

# The unforking of KDE's KHTML and Webkit

By [Troy Unrau](#) | Published 4 years ago

There is one major web rendering engine that grew entirely out of the open source world: [KHTML](#) is KDE's web renderer which was built from the ground up by the open source community with very little original corporate backing. The code was good and branches were born as a result, the best known being Webkit. Now, after years of split, KHTML and Webkit are coming together once again.

KHTML was introduced to the world as part of [KDE 2.0](#), a milestone release within the KDE community that included Konqueror, KDE's integrated KHTML-powered web browser and file manager. It was launched in a time when Netscape was failing, and the Mozilla Foundation was still struggling to build a community around the released Netscape source code. At the time, the rendering was decent for 90% of all HTML you could throw at it, but had limited support for JavaScript and other features that were emerging at the time. A good first start, though, which did not go unnoticed by the software world at large.

One of the first projects to pick up KHTML for use outside of KDE was [AtheOS](#) (now known as [Syllable](#)), which abstracted KHTML to produce their own web browser, known as [Abrowse](#). KDE did not know at the time that the interest was actually much higher. Apple took a fork of KHTML and worked privately on its own branch of it for years, abstracting the KDE and Qt specific code until they produced Webkit for the purposes of powering Safari. In that time, KHTML and Webkit grew apart. By the time Apple had launched the first public releases of Safari, its improvements had become [difficult to move back into KHTML](#). Apple also had an interesting time trying to discover how to work within the open source community within which they derived their code. Flame wars erupted and settled down. Some of the KHTML coders were granted access to Webkit's source tree and for a while, we all lived peacefully with one another.

But Apple wasn't the only interested party. Nokia took a [fork of Webkit](#) to power their mobile browsers (citing reduced memory and performance requirements, as well as clean code). They produced a fork that was adjusted to use GTK, but it was a fork nonetheless. They also have a separate implementation of Webkit that is seen in their Series 60 mobile phone, which is estimated to have a distribution of 10 million devices. Other companies have started to develop Webkit or KHTML forks internally as well, although they tend to more frequently start their fork from Webkit due to the code already having been abstracted.

On top of that, Webkit-derived browsers are starting to get a good share of the browser market either via their near monopoly of Mac surfers, or the use of Webkit on cell phones. While there were some previous efforts to make a KHTML or Webkit browser on Windows, Apple's decision to release Safari on Windows will really drive up website compatibility. Web developers are starting to pay attention to compatibility standards (thanks largely in part to the declining market for Internet Explorer) and programming for Webkit is fairly easy if you program for the web standards. However, the developers' fear of continued forking is starting to pull on some of the various involved entities, with Apple increasingly making overtures to those using Webkit to send their improvements upstream. The problem is that upstream from Apple is really KDE, which contains the original code that has not been abstracted at all.

Recently Abrowse, one of the first KHTML derivative browsers [switched over](#) to Webkit to take advantage of its rising market share and improved website compatibility. GNOME's Epiphany browser also [recently implemented](#) the ability to use Webkit as its rendering core.

Well, this is a tide that KDE is having trouble swimming against. During the Akademy 2007 developer conference in Glasgow, several KDE developers sat down to help decide a reasonable course of action with regard to Webkit and KHTML. During the keynote delivered by Trolltech's Lars Knoll, it was revealed that the company was working on bringing Webkit into Qt proper which has encouraged KDE to play its hand. The decision had to be made as to whether

to stick with the home grown KHTML, or move to Webkit, KHTML's fully grown offspring. As with many decisions in the open source community, this decision was not made by an open vote, but by consensus. In this case, the debate flared up at a Glasgow pub where the developers were free and open to speak their minds. While there are still a few reservations, the consensus is to develop a Webkit KPart for embedding into Konqueror at the earliest opportunity and to take a more active role in the development of Webkit itself. This was hinted at earlier in an Ars [interview with Lars Knoll](#), but now it is more or less the official word.

Now, KHTML won't be deleted right away since there are features in it that need to be ported into Webkit. For example, KHTML (in KDE 4) [implements](#) portions of the definition of the CSS3 standard, which will need to be adopted into Webkit and so forth. But the big deal is that the coders that invented the underlying layers that power Konqueror, some Nokia browsers, Abrowse, Safari, Adobe's [Air](#), and now Epiphany and a few other projects that are in the works, are now back in the fold. Additionally, Trolltech has announced that they are including Webkit in their upcoming Qt 4.4 release which means that a major, cross-platform toolkit now permits anyone to use the Webkit rendering engine where ever they need to render some HTML.

In open source terms, this may be as big of a deal as the gcc and egcs merger of yonder days. KHTML and Webkit are definitely coming of age. The KDE developers, responsible for the original creation of KHTML, are dedicated to seeing this unforking happen and are taking a leading role in that effort.

In related news, some Webkit developers are planning a (still unannounced) logo design competition. This should pit the best of everyone ranging from Apple's community artists to those in the open source world against each other. May the best logo win!