

APPENDIX A

Anticipation of the '552 Patent

I. EXPLANATION OF PERTINENCE AND MANNER OF APPLYING CITED PRIOR ART TO THE ASSERTED CLAIMS

A. Summary Of The Claimed Subject Matter Of The '552 Patent For Analysis of Invalidity

The asserted claims of the '552 Patent can be broken into "server-side" claims (set 1) and client-side" claims (set 2):

Set	Claims	"program" claims		"data table" claims	
		method	system	method	system
1	Generating compact difference result using invariant references (server-side)	8 9-11, 28*	21 22-24	42 43-45, 62*	55 56-58
2	Performing update using invariant references (client-side)	12 13, 29-31	25 32-34	46 47, 63-65	59 60, 66-68

* Claims 28 and 62 are multiple dependent product-by-process claims.

Since many of the asserted claims are nearly identical to one another, the analysis of all of the asserted independent claims for invalidity can be simplified into analysis of two representative claims from each set as follows:

- Set 1: Server-side claim 8 – same analysis applies to claims 21, 42, and 55
- Set 2: Client-side claim 12 – same analysis applies to claims 25, 46, and 59

The asserted dependent claims add insignificant subject matter to the independent claims and are of three types. First, there are dependent claims that call for sending the compact difference result results via a communications network or the Internet (dependent claims 9, 10, 22, 23, 32, 33, 43, 44, 63, 64, 59, 56, 57, 66, and 67). Second, there are dependent claims that call for storing the compact difference result on a storage medium (dependent claims 11, 24, 31, 34, 45, 58, 60, and 65). Finally, there are multiple dependent product-by-process claims (claims 28 and 62) that call for a processing device having a storage medium that holds the compact difference result generated by the methods of claims 8-11 or 42-45.

B. Explanation Of Cited Prior Art And Its Application With Respect To Invalidity Based On Anticipation

1. Anticipation Of Server-Side Independent Claims 8, 21, 42, And 55, And Dependent Claims 11, 24, 28, 45, 58, 62/42, And 62/45 Based On The Wetmore ‘713 Patent

For the reasons set forth below, the independent server-side claims (*i.e.*, claims 8, 21, 42, and 55) and dependent server-side claims 11, 24, 28/8, 28/11, 45, 58, 62/42, and 62/45 of the ‘552 Patent are invalid under 35 U.S.C. § 102(b) over the Wetmore ‘713 Patent. These claims are the server-side independent claims, plus dependant claims relating to storing the compact difference result on a storage medium.

As discussed above, the Wetmore ‘713 Patent meets all of the limitations recited in these claims, which is shown in detail in the claim charts below:

Set 1: Server-Side Independent Claims 8, 21, 42, and 55:

Claim Element	Claim Element	The Wetmore ‘713 Patent
8. A method for generating a compact difference result between an old executable program and a new executable program; each program including reference entries that contain reference that refer to other entries in the program; the method comprising the steps of:	21. A system for generating a compact difference result between an old executable program and a new executable program; each program including reference entries that contain reference that refer to other entries in the program; the system comprising a processing device capable of:	As discussed in more detail in each step below, the Wetmore ‘713 Patent generates a compact difference result (vector patch resource) between an old executable program (old non-vectorized program) and a new executable program (new non-vectorized program). Each program (old and new non-vectorized programs) including reference entries (“jump” entry points) that contain reference that refer to other entries in the program.
(a) generating a modified old program utilizing at least said old program;	(a) generating a modified old program utilizing at least said old program;	During vectorization, the Wetmore ‘713 Patent generates a modified old program (vectorized old program) utilizing at least the old program (the old non-vectorized program). (Wetmore ‘713 Patent col.4 ll.38-39; col.5 ll.1-3, 10-17, 18-31; FIG. 4.)
(b) generating a modified new program utilizing at	(b) generating a modified new program utilizing	During vectorization, the Wetmore ‘713 Patent generates a modified new program (vectorized new program) utilizing at least the new program (the new

Claim Element	Claim Element	The Wetmore '713 Patent
least said new program,	at least said new program,	non-vectorized program). (Wetmore '713 Patent col.4 ll.38-39; col.5 ll.1-3, 10-17; col.10 ll.6-14, col.11 ll.2-12; FIG. 4.)
<p>said modified old program and modified new program have at least the following characteristics:</p> <p>(i) substantially each reference in an entry in said old program that is different than corresponding entry in said new program due to delete/insert modifications that form part of the transition between said old program and new program are reflected as invariant references in the corresponding entries in said modified old and modified new programs;</p>	<p>said modified old program and modified new program have at least the following characteristics:</p> <p>(i) substantially each reference in an entry in said old program that is different than corresponding entry in said new program due to delete/insert modifications that form part of the transition between said old program and new program are reflected as invariant references in the corresponding entries in said modified old and modified new programs;</p>	<p>The modified old program (vectorized old program) and modified new program (vectorized new program) have at least the following characteristics:</p> <p>Substantially each reference (the location in the ROM code in the jump entry point) in an entry (in each "jump" entry point) in the old program (the old non-vectorized program) that is different than corresponding entry in said new program (the new non-vectorized program) due to delete/insert modifications that form part of the transition between the old program and new program (<i>e.g.</i>, routines in the Wetmore '713 Patent can be deleted or inserted), are reflected as invariant references (table pointers with an offset) in the corresponding entries in the modified old and modified new programs. (Wetmore '713 Patent col.5 ll.18-56; col.6 l.45 – col.8 l.52; FIGS. 3-5.)</p> <p>Invariant references: Invariant references are discussed in the '552 Patent and an example is given of providing <i>invariant references</i> by generating modified old and new programs wherein the address references in entries are replaced by label marks. Namely, the '552 Patent states that: "the invention aims at generating a modified old program and a modified new program, wherein the difference in references in corresponding entries in said new and old programs as explained above, will be reflected as <i>invariant entries</i> in the modified old and new programs. The net effect is that the <i>invariant reference entries</i> (between the modified old program and the modified new program), will not appear in the difference result, thereby reducing its size as compared to a conventional difference result obtained by using hitherto known techniques." ('552 Patent. col.3 ll.36-46 (emphasis added)). The '552 Patent also states: "Those versed in the art will readily appreciate that according to the invention, it is desired to neutralize this change, since it has occurred solely due to the fact that other entries have been affected (<i>i.e.</i>, entries 7 to 9). It is accordingly an object of the invention to give rise to a</p>

Claim Element	Claim Element	The Wetmore '713 Patent
		<p>situation where modifications of this kind will be modified to <i>invariant references</i> with the obvious consequence that they are not reflected in the difference result, thereby keeping the latter relatively compact.” (‘552 Patent col.10 ll.7-15 (emphasis added).) Further, the ‘552 Patent notes that for the particular embodiment of FIGS. 1 and 2, “the desired <i>invariant references</i> are accomplished by generating modified old and new programs wherein address references in entries are replaced by label marks as follows ...” (‘552 Patent col.10 ll.47-50 (emphasis added).) The Patentee also provided its claim construction for “invariant references” as “References that are the same. <i>See, e.g.</i>, ‘552 Patent 10:10-15.” (Edwards Decl. Exh. A.)</p> <p>Thus, the vectorization process in the Wetmore ‘713 Patent that results in vectorized programs having table pointers with an offset are “invariant references” under the ‘552 Patent’s label marks example and also under the Patent Owner’s construction, since the table pointers with offsets will be “the same” in both the vectorized old program and vectorized new program.</p>
(c) generating said compact difference result utilizing at least said modified new program and modified old program.	(c) generating said compact difference result utilizing at least said modified new program and modified old program.	The Wetmore ‘713 Patent generates the compact difference result (the vector patch resource) utilizing at least the modified new program (vectorized new program) and modified old program (vectorized old program). (Wetmore ‘713 Patent col.10 l.62 - col.11 l.12.)

Claim Element	Claim Element	The Wetmore '713 Patent
42. A method for generating a compact difference result between an old data table and a new data table; each data table including reference entries that contain reference that refer to other entries in the data table; the	55. A system for generating a compact difference result between an old data table and a new data table; each data table including reference entries that contain reference that refer to other entries in the data	<p>As discussed in more detail in each step below, the Wetmore ‘713 Patent generates a compact difference result (vector patch resource) between an old data table (old non-vectorized program) and a new data table (new non-vectorized program). Each data table (old and new non-vectorized programs) includes reference entries (“jump” entry points) that contain references that refer to other entries in the data table.</p> <p>Data table: The ‘552 Patent defines a “data table” as “a table of entries, each may have a different size.” (‘552 Patent col.2 ll.33-34.) The ‘552 Patent also explains that “a data table can be an executable</p>

Claim Element	Claim Element	The Wetmore '713 Patent
method comprising the steps of:	table; the system comprising a processing device capable of:	<p>program either as a loaded program in machine-memory or as an executable-file. In this example, entries are individual machine instructions of the program or the individual data elements used by the program.” (‘552 Patent col.2 ll.61-65.) The Patentee also provided its claim construction for “data table” as “a table of entries, where an entry is an addressable unit within the data table. An executable program is one example of a data table. <i>See, e.g.,</i> ‘552 Patent 2:33-36; 2:61-63.” (Edwards Decl. Exh. A.) Thus, the ROM based code of the Wetmore ‘713 Patent qualifies as a data table.</p> <p>Reference entry: The ‘552 Patent states in the Glossary that “Entries that include references are designated also as reference entries.” (‘552 Patent col.2 ll.46-47.) The Patentee also provided its claim construction for “reference entry” as “An addressable unit containing data that includes a reference. <i>See, e.g.,</i> ‘552 Patent 2:35-36; 2:46-47.” (Edwards Decl. Exh. A.) In Wetmore, the entry points (304) may be “jump[s] to a label somewhere else in the ROM code.” (Wetmore ‘713 Patent col.5 ll.38-39.) Under the ‘552 Patent’s definition of “reference entries,” the Wetmore ‘713 Patent’s “jump” entry points (reference entries) include references to labels somewhere else in the ROM code.</p> <p>Reference: A “reference” is defined in the ‘552 Patent to be “a part of the data appearing in an entry in the data table which is used to refer to some other entry from the same data table. A reference can be either an address or a number used to compute an address.” (‘552 Patent col.2 ll.42-45.) The Patentee also provided the identical claim construction for “reference” as “Part of the data appearing in an entry in the data table which is used to refer to some other entry from the same data table. A reference can be either an address or a number used to compute an address. <i>See, e.g.,</i> ‘552 Patent 2:42-45.” (Edwards Decl. Exh. A.) Under this definition, in the Wetmore ‘713 Patent, all of the jump entry points in the old non-vectorized program would be all of the references.</p> <p>Entry: The ‘552 Patent states in the Glossary that “a data table includes entries, each of which is an addressable unit that contains data.” (‘552 Patent</p>

Claim Element	Claim Element	The Wetmore '713 Patent
		col.2 ll.35-36.)
(a) generating a modified old data table utilizing at least said old data table;	(a) generating a modified old data table utilizing at least said old data table;	During vectorization, the Wetmore '713 Patent generates a modified old data table (vectorized old program) utilizing at least the old data table (the old non-vectorized program). (Wetmore '713 Patent col.4 ll.38-39; col.5 ll.1-3, 10-17, 18-31; FIG. 4.)
(b) generating a modified new data table utilizing at least said new data table,	(b) generating a modified new data table utilizing at least said new data table,	During vectorization, the Wetmore '713 Patent generates a modified new data table (vectorized new program) utilizing at least the new data table (the new non-vectorized program). (Wetmore '713 Patent col.4 ll.38-39; col.5 ll.1-3, 10-17; col.10 ll.6-14, col.11 ll.2-12; FIG. 4.)
said modified old data table and modified new data table have at least the following characteristics:	said modified old data table and modified new data table have at least the following characteristics:	The modified old data table (vectorized old program) and modified new data table (vectorized new program) have at least the following characteristics:
(i) substantially each reference in an entry in said old data table that is different than corresponding entry in said new data table due to delete/insert modifications that form part of the transition between said old data table and new data table are reflected as invariant references in the corresponding entries in said modified old and modified new data tables;	(i) substantially each reference in an entry in said old data table that is different than corresponding entry in said new data table due to delete/insert modifications that form part of the transition between said old data table and new data table are reflected as invariant references in the corresponding entries in said modified old and modified new data tables;	<p>Substantially each reference (the location in the ROM code in the jump entry point) in an entry (in each "jump" entry point) in the old data table (the old non-vectorized program) that is different than corresponding entry in said new data table (the new non-vectorized program) due to delete/insert modifications that form part of the transition between the old data table and new data table (<i>e.g.</i>, routines in the Wetmore '713 Patent can be deleted or inserted), are reflected as invariant references (table pointers with an offset) in the corresponding entries in the modified old and modified new data tables. (Wetmore '713 Patent col.5 ll.18-56; col.6 1.45 – col.8 1.52; FIGS. 3-5.)</p> <p>Invariant references: Invariant references are discussed in the '552 Patent and an example is given providing <i>invariant references</i> by generating modified old and new programs wherein the address references in entries are replaced by label marks. Namely, the '552 Patent states that: "the invention aims at generating a modified old program and a modified new program, wherein the difference in references in corresponding entries in said new and old programs as explained above, will be reflected as <i>invariant entries</i> in the modified old and new programs. The net effect is that the <i>invariant reference entries</i> (between the modified old program and the modified new program),</p>

Claim Element	Claim Element	The Wetmore '713 Patent
		<p>will not appear in the difference result, thereby reducing its size as compared to a conventional difference result obtained by using hitherto known techniques.” (‘552 Patent. col.3 ll.36-46 (emphasis added)). The ‘552 Patent also states: “Those versed in the art will readily appreciate that according to the invention, it is desired to neutralize this change, since it has occurred solely due to the fact that other entries have been affected (<i>i.e.</i>, entries 7 to 9). It is accordingly an object of the invention to give rise to a situation where modifications of this kind will be modified to <i>invariant references</i> with the obvious consequence that they are not reflected in the difference result, thereby keeping the latter relatively compact.” (‘552 Patent. col.10 ll.7-15.) Further, the ‘552 Patent notes that for the particular embodiment of FIGS. 1 and 2, “the desired <i>invariant references</i> are accomplished by generating modified old and new programs wherein address references in entries are replaced by label marks as follows” (‘552 Patent. col.10 ll.47-50.) The Patentee also provided its claim construction for “invariant references” as “References that are the same. <i>See, e.g.</i>, ‘552 Patent 10:10-15.” (Edwards Decl. Exh. A.)</p> <p>Thus, the vectorization process in the Wetmore ‘713 Patent that results in vectorized programs (data table) having table pointers with offsets are “invariant references” under the ‘552 Patent’s label marks example and also under the Patent Owner’s construction, since the table pointers with offset will be “the same” in both the vectorized old program and vectorized new program.</p>
(c) generating said compact difference result utilizing at least said modified new data table and modified old data table.	(c) generating said compact difference result utilizing at least said modified new data table and modified old data table.	The Wetmore ‘713 Patent generates the compact difference result (the vector patch resource) utilizing at least the modified new data table (vectorized new program) and modified old data table (vectorized old program). (Wetmore ‘713 Patent col.10 l.62 - col.11 l.12.)

Server-Side Dependent Claims 11 and 24:

Claim Element	The Wetmore ‘713 Patent
11. The method of claim 8, further comprising the step of: (d) storing said	To perform patching, the Wetmore ‘713 Patent explains that the client is provided

Claim Element	The Wetmore '713 Patent
compact difference result on a storage medium.	with a loading routine via the system disk that loads the proper vector resource patches following a version check. (Wetmore '713 Patent col.11 ll.34-40.) Thus, the compact difference result (vector patch resource) is stored on a storage medium (system disk).
24. The system of claim 21, wherein said processor is further capable of storing said compact difference result on a storage medium.	

Server-Side Dependent Claims 28/8 and 28/11:

Claim Element	The Wetmore '713 Patent
28 [28/8, 28/11]. A processing device having associated therewith a storage medium which holds compact difference result data that was generated by the method of anyone of claims 8 to 11.	The Wetmore '713 Patent provides a processing device having associated therewith a storage medium (system disk) which holds compact difference result data (vector patch resource) that was generated by the method of claims 8 and 11.

Server-Side Dependent Claims 45 and 58:

Claim Element	The Wetmore '713 Patent
45. The method of claim 42, further comprising the step of: (d) storing said compact difference result on a storage medium.	To perform patching, the Wetmore '713 Patent explains that the client is provided with a loading routine via the system disk that loads the proper vector resource patches following a version check. (Wetmore '713 Patent col.11 ll.34-40.) Thus, the compact difference result (vector patch resource) is stored on a storage medium (system disk).
58. The system of claim 55, wherein said processor is further capable of storing said compact difference result on a storage medium.	

Server-Side Dependent Claims 62/42 and 62/45:

Claim Element	The Wetmore '713 Patent
62 [62/42, 62/45]. A processing device having associated therewith a storage medium which holds compact difference result data that was generated by the method of anyone of claims 42 to 45.	The Wetmore '713 Patent provides a processing device having associated therewith a storage medium (system disk) which holds compact difference result data (vector patch resource) that was generated by the method of claims 42-45.

Accordingly, each and every limitation the server-side independent claims (*i.e.*, claims 8, 21, 42, and 55) and server-side dependent claims 11, 24, 28/8, 28/11, 45, 58, 62/42, and 62/45 of the '552 Patent is met by the Wetmore '713 Patent, and therefore, these claims are anticipated by the Wetmore '713 Patent under 35 U.S.C. § 102(b).