

# DATASIM Testing and Evaluation Plan

Data and Training Analytics Simulated Input Modeler

27 May 2021

This work was supported by the U.S. Advanced Distributed Learning (ADL) Initiative (HQ003419C0061). 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. Government is authorized to reproduce and distribute reprints for Government purposes.



**Distribution Statement A**

Approved for public release: distribution unlimited.

# DATASIM Testing and Evaluation Plan

Data and Training Analytics Simulated Input Modeler

Prepared by Shelly Blake-Plock, PI, Yet Analytics, Inc.

Submitted to ADL on February 16, 2021

## 1. Purpose of this Document

The purpose of this document is to explore the methodology used to test the TRL5 Beta version of the Data and Training Analytics Simulated Input Modeler (DATASIM).

At this stage of the project, the goals of testing are to demonstrate that the prototype is capable of performing the stated goals of the project and that all the features are integrated and working together in a relevant real life or simulated environment. As a result of this testing, it can also be determined which areas need additional work to move the prototype to a full implementation.

<b>1. Purpose of this Document</b>	<b>1</b>
<b>2. About DATASIM</b>	<b>3</b>
<b>3. Testing Goals</b>	<b>3</b>
3.1 Alignment Behavior	3
3.1.1 Groups and Roles	3
3.1.2 Zero Statement Alignment	4
3.1.3 Object Override Alignment	4
3.2 Simulation Repeatability	4
3.3 Micro-Scale Fidelity	4
3.4 LRS Cross-compatibility	4
3.5 Simulation Performance	5
3.6 Visual Validation	5
3.6.1 Types of Errors	5
3.6.1.1 Parameter	5
3.6.1.2 Personae	6

3.6.1.3 xAPI Profile	6
3.6.1.4 Alignments	7
3.7 User Interface	7
<b>4. Testing Methodology</b>	<b>7</b>
4.1 Alignment Behavior	7
4.1.1 Groups and Roles Alignments	8
4.1.2 Zero Statement Alignment	8
4.1.3 Object Override Alignment	8
4.2 Simulation Repeatability	8
4.3 Profile Compatibility	9
4.4 Micro-Scale Fidelity	9
4.5 LRS Cross-compatibility	9
4.6 Simulation Performance	9
4.6.1 Simulation Throughput	10
4.6.2 Clustered Generation and Scaling Rate	10
4.7 Validation Testing	10
4.7.1 Parameters	10
4.7.2 xAPI Profiles	10
4.7.3 Personae	11
4.7.4 Alignments	12
4.8 User Interface	12
4.8.1 Parameters Page	12
4.8.2 Personae Page	13
4.8.3 Alignment Page	13
<b>5. Conclusion</b>	<b>14</b>
6.1 Yet Calibration Profile	15
6.2 Master Object Model Profile	15

## 2. About DATASIM

DATASIM is a software project capable of running simulations resulting in the generation of realistic xAPI data. The environment uses xAPI Profiles to model behavior for a cohort of simulated Actors. These datasets may be used to benchmark and stress-test components of the Total Learning Architecture (TLA) and other distributed learning projects.

Additionally, DATASIM helps learning scientists, engineers, ISDs, IT staff and decision-making stakeholders to determine the effectiveness of xAPI data design and implementation across the TLA.

DATASIM is open source under the Apache License Version 2 and is available on GitHub here:

Backend <<https://github.com/yetanalytics/datasim>>

Frontend <<https://github.com/yetanalytics/datasim-ui>>

## 3. Testing Goals

In this section, the aspects of the DATASIM Framework that will be tested as a part of this exercise are outlined. The conclusion explores what potential test results may mean about DATASIM capabilities, and what they may suggest concerning areas of further research and development.

### 3.1 Alignment Behavior

The Alignments are a way of influencing the behavior of a specific Actor within a simulation. The goal of this area of testing is to confirm that Alignments can be made for groups and roles as well as enforce situational Alignments to override the Object within a sequence of statements or create no statements in certain scenarios.

#### 3.1.1 Groups and Roles

Groups and roles can be allocated to specific Actors, so that the statements being generated for those Actors take into account the weight of the group or role they are assigned. The Alignment Specification allows groups and roles to affect the final weight of an individual Actor's Activities that will be associated. The success of this test will be gauged by the resulting statements accounting and averaging the weight of all of these Alignments. To appropriately test group and role Alignments, it must be a multi-level testing of each aspect individually with the Actor and together with varying weights for each.

### **3.1.2 Zero Statement Alignment**

The Zero Statement Alignment allows a parameter to be set to disallow certain Activities within the simulation for a given Actor, Group, or Role. The goal of this test is to verify no statements are generated when a Zero Statement Alignment is present. Setting one or multiple Alignments where the weight is exactly -1 will yield no statements for that Activity-Actor combination.

### **3.1.3 Object Override Alignment**

DATASIM will allow the user to enforce a particular Object to be used within a generated Statement. This test will be considered successful if the resulting statements only contain the Object described in the Object Override Alignment. An additional test will be conducted using an Actor as the Object to demonstrate statements can be overwritten to read '*Actor verb Actor*'. This will display the capability to not only override an Object, but to make that Object a Personae.

## **3.2 Simulation Repeatability**

In order to be useful in a research context it is a requirement that DATASIM always produce the same xAPI dataset when provided with the same inputs. The goal of this area of testing is to establish that the simulations are deterministic and will consistently produce the same results when given the same inputs, and different results when provided with slightly different inputs.

In this test simulation, the aspects of the simulation will remain the same for the control and then small changes will be made to the parameters between each control to ensure the consistency of the control simulation. Parameters will then also be reverted to the previous state to ensure they produce earlier datasets.

## **3.3 Micro-Scale Fidelity**

Micro-Scale fidelity measures the degree of exactness with which a small scale simulation is run. In a real classroom scenario, an Instructor would find value in seeing realistic data from a small group and time frame like a classroom setting. By testing the results of smaller data sets, someone creating learning content would be able to review the test simulation and review if the findings are realistic to a live classroom situation. This type of testing will be useful from the xAPI Profile designer perspective.

## **3.4 LRS Cross-compatibility**

In order to be useful as an aid in high volume load testing, and to be more broadly applicable to the xAPI community, DATASIM's output must be compatible with any ADL LRS Test Suite conformant LRS. The goal of this area of testing is to have DATASIM generate a stream of Statements to a number of LRS implementations and report on the results.

The test will consist of multiple major LRS providers in the industry and their ability to use datasets generated by DATASIM.

## **3.5 Simulation Performance**

DATASIM was designed to be efficient and have a throughput which allows for the generation of a large volume of Statements in a short time period. The goal of this area of testing is to establish the upper limits of throughput and total generation for this Beta implementation of DATASIM.

### **Clustered Generation**

The ability to create datasets through multiple coordinated systems has enabled the volume of data to grow with the needs of the user. The testing will establish whether multi-node generation is possible and what the scaling factor's effect on overall throughput could be when adding additional resources.

This research into scaling rate will test the output capacity at multiple levels of multi-node generation. This means running simulations with  $n$ ,  $n+1$ ,  $n+2$ ,  $n+i$  and tracking the curve as it increases to calculate the data production volume. This test will find what capacity DATASIM has to compute more quickly and at a high rate.

## **3.6 Visual Validation**

The user is notified when an error has occurred validating the specifications of the simulation. In order for the user to remediate each component, all of the errors appear in a text field at the top of the page. Each type of error will be rendered individually alongside other errors to ensure the correct simulation specification errors appear when a certain mistake is made.

The main reason for the visual validation feature is to allow the simulation configurations to be readable and allow a user to find the appropriate error in their inputs by the information provided in the validation messages.

### **3.6.1 Types of Errors**

#### **3.6.1.1 Parameter**

There are multiple errors that can cause a simulation not to execute. The Parameters of a simulation are now added through text fields and a clickable widget to add the dates. This greatly reduced the likelihood of user error, but the user still has the opportunity to input an incorrect value. This test will ensure that the parameter errors appear and have a clear path to fix the issue.

### 3.6.1.2 Personae

With the addition of groups and roles, there is a need to test different types of group hierarchy against multiple types of roles to test the possible errors that could prevent a simulation from running. The goal of this test is to validate the resulting statements are following the group and role hierarchy and that statements can be generated for groups themselves.

### 3.6.1.3 xAPI Profile

The xAPI Profile validation testing will show the violation type with a description and the location within xAPI Profile(s) for a variety of errors. There are too many error configurations to test all of the scenarios that could occur while building xAPI Profiles in DATASIM. This test will be done with the most common errors made when creating a xAPI Profile to ensure the location within the xAPI Profile and error message are correct.

#### Syntax

The xAPI Profile Syntax is outlined in the DATASIM documentation in GitHub. Testing the error validation for syntax will entail creating varying errors within the same simulation based on the specifications for each aspect of the xAPI Profile.

- Basic syntax Validation
  - JSON-LD<sup>1</sup>
  - Well Structured
- xAPI Profile Specification Validation
  - xAPI Profile metadata
  - Concept Specification<sup>2</sup>
  - Statement Templates<sup>3</sup>
  - Patterns<sup>4</sup>

#### Structure

The Visual Error Validation Panel appears at the top of the page in a plain text panel and displays the area in the xAPI Profile that needs attention, a description of the error, and will eventually offer suggestions to remediate the error. The user may need to scroll through the panel to see all of this information.

#### Specification

In the new error message user interface, the error messages will contain the type of error, where in the input the error occurred and how to correct the error(s).

---

<sup>1</sup> <https://github.com/adlnet/xapi-profiles/blob/master/xapi-profiles-structure.md#structure>

<sup>2</sup> <https://github.com/adlnet/xapi-profiles/blob/master/xapi-profiles-structure.md#concepts>

<sup>3</sup> <https://github.com/adlnet/xapi-profiles/blob/master/xapi-profiles-structure.md#statement-templates>

<sup>4</sup> <https://github.com/adlnet/xapi-profiles/blob/master/xapi-profiles-structure.md#patterns>

### **3.6.1.4 Alignments**

The goal of this test would be to create each Alignment error and ensure the text field displays the visual validation. Another test will remove a required Personae to ensure the existence of Alignment IRIs and Personae in associated inputs.

## **3.7 User Interface**

The User Interface testing will mainly consist of demonstrating each section of the simulation building is operating as expected and the graphical representation is accurate when viewed in the Advanced Mode (JSON).

### **Parameter Page**

The Parameter interface allows the user to add the dates, timezone, numerical Max Statement and numerical Seed value using text fields and selectable input. The goal of this test will be to create multiple simulations using the new graphical user interface and confirm the information has been updated in the Advanced Mode (JSON).

### **Personae Page**

The new user interface allows the user to easily add Actors into a Group and assign them to a Role. This test will set out to create multiple team hierarchy scenarios to show the viability of having flexible team structures within the simulation.

### **Alignment Page**

Alignments can now be added via a series of dropdown menus to select the Personae and Verb with a sliding bar to assign the weight between -1 and 1. In order to test this, multiple Alignments will be created in the new user interface using Actors, groups and roles as the Personae and varying weights. The goal will be to verify each of the Alignments accurately generated the correct amount of statements according to simulation specifications.

# **4. Testing Methodology**

## **4.1 Alignment Behavior**

In order to test the behavior of simulations with respect to Alignments, for these tests everything will be kept the same as in the previous experiment. In multiple simulation runs, we will examine the output of the relative number of dataset records corresponding to an Activity and Actor over a range of Alignment weight values. We will then plot the relationship to weight and aligned statement generation volume with the expectation that we see a direct relationship between weight and volume.

### 4.1.1 Groups and Roles Alignments

In this test, the user will associate the Actor 'Persona 1' with a role and place this Actor in a group. Alignments with identical Activities will be created for the Actor, the group, and the role. This test will show Alignments can be made for Actors, groups, and roles and those Alignments will result in the accurate mean weight of all the associated Alignments.

Actor 1 'Persona 1'	<a href="http://example.com/xapi/assessed">http://example.com/xapi/assessed</a>	.5
Role 1	<a href="http://example.com/xapi/assessed">http://example.com/xapi/assessed</a>	-.25
Group 1	<a href="http://example.com/xapi/assessed">http://example.com/xapi/assessed</a>	-.3

#### Desired Results

Persona 1 final 'assessed' weight  $(.5 + -.25 + -.3) / 3 = -.01666666$

### 4.1.2 Zero Statement Alignment

This must be tested in two ways: using multiple Alignment types individually in addition to simultaneous zero statement Alignments within one simulation. To test this, an Alignment will be created with a weight of -1 for a particular Activity. In total 10 simulations will be run, one Zero Statement Alignment simulation and then 9 simulations with multiple varying Zero Statement Alignments to ensure there are no statements generated for any Alignments weighted -1.

### 4.1.3 Object Override Alignment

In order to test this behavior, an Alignment will be written to override the Object to always be Persona 1 within a Statement Template. This will demonstrate that the statements for this Activity all have the same Object as specified in the Alignment.

#### Desired Results

*Resulting Statements: 'Persona 2' Evaluated 'Persona 1'*

## 4.2 Simulation Repeatability

In this test, the each xAPI Profiles resulting dataset from 100 simulations (50 each on 2 different hardware deployments) were stored. New xAPI Profile simulations were tested where the seed was changed for every set of simulations run. In additional testing, the configuration of the simulation will be altered using higher test sets, differing amounts of computing power and Parameter changes.

### 4.3 Profile Compatibility

The user will run a simulation using each of the xAPI Profiles below to test if they are able to run a simulation in DATASIM that conforms to the simulation specifications. Each xAPI Profile will be run using the same Parameter and Personae specifications for every test.

xAPI Profiles
cmi5
Yet Calibration Profile
TCCC
Master Object Model Profile
STEEL-R Profile

### 4.4 Micro-Scale Fidelity

To test the fidelity of a small batch of Actors in a short time span, the goal will be to simulate a 90 min activity in a group of 35 Personae. This simulation will be run 100 times with varying Alignments.

**This will demonstrate that a simulation setup with realistic classroom specifications will output realistic results.**

Instructors may see value in a single class scenario run through a normal class duration to identify if the resulting data is realistic. This would not predict the simulation Actor's behavior, but shows within the context of a discrete course if it aligns with the type of metrics that the Instructor needs. This would ultimately show which Activities are being engaged and which are not.

### 4.5 LRS Cross-compatibility

DATASIM will be tested with both Yet Analytics SaaS Pro and Cloud LRS as well as three industry LRS providers. These additional LRS systems will include Rustici SCORM Cloud LRS, Learning Pool (Learning Locker) and Watershed. The test will determine if none, some, or all statements are successfully posted to the respective LRS implementations.

### 4.6 Simulation Performance

For this area of testing, two types of testing will take place to validate distributed generation among multiple machines. The test will gradually increase in the number of machines from 2 to

10. This will demonstrate the rate at which the simulation is generating among differing hardware setups.

### **4.6.1 Simulation Throughput**

This test will indicate the average measured throughput of generation across 50 simulations each on this diagnostic hardware setup with Extensions and without Extensions. The results will reflect the statements per second rate for each of the hardware setups in this test.

- With Extension
- Without Extension

### **4.6.2 Clustered Generation and Scaling Rate**

In this test, 30 simulations will be run for 8 hour with multi-node generation with an increasing number of machines. The simulation will have the same parameters and will be used for each test simulation. This test will be run 3 times with 1, 2,..., 9 and 10 identical computing power machines. The desired result for this test will be to determine that distributed generation allows for more statements to be created in a short amount of time.

#### **Desired Result**

# statements can be generated in x hours with n nodes

## **4.7 Validation Testing**

### **4.7.1 Parameters**

In the new user interface, the user can toggle between JSON and a graphical user interface. This test will demonstrate the ability to create accurate errors when the user adds the Start and End Date set where the End Date is before the Start Date. In addition to testing the Start and End Date, a test will be run to create a formatting error with the date field. We would also set the Max Statement field to zero, to be sure the error occurs. The last Parameter test that will be run is an invalid Time Zone. This will be less likely to occur due to the graphical drop down interface, but could occur in the Advanced Mode.

### **4.7.2 xAPI Profiles**

In this section, the xAPI Profile Structure, Syntax, and Specification testing is outlined. The Structure test will assure the correct validation structure appears in the error text panel. The Syntax reviews the types of xAPI Profile error that could be generated and the Specification will be tested to ensure all xAPI Profile Specifications have been met and if not, the correct error is generated.

The xAPI Profile Structure Validations have a layout that allows the user to see the location, area of the xAPI Profile which needs attention, and the specification that needs to be remedied.

Each error message begins with the basic information about the error within the simulation, then shows the '^' below the area that needs to be remediated and the specification requirement that was not met.

**Example:**

Error Detected in: [:profiles :concepts "ResultExtension" :inScheme]

**Error Structure**

^ **Error Detected in:** [:(section of the simulation) :(the element that need attention): (the value that is causing the error)]

Spec failed

```
...  
...  
Line with the error  
^^^^^^^^^^^^^^^^^^^^  
...
```

**Should satisfy**

```
;;  
;;failed spec. Reason for the error  
;;  
;;should satisfy  
;;  
;; Specification that is not being met  
-----
```

Detected ## error(s)

### 4.7.3 Personae

In this test, the simulation will be set up with 4 Personae, 2 Roles, and 1 Group. Each scenario will be run 20 times with varying group and role attribution. After the first 'control' simulation, the remaining simulations will have common errors made in the user interface to verify the correct errors are being generated.

**Personae Specification**

- Actor 1 is assigned to Role 1 in Group 1
- Actor 2 is assigned to Role 2 in Group 1
- Actor 3 is assigned to Role 1 in Group 1
- Actor 4 is assigned to Role 2 in Group 1

## 4.7.4 Alignments

This test will take a 'Control' simulation and change certain aspects to create 10 validation errors within the Alignment configuration. The Actors, Roles, and Groups used in this test will be the same as the Personae Validation Testing.

Actor 1	<a href="http://example.com/xapi/assessed">http://example.com/xapi/assessed</a>	.5
Actor 2	<a href="http://example.com/xapi/assessed">http://example.com/xapi/assessed</a>	-.5
Actor 3	<a href="http://example.com/xapi/assessed">http://example.com/xapi/assessed</a>	.5
Actor 4	<a href="http://example.com/xapi/assessed">http://example.com/xapi/assessed</a>	.75
Role 1	<a href="http://example.com/xapi/assessed">http://example.com/xapi/assessed</a>	-.25
Role 2	<a href="http://example.com/xapi/assessed">http://example.com/xapi/assessed</a>	.25
Group 1	<a href="http://example.com/xapi/assessed">http://example.com/xapi/assessed</a>	-.3

### Desired 'Control' Results

Actor 1 final 'assessed' weight  $(.5 + -.25 + -.3) / 3 = -.01666667$

Actor 2 final 'assessed' weight  $(-.5 + .25 + -.3) / 3 = -.18333333$

Actor 3 final 'assessed' weight  $(.5 + -.25 + -.3) / 3 = -.01666667$

Actor 4 final 'assessed' weight  $(.75 + .25 + -.3) / 3 = .23333333$

## 4.8 User Interface

The goal of these User Interface tests is to create simulation specifications using the new graphical user interface. Each section below will outline the actions that will be taken to confirm a user can accurately build the simulation specifications within DATASIM.

### 4.8.1 Parameters Page

The following tests will be run to confirm the Parameter feature within the user interface is functioning as expected.

- Select Start and End Date - ensure the dates are in sequence and have the correct format.
- Select Timezone from dropdown - the timezone should be reflected in all statements for the correct hour discrepancy EST- America / New York (GMT -05:00)
- Enter Max Statements integer value into text field- ensure cannot be 0
- Enter Seed numeric value into the text field- ensure cannot be null or 0

### 4.8.2 Personae Page

In this test simulation, the user will create 9 Actors within DATASIM Personae Section, add all 9 to Group 1 (change name to Squad A). Actor 1 will be renamed to Squad Leader, assigned to Role 1 and will be associated with Group 1. The remaining Actors will have Role 2, 3, 4, 5 assigned to 2 Actors in their respective sub-teams. This will create a realistic hierarchy to test the group and role attributes.

Group 1 (Squad A)	Actor 1 (Squad Leader)	Role 1
	Actor 2	Role 2
	Actor 3	Role 3
	Actor 4	Role 4
	Actor 5	Role 5
	Actor 6	Role 2
	Actor 7	Role 3
	Actor 8	Role 4
	Actor 9	Role 5

### 4.8.3 Alignment Page

This test will add 10 Alignments to a simulation. Some of these Alignments will be for the same Activity, for different Agents. The user will determine if all of the Activities from the Statement Templates and all of the available Agents from the MoM Profile are present in the Basic Mode. In all of these test Alignments, the user will switch from Basic to Advanced to confirm the information is changing in the JSON version.

#### Personae Specifications

Actor 1 will be assigned to Role 1 in Group 1  
Actor 2 will be assigned to Role 2 in Group 1  
Actor 3 will be assigned to Role 1 in Group 1  
Actor 4 will be assigned to Role 2 in Group 1

#### Alignment Specifications

Actor 1	assessed	.5
Actor 2	evaluated	.25

Actor 3	launched	.5
Actor 4	opened	-.5
Role 1	evaluated	.75
Role 1	assessed	.25
Role 2	launched	-.75
Role 2	opened	1
Group 1	assessed	-.5
Group 2	evaluated	.25
Group 2	opened	-1

## 5. Conclusion

Following the Beta DATASIM platform Testing and Evaluation Plan will reveal that the current prototype allows the user to build simulations, correct validation errors, and do so through a guided user interface. The results of this testing, which will be completed and detailed in the Final Report, will assist the development team in the next iteration of DATASIM.

The Alignment Behaviors introduced in this iteration include the ability to assign groups and roles to an Actor, Zero-statement Alignments, and Object Override Alignments. These Alignments allow the user to build better simulations, closer to real life scenarios. The expectation is that these tests will help to show DATASIM is capable of more fine-grained control over simulation modeling.

Micro-Scale fidelity will measure the scale of accuracy of a small scale simulation. This test will show how the system can be used to explore and model activity engagement with xAPI Profiles in a classroom environment.

With the added computing power of clustered generation, these tests will demonstrate the increasing capacity for xAPI Statement generation. The visual validation error interface will allow the user to find and correct any issues within the Personae, Alignment, Profiles, or Parameters. By recreating the most common errors made within the Profile(s), this test will prove the veracity of the error messaging feature. DATASIM users can now edit and update simulation specifications using a clickable graphic user interface. These tests set out to show the new features and user interface have been implemented successfully and will be ready for the next phase of development.

Overall the expectation is that these tests will demonstrate the DATASIM platform's increased readiness to be used in an increased variety of simulation scenarios and by a much broader user base than the Alpha implementation.

## **6. Appendix**

### **6.1 Yet Calibration Profile**

The Yet Calibration Profile, used for simplified simulation behavior analysis, can be found here: [https://github.com/yetanalytics/datasim/blob/calibration\\_profile/dev-resources/profiles/calibration/calibration.jsonld](https://github.com/yetanalytics/datasim/blob/calibration_profile/dev-resources/profiles/calibration/calibration.jsonld)

### **6.2 Master Object Model Profile**

The conformant version of the Master Object Model (MOM) Profile used for generation in this effort can be found here: <https://adlnet.gov/resources/publications/2020/04/2019-Total-Learning-Architecture-Report/>