

1 December 2008

By: Marius Oiaga, Technology News Editor



Windows 7

[User Interface Automation Evolution in Windows 7](#)

The superlative of Vista

Microsoft is leaving no components of the Windows client "untuned" in the evolution from Windows Vista to [Windows 7](#). Loyal to the strategy of building the next Windows operating system as a superlative of Vista in all aspects, the Redmond company's overhauling efforts span from Win7's under-the-hood architecture to the GUI on the surface, and to the natural user interface interaction model. Michael Bernstein, the development lead for Accessibility and Speech Recognition experiences in Windows 7, pointed to the Microsoft User Interface Automation as the evolution of the foundation for accessibility in the next iteration of the Windows client.

"In Windows 7, we invested in improving the performance of the UI Automation system, and created a new, native-code API for UI Automation, to make sure that it can be used effectively by a wide range of assistive technology software," Bernstein [explained](#).

"Now applications written in C++, as well as those written using the .NET Framework, can take advantage of UI Automation. We also did a bunch of work to make sure that the UI Automation system was integrated even more closely with the legacy Microsoft Active Accessibility (MSAA) system, and developed new bridging techniques between the best of the new and the old technologies."

The Windows accessibility framework, intimately connected with the Windows Presentation Framework, is designed to take the manipulation of the UI beyond standard input via assistive technology products. A collection of applications programming interfaces and a set of default features, Windows Accessibility is designed to open up the operating system to as much people as possible, regardless of physical or cognitive disabilities.

"UI Automation Clients can read Accessibility information from MSAA applications, and vice versa, to ensure maximum Accessibility, regardless of which accessibility API an application used originally," Bernstein added. "Since the UI Automation and MSAA systems cooperate closely in many scenarios, we decided to name the combination of the two, calling it the Windows Automation API. This architecture forms the foundation for the rest of our Accessibility effort, and we're pleased to have this Accessibility foundation [Windows 7](#)."