



- Joe Hewitt: web development evolved
- Integrated with Firefox:
 - at your finger tips, instant on
 - direct reflection of DOM, CSS
 - inspect, direct manipulation of debugger
- Huge hit with Web 2.0 developers
- Open sourced Dec. 2006
- Active development ended w/1.05 ~ May 2007



- Firebug awesome, but no `eval()` debugging
 - `eval()` widely used in Web 2.0
 - Basis for Dojo toolkit, JSON data
 - No debugging support for `eval()` in Firebug or Venkman
- How to extend Firebug to support `eval()` ?
 - Ok, in reality it looked like a bug fix at first...



Example code, dynamic Javascript eval

```
// Load the code dynamically
var req = new XMLHttpRequest();
req.open('GET','payloadCatsAndDucks.js', false);
req.send(null);
eval( req.responseText ); // line 23

// Now use the code
prod( new duck() );
prod( new cat() );
prod(); // call point of error
```



```
// Load the code dynamically
var req = new XMLHttpRequest();
req.open('GET','payloadCatsAndDucks.js', false);
req.send(null);
eval( req.responseText); // line 23

// Now use the code
prod( new duck());
prod( new cat());
prod(); // call point of error

function prod( animal) { // line
    animal.say();
};

var duck = function() {
    this.say = function() {
        dump('quack\n');
    };
}

var cat = function() {
    this.say = function() {
        dump('meow\n');
    };
}
```

The `eval()` code looks overlaid on caller



- `jsd.scriptHook(script)` (script means "function" in Mozilla JS)
 - script can be top-level, nested, event, eval, or Function
 - both top-level and eval scripts have no `functionName`
 - top-level scripts are on top of the stack, eval are not: we have a test
 - Ooops: no stack frame in this scriptHook!
 - Ok, set a breakpoint in the script at PC=0, return
- `jsd.breakpointHook(frame, type, rv)`
 - Remove our breakpoint, use `frame` to decide if we have an eval
 - Create a unique ID for buffer, `data:` URL
 - Store `eval` source and metadata in side table
- Breakpoints, stack traces, error messages:
 - Detect eval-frames, use side table



- Ok, its big hack, but it works!
 - Did I mention that `eval()` is critical to Web 2.0?
 - Did I mention that Firebug 1.0 was open source but not active?
 - How to complete contribution?
- Create 1.1 to contribute `eval()` debugging
 - Build alpha versions
 - Incorporate key bug fixes to draw users
 - Contribute to discussion group to build trust
 - Recruit to broaden impact: Max Stepanov from Aptana joins
 - Beta releases, respond to bug reports

Demo time



■ Features

- eval()
 - executable lines marked
 - Call stack side panel
 - Firefox 3.0 compatibility
 - New XHR spy implementation
 - extended plugin architecture
 - editor interaction API
 - bug fixes
-
- Firebug 1.1.0b7 1471 downloads in Oct. 2007
 - Firebug-1.1.0b9 will be 1.1.0. Probably.
-
- ...but it's slow



■ Performance

- create performance tests (done)
- MD5 instead of encodeURIComponent (done)
- Delay add line numbers (in debug)
- re-do script side tables
- performance analyze
- static/dynamic user interaction (?)

■ new Function() (done)

■ bug fixes



- Expand dev team *Not announced!*
 - John Barton, IBM Almaden
 - Max Stepanov [Aptana](#), JSD expert
 - [Steve Souders](#), the YSlow guy, Yahoo
 - [Dion Almaer](#) Google, Ajaxian founder
 - [Joe Hewitt](#) Facebook, Firebug creator
 - TBD Mozilla, the Firefox people
- Invest time to create corporate sponsors
 - To sustain Firebug open source effort
 - On-site demos/talks
 - Encourage company-specific plugins
 - Intern program



Chromebug: Firebug for Chrome

- Multi-window/multi-DOM support
- **chrome** urls, Firefox components, extensions
- Avoid eating your own arm
- Goal: do more than Venkman+DOMInspector
- demo



More ambitious features

- Break on mutation events
- Event/Execution monitoring
- Direct JSON support
- Update w/o reload
- In-place *Need to size and prioritize these*
edit
assistance
- Integrated JS/C++ crash analysis
- GC/Memory footprint tools
- ...could go on for pages



- jsDebug 2.0 for Web 2.0
 - JSD Built in support for dynamics
 - redesign API (stack frames pls!)
 - Writable PC/frames
 - All-event hook
 - GC/Memory hooks
 - Migrate firebug-service via shim API
- Mozilla 2
 - Debug Support for Tamarin
- Webkit
 - implement jsDebug 2.0 for webkit
 - primary focus would be mobile



- Source [svn at code.google.com](http://svn.atcode.google.com)

- trunk - 1.05 (stable)
- branches/eval - 1.1.0b9 (beta, no tracing)
- branches/firebug1.1 - 1.1.0 (beta w/tracing)
- branches/firebug1.2 - 1.2aX (unstable)

- Downloads

- 1.0: getfirebug.com Nice, compelling, easy install
- 1.1.0b: fireclipse.xucia.com obscure, adequate

- Documentation

- [Bug tracking on code.google.com](#) kinda lame
- [Discussion group](#)
- [FAQ](#) short
- [1.0 docs](#) nice but Needs up date for 1.1



Firebug 1.1.1

- Bug fix/contained feature migration track
- Merge diffs from 1.2
- Pro: shorter release cycle
- Con: shorter release cycle
- Con: patch management overhead
- Summary: would require a dedicated engineer

Would be nice, but its not practical now



Firebug 1.2feature1a

- Create feature branches
- Each feature branch to track 1.2, merge on accept
- Pro: independent feature progress
- Con: loss of opportunistic fixes
- Con: coordination on main branch
- Summary: a good choice for mixed-commitment team



- A growing, diverse group relies on Firebug
 - Need a stable fallback release
- Complex, multiple use-model tool
 - Plan for long beta cycle
- Many cool short and long range ideas
 - Cool ideas need user input too
- Proposal: Four download pages on GetFirebug
 - Firebug Museum: old stable releases (no updates)
 - Firebug Stable: one stable release
 - Firebug Dev: beta release (updates to self then stable)
 - Firebug Labs: experimental features (updates to self then beta on merge)



- Joe Hewitt's Firebug is web development evolved
 - but it's a bug-eat-bug world out there
- Development tools need our support
 - Javascript is not a toy language, it's a vital economic engine
 - We need to build more and better development tools for Javascript
 - We need to attract JS development tools developers!