www.fotolia.de, Andrea Danti

The Aptana Free developer environment for AJAX

WEB BUILDING

Aptana is an extremely convenient GUI for AJAX development. You can use this versatile IDE as a stand-alone solution or an Eclipse plugin.

BY KRISTIAN KISSLING

Not so long ago, web programmers could use a simple text editor to develop simple HTML pages that maybe even included some Javascript or CSS. But Web 2.0, and the ever-increasing complexity of websites, have sharpened the demands placed on development tools. Aptana [1] is an Open Source, multi-platform IDE for AJAX web development. According to founder Paul Colton, "When we launched our project at the end of 2005, we asked many developers, and what we found out was that most of them used Notepad, Emacs, and Vi. In other words, nobody had a complete IDE for Web 2.0. So we went about changing that."

Some users compare Aptana with Macromedia's Dreamweaver. Aptana is an Eclipse-based IDE that you can use to create both AJAX and Aflax elements. The Aptana IDE is released under the Eclipse Public License 1.0.

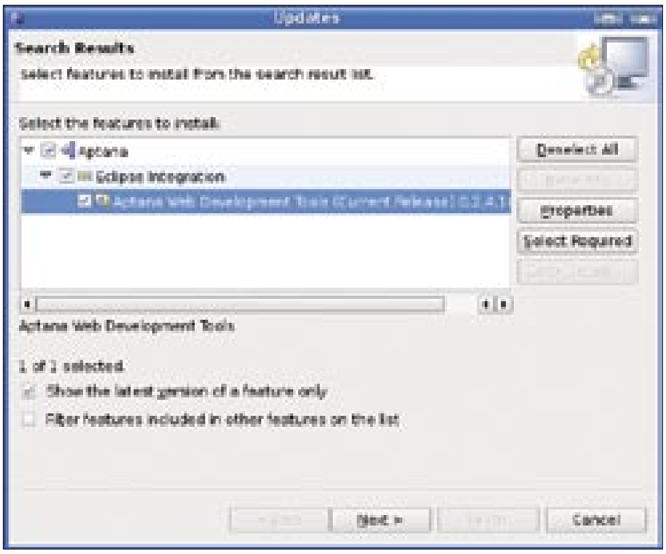
The current version of the IDE, 0.2.4, has a number of sophisticated features, in particular, Code Assist, which supports frameworks like Dojo, Mochikit, and Prototype. The integrated Scriptdoc tool makes it easier for developers to document their own code. As of this writing, the IDE is available in English only, although there are plans to add support for other languages. The team is also working on PHP and ASP support [3].

Installation

Aptana comes in either a "Current" version or a "Next" version [4]. The latter has more up-to-date features but might be unstable. Before you can launch Aptana, there are one or two manual tasks to complete.

Ubuntu users will need to run Apt to install the Mozilla software from an external package archive. To do so, become root, open */etc/apt/sources.list*, and look for the line that reads *# deb http://de.archive.ubuntu.com/ubuntu/dapper universe*. Comment out this line and the line that follows. At the end of these two lines, add the words *universe* and *multiverse*, separated by a blank. Now save the modified file.

The next step is to update the package sources: *sudo apt-get update* will take care of this. Then install Mozilla by giv-



Updates

Search Results

Select features to install from the search result list.

Select the features to install:

- ☑ Aptana
  - ☑ Eclipse Integration
    - ☑ Aptana Web Development Tools (Current Release) 0.2.4.1

Aptana Web Development Tools

1 of 1 selected

☑ Show the latest version of a feature only

□ Filter features included in other features on the list

Figure 1: If you opt to run Aptana as an Eclipse plugin on Ubuntu, Eclipse will automatically retrieve the software off the Web at the click of your mouse.

ing the *sudo apt-get install mozilla* command, before going on to set an environmental variable with the path to the Mozilla libraries:

```
export MOZILLA_FIVE_HOME=↵
/usr/lib/mozilla
```

If you can't launch Aptana, you might be missing this path. To make the path persistent, just add this line to your *~/.bashrc*. The IDE also requires Java 1.4.2, although Ubuntu has this by default.

The install on Suse is less intuitive: in fact, we were unable to convince the software to run as an Eclipse plugin (see the "Aptana as an Eclipse Plugin" box) in our lab, although the stand-alone variant was quite happy to do so. For Suse 10.0, you first need to install a Java environment, which you can download from an external package source.

To add the source, launch YaST and select the *FTP* option in *Change Installation Source | Add*. Enter *ftp3.gwdg.de* in the *Server name* line and the path in the line below: *pub/opensuse/distribution/SL-OSS-factory/inst-source*. Suse will take its time reading the package list from the new source, but when it's done, you can go on to install *java-1_4_2-gcj-compat*, *libgcj* and *mozilla* in the normal way.

See the Aptana website or the documentation for your own Linux distro for information on installing Aptana for other Linux variants.

Public Viewing

Just like Eclipse, the Aptana GUI uses views – small dynamic windows that share the available space (Figure 2). Every view has a handle with labels. Double clicking a name expands the view to fill the screen. You can hold down the left mouse button and drag the mouse to move the whole view to a different position. An arrow with a number shows you where windows are hiding other windows: if you scale up one view, the other views automatically scale down.

The Outline view, which is located on the right, shows the code elements for a page in a tree structure. Below this, you can see the Action view, which manages code snippets, as well as templates for CSS, HTML, Javascript, and XML. To discover what the Action view does, just read on. The large area in the center is the workspace, where you can write, modify, and test your source code. Clicking *Preview* in the lower left corner of the window launches the preview service and renders your page in a browser window.

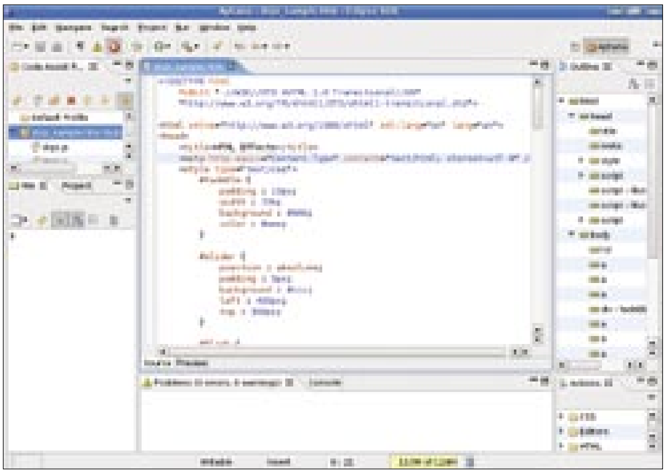


Figure 2: Views share the available space democratically. The tabs let you move these dynamic windows to different positions.

The Problem view below the main window shows script errors or invalid tags; the Scripting console is hidden in a view behind this. The integrated *File* manager, located on the lower left, can be used to browse folders for scripts and libraries and to create new Javascript and CSS files. The Project view helps you keep track of your projects. The *Code Assist Profiles* in the window above this view helps manage the project libraries, no matter whether you wrote them yourself or imported external libraries from Dojo or Prototype.

The *Walkthroughs* view appears on the right side of the screen. It dynamically loads various AJAX and Aflax tutorials off the Web, letting developers view the tutorials at the click of a mouse.

Aptana also has a fairly comprehensive help browser, which you access by selecting *Help | Help Contents* (Figure 3). The browser not only has a couple of intuitive step-by-step guides, but also the references for CSS, Javascript, HTML DOM Level 0 and 2, and the **Mochikit** index. This is a very useful collection of documents, especially for newcomers.

If a view accidentally disappears from the screen, you can restore it, or any other view, by selecting *Window | Show View | Other*. Aptana does not display some views by default, as they take up space, and Aptana handles them separately within the development process, by means of *Perspectives*. Selecting *Window | Open Perspective | Other* takes you to four different perspectives.

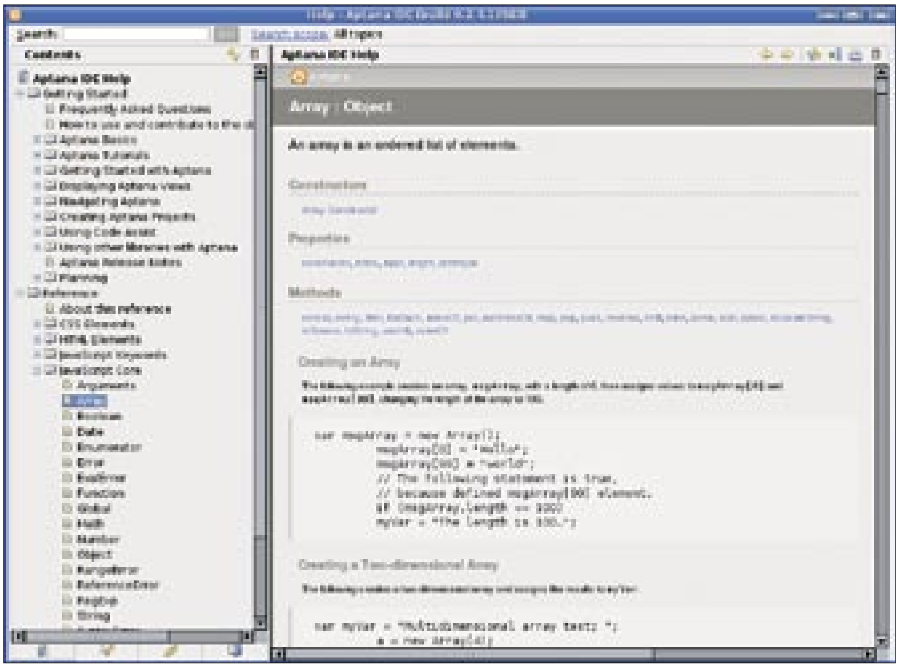
Selecting a perspective changes the appearance of the IDE, not only by reorganizing the current views, but also by displaying new views, such as the variable or breakpoint views in the *Debug*

Aptana as an Eclipse Plugin

Running Aptana as an Eclipse plugin gives you a couple of additional Eclipse features, which let you program in languages such as Java or C. Although Aptana officially supports Eclipse 3.1, it will also run with Eclipse 3.2, give or take a couple of restrictions. You can install the IDE from the Ubuntu Multiverse by giving the *sudo apt-get install eclipse eclipse-jdt* command. You do not need to install Aptana: Eclipse will automatically download Aptana off the Web.

To launch the software, just type *eclipse* on the command line. Eclipse will first prompt you to specify a path for the *Workspace*. Now close the welcome screen to access the menu. Select *Open Perspective | Other | Debug*, and then *Help | Software Updates | Find and Install*. This pops up a dialog where you can use the function *Search for new features to install*. Click the *New Remote Site* button to display an input box where you can type Aptana as the *Name:*, followed by the *URL: http://update.aptana.com/update/*.

Complete this process by clicking *Finish*, and in the dropdown menu that follows, select *Aptana Development Tools*, then *Next* (Figure 1). Accept the license, and press *Next* and *Finish*. *Install All* then tells Eclipse to install Aptana. To open Aptana, relaunch Eclipse, and after doing so, select *Window | Open Perspective | Other*, and finally, *Aptana*. Although Eclipse output a couple of error messages in our lab, the IDE still worked perfectly.



Array : Object

An array is an ordered list of elements.

Construction

Properties

Methods

Creating an Array

```
var msgArray = new Array();
        msgArray[0] = "Hello";
        msgArray[99] = "world";
        // The following statement is true,
        // because defined msgArray[99] element.
        if (msgArray.length == 100)
        myVar = "The length is 100.";
```

Creating a Two-dimensional Array

```
var myVar = "Multidimensional array test; ";
        a = new Array(4);
```

Figure 3: The help browser is useful, as it not only explains how to work with Aptana but also provides references for CSS and Javascript.

perspective (Figure 4). Developers have the freedom to set up perspectives, defining window sizes and positions. Once you have set up a perspective to your liking, you can save it by selecting *Window | Save Perspective As*.

Beautiful Things

But how can this flexible GUI really help you become a better programmer? The answer is with innumerable features such as Code Assist, outlining, code completion, syntax highlighting, and the integration of popular Javascript libraries. Just like with other IDEs, you can group your files with a project that Aptana will store in the *Workspace* folder. Projects can be exported and imported as needed.

Projects can help you keep track of your work, especially if you are involved in developing a major website. Clicking *File | New | Project* opens a new project. You can opt for a *Simple Project*, or an *AJAX Library Project*, which you will find below the *Web Library Project* entry. Enter an intuitive name, and then decide which libraries you need for your AJAX project (Figure 5).

Aptana creates a sample file as a starting point. If you select Dojo as your Javascript library, the file will be titled *dojo_sample.htm*. It appears at the bottom left of the project view, with the source code shown in the central workspace. The accompanying Javascript files, *dojo.js* and *html.js*, appear at the top left in *Code Assist Profiles*, where you can double click to edit them.

Time to write your own program: as the name suggests Code Assist will help you do so. For example, if you delete the first curly bracket next to the *#fadeElm* CSS selector in the *dojo_sample.html* file and start to type, a dialog will pop up, showing you which CSS properties the selector supports, and which of these properties are supported by popular browsers (Figure 5).

Every browser interprets the CSS specifications in a slightly different way, so this kind of information is invaluable. You can press the arrow keys to navigate the dialog. If you select an item, a tool tip is displayed next to it to explain what the element actually does. Pressing [Enter] confirms the current selection and tells Aptana to insert the element next to the cursor.

Code Assist will complete various selectors, classes, and functions in this way, assuming it has enough information on them. Of course, it will not understand any functions you have written yourself. But why not use Scriptdoc to supply missing details? Scriptdoc marks up the data from the Javascript documents in the workspace for Code Assist. For this to work, all you need to do is to document your code according to a fixed schema.

The Outline view helps you keep track of longer sections of code. Normally, tags and functions are organized hierarchically in the source code. Outline will detect these elements, and their relations, and map them (Figure 6). If you click a specific element, the cursor will jump to the corresponding position in the source code. Small icons next to the tags tell you if you are looking at a function, variable, or selector. Clicking a tag tells you what variables and functions are hiding behind it. However, Outline is only for navigation; you can't modify entries in this view.

Syntax highlighting is another Aptana feature that will help you write code. The IDE highlights various elements in the program in different colors depending on whether you are looking at a value, a comment, or just plain old text.

GLOSSARY

Mochikit: A widespread and well-documented suite of Javascript libraries, designed and tested for popular browsers such as Safari 2.0.2, Firefox 1.0.7, Firefox 1.5b2, Internet Explorer 6, and Opera 8.5.

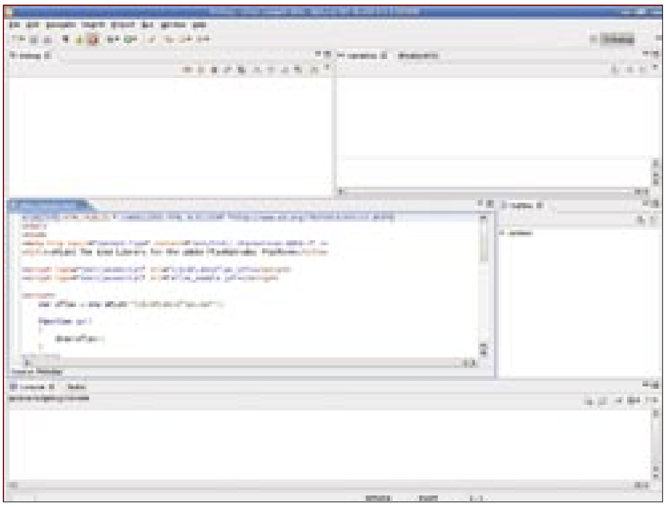
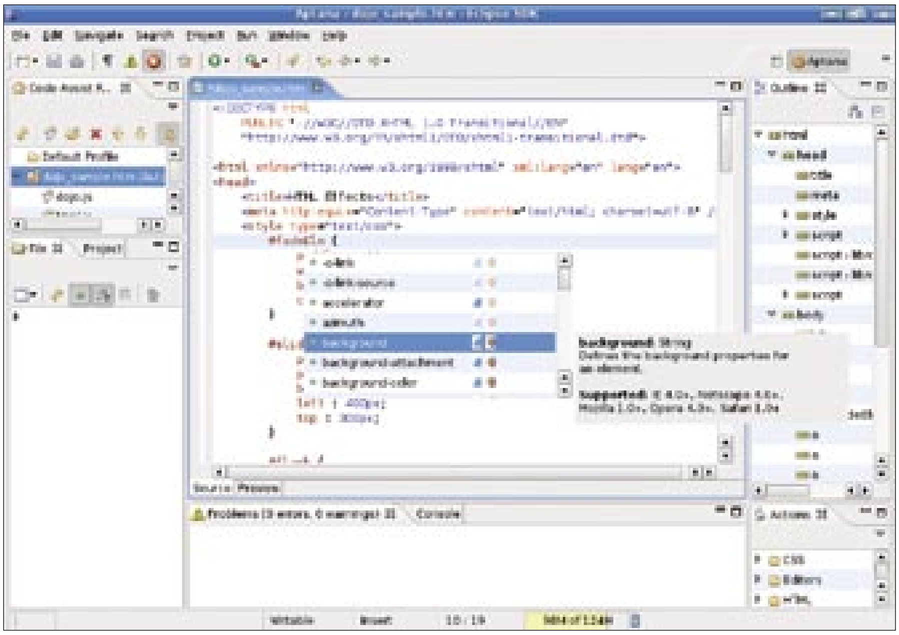


Figure 4: The perspective not only changes the way the windows are organized; it also opens new views with different functions.



```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>HTML Effects</title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" /
    <style type="text/css">
        #fadeIn {
            accelerator
            azimuth
            background
            background-attachment
            background-color
            left : 400px;
            top : 300px;
        }
```

background: String
Defines the background properties for an element.

Supported: IE 4.0+, Netscape 6.0+, Mozilla 1.0+, Opera 4.0+, Safari 1.0+

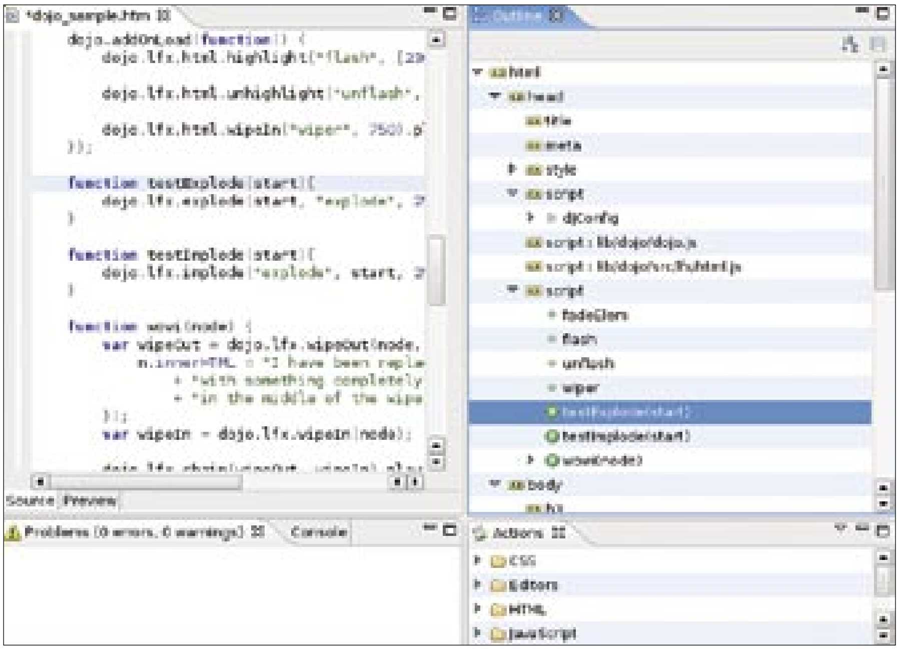
Figure 5: Code Assist shows you the properties supported by a selector and also tells you which browsers have the required support.

This gives programmers an excellent overview. You can change the colors – depending on the file type – by selecting *Window | Preferences | Aptana | Editors*.

Snippets, in the Action view, give you quick access to bits of code that you use frequently. Select a piece of text in the code, and then click the snippet you need to wrap the text in the tagging of your choice. The software has a couple of HTML templates, although the feature is actually designed for you to add your own snippets.

Aptana lets you compare two documents, or the current document with a previous version. To do so, right click the source code, and select *Compare with | Each Other*. The same menu has an entry for *Team | Apply Patch*, which lets you install patches. As the word *Team* suggests, you can store the project on a network drive to support collaborative development.

Plugins give you the ability to extend the IDE: support for Ruby On Rails is already available, and it goes by the name of RadRails. The Aptana developers are planning plugins for PHP and ASP.



```
dojo.addOnLoad(function() {
    dojo.lfx.html.highlight("flash", [23

    dojo.lfx.html.unhighlight("unflash",

    dojo.lfx.html.wipeIn("wiper", 7500).p
});

function testExplode(start){
    dojo.lfx.explode(start, "explode", 2
}

function testImplode(start){
    dojo.lfx.implode("explode", start, 2
}

function wow1(node) {
    var wipeOut = dojo.lfx.wipeOut(node,
        n.innerHTML = "I have been repla
            + "with something completely
            + "in the middle of the wipe
    });
    var wipeIn = dojo.lfx.wipeIn(node);
```

Figure 6: Outline helps you keep track of the structure of a document by grouping elements hierarchically.

Preferences

Of course, Aptana gives you a number of options for setting up the environment to your liking. Selecting *Window | Preferences* takes you to a tab where you can fine tune the environment. *Appearance* lets you modify Aptana's colors and views. The *Keys* section below *Editors* is also important, giving you the ability to define keyboard shortcuts for Aptana to enable even quicker access to frequently needed functions.

You can also assign file formats to specific editors: for example, if you don't have an editor for XUL files, you can assign them to the XML Editor. *Startup and Shutdown* let you say which plugins you want the IDE to load automatically. *Workspace* lets you set how often you want the software to create a backup copy of your data, and how many steps you would like to be able to undo.

Conclusions

Aptana is just at the start of its journey, but this free tool is already starting to pick up speed. It owes many of the interesting features to the Eclipse developers, but the way it integrates libraries, Code Assist, and other features, make Aptana a very useful tool for people who prefer not to use a text editor for their work.

Aptana is suitable for beginners and advanced programmers interested in AJAX. Performance hits in large-scale projects detract from the excellent all round impression. Aptana does not have the same feature scope as Quanta Plus as of this writing – features such as localization, FTP support, and Unicode detection are still on the project's to-do list. Despite this, Aptana scores top marks with respect to usability and its support for numerous Javascript functions. ■

INFO

[1] Aptana project homepage: *http://www.aptana.com*

[2] Aflax demos: *http://www.aflax.org/demos.htm*

[3] Aptana roadmap: *http://www.aptana.com/docs/index.php/Aptana_Roadmap*

[4] IDE download page: *http://www.aptana.com/download_linux.php#next*