1. A number of potential names have been put forward for the new "Titanium Desktop". Below are some of the most popular suggestions. Pick one.

		Response Percent	Response Count
DeskWrap		2.9%	2
Tide		43.5%	30
AppTide		17.4%	12
Triceradesktops		0.0%	0
Vanadium		36.2%	25
	Other (please specify) Show Responses		15

mobile bypass
29/2/2012 13:10 View Responses
Chimera
28/2/2012 21:22 View Responses
TiDesktop
28/2/2012 16:51 View Responses
Anything other than these
28/2/2012 9:27 View Responses
Last thing we need is a new branding. Just called it Titanium Desktop.
27/2/2012 15:42 View Responses
Desketeer "One for all"
27/2/2012 13:09 View Responses
tide might be an issue for search due to Tide the detergent. also, we should be moving away from titanium name.
27/2/2012 11:15 View Responses
Is Tide trademarked?
26/2/2012 20:47 View Responses
TiDesk
25/2/2012 18:23 View Responses
KrytonDesktop because it is 'super' and still in 'elemental' theme. This question required an answer choice from above rather than choosing other.
25/2/2012 4:31 View Responses
Zirconium
24/2/2012 23:13 View Responses
I don't know... names are not too important.
24/2/2012 22:45 View Responses
Would like to see a consistent product lineup naming scheme (i.e. the platform for AppleTV, GoogleTV, etc. could be called Scandium)
24/2/2012 19:30 View Responses
AppTiDe also
24/2/2012 18:40 View Responses
Keep it,or use another physical element name
24/2/2012 18:34 View Responses

2. How important to you is the continued inclusion of server side languages (Python, Ruby, PHP)? If important, please explain why.

		Response Percent	Response Count
Very Important		26.1%	18
Important		26.1%	18
Take it or leave it		23.2%	16
Would prefer to see server-side languages removed		24.6%	17
		Additional reasons/comments Show Responses	22

It's useless without scripting support.
29/2/2012 13:10 View Responses

We are currently heavily invested in python with our current product and wish to certainly maintain python support.
29/2/2012 12:28 View Responses

Some of the File APIs in Titanium Desktop were under-documented so I had to use Ruby's STDLib. Actually - having Ruby at my disposal simplified the code in quite few places.
29/2/2012 10:09 View Responses

I am a newbie, but have spent some time learning PHP and would like for it to stay. I use Javascript for the client side.
28/2/2012 16:51 View Responses

Better familiarity with Python over JS, also Python's larger standard library
26/2/2012 21:38 View Responses

Keep Python and Ruby.
26/2/2012 19:52 View Responses

only if nodeJS comes in as a replacement (although it's not really comparable)
26/2/2012 5:52 View Responses

The only important is node.js
25/2/2012 23:04 View Responses

performance = fast + lightweight no need to learn another languages html5 already have the power html+javascript+css3+sqlite
25/2/2012 18:23 View Responses

I prefer JS!
25/2/2012 17:08 View Responses

Node.js inclusion would be perfect. Or some custom solution, with module support.
25/2/2012 10:57 View Responses

Would prefer a focus on Javascript, to keep the product lean & focused as well as make it easier to learn/understand from a new developers point of view.
25/2/2012 8:53 View Responses

I was personally interested in this project for the easy programming issues, Cause I want to write a program that should run on web and also on desktop too with full desktop programming features.

This is functionality I currently use and rely on. I use the kroll bridge in python primarily but is and essential bridge to functionality that does not exist in JavaScript.

If it would allow for a single code base to used on server side for a web app or embedded in a app. This does not happen automatically just because we include the language; we would have to have a library with a common api that has two version for server and client side.

I think it would be great to have them but I don't know that they are all important to the overall project. I would like to see them pulled out of the core and maybe added in like a module. They could live in their own github and people could update them as needed. Some type of pluggable modularity would be great because then someone could develop other languages etc...

I would prefer focus on JavaScript

Would like to see a focus on making a "trusted" SSB. So focus on javascript and node.js "style" plugins.

I think they complicate the platform, and come with an additional cost that I don't think is worth it.

Because, we can not do everything with javascript only, so I need python integration. For example, local file manipulation with python. Also, It is easy to embed python codes within Tide rather than wxPython GUI. So, it is very very important to include at least python.

I have extensive experience with PHP, but getting PHP to work within it is hard on occasions. Even a simple 'echo' doesn't work properly.

There are already way too much libraries that could be used by a developer to reach what he wants with server side scripts, so removing this from the tool would limit the usage of the framework.

3. Would you prefer to see a more "javascript-based" approach to Ti Desktop, similar to that employed by the Titanium Mobile SDK?

		Response Percent	Response Count
Yes		69.6%	48
No		14.5%	10
Don't Care		15.9%	11
	Other (please specify) Show Responses		6

This question needs more explanation
28/2/2012 21:22View Responses
Would this help us use Titanium Mobile?
28/2/2012 16:51View Responses
see answer to previous question. i think the JS approach is good for the long term.
26/2/2012 5:52View Responses
no more library please . . .
25/2/2012 18:23View Responses
I want to see an enhacement of JavaScript API's not a radical shift in Desktop.
25/2/2012 4:31View Responses
But I don't think that Ti Desktop should convert and compile js as native code. Should focus on being an awesome SSB instead.
24/2/2012 19:30View Responses

4. Would you prefer to see WebKit replaced by Chrome Embedded (Chromium)?

		Response Percent	Response Count
Yes		42.6%	29
No		13.2%	9
Don't Care		44.1%	30
	Comments / Supporting Reasons Show Responses		18

seems like a good idea.. should be easier to work with
29/2/2012 22:31View Responses
webkit or chromium - we are interested in seeing a working solution that fixes current webkit issues in TI desktop.
29/2/2012 12:28View Responses
Chrome Embedded (Chromium) sounds good, but I do not want to "slow down" the Desktop development.
28/2/2012 16:51View Responses

As long as the performance and quality of rendering is as good or better it doesn't matter
28/2/2012 9:27View Responses
Better performance, better HTML5 support
27/2/2012 13:09View Responses
not entirely sure what the benefits are :)
26/2/2012 5:52View Responses
v8
25/2/2012 23:04View Responses
On OSX, at least, using embedded WebKit means I get a small binary and a rendering engine that stays up to date.
25/2/2012 19:35View Responses
keep helping our apps small fast and lightweight. costumizeable CE. v8 engine mathml parser like firefox (we are not building another browser, is'nt it?
25/2/2012 18:23View Responses
I don't really know if it would help.
25/2/2012 10:57View Responses
Whatever is going to be fastest, safest and easiest for the core devs to keep up to date.
25/2/2012 8:53View Responses
its great.
25/2/2012 8:07View Responses
Absolutely not. Let's not screw with this. WebKit is reliable and trusted and I don't want to see the project as the odd man out. We will have a hard enough time maintaining code that Apple will support for entry to AppStore. Let's not mess this up.
25/2/2012 4:31View Responses
Main reason for me is that I am using sencha touch which mostly relies on an underlying webkit browser. I think it would best to stick with what is already there unless there are powerful reasons to move to something else.
24/2/2012 22:45View Responses
I don't want us maintaining our own separate webkit fork - we'll fall behind feature wise faster.
24/2/2012 19:48View Responses
Misnomer since CEF is based on WebKit. ;) As long as inclusion of chromium embedded can be treated as a framework on all platforms to keep redistributable size down I'm all for it. Marshall seems really dedicated to the CEF project and Ti Desktop using it will just make him that much more dedicated to it.
24/2/2012 19:30View Responses
I think it should be easier to stay up to date, simplify the build process, make sure performance is good cross platform.
24/2/2012 19:25View Responses
As long as it supports the latest version from the UI point of view I don't think it should matter. If there are any chances to use node, webkit would mean to duplicate the js engine, which makes no sense (as far ar I understand). I think we should use webkit for now because we already have the foundation, and if later chromium makes more sense to make the switch
24/2/2012 18:40View Responses

5. What technologies do you want to see integrated into any future Ti Desktop product?

php
29/2/2012 22:31 View Responses
Sprinke, or some kind of update mechanism
29/2/2012 21:16 View Responses
Geo
29/2/2012 17:14 View Responses
How about make it work with *nix.
29/2/2012 13:10 View Responses
not sure
29/2/2012 12:28 View Responses
- Ti Web Mobile - CommonJs
29/2/2012 12:16 View Responses
Anything what's included in dev previews of Chromium ;-) (indexed db, sockets etc)
29/2/2012 10:09 View Responses
debugger
29/2/2012 9:12 View Responses
None
28/2/2012 21:22 View Responses
Does the Desktop have a GUI interface?
28/2/2012 16:51 View Responses
I do not know
28/2/2012 13:48 View Responses
V8 Javascript engine
28/2/2012 12:24 View Responses
Maybe support for other js frameworks like Mootools.
28/2/2012 12:22 View Responses
Adjustable menu's for Mac apps (application menu, help menu, etc.) are my current top priority. Some extended hardware access might be nice for the future, but it's complicated to do that right & cross platform, so not a very high priority for me.
28/2/2012 12:20 View Responses
unsure, not rally used it yet!
28/2/2012 11:07 View Responses
Don't know
28/2/2012 9:27 View Responses
Html5, js
27/2/2012 17:51 View Responses
Native photo processing libraries/APIs
27/2/2012 15:43 View Responses
PHP?
27/2/2012 15:42 View Responses
Including something like Node.js would be nice, if that will boost our momentum by the Node.js community, but I mostly just use HTML/JS and don't really need new technologies.

27/2/2012 13:09View Responses

n/a

27/2/2012 11:15View Responses

svg and html5 video support

27/2/2012 7:26View Responses

Background processes, native controls, modules.

27/2/2012 4:23View Responses

full native code generation like ti mobile and submission to app stores.

27/2/2012 2:21View Responses

Native UI capabilities (a la XUL)

26/2/2012 21:38View Responses

n/a

26/2/2012 21:29View Responses

Support for nodejs modules (like File, TCP, etc)... would be a common api and open up 1000s of npm modules.

26/2/2012 20:47View Responses

Besides Chrome, not sure.

26/2/2012 20:08View Responses

More OS native features.

26/2/2012 19:54View Responses

Obfuscation, crypted source code.

26/2/2012 19:52View Responses

Not sure.

26/2/2012 19:46View Responses

I like the idea of using the javascript-based approach the more I think about it.

26/2/2012 19:43View Responses

Don't Care

26/2/2012 19:25View Responses

NodeJS integration

26/2/2012 17:55View Responses

Ti compatability

26/2/2012 15:13View Responses

nodeJS would be great. can't think of anything else. HTML5 and webkit bring a lot to the table already. maybe code obfuscation? i think it might be a sticky point for some. it won't prevent anything, just make it harder.

26/2/2012 5:52View Responses

A Simplified UI framework - any

25/2/2012 23:24View Responses

node.js

25/2/2012 23:04View Responses

Printing, media playing

25/2/2012 20:57View Responses

None.

25/2/2012 19:35View Responses

Visual Javascript (i miss creating desktop apps in vb6)

25/2/2012 18:23View Responses

- Ti Web Mobile binds

25/2/2012 17:08View Responses

webcamstreaming

25/2/2012 12:14View Responses

DirectWrite on Windows, Node.js

25/2/2012 10:57View Responses

Cloud Services (a la cocoafish)

25/2/2012 10:43View Responses

Cloud services

25/2/2012 9:33View Responses

easy treading and process management, easy zeromq support

25/2/2012 9:20View Responses

Native support for Less CSS (ie compiles to plain css as part of the dev build/test cycle). Amazing debugging in whatever IDEs are supported (how about a showcase IDE built with Tide?)

25/2/2012 8:53View Responses

Not sure I know enough to answer that question in a fruitful way.

25/2/2012 8:13View Responses

Srry, I didnt understand the question.

25/2/2012 8:07View Responses

I would like to see some proper support for modules, ensure CommonJS and but most of all rather that focusing on new things, stabilize what exists, kill the bad bugs and see that the necessary sandboxing is done to satisfy Apple else there will be not developers getting apps to the AppStore.

25/2/2012 4:31View Responses

Not sure

25/2/2012 1:02View Responses

Integration with native interfaces (menus, right click, etc.)

24/2/2012 23:13View Responses

Print support - this is huge. Printing is nearly nonexistent and was a dealbreaker for one of my projects. Modular plugins - a lot could be done to make it more extensible. A module/plugin architecture could boost this into a great product.

24/2/2012 22:45View Responses

Node.js

24/2/2012 22:44View Responses

The ability to use native components like in Mobile.

24/2/2012 22:09View Responses

PhoneGap has a nice architecture for adding extensions. I would like to see a similar architecture for Tide (my preferred name) that would allow us to have similar functionality on OSX, Windows, and Linux. Having a well-defined architecture for extensions is way more important than any specific technology.

24/2/2012 20:38View Responses

Lua require() ala node

24/2/2012 20:37View Responses

I would personally like to be able to use node.js and the associated ecosystem of libraries. Would strengthen apps to have access to all that functionality, and there are binaries for all platforms now.

24/2/2012 19:48View Responses

-js plugins (such as for ZeroConf, Menus, IO, etc) for extending the SSB's functionality -vendor specific CSS (for styling more native controls, menus, views etc.)

24/2/2012 19:30
I think node.js and possibly a good NOSQL db. Something like couchdb could be useful for offline/performance/synchronization
24/2/2012 19:25
Node
24/2/2012 19:14
Packaging to various plot-forms: windows, mac, linux, .net, java.
24/2/2012 18:53
Node js
24/2/2012 18:49
CommonJS, WebGL
24/2/2012 18:41
nodejs maybe
24/2/2012 18:40
Internationalization libraries
24/2/2012 18:37
Better os system integration.
24/2/2012 18:36
Code encryption and protection, stand alone apps.
24/2/2012 18:34

6. Would you like to see Ti Desktop become more extensible via the use of external, native modules?

		Response Percent	Response Count
Yes		85.5%	59
No		0.0%	0
Don't Care		14.5%	10

7. What do you feel are the most important goals we should try to achieve in the short term?

Ability to package apps and distribute them with minmum interruption
29/2/2012 22:31
stability across OSs
29/2/2012 21:16
management long term commitment
29/2/2012 17:14
So far, Apps crash a lot, and are nearly impossible to install on Linux. Until that is fixed, this product is not too terribly helpful.
29/2/2012 13:10
updating to latest webkit version (or Chromium!) and fixing current associated bugs and issues with the webkit implementation,. Getting a bug fixed TI release out as quickly as possible,
29/2/2012 12:28

- Fix what already you have first - Initial release under new name (with bug fixes) - Chrome Embedded?

Make it clear how important TiDesktop is.

All android mobile compatibility Ui and functionality

Fast release cycle

Go for stability, then new features.

Increase the tutorial guides etc. ...

To have a stable version with the latest technologies (V8, Chromium...)

Documentation and examples. I see myself messing with the source code just to understand how some functions work, what are their parameters and so on. This should be documented.

Let's just get the basics right first: a stable dev community, bug fixing. Let's not go crazy on features until we're sure everything in the basics is as it should be.

Product that is active and a good team to maintain it

Stable and fast core. Basic apps should run with native looking performance

...

Improving APIs and making them flexible

Minor bug fixes. I love the program already

Get good leadership and programmers involved.

Getting a solid 1.2 release out. Sandboxing support for mac app store. I am running 1.2rc6.

Replace or at least upgrade webkit

Native controls and modules

make an IDE for it or provide plugins to use ti studio itself. auto or in app update like adobe air.

Stability, APIs

Improve documentation

Tools to iterate even quicker and more help w/ app discovery.

26/2/2012 20:47View Responses

Ability to hide Source Code, and Encryption

26/2/2012 20:08View Responses

Bring some web apps to the desktop, at least like a proxy for those.

26/2/2012 19:54View Responses

Stability, better api.

26/2/2012 19:52View Responses

Same approach as Mobile to transition skillset. Support for key desktop widgets.

26/2/2012 19:46View Responses

More documentation. All the documentation available is mostly focussed on the mobile development.

26/2/2012 19:43View Responses

More native (especially UI) support.

26/2/2012 19:25View Responses

Updating Webkit to the newest version, or replacing it with Chromium, andf fixing all existing issues, to make TiDesktop as stable as possible

26/2/2012 17:55View Responses

Php support

26/2/2012 15:13View Responses

establish and get consensus around vision establish officers develop roadmap

26/2/2012 5:52View Responses

Bug Fixes, Stability, Speed, etc.

25/2/2012 23:24View Responses

Solve/migrate all the technical details that are provided by appcelerator: - Build Process - App Update system - Studio (since for now is the only way to compile/preview the apps) - of course we dont need a new IDE, just migrate the part of build/console log to the command line

25/2/2012 23:04View Responses

Working with files, such as PDFs

25/2/2012 20:57View Responses

Stabilize the 1.2 release and get it out the door with the current feature set.

25/2/2012 19:35View Responses

HTML5, Fast, Lightweight, fast in development fast in production, not learning another library n another programming languages.

25/2/2012 18:23View Responses

1- Build/Package/Statistic servers (if they will keep existing) 2- Auto updates (that never worked) 3- CommonJS stuff would be awelsome (like modules to start)

25/2/2012 17:08View Responses

don´t know

25/2/2012 12:14View Responses

Create more desktop-like UX. Better font rendering. And of course source code packing - that is the most important goal for commercial use of Titanium Desktop

25/2/2012 10:57View Responses

ability to distribute on Apple Mac App Store fix the build servers so we can publish for multiple OSes

25/2/2012 10:43View Responses

Setup strategy

packaging, threading and process management on server side languages.

1. Get a new team together that is capable of moving things forward. 2. An initial release that updates Webkit and sorts any other pressing short-term needs to keep the project alive (read: Apple sandbox). 3 Some momentum

A stable 1.2 release.

A web developer should port his/her program without any hassle.

Sandboxing, killing bugs, setting bounties to eliminate many bugs that have been around for a very long time. Ensuring the Apache licensing issues are unencumbered by problems posed by TiStudio. Ensure TiDesktop development is not left out of Kroll and SDK development and consultation. Attract developers to sustain project otherwise finished before it has begun. Get a commitment to some funding from Appcelerator and to some development resources including build server, and whatever tools they can provide.

Not sure

Solid standards compatible release.

Fix critical bugs and push out a release. I think it has been year(s) since there has been a release. Setup a nice looking one page website with all pertinent information.

Chromium, Close/resolve open issues, Better documentation, a kitchensink equivalent and a blog with tutorials

To get a structure up an running with tech lead and all that mumbojumbo

Either update to the latest version of WebKit or to the latest version of Chromium. I don't care which, but get all three platforms current, please.

Fix bugs, add require(), fix OSX menu, update to python 2.7

A solid release which delivers 1.) upgraded webkit 2.) x-platform packaging 3.) strong JavaScript API support

Actually finish the 1.2.0 release with a usable Windows version. This way at least existing commercial apps that depend on Ti Desktop have a viable option while version 2.0 is being built out and they don't have to abandon it all together.

Switch to embedded chromium and make it easy to build cross-platform. Figure out a good way to integrate Node.js

Matching the current offering of Appcelerator using non-appcelerator infrastructure
24/2/2012 19:14View Responses
If possible, packaging locally to various plot-forms.
24/2/2012 18:53View Responses
Leadership, road map
24/2/2012 18:49View Responses
Update WebKit, Get rid of other languages (simplify)
24/2/2012 18:41View Responses
- update webkit and fix major bugs - compile/hide somehow the js code - make the publishing in appstore easyer
24/2/2012 18:40View Responses
More extendible infrastructure, module development.
24/2/2012 18:37View Responses
Chrome support and all current os support

8. What do you feel are the most important goals we should try to achieve in the long term?

stability across OSs
29/2/2012 21:16View Responses
compatibility with titanium
29/2/2012 17:14View Responses
Make it work.
29/2/2012 13:10View Responses
keeping up to date. regular smaller releases rather than than infrequent larger releases.
29/2/2012 12:28View Responses
Make it a strong, independent and viable plataform
29/2/2012 12:16View Responses
All android mobile compatibility Ui and functionality
29/2/2012 9:12View Responses
Performance
28/2/2012 21:22View Responses
Ease of use for developers.
28/2/2012 16:51View Responses
More integration with remote webapps. I'm using Titanium to distribute my webapps in a way users don't need to upgrade browsers, and I was forced to bypass session/cookie problems because of an iFrame, create a new interface to download and open files (Titanium does nothing when we click or submit a form that leads to a file). So more integration would be good.
28/2/2012 12:22View Responses
bug fixes, documentation and good example.
28/2/2012 11:07View Responses
Speed and performance. Ease of use. Access to attached devices (camera etc)
28/2/2012 9:27View Responses
I just like to be able to use webtechnology and reuse my code for iPad/iPhone in a WebView with Ti.Mobile/Phonegap. Most important for me is that this project will be stable and support new OS's and keeps running on

enterprise PC's/Mac's where the users have limited OS privileges. And a packaging server which allows custom images and foreign languages in the installer would be nice.

27/2/2012 13:09View Responses

great documentation.

27/2/2012 11:15View Responses

Build on computer, not server.

27/2/2012 4:23View Responses

Native UI

26/2/2012 21:38View Responses

Improve documentation

26/2/2012 21:29View Responses

Performance, Reliability and better memory usage

26/2/2012 20:08View Responses

I chose Titanium because it covered all bases - Mobile and Desktop. Learning one speeds up the ability to use the other. Should be able to re-use code between Mobile / Desktop etc. Please do not lose this.

26/2/2012 19:46View Responses

More and more native support.

26/2/2012 19:25View Responses

Extension system - low-level JS modules

26/2/2012 17:55View Responses

Ruby/ MySQL

26/2/2012 15:13View Responses

nodejs support phasing out of server side language support

26/2/2012 5:52View Responses

modules, extensions and core/sdk mainly in javascript.

25/2/2012 23:04View Responses

Easy compiling

25/2/2012 20:57View Responses

crossplatform, fast, easy, performance, no memory issues

25/2/2012 18:23View Responses

- Maybe default CSS's that resembles to the native interface of the environment to easy of integration. (to start with) - Give it a own JS UI (like ExtJs/Touch or jQuery Mobile), to be more like the Ti Mobile would make easier to code quick and good stuff. Code the UI is always the harder part.

25/2/2012 17:08View Responses

To stabilize a good community support

25/2/2012 10:57View Responses

maintain as much compatibility with Titanium Mobile as possible so we can re-use code

25/2/2012 10:43View Responses

project continuty

25/2/2012 9:20View Responses

1. Lean, mean platform based on Webkit/Chromium and JS. 2. A super simple initial install and setup. 3. Great docs - possibly something like with php, where moderated user comments can be added to the official docs? 4. Best Practices: Structuring and layout for larger projects that provide a great starting point. (MVC, various js libs, etc)

25/2/2012 8:53View Responses

Out of the box commonJS support, better performance in Windows. I think just keeping the SDK synched across the different platforms is a challenge to be tackled with pride.

25/2/2012 8:13

Web based technologies should be compatible with this programming tools.

25/2/2012 8:07

Move forward with more JavaScript API's in a way that is consistent with mobile if possible. Set up a paid support system to help fund project.

25/2/2012 4:31

Integration with Metro for Win 8, common libraries that can be used in browser and in a desktop app. Ultimately, the ability to publish an app to multiple platforms (including web, mobile web, multiple web, etc.) that feels native with NO conditional code or multiple versions.

24/2/2012 23:13

Beat adobe air!

24/2/2012 22:45

Node,js, community-driven cloud server that packages up for win, mac, linux

24/2/2012 22:44

See point 5: Native components, personally I would very much like having the ability to use js to setup native components like in the mobile sdk instead of using html and css.

24/2/2012 22:09

Cross-platform development using HTML5, CSS3, and JavaScript, with a standardized mechanism for creating extensions that can also work cross-platform.

24/2/2012 20:38

Module/plugin architecture - keep the core as light and extensible as possible

24/2/2012 19:48

Move to chromium embedded (CEF3?) and design such that continuously linking to the newest version is easy. Build js modules and vendor specific CSS/HTML. Consider Ti Destkop a "trusted" SSB (so no sandboxing).

24/2/2012 19:30

To stay on top of developing web standards and use those instead of proprietary solutions as much as possible. To possibly even implement those standards if they do not exist in Chromium such as what B2G is working on now. After that we should think about what is really still missing and make sure there aren't major gaps in what we can do vs. other desktop apps. Also - in the rapidly changing environment where mobile is taking over, find where Tide makes the most sense. Should we target Windows 8 or not? Also consider the possibility that with mobile devices getting better, it might not be a bad thing to eventually target mobile as well, similar to phonegap.

24/2/2012 19:25

Stable release, great team/community, documentation

24/2/2012 18:49

Bootstrap from the node.js community if possible -- allow npm modules, etc

24/2/2012 18:41

reach a level where this to "matter", just like tiMobile, so the people be able to build "real" desktop apps based on html/css/js

24/2/2012 18:40

More robust, html5 javascript framework

24/2/2012 18:37View Responses

Chrome, more better apis and js lije titaniun mobile without html

24/2/2012 18:36View Responses

9. Please provide any more details regarding project direction, management, or anything at all that you want.

The only App I have played with so far, has been a struggle to use, due to Titanium problems. Does not give me a warm fuzzy feeling for the future.

29/2/2012 13:10View Responses

RERO

29/2/2012 12:16View Responses

This seems to be a great opportunity to improve a Desktop Development System. And, with the knowledge gained in learning TiDesktop, I would help us use Titanium Moble because of the similarities.

28/2/2012 16:51View Responses

Keep these things transparent and allow public to see whats going on and interact.

28/2/2012 11:07View Responses

<3

27/2/2012 15:42View Responses

Linux support can be dropped and focus on OS X, Windows support.

26/2/2012 19:25View Responses

In the first place, all existing issues should be fixed, and than we should move to major changes.

26/2/2012 17:55View Responses

this textarea is to small. :) i will post latter in the discusion group

25/2/2012 23:04View Responses

no need to be perfect, don't wait to long just publish it

25/2/2012 18:23View Responses

If it just keep going will makes me happy enough. But make it adhere to CommonJs stuff maybe would give it more credibility or at least make it more cool.

25/2/2012 17:08View Responses

Maybe it would be good to have a UI designer, who will create some basic design standarts for Titanium Desktop.

25/2/2012 10:57View Responses

Add the good bits of Adobe's AIR, such as a clean update framework.

25/2/2012 8:53View Responses

Technical writers and code example creators are probably the "easy" people to get on board. I am more worried about actual SDK developers for the different platforms. Getting those people interested to get involved is a key to success.

25/2/2012 8:13View Responses

The team should provide a release a complete package for at least PHP + Java + MySql support.

25/2/2012 8:07View Responses

The SDK's, TiDesktop and Kroll must not be encumbered by the TiStudio license. These all have to be Apache 2 licensed without any impact of clause from TiStudio which is now affecting majority of Titanium developers.

25/2/2012 4:31 View Responses

Focus on code re-usability between classic web apps and desktop apps.

24/2/2012 23:13 View Responses

I just want to be able to develop my app with sencha touch or any other HTML 5 framework and throw it into Ti desktop and it works.

24/2/2012 22:45 View Responses

I think we should start with a clean website with a Q&A specific to ti desktop only, better detailed api docs, a blog with tutorials and guides, generally build a community site and once it's built, they will come ;) Dont use the same crappy infrastructure as appcelerator does, i particularly like their Q&A site... There are plenty of open source projects out there we can use like OSQA. I can help with these kinds of things if you want like setting up wordpress, a q&a, generally getting a part of the webmaster responsibilities. Panos ;)

24/2/2012 22:44 View Responses

I'm pretty happy with the direction so far and I very much appreciate your taking on the role of project lead. Somebody's got to do it and you stepped up.

24/2/2012 20:38 View Responses

I would not try to recreate Titanium Mobile in terms of a JavaScript API for native UI. At that point, you might as well just create a new deployment target in the titanium_mobile repo. This product is about lightweight, always on, HTML-based apps on the desktop.

24/2/2012 19:48 View Responses

Lets take the best of web apps (standards-based code, ease of distributability) and merge it with the best of native apps (performance, not sandboxed) and merge them together. I envision developers building apps that run in both standard web browsers and Ti Desktop, but when the app runs in Ti Desktop, its functionality is greatly extended and of course has the appearance of being completely native.

24/2/2012 19:30 View Responses

One thing I was curious about - I know this all started because TiDesktop is getting released into the wild, but perhaps we should consider upfront how we can attract a larger audience and not just old users keeping the torch lit. Our architectural decisions might play a large role in that. Should we try to piggyback on the node community, make that core, and see where it takes us? Should we try to be compatible with Chrome Apps and suddenly fill a niche for many of those developers? How soon do we want to try to make a website. We'll probably want to use github, etc.

24/2/2012 19:25 View Responses

I can help out with documentation, marketing, tutorials etc. Sharry

24/2/2012 18:49 View Responses

Docs, lot of docs and examples are needed, even if the platform is not mature, the missing docs and examples will drive users away.

24/2/2012 18:40 View Responses

-

24/2/2012 18:36 View Responses

-

24/2/2012 18:34