Engineering Windows 7

Welcome to our blog dedicated to the engineering of Microsoft Windows 7

Follow-up: Windows Desktop Search

Steven Sinofsky 23 Oct 2008 3:00 AM | **77**

The discussion and email about desktop search offered an opportunity for us to have a deeper architectural discussion about engineering Windows 7. There were a number of comments suggesting alternate implementation methods so we thought we'd discuss another approach and the various pros and cons associated with it. It offers a good example of the engineering balance we are striving for with Windows 7. Chris McConnell wrote this follow-up. --Steven (See you at the PDC in a week!)

Thanks for all the great feedback on our first blog post on **Windows Desktop Search**. I've summarized a number of points that have been made and added some comments about the architectural choices we have made and why.

Integration with the File System

As some posters have pointed out, one possible implementation is to integrate indexing with the file system so that updating a file immediately updates the indices. Windows Desktop Search takes a different approach. There are two aspects of file system integration: knowing when a file changes and actually updating the indices before a file is considered "closed" and available. On an NTFS file system, the indexer is notified whenever a file changes. The indexer never scans the NTFS file system except during the initial index. It is on the second point—updating the indices immediately when a file is closed that we made a different choice. Updating immediately has the benefit that a file is not available until it is indexed, but it also comes with a number of potential disadvantages. We chose to decouple indexing from file system operations because it allows for more flexibility while still being almost real-time. Here are some of the benefits we see in the approach we took:

1. **Fewer resources are used.** Inverted indices are global. An inverted index maps from a word found in a property to a list of every document that contains that word. Indexing a single file requires updating an index for every single unique word found in the file. A single document might then update a very large number of individual indices. Making these changes and committing them with the same robustness found on individual files would be very expensive. The design of the indexer allows scheduling and aggregating these changes so that much less work is done overall—that means less CPU and less disk I/O. The system can be more robust because indexing doesn't only happen when a file is closed—and it can even be retried if necessary.
2. **File system operations are prioritized over indexing.** Getting files robustly updated and available is necessary for applications to use them. We don't want to delay that availability by forcing the cost of indexing into file close operations. Searching over files is important, but is less important than actually working with files. We wouldn't want applications to decide individually if the indexer should be turned on or off just because they were seeking the best performance with respect to the file system.
3. **There are lots of file types.** Microsoft supplies extractors (IFilter/IPropertyHandler) for many common file types as part of Windows. There are many other file types as well so it is important to allow non-Microsoft developers to write their own extractors. In Vista (and Windows 7), these extractors run in a locked down process that ensures that they are secure and do not affect the performance of the whole system. If indexing had to happen before a file was available, then an extractor could impact (intentionally or not) all file system operations.
4. **Some files are more valuable to index then others.** If indexing happened when a file is closed, then there is no control over the order files are indexed. Decoupling allows prioritizing indexing some files over others. For example, searching for music is much more likely than searching for binary files. If both music files and binary files have changed, then the indexer ensures it indexes the music files first. Some files are not worth indexing at all for most people. Several comments suggested that we should index the whole drive. We can do that—and for those who would find it valuable it easy to add folders to be indexed. (You can also remove them, but that is much less common so that is controlled through the control panel "Indexing Options.") For most people indexing system files is just a cost—they would never search for them and would be confused if they showed up as the result of a search.
5. **Not everything is a file in single file system.** Windows is all about supporting diversity. There are many different file systems like FAT32 and CDFS and we would like to be able to search over those as well. If we integrated with only NTFS, then we would have to still have a loosely coupled system for other file systems. Many applications also have databases optimized for their own needs. For example, Outlook has a database of email. If only files were indexed, then the email in the database could not be indexed unless Outlook either compromised their experience by using files only, or complicated their implementation by duplicating everything in both the file system and the database.

Advanced Queries

Advanced Queries

A number of people expressed frustration with the lack of an advanced query UI. Microsoft has many advanced query user-interfaces in many products, but these are generally focused on well-defined query languages (SQL) or on specific domains (like the Advanced Find in Outlook). With Vista we wanted to address the query problem in a manner more familiar to people today—a single edit control. Our implementation supports a rich query language within that edit control. This is the same approach people are familiar with for web searching for both standard and advanced queries.

We had two observations that led to this approach:

1. The most important part of a search are the search terms. Usually a single term is enough (and as we know from web searching, the majority of searches are one or two words). And for refinement the file system tools of thumbnails, sorting, and/or type ahead can be used to narrow the search.
2. It is reasonable to consider a design for an advanced query UI covering property based search, but it will generally be unwieldy for all but the bravest people. As we mentioned, Windows Search covers over 300 properties by default so if you show every property then the UI is unusable. If we only show the most commonly used properties then how do you handle all of the other properties? Would properties be grouped by the common application or by attributes such as times, names, file attributes, etc.? Some of you might value the Outlook Advanced Find… interface, but there you see some of the challenges and that is within a specific domain where the grouping or related properties probably can be understood.

In designing Vista we incorporated the feedback that it is desirable to do precise queries. The approach taken in Vista was to support a rich **query language** which allows all properties and a fairly natural syntax. For example typing "from:gerald sent:today" will find all email from "Gerald" sent today! The big issue is that people do not know or the query language. In Windows 7, we have focused on helping people see how to use the query language in context. For now, you can see the following for some information on Vista's **query syntax**. Much of this syntax and experience is similar to web search that we all use today.

A number comments were about substring matches in filenames, which we do not currently support. This is part of the overall discussion about advanced queries. In order to efficiently execute queries, the indexer builds indices that are based on individual words. In Vista we introduced "searching as you type" to our search UI. Under the hood this is implemented as prefix matches on the indexed words. So when you type, 'foo', we look for all terms that start with those letters including 'food' and 'football'. Even more interesting if you type 'foo net' we will match on items that have the words 'food' and 'network' in them. (If what you really want is to match the phrase "foo net" then typing those words inside quotes will do that—another example of advanced query syntax) We have focused primarily on searching for terms found in any property, but there is no question that filenames are special. In recognition of that we support suffix queries on filenames. If you type '*food' then we will return files that end in 'food' like "GoodFood". We do this by reversing the filename and then indexing it as a word. For example, the reverse filename of "GoodFood" would be "DooFdooG" which we index as a word. The suffix query '*food" is transformed into a prefix query "doof*" over the reverse filename index—clever, no? So we support prefix matches for all properties and suffix matches for filenames, but we do not support substring matches.

Performance and Citizenship

A number of comments focused on improving performance and citizenship—and we definitely agree on this input. We are always striving to make Windows do more with fewer resources. For those who have turned off indexing all together we hope that our continued improvements will make you reconsider. Even if you organize all of your files and don't find search useful for files, perhaps you will find start menu search, email search or Internet Explorer 8 address bar search useful. We have worked hard at improving performance and citizenship across Windows. Some of this progress is visible in WS4 and soon in Windows 7. We have improved along all of our dimensions including indexing cost, battery life, citizenship, query speed and scrolling speed. We have some tremendous tools that help us track down performance problems. If you want to help, please contact **idx-help@microsoft.com** and we will tell you how to collect performance traces we can analyze so that we can continue to make improvements.

Chris McConnell

Find and Organize

Comments

anony.muos
23 Oct 2008 4:12 AM

One difference is that for desktop search, the query syntax is much more larger and comprehensive, for web search, it is relatively smaller. Why can't a UI be built that changes the single edit box? Users who want the

edit box control can use it, those who want the UI can use that. Several other third-party tools (including real-time and indexed search tools) on the market manage to ship with a powerful UI that makes searching so much easier. You say Windows Search covers over 300 properties but when why should users remember those many properties? The Outlook Advanced Find interface is too complex and an overkill but a UI that is similar to the real-time search in Windows 2000/XP can surely be developed. (Hint: Searching by file extension/file type is most most most needed in the UI and some *simple* way to search by wildcards such as M*.??x). Also, the location box is not editable/you cannot type the path into it. I don't like going inside another dialog box from the search window to set my search location for instance.

With Vista, the same window searches indexed as well as non-indexed content, but there are times when users only and only wish to perform a real-time desktop search. The functionality that covers "Include non-indexed, hidden and system files" should be reversed to say "Include indexed files".

And then fix those minor issues such as not requiring users to click a tiny round arrow to access "Advanced Search" and keeping the "Advanced Search" expanded the next time the search window is opened if it was expanded when last closed.

Anonymuos
23 Oct 2008 4:15 AM

Not sure whether this is related to the search team, but can the search box be built into the class Start menu as well so that users truly get a choice of using what Start menu to use?

Pendragon
23 Oct 2008 4:16 AM

Thanks for following up on this important topic. It's also good to see that you are listening to our comments and concerns.

lyesmith
23 Oct 2008 4:39 AM

Asus CEO Jerry Shen says Eee will not go over 10-inch screen size and will offer Windows 7 by mid-2009

http://www.dailytech.com/ASUS+CEO+Jerry+Shen+Discusses+the+Eees+Future/article13251.htm

Thats good news! Lower sys requirements, and confirmed release date!

Domenico
23 Oct 2008 5:04 AM

Windows 7 end 2009 or 2010 I hope that the PDC is made on clarity release date.

I'm sure Windows 7 will be a historic success. Short can see Notebook and UMPC with a mini Surface for the rest of us ,

The team is working hard and..

Go Team GO!!!

http://on10.net/blogs/sarahintampa/Touch-Enabled-Eee-PCs-Are-Coming/

locolorenzo
23 Oct 2008 5:11 AM

In Win 7 I have left indexing on, it is lightning fast. I am on Server 2003... ow what a difference.

What if you add a couple of options like radio buttons for those of use that have to search for dumb files like "io.h" on some obscure network drive.

Better yet give a choice of machine use in set-up for Corporate, Programmer(Scientific) or Home User, and a preset config file or filter for each.

Thank you for not using "whereis or find" from a command window.

Also being able to give feed back on this site makes me feel a greater commitment to stay with MS.

@steven thanks for this forum

har0ld
23 Oct 2008 6:01 AM

Honestly I don't see how indexing is killing performances. I had WS4 on an old Aspire2000 (1g RAM) and well, nothing to be ashamed of when you see how it makes the flow so much better.

"In Windows 7, we have focused on helping people see how to use the query language in context"

The annoying dog is back! :)

The advanced query is something people don't want to learn even with Google on web. They will with time because with it you only need a search box for all your searches. No more Outlook search, web search etc, one box to rule them all and please, make it dockable everywhere I want (This is what WDS 2.6 was and it was COOL).

Gigaplex
23 Oct 2008 6:01 AM

Apologies for beating a dead horse, but I find the lack of substring search support appalling. Let's say I want to search for a song "I Still Haven't Found What I'm Looking For" that is in the current folder (and I know it is in the current folder) and that folder is not indexed as it is an external drive. Windows Media Player in all it's wisdom defaults to prefixing file names with the track number. I don't remember what the track number is, so I have to try the reverse order search (I only just learned about this now). However, this won't work since the file is actually named "06 I Still Haven't Found What I'm Looking For [Live].mp3" and I wasn't aware that it had "[Live]" at the end. How is someone supposed to find this file? It worked fine in XP, and works fine when indexing is turned on, why id it not supported in non-indexed searches on Vista? If there is a workaround for this, I would like to know.

Aberforth
23 Oct 2008 7:21 AM

Well, the search doesn't always give the exact results- the problem is windows doesn't have a option to tag normal files (non image), like for example if I backup my project files in a RAR archive as "vs2008bkp.rar" and I forget the name and location, I'd like to tag it as "project backups". File management is the biggest problem in windows, that's why User Group based interface is highly essential instead of standard one that suits all users.

hornetfig
23 Oct 2008 7:24 AM

> find the lack of substring search support appalling. Let's say I want to search for a song "I Still Haven't Found What I'm Looking For" that is in the current folder (and I know it is in the current folder)

I think it would be reasonable to index file names in a space-delimited fashion. That is, index each word of the file name. As for inner substring matches - I'm fine if that's not supported.

> why i[s] it [substring searching] not supported in non-indexed searches on Vista? If there is a workaround for this, I would like to know.

So would I. Probably consistently, but it kind of makes search useless.

sokolum
23 Oct 2008 7:45 AM

If the technique for indexing is so much faster, is it not possible to implement that in the NTFS files system? To me sounds like the file system is outdated.

Anonymuos
23 Oct 2008 9:35 AM

@Aberforth, I agree exactly. All we can do is hope and trust MS not to needlessly remove features from the next release. I *hate* the ability to not store metadata for any file type. And there are other quirks such as not being able to sort by a column until Vista finishes it's search, no ability unless one installs an iFilter to index plain text files having some obscure extension. If the team is indeed reading the comments, then please be sure to read the comments received at windowsvistablog.com/search/SearchResults.aspx?q=Vista+Search

ak47wong
23 Oct 2008 10:15 AM

The query language is useful but I have not been able to find a way to include ratings in a search query. For example, to find all 5-star photos taken this year, I would like to say "kind:pics taken:>1/1/08 rating:5" but the rating: property seems to be undocumented and returns unpredictable results. Am I missing something here? It seems strange to give star ratings such prominence in the metadata UI and then not provide a way to search on them.

d_e
23 Oct 2008 12:02 PM

WS4 is faster than the previous versions on XP. Great job! This is the way to go.

On Vista I never noticed any indexer-related performance problems at all. I'd be very impressed if you improved the performance further :)

Also, I agree on the design choice (Metadata updates vs. indexer). BeFS used the metadata-update-approach which lead to reduced performance of all metadata-modifying operations (file creation, ...).

One of the best things in Vista is the start-menu-integreated search functionality. It's faster for me to type "notep" than navigating trough the Programs-Menu. W7 could make use of the search feature in many more ways. The "Open file" common dialog comes to mind for example. Or when selecting desktop background images. And so on...

Another thing: Sometimes I wished I could perform a regular search on Vista. Because I'm not 100% sure how the explorer search box works. Does it search in subfolders? Does it search trough the contents? Can I just search for the filename without getting search results because the content matches? Does it search hidden files? All those things were easy to answer and do with the classic XP search (not the dog-wizard). Right now I revert to using "dir" and stuff on the command line... Or is there already a better way in Vista?

graye
23 Oct 2008 1:02 PM

I've never really liked the fact that every incarnation of the search engine fails to identify the situation where you search on a file type that is not indexed.

As it is now, if I search for a word in *.vb files (which by default is not index), I get the typical "No files found" message.   A more appropriate message would be "Sorry, that file type is not indexed".

To do otherwise, is "lying to me". I essentially have to choice but to use a 3rd-party search tool to accurately find text inside files. In fact, I rarely (if ever) use the built-in search tool, since I can't trust it's results when it says "No files found".

An even better solution would be "File type not indexed, do you want to search each file anyway?"

How about an option to NOT look inside *.zip files. From the first days of Windows, I have never had a need to look inside zip files during a search.

1 2 3 4 5 »