FIREFOX ADD-ONS: MADE EASY

Use your Web skills to create powerful, restartless add-ons for Firefox the easy way – with Add-on Builder Beta and the Add-on SDK.

Add-ons empower millions of Firefox users to enhance and personalize their browsing experience. If you know how to develop in HTML, JavaScript and CSS, why not show off your Web smarts by creating an add-on of your own? With Add-on Builder and the Add-on SDK, we've developed a wide range of tools to get you going, from building right through to publishing your add-on and sharing it with the world.

Your Choice of Build Environment...

Add-on Builder Beta

Add-on Builder is a Web-based development environment that provides additional functionality for working with the Add-on SDK.

» Edit code in a feature-rich environment

» Test your add-ons instantly

» Use built-in version control & sharing

Add-on SDK

The Add-on SDK is set of APIs bundled with a command line utility that allows you to develop add-ons with your own tool chain.
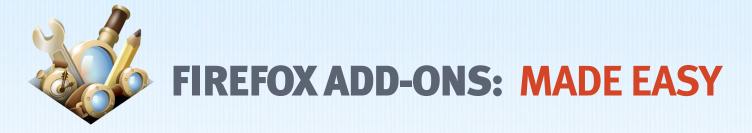
» Choose your own code editor

» Access your development files locally

» Get greater control over SDK build steps

There's Nothing To It

To make add-ons, you don't need to know your way around C++, XUL or the like. You just need some basic Web development skills in Javascript, HTML and CSS — that's it! And with Builder, you don't even need to set up a build environment. Just log in and start developing. All the hard parts are done for you.

Great Developer Features

All add-ons built with the Add-on SDK or with Builder are restartless by default. That means that users can install your add-on without needing to stop what they're doing in Firefox. Mozilla has also worked really hard to make sure that key aspects of add-on functionality are available through a library of easy-to-implement APIs. If the API you need isn't available, you can also have access to custom built APIs with other developers, too.

For more information visit:

http://addons.mozilla.org/builder

Product FAQs

What is the Add-on Builder Beta?

The Add-on Builder Beta is an online development environment which you can use to make add-ons with your web browser. It provides a code editor, a file browser, and tools to help test and package your add-on right in the browser.

What is the Add-on SDK?

The Add-on SDK is a downloadable software development kit for building Firefox add-ons in a command line interface environment. It includes high-level APIs for interacting with and modifying web pages and Firefox's user interface along with tools for creating, testing, and packaging add-ons.

What is the Jetpack SDK?

Jetpack SDK is the old name for the Add-on SDK.

What are the benefits of using the Add-on SDK and Add-on Builder Beta?

These tools lets you build Firefox add-ons using common web technologies like HTML, JavaScript and CSS, and it includes tools to make it easy to create, test, and package add-ons.

The tools are designed to produce add-ons that will be forwards-compatible with future versions of Firefox, so you won't need to update your add-on every time a new version of Firefox is released. Add-on SDK and Add-on Builder-based add-ons benefit from a security model that limits the harm that can be caused by a vulnerability in add-on code.

Finally, the Add-on SDK and Add-on Builder implement the CommonJS Packages and Modules specifications to enable a rich ecosystem of third-party APIs that developers can use to build their add-ons in addition to the APIs built into the Add-on SDK itself. We maintain a page listing some of the extra third-party APIs you can use in your add-on.

What are the benefits to users of add-ons created with the Add-on SDK and Add-on Builder?

Add-ons created with the Add-on Builder or the Add-on SDK are restartless by default, meaning users can install and remove add-ons built with these tools instantly, without interrupting their browser experience through a browser restart. This makes it easier to try add-ons and personalize their browsing experience. Developers and users alike also will not have to worry about add-on compatibility with new versions of Firefox. Finally, add-ons built with these tools will soon load in separate processes, so slow-running add-ons won't slow down Firefox itself.

For more information visit:

FIREFOX ADD-ONS: MADE EASY

Can I do everything with the Add-on SDK and Add-on Builder Beta that I could with a XUL-based add-on? Should I rewrite my XUL-based add-on now?

You can do nearly everything with an Add-on SDK and Add-on Builder based add-on that you can do with a XUL-based add-on, including:

- » modifying XUL windows
- » accessing XPCOM components
- » loading your own XPCOM components, both binary and JavaScript

You have to do some things differently: in particular, you can't use XUL overlays in Builder / SDK-based add-ons because they are not compatible with restartless add-ons. Instead, you have to use DOM manipulation.

Add-on Builder and the Add-on SDK are aimed primarily at Web developers who wish to write add-ons but are unfamiliar with the traditional XUL-based approach. So it supports a set of easy-to-use high level JavaScript APIs which address the most common use cases encountered by add-on developers. Using only the high level APIs, you can't do everything a XUL-based add-on can do.

After shipping version 1.0 of the Add-on SDK and the Add-on Builder Beta, one of our top priorities is to help developers migrate from XUL-based add-ons to Builder/SDK based add-ons, so implementing advanced add-ons will become much easier.

So if you're a XUL add-on developer, an early adopter, and are willing to participate in the development process, then now is a great time to try out the Add-on SDK and Add-on Builder and help us figure out the right set of features to enable you to make the transition.

What do I need to know to write an add-on using the Add-on Builder Beta and Add-on SDK?

You'll develop add-ons using JavaScript, and you'll interact with web content via the DOM. You'll use HTML and CSS to structure and style your own user interface components.

What are the system requirements for the Add-on SDK?

You need Python 2.5, 2.6, or 2.7, and Firefox version 5 or later. If you're running Windows, MozillaBuild, while not mandatory, will install the correct version of Python and the MSYS package, which will make it easier to work with the SDK.

Will the Add-on SDK be fully integrated into Firefox?

The Add-on SDK is a separate product from Firefox, and there are no plans to integrate its functionality directly into Firefox. However, the Add-on Builder, Add-on SDK and Firefox teams work closely to ensure Firefox provides a great set of extension capabilities for SDK APIs to expose to add-on developers.

For more information visit:
http://addons.mozilla.org/builder

FIREFOX ADD-ONS: MADE EASY

Do users need to download or install anything to use SDK-based add-ons?

No. Users only need to install the SDK-based add-ons themselves, which are all hosted in the same place as traditional XUL-based add-ons: addons.mozilla.org.

Will Mozilla stop supporting XUL-based add-ons?

Mozilla has no plans to stop supporting XUL-based add-ons, and it would be premature to start making such plans, as the vast majority of Firefox add-ons are XUL-based. If at some point in the future that ratio changes, and the vast majority of Firefox add-ons become SDK-based, then it may make sense to consider deprecating XUL-based add-ons.

What about Firefox Mobile?

The Jetpack team's current focus is creating the best possible experience for developers of Firefox add-ons for desktop. We are well aware, however, of the importance of Firefox for mobile to Mozilla and its users and we plan to prioritize Firefox add-on development for mobile in a future release of the SDK.

What is Jetpack?

Jetpack is a project to make it easy to build Firefox add-ons using common web technologies like HTML, JavaScript, and CSS. Our goal is to enable anyone who can build a web site to participate in making the Web a better place to work, communicate, and play.

I have a Google Chrome plugin/Safari add-on. Will it work as a Firefox add-on?

You will need to do some porting, but the changes should be relatively straightforward. David Morgan has written an extension for Chrome and Firefox that shares much of the code: http://code.google.com/p/scry/.