ExpectedFailures

The goal of the Expected Failures project is to deal with the unavoidable reality that stdcxx has failures in its test suites that cannot easily be fixed or worked around, either due to hard problems in the project itself or in some third party dependency such as a compiler or an operating system. The solution is to indicate these kinds of "expected" failures in test results in a way that makes it possible to distinguish them from all the others (i.e., the unexpected kind).

To that end we categorize the kinds of failures that we encounter and assign each a code and color in the test result pages. Since the same test can fail with different symptoms on different platforms or even different versions of the same platform, it seems best to decouple this information from the test itself and store it separately. The process then goes like this. As each failure is analyzed it either gets fixed or it becomes expected. The nature of each expected failure (compilation error, signal name, non-zero exit status, etc.) is entered in a special file along with the platforms it's observed on and with the bug key. Each time an new entry to the file is added the bug key is also mentioned in the Subversion change comment so that the entry is cross-referenced in the issue in Jira. This special file is then referenced when we generate our test results so that the expected failures can be distinguished from any others on our web site.

You can see how this works in practice on one of our result pages, for instance: http://stdcxx.apache.org/builds/4.2.x/aix-5.3-ppc-vacpp-9.0.html

The kinds (categories) of failures we distinguish are described in Codes section on the same page.

The special file where we enter expected failures is called `xfail.txt`. See, for example, the xfail.txt file on the 4.2.x branch.

An example of a change that added an expected failure for the problem described in STDCXX-1009 is r693416

What's not supported in this scheme yet is the ability to denote just a subset of a test's assertions as expected to fail. The solution to the problem that we discussed but haven't had time to implement yet is to assign each assertion a sequential number and use it to denote those that are expected fail for known reasons using a comma separated list.