

Coding Manual: Institutional Statements in OSS Development

We are reading and annotating (labeling) a large number of emails between programmers on open source projects to extract “institutional statements” such as rules about how the project is governed. In our annotation exercises, we would like to explore Institutional Statements in the interactions, transactions and outcomes in open source development, and particularly focus on the context of Apache incubator ecosystem and projects involved.

Definition of an Institutional Statement (IS):

You will be labeling all institutional statements you encounter. Here is the full definition of an institutional statement:

An institutional statement is a shared linguistic constraint or opportunity that prescribes, permits, or advises actions or outcomes for actors (both individual and corporate) (Crawford and Ostrom, 1995, 583).

More about Institutional Statements

Institutional statements come in many forms. You won’t be labeling them according to these forms, but knowing their variety will make them easier to identify.

First, they can include *strategies*, *norms*, and *rules*. Strategies involve outlines of directives/plans/course of action, but do not include explicit consequences or any indication that someone “should”, “must”, or “must not” perform the action in question. Norms are just like strategies, except they do have the normative should/must implication: they recommend a course of action. However, they have no formal consequence associated with that action. Rules are like norms, except they also have an explicit consequence for complying or failing to comply. The point of this summary is that strategies, norms, and rules are all examples of institutional statements, and you will encounter statements that run the range of normativity.

Another distinction is that between *regulatory* and *constitutive* institutional statements. Regulatory statements can be seen as indicating the behavior of agents in the institution. They are often phrased in terms of how some agent can, should or must behave. Constitutive statements define or introduce or state general aims or values of the institution. They can define a term or manifest an entity (define an entity into existence: “There shall be a Congress”). Again, the point of introducing this distinction is to illustrate the variety that institutional statements exhibit. You will encounter both kinds.

Coding Labels

We will be searching through emails for three types of institutional statement: general **Institution-focused**, general **Artifact** and **No-label**. All three will fit the most abstract definition of institutional statement, but they will have different focuses.

Institution-focused: A sentence that is institution-focused is about the community in general, its processes and systems and positions. Statements about it will relate to the “we” of the project, whether explicitly or implicitly, or they may be about the “I” or “you” of a formal position in the community (rather than the “I” or “you” of a specific individual). It can include value claims about what the community aspires towards. It might refer to the “it” of the project, the code artifact, but it is not about the code, it is about the organization of the project and people around the code, processes for contributing to the code, and possibly including technical systems that support building the code such as documentation, installation, and repositories. Institution-focused statements are general, about kinds of things that happen, and things that tend to happen, not about instances, events, or situations that are not expected to happen again.

Artifact-focused: A sentence that is artifactual, rather than being about the general “we” of the project is about the general “it”: the code of the project, the code artifact being collectively built. These sentences will fit the most general definition of an institutional statement, but they will be about the code and its architecture and infrastructure and design, rather than about the community and how it runs (Institution-focused). However, they are also not about a specific line of code, or setting, or bug: they are high level, about its features and functionality. They are not about other software projects, except how the core project interfaces with them, either in interactions or as dependencies. You will avoid statements about how specific parts of the code artifact relate to an individual (Specific, below). Artifact statements

are not opinions, they are more like descriptions or sometimes speculations, although if they are specifically about best practices *within the community*, they may be Institution-focused. They can include statements about general programming best practices — technical advice, protocols, rules, strategies and statements which broadly apply to project development — and so may appear mixed in with advice that does not get this general label, but instead relates to individuals' specific circumstances.

No-label: This label shall be used to mark off the first sentence of any email you believe, in its entirety, does not contain a single statement under the Institutional or Artifact categories. That way we can make sure you didn't miss an email, and it just doesn't have any statements

One type of No Label sentence that helps to be aware of is what we can Specific. It is specific-focused about an “I” or a “you” of a specific individual or is about a specific situation. Specific statements are effectively “one-offs.” They can include peer-to-peer advice, opinions, impressions, or desires for a particular exchange. It might be about a specific person's special circumstance or problem, or pertain to particular incidents and instances which may be a part of the project development. Also, issue/failure resolution, tips, and good practice recommendations. A Specific statement is likely to be describing a situation that is unique to the moment, and not being treated as if it will happen again. Statements about actual or potential regular or typical occurrences, and the processes for handling them, are general, not specific. Statements that point an individual to a general process are general.

Nuance and guidelines

This definition of IS was written with policy documents in mind, not conversational emails. Because people do not talk in institutional statements, it is necessary to refine this definition further:

1. Some of these interpretations rely on domain knowledge, both the specifics of each project, and programming generally. You don't have that knowledge, and that's OK. Make your best guess.
2. Keeping in mind the overall context of a whole email or thread of emails is vital for assigning appropriate labels. It sometimes helps to check to see what project's mailing list an email is from.
3. Don't be surprised if Institution-focused statements are rare. Out of 50 emails, it may be that you only find 10-15 that fit the Institution-focused category, or fewer.
4. Automatically generated text, from logs, email footers, or copied code can be ignored. Some automatically generated text is Institutional, however, such as when an automatic printout informs a user how or where to perform an institutional process. Knowing what text is automatically generated requires domain knowledge, so if you accidentally label too much, that is OK.

5. If you aren't sure about a label, use the [discussions sheet](#) to list the contentious example, and your notes about it, instead of labeling it right away.
6. If you aren't sure between Artifact-focused and Specific, you can err in the direction of Specific. If you aren't sure between Institution-focused and another category, after careful consideration, **mark off the instance as an Institutional statement for further expert review.**

We may add more guidance here as we all gain more experience with the task.

Examples

We went through some of the initial batches of annotations submitted by coders, and following are some directions we reached that might aid in future labeling tasks.

Example	IS Label and justification
<i>The purpose of the DFDL schema is to specify the properties of a data format</i>	ARTIFACT. This satisfies the definition of IS, and is primarily about the code artifact and programming. You may have had to Google quickly to learn that DFDL is something being developed in part in this project.
<i>And by default in system configuration state store is disabled</i>	ARTIFACT. This is about code and programming. aka a technical specification.
<i>So we end up holding on to data and using up memory just in case we need that data later</i>	ARTIFACT.
<i>What you can't do is define a type completely with all representation types, and then evolve them and change your mind in derived types</i>	ARTIFACT. This might also be Specific. It depends if the context indicates a general description of a principle, or a comment or critique in a specific context.
<i>While the spec may say comments should be stripped, I have seen that formats get extended with things that are placed into structured comments, so the spec de-facto evolves to need the comments preserved</i>	ARTIFACT. An specific individual is describing their own experiences, but the subject is general aspects of the artifact.
<i>Support for much larger files is something we hope to be able to support in the future</i>	ARTIFACT. The “we” helps make this general.
<i>In this case only full lines can be comments</i>	SPECIFIC. It describes a single, specific case
<i>May be making the images public would help</i>	SPECIFIC. this is specific: it's not a decision to make certain types of data public, but to make it public in one instance(“the” images); for one specific application
<i>Designing schemas to limit backtracking can help with this, but may not be possible in some cases</i>	ARTIFACT. Very general textbook advice, not project-specific statements, even if it relates to a specific problem.
<i>Though, if a format is found where a different</i>	SPECIFIC. This is difficult, because it is describing a future

<i>representation is needed, we can always discuss adding it</i>	instance, which may suggest a general issue, or may be offering another, different specific conversation about this issue that might connect it to a general aspect of the Artifact
--	---

Example	IS Label and justification
<i>This e-mail and any attachment is intended for the party to which it is addressed and may contain confidential information or be subject to professional privilege</i>	NO LABEL. This would be an Institution-focused statement, except it is automatically generated, being part of an email footer.
<i>If you are not the intended recipient, please contact the sender and delete all copies</i>	NO LABEL.
<i>But it's a pretty large change with some technical complexities that we haven't had a chance to fully work through yet</i>	SPECIFIC. This feels like an individual observation.
<i>Please check your network connectivity or contact the administrator for assistance</i>	INSTITUTION. It is tricky because it is being mentioned as it relates to a specific case, but the advice communicated is the general advice that would be offered for all problems of this type: "ask for help".
<i>Unless specified, we will have an interactive open agenda with a mix of:</i> - Meet and greet - Quick overview of any new major features - Community discussion (issues that anyone is facing / feature asks / etc)	SPECIFIC. This is about a specific instance or circumstance, not about general processes. It would be a "YES" if it started "The agendas of our interaction open meetings are typically as follows: ..." or if there was any indication or inference from the phrasing that all meetings are this way.
<i>This is actually the expected behavior; though it's maybe not always desired</i>	ARTIFACT. this is about the code/product, not the institution
<i>Of course we can change font or colors</i>	SPECIFIC. This is about a specific instance or circumstance, not about general processes
<i>While incubation status is not necessarily a reflection of the completeness or stability of the code, it does indicate that the project has yet to be fully endorsed by the ASF</i>	INSTITUTION. general ASF policy towards project ratification

Example	IS Label and justification
<i>If we do not hear back from the original contributor within a week, or the</i> > contributor is not interested in working on it anymore: we will close the > PR and attach it's diff to the corresponding JIRA issue (for reference of > anyone who wants to work on it in the future)	INSTITUTION. relates to a specific scenario/person, but seems to be an application of a general policy to that specific case.
<i>My tutorial on processing binary files shows how to use DFDL schema to process Windows EXE files, byte by byte, bit by bit</i>	INSTITUTION. This sounds too specifically about the code, but a tutorial is inherently for a general task that many have to learn, so it must refer to a general process

	for learning how to do something. Still, this example is hard: it could be an Artifact statement because it relates to something general about the code. Ultimately, tutorials are about the code artifact, but this statement is about the tutorial, not the artifact.
<i>Incubation is required of all newly accepted projects until a further review indicates that the infrastructure, communications, and decision making process have stabilized in a manner consistent with other successful ASF projects</i>	INSTITUTION. Talks of processes and the projects as organizations.
<i>In order to subscribe to the lists, you need to send your subscription request to "user-subscribe@ariatosca.incubator.apache.org" and "dev-subscribe@ariatosca.incubator.apache.org", rather than to the actual lists</i>	INSTITUTION. Instructions to project participants

I will post a question to the SAXON mailing list	SPECIFIC. Acting on behalf of himself or his team
I haven't found anything like this and I'm hoping someone in the community may know where I can find an example/tutorial	SPECIFIC.
I have come to think of it as a flaw in dfdl	SPECIFIC. Individual deduction.
I just feel the length of stream is too long, just personal opinion	SPECIFIC. Opinion
I have not been able yet to reproduce safely	SPECIFIC.
To enable state store in CLI apps either enable them-- in your system configuration or enable it from your CLI app using this.sysConfig(ConfigurationKeys.STATE_STORE_ENABLED, "true")	SPECIFIC. Instruction for a specific issue
Please check your configuration for mapreduce.framework.name<http://mapreduce.framework.name> and the correspond server addresses	NO LABEL. This would be a Specific statement, except it is automatically generated, being the pasted output of a program that was run.

Updates to package.json and package-lock.json to resolve downstream Prototype Pollution Vulnerabilities in dev dependencies	ARTIFACT. General update on code artifact
The purpose of the DFDL schema is to specify the properties of a data format	ARTIFACT. Statement on project component/artifact
Using gobblin_0.10.0, we want to use module, KafkaSimpleJsonExtractor.java in order to see	ARTIFACT. General strategy around the artifact

both kafka record keys and values	
The architecture diagram leads us to think that the exactor is the point where both Kafka key and value are visible in the Gobblin pipeline: https://gobblin.readthedocs.io/en/latest/Gobblin-Architecture/#gobblin-constructs	ARTIFACT.
Apache UserALE.js is suited for business analytics and enterprise application monitoring, but built for precision workflow analysis, UI/UX research and user-testing	ARTIFACT. General aspirational statement about the artifact
Apache Flagon is an effort undergoing incubation at The Apache Software Foundation (ASF), sponsored by the Apache Incubator project	INSTITUTION.
The gist of the > issue is that because not all fields fall nicely on 1-byte divisions, different > pieces of a field will get jumbled if you read the data as a linear stream from > memory	SPECIFIC. Looks like a very specific issue discussion.
While the spec may say comments should be stripped, I have seen that formats get extended with things that are placed into structured comments, so the spec de-facto evolves to need the comments preserved	ARTIFACT.
Added skipping feature to MultiWorkUnits: > https://github.com/apache/incubator-gobblin/pull/1521	SPECIFIC. or NO LABEL if it is a log output
The Saxon web site (https://www.saxonica.com/html/documentation/extensibility/functions/staticmethods.html) says that, using Saxon, an XSLT program can call static Java methods	ARTIFACT.

Highlighting:

Your main work will be reading texts, highlighting the ones that fit a tag, and selecting the appropriate tag for the highlighted text.

- 1. When highlighting a statement, please make a clean selection of a statement, making sure to exclude any preceding or trailing periods, punctuation, blank spaces, etc.**
- 2. When selecting a set of contiguous statements as IS, please select each sentence separately to label as IS, instead of as a chunk.**

3. **System generated messages, code snippets, compiler logs, errors, stack trace reports etc do not ordinarily qualify under any of the institutional statement categories.**
4. **You are expected to annotate labels in not just the email body, but also emails contained in trailing threads, or emails quoted in reply.**
 - a. Email content in trailing threads can begin with special characters like “>”. They will even break up sentences. It is OK to highlight those.
5. Once you have annotated all Institutional Statements in an email, proceed to the next. Make sure your annotations are saved properly.

We assume that you can complete about 50 emails per hour. There should be 75-150 posts all in all available for each student to work on. Please inform us if you find otherwise.

Tips and advice

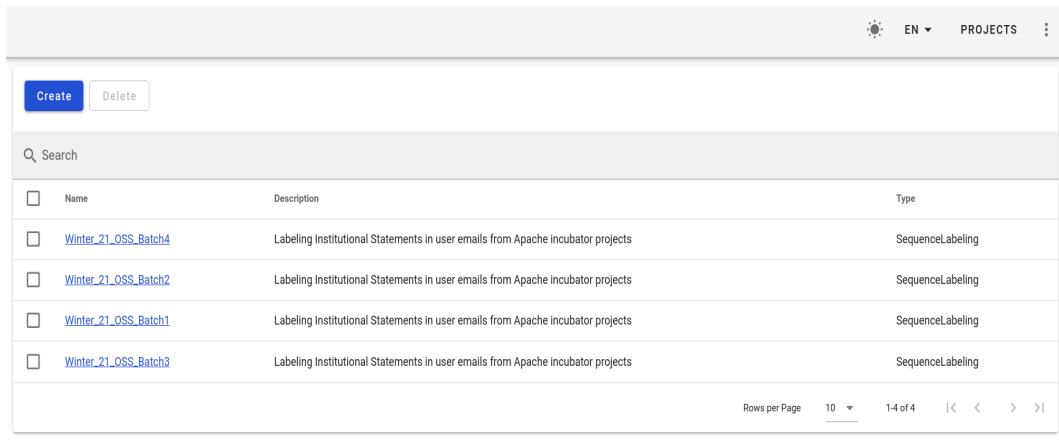
1. The general content you will be working with will be very much technical as well as institutional. When deciding if a certain context fits a label (say artifact vs institutional) and you are unsure if the statements in question are technical or not, it helps to Google key terms/parts of the statement. Usually the search results are a helpful indication of whether the statement is associated with development related contexts, or organizational aspects/operation/directives.
2. Questions are almost never institutional statements. Proposals and suggestions as well.
3. The URL to the original email should be visible in the right hand panel, for reference. This is sometimes useful to refer to, because, among other things, it tells you which of the many product names you read is the name of the group's main product.

About Doccano:

We are using a special platform that makes it easier to label text. It is called Doccano.

Every student will be provided access in the role of an annotator. To begin:

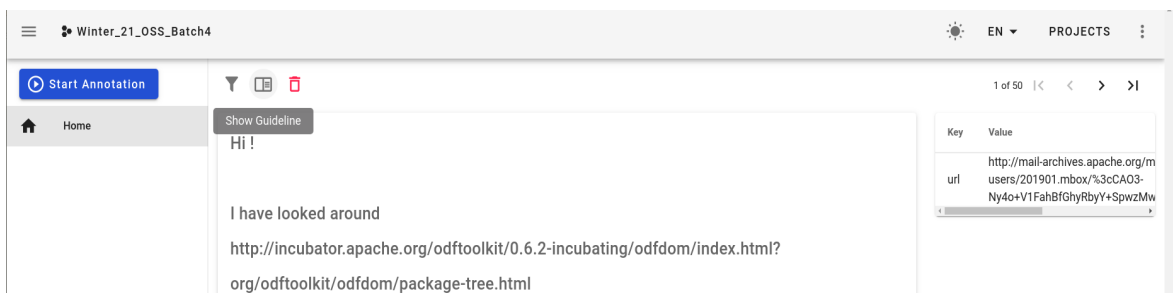
6. Got to login, you shall be prompted to enter credentials.
Currently, your username has been set the same as your password. Your username is your last name in lowercase.
7. Once in, you should be able to view the annotation project assigned to you, under **Projects**. Enter by selecting it.



Name	Description	Type
Winter_21_OSS_Batch4	Labeling Institutional Statements in user emails from Apache incubator projects	SequenceLabeling
Winter_21_OSS_Batch2	Labeling Institutional Statements in user emails from Apache incubator projects	SequenceLabeling
Winter_21_OSS_Batch1	Labeling Institutional Statements in user emails from Apache incubator projects	SequenceLabeling
Winter_21_OSS_Batch3	Labeling Institutional Statements in user emails from Apache incubator projects	SequenceLabeling

You should see one project assigned to you as of now. Please [inform me](#) if you find none, or more than one visible on your dashboard.

- You can now start labeling emails for Institutional statements through the “Start Annotation” button at the upper left. You will be presented with a series of posts to be labeled. Label is color coded in orange and named “Institutional Statement”. A link to this guide can be accessed from the panel above the email.



Start Annotation

Home

Show Guideline

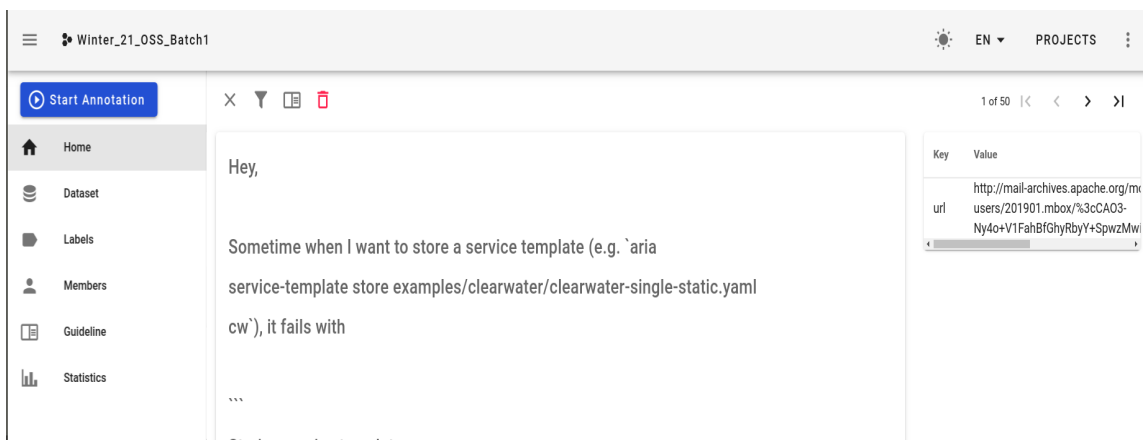
Hi !

I have looked around

<http://incubator.apache.org/odftoolkit/0.6.2-incubating/odfdom/index.html?org/odftoolkit/odfdom/package-tree.html>

Key	Value
url	http://mail-archives.apache.org/musers/201901.mbox/%3cCA03-Ny4o+Y1FahBfGhyRbyY+SpwzMW

- Posts already labeled will be present too, along with tags you have previously assigned them. You can skip forward to find a post to start working on.



Start Annotation

Home

Dataset

Labels

Members

Guideline

Statistics

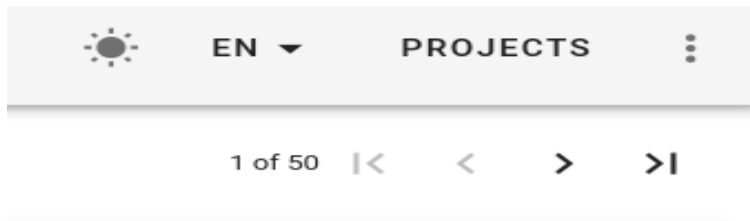
Hey,

Sometime when I want to store a service template (e.g. `aria service-template store examples/clearwater/clearwater-single-static.yaml cw`), it fails with

...

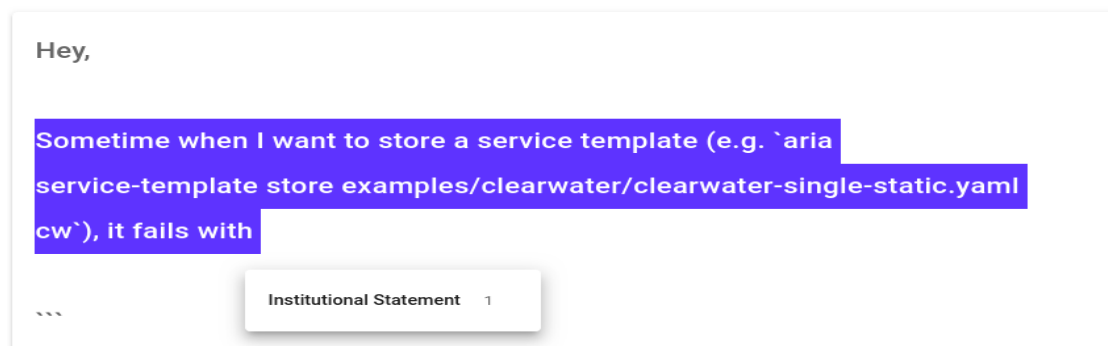
Key	Value
url	http://mail-archives.apache.org/musers/201901.mbox/%3cCA03-Ny4o+Y1FahBfGhyRbyY+SpwzMW

10.

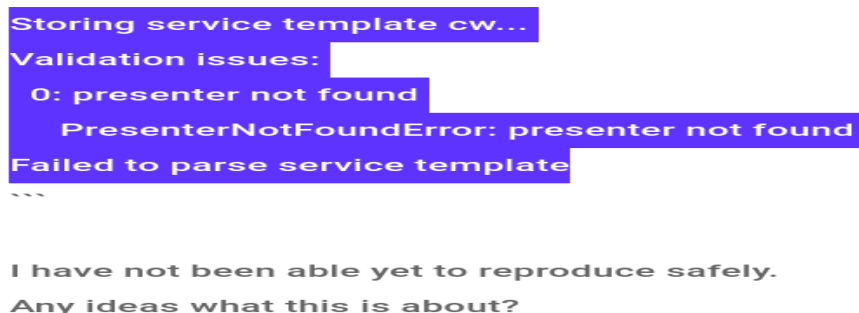


Some challenges:

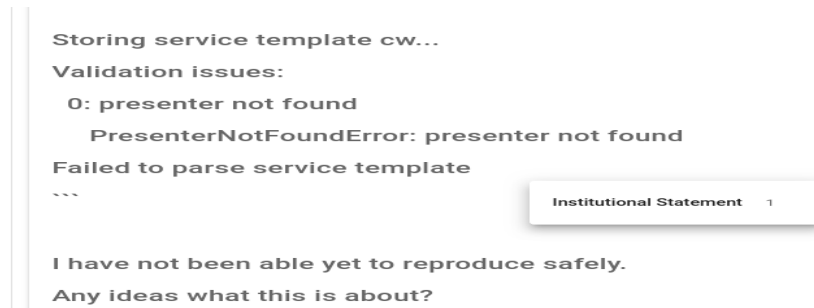
1. The interface becomes unresponsive if kept idle for long. Refresh and/or logout to start again.
2. The hovering labels panel may sometimes not appear right away.



Deselect your selection by clicking anywhere outside the highlight.



The label panel should be back now. Click on the intended choice of label from the panel.



Do however make sure that you have annotated every email in your batch. No matter where you started off. It helps by running through all the emails in the batch at least once before the deadline, to make sure nothing has been skipped.