requirement and is avoided whenever possible.	Association (SA) guidance, xAPI MUSTs were changed to SHALLs. Although MUST is becoming the standard for requirements in technical specifications, IEEE SA mandates the use of SHALL. xAPI was updated to follow IEEE SA guidance at https://standards.ieee.org/develop/drafting standard/write.html. Complete definitions of MUST, SHOULD, MAY, MUST NOT and SHOULD NOT are found in RFC 2119. IEEE adheres to these definitions. Note that MUST and SHALL are equivalent.

LRS	**NO IMPACT**

Client (LRP|LRC)	**NO IMPACT**

g3 Minor Spelling and Grammar

The editors fixed typos while editing the specification. These were typically issues like spelling and grammar errors, none of which impact the requirements of the specification.

The change log record for this item is:

ID	xAPI Version 1.0.3 Description	IEEE 9274.1.1 Updated Description
g3	N/A	During the reorganization and while addressing the additional changes listed below, several updates were made to the grammar and formatting of the specification. These changes do not affect the technical aspects of the xAPI standard.

LRS	**NO IMPACT**

Client (LRP|LRC)	NO IMPACT

g4 Version update to 2.0.0

The version of the xAPI was updated as part of this change. As a result, the LRP shall include an HTTP header with the name X-Experience-API-Version with the value of '2.0.0'.

The change log record for this item is:

ID	xAPI Version 1.0.3 Description	IEEE 9274.1.1 Updated Description
g4	The original specification included the following information regarding `X-Experience-API-Version` Every request from a Client and every response from the LRS includes an HTTP header with the name `X-Experience-API-Version` and the version as the value. For example, `X-Experience-API-Version : 1.0.3` for version 1.0.3; see the Revision History for the current version of this specification.	The standard includes the following information regarding `X-Experience-API-Version` Every request to the LRS and every response from the LRS shall include an HTTP header with the name `X-Experience-API-Version` and the version as the value. For example, `X-Experience-API-Version: 2.0.0` for version 2.0.0

LRS	MINIMAL IMPACT

LRS developers will need to update their LRS to test for the `X-Experience-API-Version` header with the value of `2.0.0`. Other versions may also be supported and are outside the scope of this document.

Client (LRP|LRC)	MINIMAL IMPACT

LRP/LRC developers will need to update their client software, Learning Record Consumers (LRC) and Learning Record Providers (LRP), to include the header `X-Experience-API-Version` with a value of `2.0.0` when making API calls to an xAPI endpoint if they are supporting 9274.1.1. Other versions may also be supported and are outside the scope of this document.

3.2. “Should *” Updates

s1-s4 Additional Properties

In xAPI Version 1.0.3, the specification included a recommendation that additional properties should not be added to xAPI Statements. IEEE 9274.1.1 updates this recommendation, instead mandating that an LRP SHALL not add additional properties to Statements, unless they are added as extensions, where extensions are permitted and an LRS shall reject a Statement with additional properties other than extensions in the locations where extensions are allowed.

The change log records for this item are:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s1	From Section 2.2 Additional properties SHOULD* NOT be added to Statements unless explicitly allowed by this specification.	From Section 5.2.1 An LRP SHALL not add additional properties to Statements
s2	From Section 2.2 Additional properties SHOULD* NOT be added to Statements and other objects unless explicitly allowed by this specification and the LRS SHOULD* reject Statements containing such additional properties.	From Section 5.2.1 An LRP SHALL not add additional properties to Statements
s3	From Section 2.2 Additional properties SHOULD* NOT be added to Statements and other objects unless explicitly allowed by this specification and the LRS SHOULD* reject Statements containing such additional properties	From Section 4.2.1 An LRS shall reject a Statement with additional properties other than extensions in the locations where extensions are allowed
s4	From Section 2.4 Additional properties not listed here SHOULD* NOT be added to this object	Section N/A *Removed as requirement is handled via other requirements (SHALLs)*

	and each property MUST occur only once. 3.14	

LRS **MINIMAL IMPACT**

LRS developers will need to update their LRS to reject Statements that contain properties not defined in the xAPI specification and not permitted as an extension.

Client (LRP|LRC) **MINIMAL IMPACT**

LRP/LRC developers will need to update their client software to ensure it does not add properties not defined in the xAPI specification except when adding them to valid extension locations.

s5 Including Display on Query with Format of ids

In xAPI Version 1.0.3, the specification included a recommendation that Statements should not include the `display` property when getting Statements with a Format of `ids`. IEEE 9274.1.1 updates this recommendation, instead mandating that Statements returned from a GET request with format of `ids` shall not include verb `display` property.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s5	From Section 2.4.3 When queried for Statements with a Format of ids, the LRS SHOULD* NOT include the `display` property	Section N/A *Removed as requirement is handled in the positive sense in other tables*

LRS **MINIMAL IMPACT**

LRS developers will need to update their LRS to only return the verb id when the requested format is `ids`.

Client (LRP|LRC) **MINIMAL IMPACT**

LRP/LRC developers will need to update their client software to not expect a display property when they request Statements and use the format of `ids`. If the software needs the display property, developers should request a format of `exact` or `canonical`.

s6 Canonical Display on Query with Format of canonical

In xAPI Version 1.0.3, the specification included a recommendation that an LRS, when queried for Statements, should return a canonical display for the verb. This requirement was removed from the specification as it is handled in other sections (see s8).

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s6	From Section 2.4.3 When queried for Statements with a Format of `canonical`, the LRS SHOULD* return a canonical Display for that Verb.	Section N/A *Removed as requirement is handled in the positive sense in other tables*

LRS	**NO IMPACT**

Client (LRP|LRC)	**NO IMPACT**

s7 Returning Canonical Language Map on Query with Format of canonical

In xAPI Version 1.0.3, the specification included a recommendation that the LRS should return a canonical version of a language map when `canonical` format is used to retrieve statements and the LRS maintains a canonical version. IEEE 9274.1.1 updates this recommendation by indicating that the LRS may maintain a canonical version of any language map and return this when the `canonical` format is used to retrieve Statements.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s7	From Section 2.1.3 If the LRS maintains a canonical version of a language map, it SHOULD* return this canonical language map when `canonical` format is used to retrieve Statements	From Section 4.1.6.1 The LRS may maintain a canonical version of any language map and return this when `canonical` format is used to retrieve Statements.

LRS	MINIMAL IMPACT

LRS developers may decide to maintain a canonical version of the verb display and must return this value when format `canonical` is requested.

Client (LRP|LRC)	NO IMPACT

LRP/LRC developers should be aware that the display value of the verb in the Statements may not be the same as was used when storing the Statement due to the LRS being permitted to maintain a canonical version of the verb display value.

s8 Return One Language within Each Language Map

In xAPI Version 1.0.3, the specification included a recommendation that LRSs should only return one language within each language map for which it returns a canonical map. IEEE mandates this behavior by mandating that the LRS shall return only one language within each language map for which it returns a canonical map.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s8	From Section 2.1.3 The LRS SHOULD* return only one language within each language map for which it returns a canonical map.	From Section 4.1.6.1 The LRS shall return only one language within each language map for which it returns a canonical map.

LRS	MINIMAL IMPACT

LRS developers will need to update their LRS to only return one language for each language map type property when the requested format is canonical.

Client (LRP|LRC)	MINIMAL IMPACT

LRP/LRC developers will need to update their client software to expect only a single language in each language map type property when Statements are requested in canonical format.

s9 Response Pattern Character Limits

In xAPI Version 1.0.3, the specification included a recommendation that LRS should not enforce character limits relating to response patterns. In IEEE 9274.1.1, the SHOULD* NOT language

regarding character limits on `cmi.interaction` response patterns was removed, meaning that there are no restrictions on the length of a response pattern.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s9	From Section 2.4.4.1 The LRS SHOULD* NOT enforce character limits relating to response patterns.	Section N/A *Removed as a requirement*

LRS	MINIMAL IMPACT

If an LRS previously enforced character limits on response patterns, then LRS developers will need to update the implementation to allow any character length for `cmi.interaction` response patterns.

Client (LRP|LRC)	NO IMPACT

Although no change to client implementations is necessary, LRP/LRC developers may want to update their software if they were limiting the length of their response patterns.

s10 correctResponsesPattern Array Length Limit

In xAPI Version 1.0.3, the specification included a recommendation that the LRS should not limit the length of the `correctResponsesPattern` array for any `interactionType`. In IEEE 9274.1.1, the SHOULD* NOT language regarding array length limits on `cmi.interaction` response patterns was removed, meaning that there are no restrictions on the length of a response pattern array.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s10	From Section 2.4.4.1 The LRS SHOULD* NOT limit the length of the `correctResponsesPattern` array for any `interactionType`	Section N/A *Removed as a requirement*

LRS	MINIMAL IMPACT

If an LRS previously enforced length limits on the `correctResponsesPattern` array, then LRS developers will need to update their LRS implementations if they had previously limited the length of the `correctResponsesPattern` array.

Client (LRP|LRC)	NO IMPACT

Although no change to client implementations is necessary, LRP/LRC developers may want to update their software if they were limiting the length of their `correctResponsesPattern` array.

s11 Setting Timestamp to Stored

In xAPI Version 1.0.3, the specification included a recommendation that the LRS should set the `timestamp` property to the same value as the `stored` property if the timestamp was not provided by the LRP. In IEEE 9274.1.1, the specification was updated to require that the `timestamp` property is set to the value of the `stored` property by the LRS if no `timestamp` property value was provided by the client.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s11	From Section 2.4.7 The `timestamp` property SHOULD* be set by the LRS to the value of the `stored` property if not provided.	From Section 4.2.4.2 The LRS shall set the `timestamp` property to the value of the `stored` property if not provided.

LRS	MINIMAL IMPACT

If the LRS did not set the `timestamp` property to the stored value if the timestamp was not provided by the LRP, then LRS developers will need to update their LRS implementation to set the timestamp to the value of the `stored` property.

Client (LRP|LRC)	NO IMPACT

s12 Timestamp with Greater Value than Current Time

In xAPI Version 1.0.3, the specification included a recommendation that the LRS should not

reject a future timestamp to prevent issues due to clock error. In IEEE 9274.1.1, the specification was updated to require that an LRS accept timestamp values greater than the current time.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s12	From Section 2.4.7 An LRS SHOULD* NOT reject a timestamp for having a greater value than the current time, to prevent issues due to clock errors.	From Section 4.2.4.1 An LRS shall not reject a timestamp for having a greater value than the current time, within an acceptable margin of error (intentionally not specified in this document)

LRS **MINIMAL IMPACT**

LRS developers will need to ensure that their implementation allows for timestamps that are in the future to an unspecified degree.

Client (LRP|LRC) **NO IMPACT**

s13, s14 JWS Compact Serialization to Create JSON Web Signature

In xAPI Version 1.0.3, the specification included a recommendation that the JWS Compact Serialization should be used to create the JSON web signature. In IEEE 9274.1.1, the specification was updated to require that JWS Compact Serialization be used when making JSON web signatures.

This format is what the xAPI specification used for its examples in previous versions, so this may not impact current implementations.

The change log records for this item are:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s13	From Section 2.6 JWS Compact Serialization SHOULD* be used to create the JSON web signature.	From Section 4.2.6 JWS Compact Serialization shall be used to create the JSON web signature. Use of

	Use of JWS JSON Serialization is strongly discouraged, is unlikely to be interoperable with other systems, and will be forbidden in a future version of this specification.	JWS JSON Serialization is strongly discouraged, is unlikely to be interoperable with other systems, and will be forbidden in a future version of this specification.
s14	From Section 2.6 JWS Compact Serialization SHOULD* be used to create the JSON web signature. Use of JWS JSON Serialization is strongly discouraged, is unlikely to be interoperable with other systems, and will be forbidden in a future version of this specification.	From Section 5.2.6 JWS Compact Serialization shall be used to create the JSON web signature. Use of JWS JSON Serialization is strongly discouraged, is unlikely to be interoperable with other systems, and will be forbidden in a future version of this specification.

LRS **MINIMAL IMPACT**

If not currently supported, LRS developers will need to update their LRS to only use JWS Compact Serialization when handling JSON web signatures.

Client (LRP|LRC) **MINIMAL IMPACT**

LRP/LRC developers may need to update their client software to use JWS Compact Serialization when using JSON web signatures.

s15 Using IRIs a Metadata Provider Controls

In xAPI Version 1.0.3, the specification included a recommendation that Metadata Providers defining new IRIs should only use IRIs they control or have permission from the controller to use. In IEEE 9274.1.1, the specification was updated to require that Learning Record Providers defining new IRIs should only use IRIs they control or have permission from the controller to use.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s15	From Section 3.1 Metadata Providers defining new IRIs SHOULD* only use IRIs they control or	From Section 5.2.7 Learning Record Providers defining new IRIs should only use IRIs they control or

	have permission from the controller to use.	have permission from the controller to use.

LRS	NO IMPACT

Client (LRP|LRC)	MINIMAL IMPACT

If LRP/LRC developers do not own or have permission to use IRIs, then LRP/LRC developers will need to update their client software to use IRIs that they own or have permission to use.

s16 Re-using Identifiers

In xAPI Version 1.0.3, the specification included a recommendation that Metadata Providers SHOULD* ensure that the exact character equivalent IRI is used. In IEEE 9274.1.1, the specification was updated to remove the requirement as it is unclear and unnecessary.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s16	From Section 3.1 When re-using an existing identifier, Metadata Providers SHOULD* ensure that the exact character equivalent IRI is used.	Section N/A *Unclear and unnecessary. Requirement removed.*

LRS	NO IMPACT

If any checks were made on IRIs to ensure they were exact matches to existing IRIs, these will need to be removed by LRS developers.

Client (LRP|LRC)	NO IMPACT

s17, s18 IRI Simple String Comparison

In xAPI Version 1.0.3, the specification included a recommendation for an LRS regarding the comparison of IRIs. In IEEE 9274.1.1, the IRI comparison requirement was updated with clearer language.

The change log records for this item are:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s17	From Section 3.1 When storing or comparing IRIs, LRSs SHOULD* handle them only by using one or more of the approaches described in 5.3.1 (Simple String Comparison) and 5.3.2 (Syntax-Based Normalization) of RFC 3987, and SHOULD* NOT handle them using any approaches described in 5.3.3 (Scheme-Based Normalization) or 5.3.4 (Protocol-Based Normalization) of the same RFC, or any other approaches.	From Section 4.2.7 When storing or comparing IRIs, LRSs shall handle them only by using one or more of the approaches described in 5.3.1 (Simple String Comparison) and 5.3.2 (Syntax-Based Normalization) of RFC 3987.
s18	From Section 3.1 LRSs SHOULD* apply the same IRI comparison and normalization rules with all IRIs in parameters and fields defined to contain IRIs.	Section N/A *Handled by change s17*

LRS	**MINIMAL IMPACT**

The requirements for IRI comparison now require LRSs to implement specific comparison algorithms defined in RFC 3987. If not currently implemented, LRS developers will need to update their LRS to follow the RFC.

Client (LRP|LRC)	**NO IMPACT**

s19 Timestamp in RFC 3339

In xAPI Version 1.0.3, the specification included a recommendation that a timestamp should be expressed using the format described in RFC 3339. In IEEE 9274.1.1, the specification was updated to require RFC 3339 format.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text

s19	From Section 4.5 A Timestamp SHOULD* be expressed using the format described in RFC 3339, which is a profile of ISO 8601	From Section 5.2 A Timestamp shall be expressed using the format described in RFC 3339, which is a profile of ISO 8601.

LRS	**MINIMAL IMPACT**

If not currently implemented, LRS developers will need to update their LRS to validate the format defined in RFC 3339.

Client (LRP|LRC)	**MINIMAL IMPACT**

If not currently implemented, LRP/LRC developers will need to update their client software to follow the rules of RFC 3339.

s20 Timestamp Including Timezone

In xAPI Version 1.0.3, the specification included a recommendation that timestamps should include the time zone. In IEEE 9274.1.1, the specification was updated to require timestamps to be formatted in Coordinated Universal Time (UTC) time, instead of including a time zone.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s20	From Section 4.5 A Timestamp SHOULD* include the time zone.	From Section 5.2.7 A Timestamp *shall* be formatted to UTC.

LRS	**MINIMAL IMPACT**

If not currently implemented, LRS developers will need their LRS to convert timestamps to UTC.

Client (LRP|LRC)	**NO IMPACT**

LRP/LRC developers need to be aware that LRSs are required to convert timestamps to UTC time, despite the value the client sets.

s21 Returning Timestamp with Different Time Zone

In xAPI Version 1.0.3, the specification allowed an LRS to return a timestamp using a different time zone than the one originally provided in the statement. In IEEE 9274.1.1, the standard was updated to remove this statement.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s21	From Section 4.5 If the Timestamp includes a time zone, the LRS MAY return the Timestamp using a different time zone to the one originally used in the Statement so long as the point in time referenced is not affected.	Section N/A *Removed*

LRS	**MINIMAL IMPACT**

LRS developers will need to update their LRS if they are using time zone offsets.

Client (LRP|LRC)	**NO IMPACT**

LRP/LRC developers need to be aware that LRSs are required to convert timestamps to UTC time and that returned timestamps will not include time zone offsets.

s22 Timestamp in UTC

In xAPI Version 1.0.3, the specification included a recommendation that LRSs should return the timestamp in the UTC time zone. In IEEE 9274.1.1, the specification was updated such that an LRS shall convert Timestamps to UTC rather than rejecting Statements that send Timestamps not in UTC form.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s22	From Section 4.5 The LRS SHOULD* return the Timestamp in UTC time zone.	From Section 4.2.7 An LRS shall convert Timestamps to UTC rather than rejecting Statements that send Timestamps not in UTC form

LRS	MINIMAL IMPACT

If not currently implemented, LRS developers will need to update their LRS to both accept timestamps in formats other than UTC and to convert those to UTC.

Client (LRP|LRC)	NO IMPACT

LRP/LRC developers need to be aware that LRSs are required to convert timestamps to UTC time and that returned timestamps will not include time zone offsets.

s23 Truncating Durations

In xAPI Version 1.0.3, the specification included a recommendation that LRSs should not reject a request with durations of more than 0.01 second precision. In IEEE 9274.1.1, the specification was updated such that the LRS shall not reject the request but may truncate the `duration` property to 0.01 second precision when receiving a duration with more than 0.01 second precision.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s23	From Section 4.6 On receiving a Duration with more than 0.01 second precision, the LRS SHOULD* NOT reject the request but MAY truncate the `duration` property to 0.01 second precision	From Section 4.2.7 On receiving a Duration with more than 0.01 second precision, the LRS shall not reject the request but may truncate the `duration` property to 0.01 second precision.

LRS	MINIMAL IMPACT

If not currently implemented, LRS developers will need to update their LRS to accept durations with a greater than 0.01 second precision.

Client (LRP|LRC)	NO IMPACT

LRP/LRC developers need to be aware that LRSs are required to accept durations with greater than 0.01 second precision but they are permitted to truncate.

s24, s25 Duration Comparison Precision

In xAPI Version 1.0.3, the specification included a recommendation that precision beyond 0.01 second should not be included in the comparison. In IEEE 9274.1.1, the specification was updated to make this a requirement.

The change log records for this item are:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s24	From Section 4.6 When comparing Durations, any precision beyond 0.01 second precision SHOULD* NOT be included in the comparison	From Section 5.2.7 When comparing Durations (or Statements containing them), any precision beyond 0.01 second precision shall not be included in the comparison
s25	From Section 4.6 When comparing Durations, any precision beyond 0.01 second precision SHOULD* NOT be included in the comparison	From Section 4.2.7 When comparing Durations (or Statements containing them), any precision beyond 0.01 second precision shall not be included in the comparison.

LRS **MINIMAL IMPACT**

If not currently implemented, LRS developers will need to update their LRS to make comparisons only at a precision of 0.01.

Client (LRP|LRC) **MINIMAL IMPACT**

LRP/LRC developers may need to update their client software to make comparisons only at a precision of 0.01.

s26-s33 Alternate Request Syntax Removal

In xAPI Version 1.0.3, the specification included information on an alternate request syntax to handle some legacy browsers. In IEEE 9274.1.1, the Alternate Request Syntax section has been removed. Specifically noted in the xAPI spec, IE 8 and IE 9 used a cross domain request API that did not fully support the xAPI endpoints. However, since the writing of xAPI version 1.0.3, support for IE 8 and IE 9 has ended and modern browsers all support the basic HTTP requests required to use the xAPI endpoints.

The change log records for this item are:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s26	From Section 1.3 The Learning Record Provider SHOULD* still include a `Content-Type` header (in the HTTP header) for this type of request with a value of `application/x-www-form-urlencoded`	From Section 5.1.1 *Clarified in new Headers section*
s27	From Section 1.3 The Learning Record Provider SHOULD* still include a `Content-Type` header (in the HTTP header) for this type of request with a value of `application/x-www-form-urlencoded`	From Section 4.1.1 *Clarified in new Headers section*
s28	From Section 1.3 The `Content-Type` form parameter SHOULD* specify the content type of the content within the content form parameter	From Section 5.1.1 *Clarified in new Headers section*
s29	From Section 1.3 The `Content-Type` form parameter SHOULD* specify the content type of the content within the content form parameter	From Section 4.1.1 *Clarified in new Headers section*
s30	From Section 1.3 The Learning Record Provider SHOULD* still include a `Content-Length` header (in the HTTP header) for this type of request indicating the overall length of the request's content	From Section 5.1.1 *Clarified in new Headers section*
s31	From Section 1.3 The Learning Record Provider SHOULD* still include a `Content-Length` header	From Section 4.1.1 *Clarified in new Headers section*

	(in the HTTP header) for this type of request indicating the overall length of the request's content	
s32	From Section 1.3 The `Content-Length` form parameter SHOULD* specify the length of the content within the content form parameter and will therefore be a lower figure than the length listed in the	From Section 5.1.1 *Clarified in new Headers section*
s33	From Section 1.3 The `Content-Length` form parameter SHOULD* specify the length of the content within the content form parameter and will therefore be a lower figure than the length listed in the `Content-Length` header	From Section 4.1.1 *Clarified in new Headers section*

LRS	**MINIMAL IMPACT**

LRS developers will need to remove support for the alternate request syntax in their LRS.

Client (LRP|LRC)	**MAJOR IMPACT**

LRP/LRC developers will not need to make any changes if they already use the regular HTTP request syntax. However, developers who are supporting older browsers will either need to update their software or continue to use xAPI version 1.0.3 for the alternate syntax format.

s34 Accepting Batches with Document Type multipart/mixed

In xAPI Version 1.0.3, the specification included a recommendation that LRS accept batches of Statements with no Attachment Objects when receiving a PUT or POST with a document type of `multipart/mixed`. In IEEE 9274.1.1 the specification was updated to mandate an LRS accept batches of Statements which contain no Attachment Objects when receiving a PUT or POST with a document type of `multipart/mixed`.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s34	From Section 1.5.2 When receiving a PUT or POST with a document type of `multipart/mixed`, an LRS SHOULD* accept batches of Statements which contain no Attachment Objects	From Section 4.1.3 When receiving a PUT or POST with a document type of `multipart/mixed`, an LRS shall accept batches of Statements which contain no Attachment Objects.

LRS	MINIMAL IMPACT

If not currently implemented, LRS developers will need to update their LRS to accept `multipart/mixed` content types without attachments.

Client (LRP|LRC)	NO IMPACT

s35 Accepting Batches with Attachment Objects with fileUrl

In xAPI Version 1.0.3, the specification included a recommendation that an LRS accept batches of statements which contain only Attachment Objects with a populated `fileUrl` when receiving a PUT or POST with a document type of `multipart/mixed`. In IEEE 9274.1.1, the specification was updated to mandate an LRS accept batches of Statements which contain only Attachment Objects with a populated `fileUrl` when receiving a PUT or POST with a document type of `multipart/mixed`.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s35	From Section 1.5.2 When receiving a PUT or POST with a document type of `multipart/mixed`, an LRS SHOULD* accept batches of Statements which contain only Attachment Objects with a populated `fileUrl`	From Section 4.1.3 When receiving a PUT or POST with a document type of `multipart/mixed`, an LRS shall accept batches of Statements which contain only Attachment Objects with a populated `fileUrl`.

LRS	MINIMAL IMPACT

If not currently implemented, LRS developers will need to update their LRS to accept `multipart/mixed` type batches of Statements which contain only Attachment Objects with a populated `fileUrl`.

Client (LRP|LRC)	NO IMPACT

s36, s37 Rejecting Statement Batch with ID Collision

In xAPI Version 1.0.3, the specification included recommendations for an LRS when handling ID collision in batches of statements. In IEEE 9274.1.1, the specification was updated to require the LRS to reject the batch and return `400 Bad Request` when it receives a batch of Statements containing two or more Statements with the same id.

The change log records for this item are:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s36	From Section 2.1.1 If the LRS receives a batch of Statements containing two or more Statements with the same id, it SHOULD* reject the batch and return `400 Bad Request`	Section N/A
s37	From Section 2.1.2 If the LRS receives a batch of Statements containing two or more Statements with the same id, it SHOULD* reject the batch and return `400 Bad Request`	From Section 4.1.6.1 If the LRS receives a batch of Statements containing two or more Statements with the same id, it shall reject the batch and return `400 Bad Request`.

LRS	MINIMAL IMPACT

If not currently implemented, LRS developers will need to update their LRS to look for statements in a batch with the same id and reject the batch if duplicate ids are found.

Client (LRP|LRC)	MINIMAL IMPACT

If not currently implemented, LRP/LRC developers will need to update their client software to ensure that statements in a batch do not include duplicate ids.

s38 Last-Modified Matching Stored

In xAPI Version 1.0.3, the specification included a recommendation that LRSs include a `Last-Modified` header which matches the `stored` Timestamp of the Statement. In IEEE 9274.1.1, the specification was updated to require the LRS to include, in the `Last-Modified` header, the most recent (maximum) `stored` property of any of the returned statement(s).

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s38	From Section 2.1.3 The LRS SHOULD* include a `Last-Modified` header which matches the `stored` Timestamp of the Statement	From Section 4.1.6.1 The LRS shall include, in the `Last-Modified` header, the most recent (maximum) `stored` property of any of the returned statement(s).

LRS **MINIMAL IMPACT**

If not currently implemented, LRS developers will need to update their LRS to set the `Last-Modified` header to the most recent `stored` property value of the returned statements.

Client (LRP|LRC) **NO IMPACT**

s39 Last-Modified Header on Single Activity State Document Return

In xAPI Version 1.0.3, the specification included a recommendation that an LRS include a `Last-Modified` header indicating when the document was last modified when returning a single document. In IEEE 9274.1.1, the specification was updated to require the LRS to include a `Last-Modified` header indicating when the Activity State document was last modified.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s39	From Section 2.2 When returning a single document, the LRS SHOULD* include a `Last-`	From Section 4.1.6.2 The LRS shall include a `Last-Modified` header indicating when the document was

	`Modified` header indicating when the document was last modified	last modified.

LRS	MINIMAL IMPACT

If not currently implemented, LRS developers will need to update their LRS to set the `Last-Modified` header to the date when the Activity State was last modified.

Client (LRP|LRC)	NO IMPACT

s40 Last-Modified Header on Single Agent Profile Document Return

In xAPI Version 1.0.3, the specification included a recommendation that an LRS include a `Last-Modified` header indicating when the document was last modified when returning a single document. In IEEE 9274.1.1, the specification was updated to mandate the LRS to include a `Last-Modified` header indicating when the Agent Profile document was last modified.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s40	From Section 2.2 When returning a single document, the LRS SHOULD* include a `Last-Modified` header indicating when the document was last modified	From Section 4.1.6.5 The LRS shall include a `Last-Modified` header indicating when the document was last modified.

LRS	MINIMAL IMPACT

If not currently implemented, LRS developers will need to update their LRS to set the `Last-Modified` header to the date when the Agent Profile was last modified.

Client (LRP|LRC)	NO IMPACT

s41 Last-Modified Header on Single Activity Profile Document Return

In xAPI Version 1.0.3, the specification included a recommendation that the LRS include a `Last-Modified` header indicating when the document was last modified when returning a single document. In IEEE 9274.1.1, the specification was updated to mandate the LRS include a `Last-Modified` header indicating when the Activity Profile document was last modified.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s41	From Section 2.2 When returning a single document, the LRS SHOULD* include a `Last-Modified` header indicating when the document was last modified	From Section 4.1.6.6 The LRS shall include a `Last-Modified` header indicating when the document was last modified.

LRS	MINIMAL IMPACT

If not currently implemented, LRS developers will need to update their LRS to set the `Last-Modified` header to the date when the Activity Profile was last modified.

Client (LRP|LRC)	NO IMPACT

s42 Last-Modified on multiple document return

In xAPI Version 1.0.3, the specification included a recommendation that the LRS include a `Last-Modified` header indicating when the most recently modified document was last modified when returning multiple documents. In IEEE 9274.1.1, the requirement was removed.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s42	From Section 2.2 When returning multiple documents, the LRS SHOULD* include a `Last-Modified` header indicating when the	Section N/A *Removed. Handled by other requirements*

	most recently modified document was last modified	

LRS	**NO IMPACT**

Client (LRP|LRC)	**NO IMPACT**

s43 Return an Activity Object when unknown

In xAPI Version 1.0.3, the specification included a recommendation that the LRS return an Activity Object when queried, even if the LRS does not have a canonical definition of the Activity. In IEEE 9274.1.1, the specification was updated to mandate that If an LRS does not have a canonical definition of the Activity to return, the LRS shall still return an Activity Object when queried.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s43	From Section 2.5 If an LRS does not have a canonical definition of the Activity to return, the LRS SHOULD* still return an Activity Object when queried	From Section 4.1.6.4 If an LRS does not have a canonical definition of the Activity to return, the LRS shall still return an Activity Object when queried

LRS	**MINIMAL IMPACT**

If not currently implemented, LRS developers will need to update their LRS to return an Activity Object with at least the id used in the request.

Client (LRP|LRC)	**MINIMAL IMPACT**

LRP/LRC developers need to be aware that LRSs will return an activity object to all activity endpoint requests.

s44-s46 Concurrency Header Use on PUT and POST Document Endpoints

In xAPI Version 1.0.3, the specification included recommendations for concurrency headers on PUT and POST requests to document endpoints. In IEEE 9274.1.1, the specification was updated to mandate LRP include the appropriate header.

The change log records for this item are:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s44	From Section 3.1 A Client making a POST request to either the Agent Profile Resource or Activity Profile Resource SHOULD* include the `If-Match` header or the `If-None-Match` header	From Section 5.1.4 An LRP making a POST request to either the Agent Profile Resource or Activity Profile Resource shall include the `If-Match` header or the `If-None-Match` header.
s45	From Section 3.1 A Client making a POST request to either the Agent Profile Resource or Activity Profile Resource SHOULD* include the `If-Match` header or the `If-None-Match` header	From Section 5.1.6.5 An LRP making a POST request to this resource shall include the `If-Match` header or the `If-None-Match` header.
s46	From Section 3.1 A Client making a POST request to either the Agent Profile Resource or Activity Profile Resource SHOULD* include the `If-Match` header or the `If-None-Match` header	From Section 5.1.6.6 An LRP making a POST request to this resource shall include the `If-Match` header or the `If-None-Match` header.

LRS	**NO IMPACT**

Client (LRP|LRC)	**MINIMAL IMPACT**

If not currently implemented, LRP/LRC developers will need to update their software to send currency headers with each document endpoint POST or PUT request.

s47-s49 Concurrency Header Use on DELETE Document Endpoints

In xAPI Version 1.0.3, the specification included recommendations on currency headers for DELETE requests on document endpoints. In IEEE 9274.1.1, the specification was updated such that an LRP making a DELETE request to the State Resource, Agent Profile Resource or Activity Profile Resource shall include the `If-Match` header.

The change log records for this item are:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s47	From Section 3.1 A Client making a DELETE request to either the Agent Profile Resource or Activity Profile Resource SHOULD* include the `If-Match` header	From Section 5.1.4 An LRP making a DELETE request to either the Agent Profile Resource or Activity Profile Resource shall include the `If-Match` header.
s48	From Section 3.1 A Client making a DELETE request to either the Agent Profile Resource or Activity Profile Resource SHOULD* include the `If-Match` header	From Section 5.1.6.5 An LRP making a DELETE request to this resource SHALL include the `If-Match` header.
s49	From Section 3.1 A Client making a DELETE request to either the Agent Profile Resource or Activity Profile Resource SHOULD* include the `If-Match` header	From Section 5.1.6.6 An LRP making a DELETE request to this resource SHALL include the `If-Match` header.

LRS **NO IMPACT**

Client (LRP|LRC) **MINIMAL IMPACT**

If not currently implemented, LRP/LRC developers will need to update their software to send the `If-Match` header with each document endpoint DELETE request.

s50, s51 LRS ETag If-Match / If-None-Match header

In xAPI Version 1.0.3, the specification included recommendations regarding ETags. In IEEE 9274.1.1, the specification was updated such that an LRS responding to a PUT, POST, or DELETE request shall handle the `If-Match` header as described in RFC2616, HTTP 1.1 if it contains an ETag, in order to detect modifications made after the document was last fetched.

The change log records for this item are:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s50	From Section 3.1 An LRS responding to a POST or DELETE request SHOULD* handle the `If-Match` header as described in RFC2616, HTTP 1.1 if it contains an ETag, in order to detect modifications made after the Client last fetched the document	From Section 4.1.4 An LRS responding to a PUT, POST, or DELETE request shall handle the `If-Match` header as described in RFC2616, HTTP 1.1 if it contains an ETag, in order to detect modifications made after the document was last fetched.
s51	From Section 3.1 An LRS responding to a POST request SHOULD* handle the `If-None-Match` header as described in RFC2616, HTTP 1.1 if it contains "`*`", in order to detect when there is a resource present that the Client is not aware of.	Section N/A *Removed. Clarified in s-49*

LRS **MINIMAL IMPACT**

If not currently implemented, LRS developers will need to update their LRS to verify the concurrency header value matches the target resource.

Client (LRP|LRC) **NO IMPACT**

s52 Header Precondition Failed Response

In xAPI Version 1.0.3, the specification included a recommendation if any POST or DELETE header preconditions fail. In IEEE 9274.1.1, the specification was updated such that If the header precondition in either of the request cases above fails, the LRS:

- shall return HTTP status `412 Precondition Failed`.
- shall not make a modification to the resource.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s52	From Section 3.1	From Section 4.1.4

	If the header precondition in any of the POST or DELETE request cases above fails, the LRS: • SHOULD* return HTTP status `412 Precondition Failed`. • SHOULD* NOT make a modification to the resource.	*Note: Combined with PUT section as they are now identical* If the header precondition in either of the request cases above fails, the LRS: • shall return HTTP status `412 Precondition Failed`. • shall not make a modification to the resource.

LRS	**MINIMAL IMPACT**

If not currently implemented, LRS developers will need to update their LRS to respond with `412 Precondition` Failed if the value in the `If-Match` or `If-None-Match` header does not match the one associated with the target resource.

Client (LRP|LRC)	**MINIMAL IMPACT**

LRP/LRC developers need to be aware that LRSs will return `412 Precondition Failed` if the value in the `If-Match` or `If-None-Match` header does not match the one associated with the target resource.

s53 Client use of ETag

In xAPI Version 1.0.3, the specification included a recommendation that clients use the ETag value provided by the LRS rather than calculating it themselves. In IEEE 9274.1.1, the specification was updated such that an LRP shall use the ETag value provided by the LRS rather than calculating it themselves.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s53	From Section 3.1 Clients SHOULD* use the ETag value provided by the LRS rather than calculating it themselves.	From Section 5.1.4 An LRP shall use the ETag value provided by the LRS rather than calculating it themselves.

LRS	NO IMPACT

Client (LRP|LRC)	MINIMAL IMPACT

If not currently implemented, LRP/LRC developers will need to update their software to send the ETag value provided by the LRS instead of calculating the value.

s54 Removal of ETag Calculation Algorithm

In xAPI Version 1.0.3, the specification included a recommendation on an ETag calculation algorithm. In IEEE 9274.1.1, this recommendation was removed from the specification. Content shall use the ETag value provided by the LRS.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s54	From Section 3.1 An LRS responding to a GET request without using a transfer encoding or using the identity transfer encoding MUST calculate the value of the ETag header to be a hexadecimal string of the SHA-1 digest of the contents. This hexadecimal string SHOULD be rendered using numbers and lowercase characters only; uppercase characters SHOULD NOT be used. The requirement to calculate the ETag this way will be removed in a future version of the specification.	Section N/A *Removed*

LRS	NO IMPACT

Client (LRP|LRC)	MINIMAL IMPACT

If not currently implemented, LRP/LRC developers will need to update their software to send the ETag value provided by the LRS instead of calculating the value.

s55 Removal of Content Type Requirement

In xAPI Version 1.0.3, the specification included a recommendation on the content type and the rules for a `400 Bad Request` error status. In IEEE 9274.1.1, the specification was updated to remove these requirements. These requirements were either specific to the Alternate Request Syntax, which was removed, or about content types in general, which is covered in other parts of the specification.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s55	From Section 3.2 The LRS SHOULD* reject any request with `400 Bad Request` status where the content type header does not match the content included in the request or where the structure of the request does not match the structure outlined in this specification for a particular content type. For example, if the content of the request is formatted as JSON, the content type is expected to be `application/json`. If the content type is `application/x-www-form-urlencoded` it is expected that the request will include a method parameter as outlined in Alternate Request Syntax.	Section N/A *Removed*

LRS	**NO IMPACT**
Client (LRP|LRC)	**NO IMPACT**

s56 LRS Configuration for Test Suite

In xAPI Version 1.0.3, the specification included recommendations for the purposes of conformant testing. In IEEE 9274.1.1, requirements regarding configurability of the LRS have been removed and consolidated into section 4.1.9 Security.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s56	From Section 3.2 The following requirements exist for the purposes of conformance testing, to ensure that any limitations or permissions implemented by the LRS do not affect the running of conformance testing software. ● The LRS SHOULD* be configurable not to reject any requests from a particular set of credentials on the basis of permissions. This set of credentials SHOULD* be used for conformance testing but MAY be deleted/deactivated on live systems. ● The LRS MUST be configurable to accept Attachments, Statements or documents of any reasonable size (see above). ● The LRS MUST be configurable to accept requests at any reasonable rate.	From Section 4.1.5 *Appropriate handling of these requirements is now included in the Error Codes section of the new standard*

LRS	**NO IMPACT**

The LRS is still required to be configurable to support the Conformance Test Suite.

Client (LRP|LRC)	**NO IMPACT**

s57 Header Value for Basic Authentication

In xAPI Version 1.0.3, the specification included a recommendation on the header value for basic authentication. In IEEE 9274.1.1, details about authentication have been removed and are under consideration at the xAPI cybersecurity working group.

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s57	From Section 4.1 Requests SHOULD* include headers for HTTP Basic Authentication based on a username and password each consisting of an empty string. In this case the HTTP Basic Authentication header will be `Basic` followed by a base64 encoded version of the string :. This results in the string `Basic Og==`.	Section N/A *Removed and under consideration in the cybersecurity xAPI sub-group*

LRS	**NO IMPACT**

LRSs can continue to use current authentication methods.

Client (LRP|LRC)	**NO IMPACT**

s58 Voiding a Missing Statement

In xAPI Version 1.0.3, the specification included a recommendation that the LRS SHOULD NOT* reject the request on the grounds of the Object of that voiding Statement not being present when receiving a Statement that voids another. In IEEE 9271.1.1, the specification was updated such that the LRS shall not reject a Statement that uses the voided verb if it cannot find the id of the Object of that Statement (nor does the LRS have to try to find it)

The change log record for this item is:

ID	xAPI Version 1.0.3 Text	IEEE 9274.1.1 Updated Text
s58	From Section 2.3.2 Upon receiving a Statement that voids another, the LRS SHOULD NOT* reject the request on the grounds of the Object of that voiding Statement not being present.	From Section 4.2.4.1 The LRS shall not reject a Statement that uses the voided verb if it cannot find the id of the Object of that Statement (nor does the LRS have to try to find it)

LRS	**MINIMAL IMPACT**

If not currently implemented, LRS developers will need to update their implementation to not reject statements that void a statement that does not exist on their LRS.

Client (LRP|LRC)	**NO IMPACT**

3.3. Cybersecurity Updates

In xAPI Version 1.0.3, the specification included incomplete information on xAPI cybersecurity. IEEE 9274.1.1 specification does not include information about authentication and security. Information in the original specification was meant to serve as an example and not the only way to handle security and authentication. As with most web services, security and authentication measures can differ based on the use case. Since there is no one size fits all solution, the information in the original specification was deemed confusing by implementers. The IEEE xAPI cybersecurity working group is drafting a separate recommended practice covering these topics.

The change log record for this item is:

ID	xAPI Version 1.0.3 Information	IEEE 9274.1.1 Updated Information
c1	Specification contained information on authentication including OAuth 1.0 and HTTP Basic.	Standard removed information about authentication and security. Information in the original specification was meant to serve as an example and not the only way to handle security and authentication. As with most web services, security and authentication measures can differ based on the use case. Since there is no one size fits all solution, the information in the original specification was deemed confusing by implementers. The IEEE xAPI sub-group on cybersecurity is currently working to publish a separate guide covering these topics.

3.4. Context Agents and Content Groups Updates

x1 Context Agents

IEEE 9274.1.1 includes a new context array for `contextAgents` (in section 4.2.2.5 and section 5.2.2.5). This is an array of `contextAgent` objects including an `objectType`, `agent` and `relevantTypes` (array of `type` IRIs). Although `context.instructor` from version 1.0.3 is still present, it is considered deprecated. Implementers should make use of a `contextAgent` with an instructor `relevantType` IRI for this purpose. The intent of `contextAgent` is to allow additional actors to be included as context of a statement beyond just the instructor allowed in version 1.0.3

The change log record for this item is:

ID	xAPI Version 1.0.3 Information	IEEE 9274.1.1 Updated Information
x1	The Statement data structure includes a property for the instructor (`context.instructor` : `Agent`) in section 2.4.6.	The IEEE standard includes a new context array for `contextAgents` (in section 4.2.2.5 and section 5.2.2.5). This is an array of `contextAgent` objects including an `objectType`, agent and `relevantTypes` (array of `type` IRIs). Although `context.instructor` from version 1.0.3 is still present, it is considered deprecated. Implementers should make use of a `contextAgent` with an instructor `relevantType` IRI for this purpose. The intent of `contextAgent` is to allow additional actors to be included as context of a statement beyond just the instructor allowed in version 1.0.3

LRS	**MAJOR IMPACT**

LRS developers will need to implement all of the Context Agent requirements from the new specification.

Client (LRP|LRC)	**NO IMPACT**

x2 Context Groups

IEEE 9274.1.1 includes a new context array for `contextGroups` (in section 4.2.2.5 and section 5.2.2.5). This is an array of `contextGroup` objects including an `objectType`, `Group`, and `relevantTypes` (array of `type` IRIs). Although `context.team` from version 1.0.3 is still present, it is considered deprecated. Implementers should make use of a `contextGroup` with a team `relevantType` IRI for this purpose. The intent of `contextGroups` is to allow additional groups to be included as context of a statement beyond just the team allowed in version 1.0.3

The change log record for this item is:

ID	xAPI Version 1.0.3 Information	IEEE 9274.1.1 Updated Information
x2	The Statement data structure includes a property for team (`context.team` : `Group`) in section 2.4.6.	The IEEE standard includes a new context array for `contextGroups` (in section 4.2.2.5 and section 5.2.2.5). This is an array of `contextGroup` objects including an `objectType`, `Group`, and `relevantTypes` (array of `type` IRIs). Although `context.team` from version 1.0.3 is still present, it is considered deprecated. Implementers should make use of a `contextGroup` with a team `relevantType` IRI for this purpose. The intent of `contextGroups` is to allow additional groups to be included as context of a statement beyond just the team allowed in version 1.0.3

LRS	**MAJOR IMPACT**

LRS developers will need to implement all of the Context Groups requirements from the new specification.

Client (LRP|LRC)	**NO IMPACT**

APPENDIX A-ALL MAJOR TECHNICAL IMPACTS

LRS Major Technical Impact Updates	Client Major Technical Impact Updates
• x1 Context Agents • x2 Context Groups	• s26-s33 Alternate Request Syntax Removal

APPENDIX B-ALL MINIMAL TECHNICAL IMPACTS

LRS Minimal Technical Impact Updates	Client Minimal Technical Impact Updates
• g4 Version update to 2.0.0 • s1-s4 Additional Properties • s5 Including Display on Query with Format of ids • s7 Returning Canonical Language Map on Query with Format of canonical • s8 Return One Language within Each Language Map • s9 Response Pattern Character Limits • s10 correctResponsesPattern Array Length Limit • s11 Setting Timestamp to Stored • s12 Timestamp with Greater Value than Current Time • s13, s14 JWS Compact Serialization to Create JSON Web Signature • s17, s18 IRI Simple String Comparison • s19 Timestamp in RFC 3339 • s20 Timestamp Including Timezone • s21 Returning Timestamp with Different Timezone • s22 Timestamp in UTC • s23 Truncating Durations	• g4 Version update to 2.0.0 • s1-s4 Additional Properties • s5 Including Display on Query with Format of ids • s8 Return One Language within Each Language Map • s13, s14 JWS Compact Serialization to Create JSON Web Signature • s15 Using IRIs a Metadata Provider Controls • s19 Timestamp in RFC 3339 • s24, s25 Duration Comparison Precision • s36, s37 Rejecting Statement Batch with ID Collision • s43 Return an Activity Object when unknown • s44-s46 Concurrency Header Use on PUT and POST Document Endpoints • s47-s49 Concurrency Header Use on DELETE Document Endpoints • s52 Header Precondition Failed Response • s53 Client use of ETag • s54 Removal of ETag Calculation Algorithm

LRS Minimal Technical Impact Updates (Cont'd)	
• s24, s25 Duration Comparison Precision • s26-s33 Alternate Request Syntax Removal • s34 Accepting Batches with Document Type multipart/mixed • s35 Accepting Batches with Attachment Objects with fileUrl • s36, s37 Rejecting Statement Batch with ID Collision • s38 Last-Modified Matching Stored • s39 Last-Modified Header on Single Activity State Document Return • s40 Last-Modified Header on Single Agent Profile Document Return • s41 Last-Modified Header on Single Activity Profile Document Return • s43 Return an Activity Object when unknown • s50, s51 LRS ETag If-Match / If-None-Match header • s52 Header Precondition Failed Response • s58 Voiding a Missing Statement	