The Apache Xerces™ Project - Charter

NOTICE:

Table of contents

1 INTRODUCTION

1.1 Apache Xerces is a collaborative software development project dedicated to providing robust, full-featured, commercial-quality, and freely available XML parsers and closely related technologies on a wide variety of platforms supporting several languages. This project is managed in cooperation with various individuals worldwide (both independent and company-affiliated experts), who use the Internet to communicate, plan, and develop XML software and related documentation.

1.2 This charter briefly describes the mission, history, organization, and processes of the project.

2 MISSION

2.1 Apache Xerces exists to promote the use of XML. We view XML as a compelling paradigm that structures data as information, thereby facilitating the exchange, transformation, and presentation of knowledge. The ability to transform raw data into usable information has great potential to improve the functionality and use of information systems. We intend to build freely available XML parsers and closely related technologies in order to engender such improvements.

2.2 The Apache Xerces parsers support standard APIs (formal, de facto, or proposed). They are designed to be high performance, reliable, and easy to use. To facilitate easy porting of ideas between languages, the API's supported should be as similar as possible, given the constraints of the languages and existing architectures. Apache Xerces parsers should also be designed to work efficiently with other Apache projects that deal with XML whenever possible.

2.3 We believe that the best way to further these goals is by having both individuals and corporations collaborate on the best possible infrastructure, APIs, code, testing, and release cycles. Components must be vendor neutral and usable as core components for all.

2.4 In order to achieve a coherent architecture between Apache Xerces parsers and other components and applications, standards (formal or de facto) will be used as much as possible for both protocols and APIs. Where appropriate, experiences and lessons learned will be fed back to standards bodies in an effort to assist in the development of those standards. We will also encourage the innovation of new protocols, APIs, and components in order to seed new concepts not yet defined by standards.

3 HISTORY

3.1 The code base which formed the foundations of both the Xerces-Java and Xerces-C++ subprojects of the Apache XML Project was originally donated to Apache by IBM in 1999.

Xerces-Perl came into existence as a subproject of the Apache XML project after the Xerces-C++ community had already matured to a significant extent. All three were subprojects of the Apache XML Project until late 2004. At this time, reflecting the growth in the Apache XML project and these communities themselves, Apache Xerces became a top-level Project of the Apache Software Foundation. Apache Xerces still shares much infrastructure with the Apache XML project and the other former subprojects of Apache XML that have become projects in their own right.

4 TERMS

4.1 The ASF Board. The management board of the Apache Software Foundation.

4.2 The Project. The Apache Xerces Project; intended to refer to the source code, website and community that are Apache Xerces.

4.3 Subproject. Apache Xerces is composed of a number of subprojects which fit into one of two categories:

a) An XML parser implementation in some particular programming language. There may be multiple parsers for a given language, if the API's the parsers support are sufficiently dissimilar. At the time of writing, there is one parser for each of Java, C/C++ and Perl.

b) A set of components serving some purpose not directly pertinent to XML parsing, but which are used in related applications and are tightly bound, usually through internal API's, to one (or more) of the parser subprojects.

4.4 Product. Some deliverable (usually a binary or source package) that a subproject releases to the public. Subprojects may have multiple products.

4.5 Contributor. Anyone who makes a contribution to the development of the Apache Xerces project or a subproject.

4.6 Committer. Apache Xerces has a set of committers. Committers are contributors who have read/write access to the source code repository.

5 THE PROJECT MANAGEMENT COMMITTEE

5.1 The Apache Xerces project is managed by a core group of committers known as the Project Management Committee [PMC], which is composed of volunteers from among the active committers (see 8.3 below) from all subprojects. Each subproject must have at least one representative on the PMC, to ensure active supervision of the subproject.

5.2 The activities of the PMC are coordinated by the Chairperson, who is an officer of the corporation and reports to the Apache Board. The Chairperson will, on the request of the Apache Board, provide reports to the Board on issues related to the running of the Apache Xerces project.

5.3 The PMC has the following responsibilities:

a) Accepting new subproject proposals, voting on these proposals and creating the subproject (see SUBPROJECTS below). This is done in collaboration with the Incubator (see http://incubator.apache.org).

b) Facilitating code or other donations by individuals or companies, in collaboration with the Incubator.

c) Resolving license issues and other legal issues in conjunction with the ASF board.

d) Ensuring that administrative and infrastructure work is completed.

e) Facilitating relationships among subprojects and other Apache projects.

f) Facilitating relationships between Apache Xerces and the external world.

g) Overseeing Apache Xerces to ensure that the mission defined in this document is being fulfilled.

h) Resolving conflicts within the project.

i) Reporting to the ASF board (through the Chair) on the progress of the project.

5.4 In cases where the sub-project is unable to directly provide at least one representative on the PMC--implying that there are no active committers on that code base--then the subproject should be considered dormant, and any relevant Apache policies for dormant projects should be implemented. At the least, the subproject's status should be updated on its website.

5.5 Every 12 months, or at the request of the Board, the PMC will provide a recommendation to the Apache Board for the position of Chairperson of the PMC.

5.6 This recommendation will be made on the basis of an election held within the PMC. The election will be performed using a simple majority vote of PMC members.

5.7 Upon agreement by the Apache Board, the recommended Chairperson will, if they are not already, be appointed an officer of the corporation. See http://www.apache.org/foundation/bylaws.html for more information.

5.8 In the unlikely event that a member of the PMC becomes disruptive to the process, ceases to make codebase contributions for an extended period, or ceases to take part in PMC votes for an extended period of time, said member may be removed by unanimous vote of remaining PMC members.

5.9 The PMC is responsible for maintaining and updating this charter. Development must follow the process outlined below, so any change to the development process necessitates a change to the charter. Changes must be approved by a two-thirds majority of all members of the PMC.

6 SUBPROJECTS

6.1 When a new subproject proposal is submitted to the PMC, it may be accepted by a two-thirds vote of the PMC.

6.2 A subproject may be removed by unanimous vote of the PMC, subject to the approval of the ASF board.

7 CONTRIBUTORS

7.1 Like all Apache projects, the Apache Xerces project is a meritocracy -- the more work you do, the more you are allowed to do. Contributions will include participating in mailing lists, reporting bugs, providing patches and proposing changes to a product.

7.2 In order to ensure that all code contained in the Apache Xerces project's code repository is free of licensing, intellectual property and patent issues, any developer wishing to contribute a new feature to Xerces must either sign:

a) If contributing as an individual, sign the "Individual Contributor License Agreement (CLA)" (http://www.apache.org/licenses/icla.txt) and file a copy with the Secretary of the Corporation; or

b) If making the contribution as part of their employment responsibilities, sign the "Corporate CLA (CCLA)", (http://www.apache.org/licenses/cla-corporate.txt) and file a copy with the Secretary of the Corporation.

7.3 If the contribution in question is a small bugfix, the contributor need not sign a CLA, but need only provide the following information, attaching it to the communication containing the patch:

a) Name and employer

b) Are you the author of the code being contributed?

c) Do you have the right to grant the copyright and patent licenses for the contribution that are set forth in the ASF v.2.0 license (http://www.apache.org/licenses/LICENSE-2.0)?

d) Does your employer have any rights to code that you have written, for example, through your contract for employment? If so, has your employer given you permission to contribute the code on its behalf or waived its rights in the code?

e) Are you aware of any third-party licenses or other restrictions (such as related patents or trademarks) that could apply to your contribution? If so, what are they?

7.4 Contributors who make regular and substantial contributions may become committers as described below.

8 COMMITTERS

8.1 Each subproject has a set of committers. Committers are contributors who have read/write access to the source code repository.

8.2 Normally, a new committer is added after a contributor has been nominated by a committer and approved by at least 50 percent of the active committers for that subproject with no opposing votes. In the case that a subproject has a very small number of active committers, the PMC may choose to require a PMC resolution to approve the nomination of a contributor by one of the active committers in that subproject. All committers must have a signed Contributor License Agreement on file with the Secretary of the Corporation. Since, in most cases, contributors will already have contributed significant amounts of code, this should usually have been done before nomination.

8.3 Although committers have write access to all Apache Xerces subprojects, they are only permitted to make changes to the subprojects to which they have been elected committers. A committer may be elected to multiple subprojects, but, except that no new access need be granted, the process is the same as for any other contributor.

8.4 For the purposes of voting, committers will be classed as "active" or "inactive". Only active committers will be included in the totals used to determine the success or failure of a particular vote, and only active committers are part of the PMC.

8.5 Committers remain active as long as they are contributing code or posting to the subproject mailing lists. If a committer has neither contributed code nor posted to the subproject mailing lists in 3 months, the PMC chair may e-mail the committer, the subproject development list, and the PMC mailing list notifying the committer that they are going to be moved to inactive status. If there is no response in 72 hours, the committer will become inactive, and may be removed from the PMC mailing list.

8.6 An inactive status will not prevent a committer committing new code changes or posting to the mailing lists. Either of these activities will automatically re-activate the committer for the purposes of voting, and necessitate their addition to the PMC mailing list.

9 INFRASTRUCTURE

9.1 The Apache Xerces project relies on the Apache XML project and the Apache Infrastructure project for the following:

a) Bug Database -- This is a system for tracking bugs and feature requests.

b) Subproject Source Repositories -- These are several repositories containing both the source code and documentation for the subprojects.

c) Website -- A xerces.apache.org website will contain information about the Apache Xerces project, including documentation, downloads of releases, and this charter. Each subproject will have its own website with subproject information.

d) PMC Mailing List -- This list is for PMC business requiring confidentiality, particularly when an individual or company requests discretion. All other PMC business should be done on the general mailing list.

e) General Mailing List -- This mailing list is open to the public. It is intended for discussions that cross subprojects.

f) Subproject Mailing Lists -- Each subproject should have at least one devoted mailing list. Many subprojects may wish to have both user and development lists. The individual subprojects may decide on the exact structure of their mailing lists.

10 LICENSING

10.1 All contributions to the Apache Xerces project adhere to the Apache Software Foundation License, v.2.0 (http://www.apache.org/licenses/LICENSE-2.0)? All further contributions must be made under the same terms.

10.2 When a committer is considering integrating a contribution from a contributor who has no CLA on file with the Corporation, it is the responsibility of the committer, in consultation with the PMC, to conduct due diligence on the pedigree of the contribution under consideration; see sections 7.2 and 7.3.

11 THE DEVELOPMENT PROCESS

11.1 The development process is intentionally lightweight; like other Apache projects, the committers decide which changes may be committed to the repository. Three +1 ('yes' votes) with no -1 ('no' votes or vetoes) are needed to approve a significant code change. For efficiency, some code changes from some contributors (e.g. feature additions, bug fixes) may be approved in advance, in which case they may be committed first and changed as needed, with conflicts resolved by majority vote of the committers.

12 SUBPROJECT REQUIREMENTS

12.1 Each subproject should have a set of requirements as well as an up-to-date release plan and design document on its dedicated web page.

12.2 It is recommended that each subproject have a smoke-test system that works at least as a basic integration test.

13 RELATIONSHIP TO OTHER APACHE PROJECTS

13.1 The Apache Xerces project should work closely with other Apache projects, such as XML, Jakarta and the Apache Server, to avoid redundancy and achieve a coherent architecture among Apache Xerces and these projects.