

# Test262-ES6

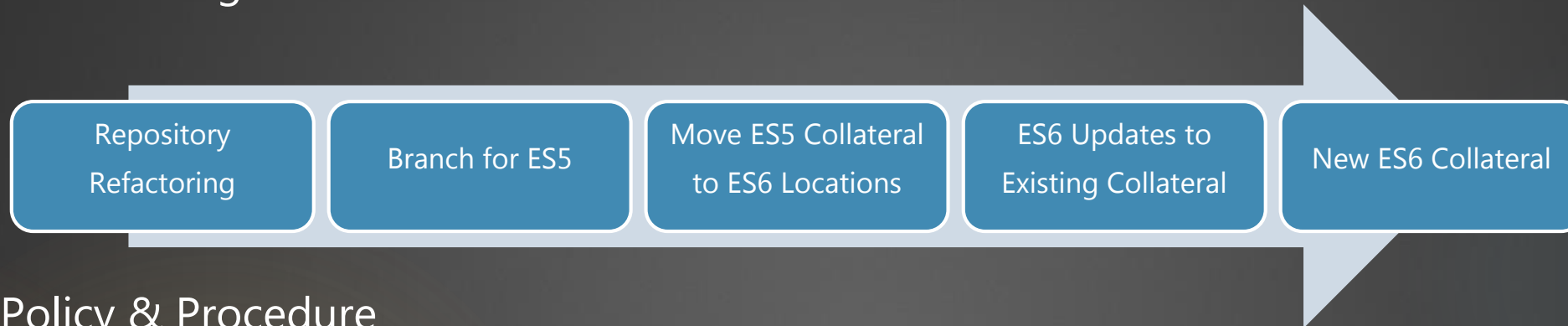
GET READY FOR ES6

# Status

- ◆ Recent update to website with a few test fixes (Thanks Adobe!)
- ◆ Loads of Work Items (test bugs, additional test collateral) tracked on [bugs.ecmascript.org](https://bugs.ecmascript.org).
- ◆ (Un)Official IRC channel set up: [#test262](https://irc.mozilla.org).
- ◆ Microsoft nearly ready to contribute test cases for `let`, `const`, `Map`, `WeakMap`, and `Set` (modulo some updates for recent spec revisions).

# Proposed changes – with patches!

## Code Changes



## Policy & Procedure

- ◆ Formalize test262-core members on the wiki
- ◆ Website refresh process
- ◆ Make algorithm step identification optional

## Website Changes

- ◆ New Endpoint for test262-es6
- ◆ New Landing Page for all ECMAScript-related collateral

# Code Changes

# Repository Refactoring

## Basics

- ◆ Proposed by Norbert Lindenberg (bug id: 575).
- ◆ Group suites together – ecma262, ecma402, best practices – and make room for future ancillary test suites
- ◆ Group test case dependencies together outside of harnessing.
- ◆ Group common harness files together (preventing duplication), and group all harness files under one location
- ◆ Make generated assets more clear.

## Details

New layout:

- ◆ tests/ecma402
- ◆ tests/ecma262
- ◆ tests/bestPractices
- ◆ tests/includes
  - ◆ Any shared test helpers go here. Packager will pick most specific include available.
- ◆ harness/shared
- ◆ harness/website
- ◆ harness/console
- ◆ tools/ (same as today)
- ◆ website/ and console/ are auto-generated directories that won't be checked in.
- ◆ docs/ and external/ untouched.

# Branch for ES5

- ◆ Goals

- ◆ As friction-free as possible for edge ECMAScript collateral
- ◆ Clearly delineate ES5 collateral from ES6 collateral.
- ◆ Ability to update collateral for ES6 without breaking ES5 collateral.
- ◆ Easily integrate fixes between branches

- ◆ Proposal

- ◆ Once refactor is complete, create ES5 branch.
- ◆ ES6 collateral goes into master
- ◆ ES5-specific fixes made to off of ES5 branch.
- ◆ Collateral fixes that cross versions can will simply be integrated.

# ES6 updates to current collateral

(depends on agreed-upon procedures)

- ◆ Programmatically move collateral from ES5 to ES6 locations
- ◆ Persist ES5 @path attribute (which should match filepath) as @es5id
- ◆ Add @es6id attribute that matches new file path
- ◆ Note: proposal to refactor/flatten sub-folders of section



# Procedure Additions & Updates



# Test262 Core

- ◆ Has commit rights
- ◆ Has access to update the website
- ◆ Documented at <http://wiki.ecmascript.org/doku.php?id=test262:coreteam>
- ◆ (Currently only has 3 names, who else has commit rights??)

(Also, pretty sure I completely hosed the wiki creating this page, the index is broken at least)

# Website Refresh Procedure

- ◆ Ensure harness runs in Top 4 rendering engines – Chrome/Opera, FF, IE, Safari.
- ◆ Examine results for drastic changes in pass rates
- ◆ Send mail to test262-discuss with changes that will be going live
- ◆ Wait 24 (?) hours
- ◆ Follow process documented here (modulo previous changes):  
[http://wiki.ecmascript.org/doku.php?id=test262:submission\\_process](http://wiki.ecmascript.org/doku.php?id=test262:submission_process)

# Optional Algorithm Step Designation

- ◆ Goal: reduce friction to contribute & reduce friction carrying forward collateral from previous versions.
- ◆ Proposal:
  - ◆ All tests MUST have either @es5id or @es6id.
  - ◆ An ID MUST contain a section identifier, and MAY contain an algorithm identifier.
- ◆ Drawbacks
  - ◆ Ensuring 100% coverage becomes much harder.

# Work Items

| Owner         | Description                                    |
|---------------|--|
| Brian Terlson | Repo Refactoring                               |
| Brian Terlson | Branch for ES6                                 |
| ?             | Create mapping of ES5 sections to ES6 sections |
| Brian Terlson | Move ES5 collateral to ES6 locations           |
| Everyone 😊    | Contribute ES6 collateral                      |

# Open Questions

- ◆ How to create cross-host-compatible collateral for realms and scripts?
- ◆ How to verify proxies broadly? Can we leverage existing collateral?
- ◆ Strict Mode: harness should run strict-agnostic tests in both modes.
- ◆ Harness should hit fast path (ie. run tests in a loop).
- ◆ Poisoning of built-ins – can we test this broadly in the harness?
- ◆ Others?