Paradata in 20 Minutes or Less

Contents

Paradata in 20 Minutes or Less
- Contents
- Terminology
- Paradata Decoded
- Summary
- Additional Examples
- Schema Quick Reference Guide
  - Change Log

The goal of this document is to get you booted up using paradata in 20 minutes or less. This guide isn't going to help you get your paradata loaded into the Learning Registry. For that, you'll need the *Learning Registry in 20 Minutes or Less* guide (and remember, use the value "payload_schema": "LR Paradata 1.0" for your paradata).

The process we used to develop the paradata model of activities, along with examples, can be found in *Modeling Paradata and Assertions as Activities*. The formal specification can be found in the *Paradata Specification*.

Please note that the examples have had their text formatted for readability purposes.

Terminology

If you don't know what paradata is, and how it relates to metadata, read this section. Otherwise, skip ahead to the next section, *Paradata Decoded*.

The term paradata itself is relatively new. It's simply used to refer to a particular kind of metadata. Specifically, paradata is usage data. How is something being used? There are lots of nuances here, but the big difference to keep in mind is that paradata involves "action" and traditional metadata involves "classification". So traditional metadata (such as Dublin Core metadata) might describe something like this:

```
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/">
  <dc:title>A Tale of Two Cities</dc:title>
  <dc:creator>Charles Dickens</dc:creator>
  <dc:language>US-en</dc:language>
</metadata>
```

Nothing wrong with that, but it refers to facts about the book: what the title is, who wrote it, what language it was written in, etc. If you want to talk about those things, use a traditional metadata format like Dublin Core. If you want to talk about *how* users interacted or used that book, use this paradata format, which is explained in greater detail below:

```
{
      "activity": {
```



```
                "actor": {
                        "objectType": "teacher",
                        "description": [
                                "high school",
                                "english"
                        ]
                },
                "verb": {
                        "action": "taught",
                        "measure": {
                                "measureType": "count",
                                "value": 15
                        },
                        "date": "2011-05-01/2011-05-31"
                },
                "object": "http://URL/to/tale-of-two-cities/"
        }
}
```

Basically this says "Over the course of May 2011, high school English teachers used the book 'A Tale of Two Cities' 15 times for teaching activities."

Paradata helps communicate when you want to talk about how people have used a book or other resource. Say you want to share the fact that some people clicked on a link on my website that leads to a page describing the book. A verb for that is "click". You may want to indicate that some people bookmarked a video for a class on literature classics. A verb for that is "bookmark". In the prior example, a teacher presented resources to a class. The verb used for that is "taught". Traditional metadata has no mechanism for communicating these kinds of things. It turns out that a group called Activity Streams (AS) has already been working on this problem for individual users. They have a really good solution for when individuals want to share that they did something specific. We like their model and have tried to use it where possible, and encourage you to do so also. For example, Activity Streams lets you talk about what individual "actors" are doing and with what, so you can talk about how "Joe Bob, a teacher, used 'A Tale of Two Cities' for teaching".

But AS doesn't let you talk about *aggregations* of activities. So you can't say in AS something like our example above, "Over the course of May 2011, high school English teachers used the book 'A Tale of Two Cities' 15 times for teaching activities". This paradata format is basically an extended and altered version of the Activity Stream JSON so it's very similar: once you learn to write one format, the other format isn't very hard to pick up.

Summary so far:

- When you want to share basic facts about something, use traditional metatdata.
- When you want to share specific actions that an individual did, use Activity Streams (and in fact AS and paradata are identical formats for this case).
- When you want to share actions that multiple people did, or multiple things that one person did, use this paradata format.

Paradata Decoded

Paradata lets you say things about activities. Let's take a look at some of the things you can say. We're going to include JSON expressions along with English language statements so that you can immediately begin to see how to actually use this format in practice. This may be a good time to review JSON in Wikipedia.

There are three main parts to a basic paradata statement: ACTOR, VERB, and OBJECT. These words work pretty much like they do in regular English grammar. An "actor" *does* "verb" *to an* "object". For example, "A teacher taught the lesson located at some URL."

Actor = A teacher
Verb = Taught
Object = The lesson located at some URL

Let's see what the JSON would look like:

```
{
	"activity": {
		"actor": "teacher",
		"verb": "taught",
		"object": "http://URL/to/lesson/"
	}
}
```

Pretty simple.

Q: What would the paradata look like if we want to say "Teachers taught this lesson 15 times"?
A: The paradata now would include "15 times" in a "measure" under "verb":

```
{
	"activity": {
		"actor": "teacher",
		"verb": {
			"action": "taught",
			"measure": {
				"measureType": "count",
				"value": 15
			}
		},
		"object": "http://URL/to/lesson/"
	}
}
```

OK, that got a bit more complicated due to the way JSON works. The verb "taught" has been replaced with a whole new JSON object. That introduces two things. First, inside the object, the term "action" is now used to hold the "verb" value of "taught". Second, "measure" has been added. "Measure" is a way to count, or otherwise measure, the verb in some appropriate manner. In this example, it's used to indicate that the the verb "taught" occured 15 times.

You could also measure actors or objects but that can get a little confusing so we don't recommend you do that unless you have a really good reason.

The above example is a fairly complete paradata expression. Let's stop for a minute and define the top level terms in more detail:

"actor" refers to the person or group who does something. Remember, if you want to talk mainly about

individual actions, you should consider using the Activity Streams format.

"verb" refers to the action that is taken. Within "verb":

- ➥ **"action"** just holds the verb value when using a JSON value for verb (as opposed to when you're just using a string value like "verb": "taught")
- ➥ **"measure"** provides a way to talk about the occurrence of the verb (whether counts, averages, ratings, or other things).

"object" refers to the thing being acted upon. The important part of an object is the URL (or URI) where you can find out about the object (or get the object, or ideally both).

Saying, "Teachers taught this resource 15 times" is interesting, but to be meaningful we need more context about what is going on. Paradata permits flexible context around things like, "What kinds of teachers are these?" or "When did this activity occur?". Let's fit answers to those questions, one at a time, into the following examples:

Q: How would we say when an activity occurred? E.g., "Teachers taught this lesson 15 times during the month of May 2011".
A: The verb would include the date that the action occurred.

```
{
      "activity": {
            "actor": "teacher",
            "verb": {
                  "action": "taught",
                  "measure": {
                        "measureType": "count",
                        "value": 15}
                  },
                  "date": "2011-05-01/2011-05-31"
            },
            "object": "http://URL/to/lesson/"
      }
}
```

We just introduced the "date" element which, as implemented here, is a single string with two substrings indicating the start and stop dates of the range. The start and stop dates are separated by a forward slash ("/"). Start and stop dates are inclusive. If you want to indicate a single date when the action occurred, use a single date inside the "date" field instead. (Note: if you're writing a parser for paradata, you need to handle the possibility that you'll get ranges and single dates from this field).

Q: How do we say that specific kinds of teachers taught this lesson? "High school English teachers taught this resource 15 times during the month of May 2011".
A: You add a more complex description of the actor.

```
{
	"activity": {
		"actor": {
			"objectType": "teacher",
			"description": [
				"high school",
				"english"
			]
		},
		"verb": {
			"action": "taught",
			"measure": {
				"measureType": "count",
				"value": 15
			},
			"date": "2011-05-01/2011-05-31"
		},
		"object": "http://URL/to/lesson/"
	}
}
```

To get the description of the actor in there, we had to convert actor from just the plain "teacher" string to a JSON object. That's the same process we went through earlier when we needed to expand the definition of "verb". In this case the term "objectType" is being used to conceptualize how to describe an actor. Its value remains "teacher", which is the string we've already been using to describe "actor". Just as "objectType" is used to generically describe an actor, so is "action" used to generically describe a "verb". Finally, we added the term "description" and used a string array to provide keywords describing the "teacher".

These keywords can be whatever you want, of course, but you'll want to try to make sure you're using terms that make sense to people who are reading your paradata! Looking at examples of how others are sharing paradata is a good way to participate in an existing community sharing information about learning activities.

Summary

We're getting close to 20 minutes now, so hopefully this has clarified how paradata JSON works. There are many more things that you can do with paradata, and remember a more complete specification with lots of examples is available in *Modeling Paradata and Assertions as Activities* (http://goo.gl/Cq6im).

Here's a recap:

1. ACTORS do things called VERBS with/to OBJECTS.
2. DESCRIPTIONS provide context to ACTORS, VERBS or OBJECTS.
3. MEASUREMENT gives data about magnitude and can be a variety of formats. In our example above we measure how many times the VERB occurred over a period of time.
4. DATES indicate when an action took place.

Additional Examples

Here are a few more examples from the main guide to help flesh things out:

Statement: "Resource @ URL X was bookmarked on Delicious by a high school physics teacher around June 2011":

```
{
	"activity": {
		"actor": {
			"objectType": "teacher",
			"description": [
				"high school",
				"physics"
			]
		},
		"verb": {
			"action": "bookmarked",
			"context": {
				"id": "http://www.delicious.com/"
			},
			"date": "2011-06-01/2011-06-30"
		},
		"object": {
			"id": "http://resourceurl/resourceX/"
		},
		"content": "Resource @ URL X was bookmarked on Delicious by a high
		school physics teacher around June 2011"
	}
}
```

The two new concepts here are:

1. "context": This is similar to "description" but provides the context for where the activity occurred. In this case we can tell it took place on Delicious.com.

2. "content": This is a new top level entity, which is just a human-readable string description of what the paradata says.

Statement: "Resource @ URL X was created by combining resources, X1, X2, …, XN".

```
{
	"activity": {
		"verb": "assembled from",
		"object": "http://resourceurl/resourceX/",
		"related": [
			{
				"object": {
					"objectType": "resource",
					"id": "http://resourceurl/X1/"
				}
			},
			{
				"object": {
					"objectType": "resource",
					"id": "http://resourceurl/X2/"
				}
			},
			{
```

```
                        "object": {
                              "objectType": "resource",
                              "id": "http://resourceurl/XN/"
                        }
                  }
            ],
            "content": "Resource @ URL X was created by combining
             resources,x1,x2,… xn"
}
```

Here we've also introduced two ideas. One is the *absence* of an "actor". The absence of "actor" indicates that this is an assertion (an arbitrary, free-standing fact/opinion statement) and not an activity. Second, there's a new, optional top-level item called "related" which describes a relation between the object in the statement and the other objects. In this case, it indicates that the "related" objects, resources X1, X2, … XN, were combined to create the main object resourceX.

Related things are added as a top-level element to the model. Related things are objects, all of which are the same type.

Statement: "Organization A asserts Resource X supercedes Resource Y."

```
{
      "activity": {
            "actor": "organizationA",
            "verb": "supercedes",
            "object": "http://resourceurl/resourceX/",
            "related": ["http://resourceurl/resourceY/"
            ],
            "content": "Organization A asserts Resource X supercedes Resource Y"
      }
}
```

This example doesn't introduce any new elements; it's a straight forward combination of the actor, verb, and a pair of objects. The verb is the assertion, rather than the action of the actor. Since both the object and related object are resources, the paradata format permits this simpler form that does not require the full description of the related object.

Statement: "Organization A asserts Resource X is identical to Resource Y."

```
{
      "activity": {
            "actor": "organizationA",
            "verb": "same as",
            "object": {
                  "objectType": "resource",
                  "id": "http://resourceurl/resourceX/"
                  },
            "related": [
                  {
                  "object": {
                        "objectType": "resource",
                        "id": "http://resourceurl/resourceY/"
                        }
                  }
            ],
            "content": "Organization A asserts Resoure X is identical to
             Resource Y"
      }
}
```

This example also doesn't introduce any new elements, but it shows how to use the more complex form of describing a

related object. It's perfectly acceptable to use either the complex or simple form of the statement.

Schema Quick Reference Guide

The main elements of the paradata schema are described below. If you want all the details, see the formal *Paradata Specification*.

"actor"		Refers to the person or group who does something. Alternatively, if you want to talk mainly about individual actions, you should consider using the Activity Streams format).
{	**"objectType"**	Is used when "actor" is a JSON structure, and refers to the actor itself (the same value you would use if actor was just a string).
	"description"	Is used to provide a list of attributes that describe the actor. It can also be used to describe the action or object.
}		
"verb"		Refers to the action that is taken.
{	**"action"**	Is used when "verb" is a JSON structure, and refers to the verb itself (the same value you would use if verb was just a string).
	"measure"	Provides a way to talk about the occurrence of the verb. It is a metric appropriate in the context. It can be a count, average, rating, or other measure.
	{ **"measureType"**	Is the kind of measurement. This can be any value that makes sense, but common ones include "count" and "rating".
		"value" Is the value or magnitude of the measurement. "count" has a value that describes how many times a thing happened. "rating" has a value that indicates the average rating.
	}	
	"date"	Is either a point or period of time (aka range of dates). If it's a period of time, it contains two dates separated by a slash. Actually this field is defined by RFC3339 and ISO8601 so look it up if you want more detail.
	"context"	Provides a way of describing where the activity took place.
}		
"object"		Refers to the thing being acted upon. The important part of an object is the URL (or URI) where you can find out about the object (or get the object, or ideally both)
{	**"id"**	Is used when "object" is a JSON structure, and refers to the object itself (the same value you would use if object was just a string).
}		
"related"		Is a collection of things that relate to the paradata (usually the object). It's an array of JSON "objects".
{	**["object"]**	Objects in the list that are related to the main object.
}		

"content"	Is a string that is a human-readable sentence restating the meaning of the paradata expression as a whole.

Change Log

Version	Date	Author	Description
1.0	20110106	SM	Initial version.
1.1	20110930	KM	General edits. Added examples.