DaffodilProposal

Daffodil Proposal

⚠️ FINAL ⚠️

Abstract

Daffodil is an implementation of the Data Format Description Language (DFDL) used to convert between fixed format data and XML/JSON.

Proposal

The Data Format Description Language (DFDL) is a specification, developed by the Open Grid Forum, capable of describing many data formats, including both textual and binary, scientific and numeric, legacy and modern, commercial record-oriented, and many industry and military standards. It defines a language that is a subset of W3C XML schema to describe the logical format of the data, and annotations within the schema to describe the physical representation.

Daffodil is an open source implementation of the DFDL specification that uses these DFDL schemas to parse fixed format data into an infoset, which is most commonly represented as either XML or JSON. This allows the use of well-established XML or JSON technologies and libraries to consume, inspect, and manipulate fixed format data in existing solutions. Daffodil is also capable of the reverse by serializing or "unparsing" an XML or JSON infoset back to the original data format.

Background

Many different software solutions need to consume and manage data, including data directed routing, databases, data analysis, data cleansing, data visualizing, and more. A key aspect of such solutions is the need to transform the data into an easily consumable format. Usually, this means that for each unique data format, one develops a tool that can read and extract the necessary information, often leading to ad-hoc and data-format-specific description systems. Such systems are often proprietary, not well tested, and incompatible, leading to vendor lock-in, flawed software, and increased training costs. DFDL is a new standard, with version 1.0 completed in October of 2016, that solves these problems by defining an open standard to describe many different data formats and how to parse and unparse between the data and XML/JSON.

Two closed source implementations of DFDL currently exist. The first was created by IBM and is now part of their IBM® Integration Bus product. The second was created by the European Space Agency, called DFDL4S or "DFDL for Space" targeted at the challenges of their satellite data processing.

Around 2005, Pacific Northwest National Lab created Defuddle, built as an open source implementation and proof of concept of the draft DFDL specification and a test bed to feed new concepts into specification development. Primary development of Defuddle was eventually taken over by the National Center for Supercomputing Applications (NCSA). However, due to evolution of the DFDL specification and architectural and performance issues with Defuddle, around 2009, NCSA restarted the project with the new name of Daffodil, with a goal of implementing the complete DFDL specification. Daffodil development continued at NCSA until around 2012, at which point development slowed due to budget limitations. Shortly thereafter, primary development was picked up by Tresys Technology where it continues today, with contributions from other entities such as the Navy Research Lab, the Air Force Research Lab, MITRE, and Booz Allen Hamilton. In February of 2015, Daffodil version 1.0.0 was released, including support for the DFDL features needed to parse many common file formats. Daffodil version 2.0.0 is expected to be released in August of 2017, which will include unparse support with one-to-one parsing feature parity.

Entities including IBM, MITRE, NATO NCI Agency, Northrop-Grumman, Quark Security, Raytheon, and Tresys Technology have developed DFDL schemas for many data formats from varying technology domains, including PNG, GIF, BMP, PCAP, HL7, EDIFACT, NACHA, vCard, iCalendar, and MIL-STD-2045 , many of which are publicly available on the DFDL Schemas github. There are also a number of military-application data formats, the specifications of which are not public, which have historically been very difficult and expensive to process, and for which DFDL schemas have been created or are actively in development; these include MIL-STD-6040/USMTF ATO, MIL-STD-6017/VMF, MIL-STD-6016/NATO STANAG 5516 (aka "Link16").

Rationale

Numerous software solutions exist that consume, inspect, analyze, and transform data, many of which can be found in the Apache Software Foundation (ASF). In order for tools like these to consume new types of data, custom extensions are usually required, often with high development and testing costs. Daffodil fills a clear gap in many of these solutions, providing a simple and low cost way to transform data to XML or JSON, which many of these tools natively support already. With the upcoming 2.0.0 release, the Daffodil project will have achieved a level of functionality in both parse and unparse that, when integrated into existing solutions, could provide for a new method to quickly enable support for new data formats.

Initial Goals

- Relicense the existing code from the University of Illinois/NCSA Open Source License to the Apache License version 2.0, working with Apache Legal to ensure correctness, and with Daffodil contributors to get their permission.
- Move the existing codebase, documentation, bugs, and mailing lists to the Apache hosted infrastructure
- Establish a formal release process and schedule, allowing for dependable release cycles in a manner consistent with the Apache development process.
- Build relationships with ASF projects to add Daffodil support where appropriate
- Grow the community to establish a diversity of background and expertise.

Current Status

Meritocracy

All initial committers are familiar with the principles of meritocracy. The Daffodil project has followed the model of meritocracy in the past, providing multiple outside entities commit access based on the quality of their contributions. In order to grow the Daffodil user base and development community, we are dedicated to continuing to operate Daffodil as a meritocracy.

A key ingredient in a meritocracy of developers is open group code review. The Daffodil project has operated in this mode throughout its existence and this provides a forum to improve the code, verify code quality, and educate new developers on the code base.

Community

Daffodil has a small community of users and developers. Although primary Daffodil development is done by Tresys Technology, a handful of other contributions have come from other entities including the Navy Research Lab, the Air Force Research Lab, MITRE, and Booz Allen Hamilton. In addition to developers, multiple users of Daffodil have created DFDL schemas, including entities such as MITRE, IBM, Raytheon, Quark Security, and Tresys Technology. The DFDL Schemas github community has been created as a place for DFDL schemas to be published. The Daffodil project also makes use of mailing lists, HipChat, and Confluence Questions to build a community of users and system for support.

Core Developers

The core developers of Daffodil are employed by Tresys Technology. We will work to grow the community among a more diverse set of developers and industries.

Alignment

Daffodil was created as an open source project with a philosophy consistent with The Apache Way. A strong belief in meritocracy, community involvement in decisions, openness, and ensuring a high level of quality in code, documentation, and testing are some of our shared core beliefs.

Further, as mentioned in the Rationale section, Daffodil fills a gap that exists in many ASF projects, including NiFi, Spark, Storm, Hadoop, Tika, and others. In order for tools like these to consume new types of data, custom extensions are usually required. Rather than create such extensions, Daffodil provides an easy and standards-compliant way to transform data to XML or JSON, which many of these tools already natively support.

Known Risks

Orphaned Products

The current core developers are the leading contributors in the space of DFDL and wish to see it flourish. Though there is some risk that the initial committers all come from the same company, a goal of entering into incubation is to grow the development community to minimize the risk of reliance on a single company.

Inexperience with Open Source

The Daffodil project began as an open source project and has continued that model throughout development. This includes public bug tracking, git revision control, automated builds and tests, and a public wiki for documentation.

Additionally, the current core developers and initial committers all work for a company that relies on, believes in, promotes, and has led or contributed to many open source software projects, including SELinux Userspace, OpenSCAP, CLIP, refpolicy, setools, RPM, and others. As such, there is low risk related to inexperience with open source software and processes.

Homogeneous Developers

The proposed initial committers come from a single entity, though we are committed to growing the Daffodil development community to include a broad group of additional committers from a wide array of industries.

Reliance on Salaried Developers

The proposed initial committers are paid by their employer to contribute to the Daffodil project. We expect that Daffodil development will continue with salaried developers, and are committed to growing the community to include non-salaried developers as well.

Relationship with other Apache Projects

As mentioned in the Alignment section, Daffodil fills a clear gap in numerous other ASF projects that consume and manage large amounts of data.

As a specific example, Daffodil developers have created a Daffodil Apache NiFi Processor, currently in use in data transfer solutions, which allows one to ingest non-native data into an Apache NiFi pipeline as XML or JSON. This processor was well received by the Apache NiFi developers, with positive comments about the concise API and how it could handle non-native data. Daffodil developers have also successfully prototyped integration with Apache Spark. We believe Daffodil could provide a strong benefit to many other ASF projects that handle fixed format data. We anticipate working closely with such ASF projects to include Daffodil where applicable to increase their ability to support new data formats with minimal effort.

Daffodil also depends on existing ASF projects, including Apache Commons and Apache Xerces.

An Excessive Fascination with the Apache Brand

Although the Apache brand may certainly help to attract more contributors, publicity is not the reason for this proposal. We believe Daffodil could provide a great benefit to the ASF and the numerous data focused projects that comprise it, as described in the Rationale and Alignment sections. We hope to build a strong and vibrant community built around The Apache Way, and not dependent on a single company.

Documentation

Daffodil documentation can be found at:

- https://opensource.ncsa.illinois.edu/confluence/display/DFDL/Daffodil%3A+Open+Source+DFDL

Information about DFDL can be found at:

- https://www.ogf.org/ogf/doku.php/standards/dfdl/dfdl
- https://www.ibm.com/support/knowledgecenter/en/SSMKHH_9.0.0/com.ibm.etools.mft.doc/df20060_.htm

Public examples of DFDL Schemas can be found at:

- https://github.com/DFDLSchemas

Initial Source

The Daffodil git repo goes back to mid-2011 with approximately 20 different contributors and feedback from many users and developers. The core codebase is written in Scala and includes both a Scala and Java API, along with Javadocs and Scaladocs for API usage. The initial code will come from the git repository currently hosted by NCSA at the University of Illinois :

https://opensource.ncsa.illinois.edu/bitbucket/projects/DFDL/repos/daffodil/

Source and Intellectual Property Submission

The complete Daffodil code is licensed under the University of Illinois/NCSA Open Source License. Much of the current codebase has been developed by Tresys Technology, who is open to relicensing the code to the Apache License version 2.0 and donate the source to the ASF. Contacts at NCSA are also open to relicensing their contributions to Apache v2. We plan to contact the other contributors and ask for permission to relicense and donate their contributed code. For those that decline or we cannot contact, their code will be removed or replaced. We will work closely with Apache Legal to ensure all issues related to relicensing are acceptable.

External Dependencies

We believe all current dependencies are compatible with the ASF guidelines. Our dependency licenses come from the following license styles: Apache v2, BSD, MIT, and ICU. The list of current Daffodil dependencies and their licenses are documented here:

https://opensource.ncsa.illinois.edu/confluence/display/DFDL/Dependencies+and+Licenses

Cryptography

None

Required Resources

Mailing Lists

- commits@daffodil.incubator.apache.org
- dev@daffodil.incubator.apache.org
- private@daffodil.incubator.apache.org
- user@daffodil.incubator.apache.org

Source Control

git://git.apache.org/incubator-daffodil.git

Issue Tracking

JIRA Daffodil (DFDL)

Initial Committers

- Beth Finnegan <efinnegan at tresys dot com>
- Dave Thompson <dthompson at tresys dot com>

- Josh Adams <jadams at tresys dot com>
- Mike Beckerle <mbeckerle at tresys dot com>
- Steve Lawrence <slawrence at tresys dot com>
- Taylor Wise <twise at tresys dot com>

Affiliations

- Beth Finnegan (Tresys Technology)
- Dave Thompson (Tresys Technology)
- Josh Adams (Tresys Technology)
- Mike Beckerle (Tresys Technology)
- Steve Lawrence (Tresys Technology)
- Taylor Wise (Tresys Technology)

Sponsors

Champion

- John D. Ament

Nominated Mentors

- Dave Fisher
- John D. Ament *

Sponsoring Entity

We request the Apache Incubator to sponsor this project.