

## **Annex 7**

### **Microsoft Responses to Documentation Criticisms**

---

#### **I. INTRODUCTION**

1. My name is Thomas Pfenning. I am a Software Architect in the Windows Division, and I have had this position since August 2004. I am assigned to provide architectural support for the technical documentation that Microsoft was ordered to submit pursuant to Article 5 of the 2004 Decision (the “Technical Documentation”). I received my Ph.D. in computer science in 1995 from the University of Cologne. Since that time, I have been a program manager for numerous server and networking products at Microsoft. The following set of responses to documentation criticisms is based on my background experience, my experience with the documentation, and information I received in the course of my involvement in the documentation program.

2. Microsoft provided a preliminary version of the WSPP Technical Documentation on 11 December 2004, and then an updated version on 8 August 2005. On 11 November and 23 November 2005, Microsoft provided substantial new material in response to the Commission’s request, resulting in a further updated version that was forwarded to the Trustee. On 15 December 2005, as Microsoft had advised the Commission it would do, Microsoft made available another updated version of the Technical Documentation that it had further enhanced both on its own initiative and in response to comments received.<sup>1</sup>

3. Relative to the November 2005 release of the documentation, the December 2005 release provided clarified instructions for use and the suggested order of reading, improved consistency in the use of terminology in references to topics, the addition of hyperlinks between topics to make the documentation more easily navigable, integration of the additional documentation

---

<sup>1</sup> On 15 December 2005, the documentation was available in Redmond under substantially the same conditions as it is now available: potential licensees would have been able to conduct an evaluation of the documentation at Microsoft’s facilities that day, and they would have been able to request that Microsoft start that day creating a rights-managed version for shipment outside of Microsoft’s facilities.

provided in November into the previously-existing documentation to enable more effective searching and ease of use, and further reviews to correct typographical and spelling errors.

#### **A. Trustee Reports**

4. The Trustee produced two reports criticizing the November 2005 release of the Technical Documentation, a Preliminary Report on 30 November 2005 (“Trustee’s 1st Report”) and a Second Report on 15 December 2005 (“Trustee’s 2d Report”), as well as a document on 16 December 2005 commenting on Microsoft’s response to the Preliminary Report.

5. The Trustee’s reports on which the Statement of Objections relies are based entirely on the documentation as of 23 November 2005. The Commission issued the Statement of Objections dated 21 December 2005 without obtaining any report from the Trustee on the adequacy of the 15 December 2005 documentation.

#### **B. OTR and Competitor Reports**

6. On the basis of the Technical Documentation of 11 December 2004, OTR produced an initial report on 11 June 2005 (“First OTR Report”), criticizing the usability of the documentation and the level of explanatory detail. Then, on the basis of updated Technical Documentation of 8 August 2005, OTR produced a second report on 28 September 2005 (“Second OTR Report”), also responding to Microsoft’s comments on the First OTR Report, dated 8 July 2005. Four competitor companies, IBM, Sun Microsystems, Oracle, and Novell, conducted three-day evaluations of the Technical Documentation of 8 August 2005 and submitted reports to the Commission.

7. Many of the comments made by OTR and the competitor evaluators are outdated, as these reports are based on the documentation as of 8 August 2005, and fail to account for the substantial additions to the documentation in November 2005, as well as the improvements made prior to the December 2005 release of the documentation.

8. The Commission’s suggestion that only the DRS, FRS, and DFS protocols changed between August and December 2005 is incorrect. Because the documentation is constantly being revised, and in particular because some of the documentation is also revised in response to comments received under the MCPP program, other protocols were also revised during that

period. In all, as recorded in the “What’s New” sections of the documentation, 19 out of the 52 WSPP protocols received documentation revisions between August and December 2005.

### **C. Recurring Misunderstandings**

9. Many of the specific criticisms are based on a few central misunderstandings. First, the Trustee and others fail to recognize that the documentation is intended for use by developers with skill in the art of developing interoperable work group server operating system products and familiarity with the products with which such products would be designed to interoperate. One of the Trustee’s primary criticisms of the Technical Documentation, as set forth in the Trustee’s Preliminary Report, is that “it assume[s] prior knowledge of the Microsoft environment.”<sup>2</sup> This criticism entails the proposition that it should be possible to design a successful software implementation of complex server operating system functionalities without any experience developing software in the relevant network and programming environment. No competitor would attempt such an effort, and if it did, no amount or quality of documentation could make the effort a success.

10. The Trustee explicitly recognizes that “developing work group server operating system products able to viably compete with Microsoft’s own products” is the goal of the documentation,<sup>3</sup> but then criticizes the documentation for not including information that the developers of such products would already know or would know how to easily find. It is notable that while the Trustee criticizes Microsoft for failing to explain certain concepts, such as “context handle,” none of the competitors that reviewed the documentation made similar criticisms, despite the fact that, for example, “[‘context handle’] appears 87 times” in the documentation.<sup>4</sup> Those competitors were almost certainly either already familiar with terms such as “context handle” or easily found the extensive discussion of such terms in the freely available and searchable Microsoft Developer Network (“MSDN”) library.

11. The Trustee and others also fail to recognize that the Technical Documentation does not and should not provide complete documentation for an entire operating system product. Rather, it is designed to document those protocols needed to implement an interoperable work group

---

<sup>2</sup> Trustee’s 1st Report p. 1.

<sup>3</sup> Trustee’s 2d Report p. 5.

<sup>4</sup> Trustee’s 2d Report p. 3.

server operating system product. Thus, some details, such as the format of particular data structures, are not described in the documentation because the operation of the protocol does not depend on those details. In such cases, to specify the details would be to impose unnecessary constraints on licensees, who would otherwise be free to make different choices in these regards without affecting the successful execution of the protocols.

12. Finally, the Trustee and others fail to recognize that, given the magnitude of the documentation and the fact that it was created after the development of the operating system itself, additional information in the documentation can only be developed through the actual use of the documentation to build real products.

13. Although Microsoft has made an effort to identify and document the interactions among the different parts of the system that are likely to occur in actual operation, unexpected issues may arise as happens in any software development process. The only realistic way to address these issues—and the way such issues are always addressed in normal software development practice—is through a process of ongoing correction and support.

## **II. RESPONSES TO SPECIFIC CRITICISMS**

14. The following sections respond to the more prominent and specific criticisms of the Technical Documentation made by the Trustee, OTR, or the competitor evaluators.

### **A. Protocol Interrelationships & Server Rules**

15. OTR's assertion that server rules have not been sufficiently documented is outdated. The additional material added in November 2005 substantially augmented the existing information on server rules. For example, this material includes the topology algorithms that Microsoft developed and implemented solely to determine the optimal way to match servers to each other for the purpose of replicating Active Directory information and files. Such algorithms are executed entirely on a single system and involve no communication with other systems. The result of the algorithms, however, is the use of replication protocols to replicate information between two matched servers. Even before November 2005, the documentation described such replication protocols. The addition of the topology algorithms added information that relates these replication protocols to server rules.

16. The Trustee's and OTR's similar claims that the documentation fails to describe sufficiently the interrelationships among the protocols are unsubstantiated. Much information about the relationships among the protocols is information that would be within the knowledge of a developer of interoperable work group server operating systems, or that such a developer could readily find in publicly available materials. Annex 5, Report of Roy Hirst ("Hirst Report"), provides an example to show how a licensee might determine the appropriate combination of protocols to use for a specific task. As a practical matter, other protocol interrelationship information can and should be developed through licensee implementation experience, including through testing of combinations of Windows and licensee systems. Such testing is entirely consistent with industry practice.

### **B. Behaviours and Dependencies**

17. The Trustee's claim that the documentation lacks sufficient explanations on behaviours and dependencies is unsubstantiated. The documentation is designed to describe those behaviours and dependencies that would not be clear to licensees with knowledge of the relevant art and that are known to affect the correct execution of the WSPP protocols. However, in protocols of this complexity, it is not possible to uncover all behaviours and dependencies from the start. Rather, this information must be developed through licensee implementation experience. It is entirely consistent with industry practice for licensees to discover more complex or unexpected interactions by testing combinations of Windows and licensee systems.

18. In particular, the documentation does describe asynchronous events; more information about such events will be added to the documentation as it is developed through licensee implementation experience. For an example of an event trigger, see the DFS client-to-server documentation: "[ ]"

### **C. Error Handling**

19. In claiming that the documentation lacks sufficient explanations on error handling, the Trustee misunderstands the proper relationship between error handling and protocol execution. In general, while a protocol should define the errors that can be generated, the protocol does not define how those errors should be handled. Error handling is left to each individual implementation, and different implementations can handle errors in different ways without affecting the correctness of the protocol execution.

20. This approach is confirmed by Clements et al.: “Making a strong distinction between detecting an error condition and handling it provides greater flexibility in taking corrective action. The right place to fix a problem raised by a resource is usually the actor that invoked it, not in the resource itself. The resource detects the problem; the actor handles it.”<sup>5</sup> In this case, the resource at issue is a Windows client or server and the actor at issue is a licensee implementation. Thus, while the Windows system should report errors as defined in the protocol, the licensee implementation should remain free to handle the error as it sees fit. The omission of an error handling requirement in the protocols for interacting with Windows systems properly avoids imposing extra constraints on error handling by licensee implementations.

21. In those situations in which the protocol does dictate that errors be handled in a particular manner and this error handling is non-obvious, error handling information is described in the documentation, possibly as a server rule. For an example of a protocol that does incorporate error handling information, see the topic “Microsoft Distributed File System Protocol\DFS.”

22. Licensees do need to know whether the protocol itself should continue after an error is generated. Although this information is not explicitly stated in the documentation, the default behavior will generally be clear to licensees skilled in the relevant art. In most cases, unless the protocol describes a mechanism for error recovery, an error defined by a protocol in the documentation requires that the protocol terminate and restart.

23. In the Preliminary Report, the Trustee implies that there is no contextual information to relate error definitions to the situations in which they arise. This implication is incorrect, and indeed, in his Sufficiency Test report, the Trustee notes that the documentation for specific protocols contain sections that list the most common error return values. These sections provide the context necessary for licensees to implement the appropriate error handling mechanisms.

24. The Trustee expresses frustration with the large size of the file that lists possible Windows error codes. It is simply the case, however, that in the Windows environment, virtually any of these codes are available to a Windows system and could theoretically be returned by a Windows system. As would be understood by licensees, however, it is not necessary to handle each such error separately. After providing for those errors most relevant to a protocol, as

---

<sup>5</sup> Paul Clements et al., Documenting Software Architectures p. 234 (2003).

described in the protocol documentation, the remaining errors can be handled through a default mechanism, as for example by passing the code back to another function or system.

#### **D. Use of Void Pointers**

25. In claiming that the documentation “quite explicitly obfuscates some important details”<sup>6</sup> by using void pointers, the Trustee misunderstands the role of void pointers in the Technical Documentation. The Trustee asserts that a void pointer is “used where the *programmer* wishes to hide detail of implementation of particular data structures,”<sup>7</sup> but he is wrong to conclude from this that the *documentation* uses void pointers to withhold necessary information from licensees. On the contrary, the void pointer indicates that licensees are free to implement the relevant data structure entirely as they see fit and that the protocol imposes no restrictions on the format of the data structure.

26. A pointer is a reference to a programming object. A void pointer is a reference to an object whose format is unknown to the holder of the void pointer. The object, or data structure, must at some point have been created, and the function that created the object will know its format. However, the creating function can pass a void pointer to other functions to allow those functions to refer to the object, while essentially denying access to the object by withholding information about the format of the object from those subsequent functions. Generally this is done where subsequent functions need only pass the reference back to the creating function.

27. In the context of protocol implementation, a licensee may need to use a void pointer in one of two ways: either a protocol requires a licensee implementation to receive a void pointer or it requires the implementation to generate one. If the implementation is required to receive a void pointer, then by definition, it should not modify the object pointed to, and hence does not need to know about the format of that object. On the other hand, if the implementation is required to generate a void pointer, then the receiving system, whether a Microsoft implementation or not, should not base its actions on the format or content of the object pointed to. Thus, while the licensee implementation in this case does need to know the format of the object in order to create it, the licensee’s choice here makes no difference to the other party to the protocol transaction. The licensee is free to make any choice about the format of the data

---

<sup>6</sup> Trustee’s 1st Report p. 3.

<sup>7</sup> Trustee’s 1st Report p. 3 n.1 (emphasis added).

structure, and the protocol will continue to function as intended. In neither case does the licensee need further data structure information from Microsoft in order to implement the protocol.

28. In the Second Report, the Trustee gives a specific example of the use of a void pointer: “typedef void\* SAMPR\_HANDLE.” As suggested by the variable name, and as explained within the documentation, this void pointer is a context handle, a common and proper situation in which to use a void pointer. As further explained below, a context handle is a reference to state information stored by a server for a specific client. This reference is passed to the client, so that the client can pass it back to refer to the stored state information. Because the state information is stored on the server, the client cannot access this information and needs to know nothing about the format of this state information. Thus, a context handle is declared to be a void pointer to indicate that the client can only pass the pointer back to the server and cannot assume anything about the format of the state information held by the server. Conversely, the server can choose any format for this information as convenient, and the client will be oblivious to this choice. Other documentation discussing context handles also notes that a context handle should be a void pointer.<sup>8</sup>

29. The Trustee comments that, “There are details missing on the definition of certain important structures that appear to be of importance,”<sup>9</sup> and similarly, “there are elements of the necessary data structures which are not defined in the detail that would be necessary to allow a programmer confidently to implement them.”<sup>10</sup> To the extent that these comments are based on the use of void pointers in the documentation, they are not valid criticisms, for the reasons described above.

### **E. Explanation of Concepts & Definitions**

30. In faulting the Technical Documentation for containing unexplained concepts and undefined terms, the Trustee misunderstands the intended audience of the documentation. The Technical Documentation presumes that a licensee developer would be skilled in the art of developing interoperable work group server operating systems and that such a developer would

---

<sup>8</sup> See [http://msdn.microsoft.com/library/en-us/rpc/rpc/interface\\_development\\_using\\_context\\_handles.asp](http://msdn.microsoft.com/library/en-us/rpc/rpc/interface_development_using_context_handles.asp) and [http://www.opengroup.org/onlinepubs/9629399/chap4.htm#tagcjh\\_08\\_02\\_16\\_06](http://www.opengroup.org/onlinepubs/9629399/chap4.htm#tagcjh_08_02_16_06).

<sup>9</sup> Letter of 16 December 2005 from Prof. Neil Barrett to Cecilio Madero.

<sup>10</sup> Letter of 16 December 2005 from Prof. Neil Barrett to Cecilio Madero.

either already understand concepts common in such development or would have ready access to appropriate reference and teaching material.

31. The documentation is also designed to provide links to external conceptual material that is both relevant and useful. More such links can be added to the documentation over time to reflect the actual implementation needs of licensees. A developer familiar with this external conceptual material and possessing knowledge of the relevant art should understand concepts and terms otherwise unexplained in the Technical Documentation. Any remaining issues would be easily resolved through a process of ongoing correction and support.

32. The Technical Documentation does not presume that the knowledge of one skilled in the relevant art is limited to knowledge about public standards. To the extent that concepts specific to Microsoft systems are generally known in the art, these concepts are also presumed to be part of the knowledge that licensees will either have or will have accessible. As with other background knowledge, the documentation is designed to provide links to conceptual material covering these Microsoft-specific concepts, generally through links to Microsoft developer websites such as the Microsoft Developer Network (MSDN). More such links will be added to the documentation over time to reflect the actual implementation needs of licensees, which are not known to Microsoft at this stage.

33. The documentation builds on a base of publicly accessible Microsoft-specific concepts. The Trustee comments that “the programmer would need to be knowledgeable in Microsoft server details and this I believe should not be an assumption within documentation intended for use by non-Microsoft programmers.”<sup>11</sup> The Technical Documentation is intended for use by non-Microsoft programmers in the sense of programmers who are not insiders and who have no proprietary knowledge about Microsoft systems. The documentation assumes no knowledge about Microsoft systems other than what is publicly accessible. The Technical Documentation is not intended, however, for use by individuals who have no experience with and no knowledge of Microsoft programming concepts. To assume otherwise would require an unwieldy and unusable expansion of the documentation.

34. The Trustee notes that the term “network objects” is used in the introductory material for the Active Directory documentation, but is not expressly defined. The Trustee also notes,

---

<sup>11</sup> Letter of 16 December 2005 from Prof. Neil Barrett to Cecilio Madero.

however, that the term “appears only in the introductory section on page 1 of the document and is never repeated.”<sup>12</sup> From this, one ought to naturally (and correctly) conclude that the term is not intended to be a defined term and that the precise definition of the term has no bearing on the proper implementation of the Active Directory protocols. The term is used merely to refer to Active Directory objects that are stored in the directory. Precise details about those objects are given in the remainder of the documentation.

35. The Trustee also notes that the term “context handle” is used throughout the documentation, but is not expressly defined. The use of context handles is a fundamental client-server programming concept that is well within the knowledge of a developer of interoperable work group server operating systems and is not specific to the Windows environment. Moreover, the concept is well documented in publicly available material, both material produced by Microsoft<sup>13</sup> and material produced outside Microsoft.<sup>14</sup> As explained in that material, a context handle provides a mechanism for “keeping state associated with a given client on a server.”<sup>15</sup> A context handle is essentially a reference to a specific location on the server where the server stores state information for a specific client. This reference is initially generated by the server and then passed to the client, which passes the reference back whenever it wishes to refer to the stored state information. In this manner, the context handle is both “used for communication between a client and server” and “used for holding state information.”<sup>16</sup> The Trustee creates a false dichotomy by asking whether “the ‘context handle’ might be some communication *or* some storage mechanism?”<sup>17</sup>

36. Thus, contrary to the Trustee’s claims, information about “how big the ‘context handle’ should be” is not necessary. As explained above, the context handle is simply a pointer to a server structure that maintains client state information. Therefore, the size of the context handle itself is simply the size of a pointer on the system. The size of the object to which the context handle points is entirely up to the server implementation. The documentation does not recite a size for the server object because a specific size (or format) is not necessary for the proper

---

<sup>12</sup> Trustee’s 2d Report p. 3.

<sup>13</sup> [http://msdn.microsoft.com/library/en-us/rpc/rpc/context\\_handles.asp](http://msdn.microsoft.com/library/en-us/rpc/rpc/context_handles.asp).

<sup>14</sup> [http://www.opengroup.org/onlinepubs/9629399/chap6.htm#tagcjh\\_11\\_01\\_06](http://www.opengroup.org/onlinepubs/9629399/chap6.htm#tagcjh_11_01_06).

<sup>15</sup> [http://msdn.microsoft.com/library/en-us/rpc/rpc/context\\_handles.asp](http://msdn.microsoft.com/library/en-us/rpc/rpc/context_handles.asp).

<sup>16</sup> Trustee’s 2d Report p. 3.

<sup>17</sup> Trustee’s 2d Report p. 3 (emphasis in original).

execution of the protocols, and specifying a size would unnecessarily constrain the licensee's choices. The Trustee is ultimately correct that, if he did not already understand the concept, he would need "to search external material outside of the documentation itself in order to discover what a context handle is,"<sup>18</sup> but there is no reason to incorporate such external material into the documentation because the concept is extensively documented in external materials and is well within the knowledge of one skilled in the relevant art.

37. Another piece of information that the Trustee claims is necessary is explanatory information on the "ntstatus.h" file. Developers of interoperable work group server operating systems, however, would recognize that this file contains a list of possible return codes, known as NTSTATUS codes. In the documentation, RPC methods that use these return codes reference the ntstatus.h file in their "Return Values" section, also listing the most common codes for that context. Developers would also know that the file itself is downloadable from MSDN as part of the publicly available Platform SDK.

#### **F. Index**

38. The Trustee's claim that the documentation does not contain a "complete index" appears to be based on the Trustee's inability to locate particular terms and concepts, such as "context handle," in the index. The index, however, is designed to be useful to a developer skilled in the relevant art, in order to allow such a developer to quickly locate proprietary disclosures relevant to the developer's task. The index is not designed to allow someone without the relevant knowledge to learn the necessary concepts from the documentation itself. Indexing such background concepts would be unlikely to help licensees, and might well be distracting to them. Any remaining instances of concepts not indexed can be easily addressed through a process of ongoing correction and support.

#### **G. Access to the Microsoft Environment**

39. OTR is incorrect in suggesting that the documentation presupposes access to Microsoft APIs. Use of the documentation does not require access to Microsoft-specific programming libraries or APIs, and every effort has been made to explain all programming methods, or functions, used by the protocols. OTR may be referring to methods that are noted as never called

---

<sup>18</sup> Trustee's 2d Report p. 3.

remotely, and hence are undocumented. A programming method, or function, can be invoked either by a function on the same system, that is, locally, or a function on another system, that is, remotely. Methods that are invoked remotely must be documented, so that a licensee implementation would understand how to respond to a request to invoke that method. Methods that a Microsoft server only invokes locally, however, need not be documented, because a licensee implementation would never receive a request to invoke such a method. In the latter case, the Microsoft server is using the method only for its own internal calculations, which licensees are free to do in other ways, using methods of their own choice.

#### **H. Explanatory Memorandums**

40. The Trustee's claim that the documentation contains insufficient explanatory memorandums is unsubstantiated. The Trustee claims that "the simplest concepts have been explained at great length, whilst the more complex topics have been treated -- in the best cases -- more cursorily."<sup>19</sup> It is difficult to understand how the Trustee could have arrived at this conclusion, given that further explanatory material was specifically provided in November 2005 for the DRS, FRS, and DFS protocols, generally understood to be the most complex of the WSPP protocols. Thus, while it may be that "the content and quality of the explanatory documents differ widely from one protocol to another,"<sup>20</sup> any such difference is designed to accurately reflect the difference in the complexity of the protocols. Further explanatory information can be added to the documentation over time to reflect the actual implementation needs of licensees.

41. OTR gives as an example the explanation of the BackupKey interface, referring to a figure in Microsoft's response to the First OTR Report. However, while a "description of how the backup key information is used by either server involved implementing the DRAPI (sic)"<sup>21</sup> is not visible in the screen shot in that particular figure, such a description does exist further down the page in the documentation. It begins, "The BackupKey interface has three purposes: To back up master keys; To restore previously backed-up master keys; To retrieve the domain backup public key . . ."

---

<sup>19</sup> Trustee's 1st Report p. 3.

<sup>20</sup> Trustee's 1st Report p. 3.

<sup>21</sup> Second OTR Report p. 5.

### **I. Requirements**

42. The Trustee misunderstands the meaning of the “Requirements” section in claiming that the description of requirements is insufficient. As the Trustee notes, the Requirements section lists those Windows clients and servers that support a particular protocol. This permits a licensee to determine not only which protocols other Windows systems on the network support, but also which protocols a licensee implementation must support in order to emulate a particular version of Windows server. The Trustee faults the section for not containing “a more comprehensive list of the features that the client expects the server to provide.”<sup>22</sup> Such a list, however, is precisely the role of the remainder of the documentation, which lays out in detail the expected functions, return values, and so forth. Repeating such information in the Requirements section would be redundant.

### **J. Relevance of Links**

43. The Trustee’s concerns regarding the relevance of links to external material appear to have been based on a misunderstanding about the structure of the documentation. The Trustee points to two links to information about RPC that have nothing to do with the file “ntstatus.h.” The Trustee appears to have mistakenly combined the “Return Codes” section of the RPC method pages, under which the link to “ntstatus.h” appears, with the “Remarks” section of those same pages, under which the two reference links appear. The “Remarks” section is designed to provide contextual information for the RPC method pages as a whole, and not specifically for the “Return Codes” section of those pages. Such misunderstandings would of course be easily addressed through a process of ongoing correction and support.

### **K. Protocol State Diagrams & Message Sequencing**

44. OTR is incorrect in suggesting that the documentation lacks a sufficient description of protocol state and message sequencing. For most protocols, any protocol state or message sequencing is immediately obvious from the face of the protocol description, and need not be explicitly described. For example, a system that sends a request quite clearly then waits for a response. The documentation is designed to describe protocol state and message sequencing

---

<sup>22</sup> Trustee’s 2d Report p. 5.

where it would not be readily apparent to developers. Any remaining issues can be resolved through a process of ongoing correction and support.

45. OTR points to the Open Group's documentation, and in particular to a finite state diagram, as being "much more helpful."<sup>23</sup> However, while some developers prefer finite state diagrams for their formality in documenting sequencing information, others find this same formality less instructive than the use of more descriptive ways of documenting sequencing information. Microsoft's methods of documenting sequencing information are reasonable, and would be generally understood by developers. Remaining issues can be resolved through a process of ongoing correction and support.

#### **L. Illustrations**

46. The Trustee's claim that the documentation lacks sufficient illustrations is unsubstantiated. As with explanatory memorandums, the need for illustrations varies widely from one context to another. The documentation is designed to provide illustrations where needed; further illustrations can be added to the documentation over time to reflect the actual implementation needs of licensees.

47. The Trustee comments that "Simple inspection of available textbooks covering many of the aspects of system operation and interoperation show that illustrations are widely used by technical authors to augment textual presentation."<sup>24</sup> Textbooks are not a proper comparison for the WSPP Technical Documentation. The Technical Documentation is designed to provide a reference to supplement the knowledge of a developer already skilled in the relevant art, whereas a textbook is designed to instruct someone who lacks such knowledge and skill. Thus, a textbook will generally contain significantly more explanatory material, whether in the form of illustrations or otherwise, than protocol documentation such as the WSPP Technical Documentation.

---

<sup>23</sup> Second OTR Report p. 7.

<sup>24</sup> Letter of 16 December 2005 from Prof. Neil Barrett to Cecilio Madero.

### **M. Use of Examples**

48. The Trustee's claim that "examples and example-based explanations are virtually absent from the Technical Documentation"<sup>25</sup> is unsubstantiated. Indeed, the Trustee explicitly acknowledges that the documentation contains "code presented as an example."<sup>26</sup> The documentation shows examples of major packet and data structures, using either MIDL or C++ source examples. The examples occur in the topics where the packet or data structures are defined. Additional examples can be added to the documentation over time to reflect the actual implementation needs of licensees.

### **N. Pseudocode**

49. The Trustee's criticism that the pseudocode in the documentation does not follow a single standard format is not valid, because the use of multiple formats is in line with industry practice. A quick search of the RFC public standards data base reveals that RFC documents use pseudocode in many different formats. To take an arbitrary example, RFC 2616, RFC 908, and RFC 3404 each use pseudocode in a different format. In RFC 2616, the assignment of a value to a variable is denoted by ":="; in RFC 908, assignments are denoted by "Set"; and in RFC 3404, assignments are denoted simply by "=", which is the convention in the C programming language. Whatever the theoretical merits of standardized pseudocode, experience has shown that developers are accustomed to and capable of using pseudocode in a wide variety of formats.

### **O. Revision History**

50. OTR is incorrect in suggesting that Microsoft does not provide sufficient information to allow for backward compatibility. The documentation of a specific protocol feature explicitly lists the client and server operating systems that support that feature. If there has been an addition or modification to an existing interface, it is explicitly noted in which version of the operating system it was added or made. This is the information that a developer would need to provide for backward compatibility, and additional revision history information is not necessary.

---

<sup>25</sup> Trustee's 1st Report p. 3.

<sup>26</sup> Trustee's 1st Report p. 3. See also Letter of 16 December 2005 from Prof. Neil Barrett to Cecilio Madero ("examples of source code").

### **P. Vista Protocols**

51. The Commission and the competitor evaluators are incorrect in suggesting that Windows Vista protocols are “generally missing” from the documentation.<sup>27</sup> New protocols that were implemented as of Windows Vista Beta 1 have been documented. After the Beta 2 source code is frozen and stabilized, the Beta 2 protocol additions will also be documented. The competitor evaluators appear to have expected more changes as of the Beta 1 release. They are simply mistaken.

### **Q. Hyperlinks and References**

52. The Trustee’s concerns regarding hyperlinks and internal references were addressed in preparing the December 2005 release of the Technical Documentation. The Trustee’s comments in this regard appear targeted at the substantial amount of material added in November 2005, which was fully integrated into the December 2005 release. As of the December 2005 release, every effort was made to provide appropriate hyperlinks and references both to external contextual information such as public standards documents, as well as to internal items such as headings and technical reference material. Any remaining issues would be easily resolved through a process of ongoing correction and support.

53. These same comments also apply to the Trustee’s concerns regarding missing parts. Parts the Trustee thought to be “missing” may have been merely mislabeled. Such issues were also addressed in preparing the December 2005 release of the documentation.

54. Similarly, issues that OTR had with internal references and the viewing mechanism at the time of the First OTR Report were addressed in subsequent releases of the documentation. Indeed, the Second OTR Report acknowledged that by then, the problems identified in the First OTR Report had been addressed.<sup>28</sup>

### **R. Integration**

55. The Trustee’s concerns regarding the integration of the documentation were addressed in preparing the December 2005 release of the Technical Documentation. Material that the Trustee had received in separate files was integrated into a single file, with a single overall structure.

---

<sup>27</sup> SO ¶ 66.

<sup>28</sup> Second OTR Report p. 8.

This should resolve any problems that the Trustee had with the general structure of the documentation, the search facility, or the filenames. Any remaining issues would be easily resolved through a process of ongoing correction and support.

### **S. Proofreading**

56. The Trustee's concerns regarding typographical errors in the documentation were addressed in preparing the December 2005 release of the Technical Documentation. All content was edited prior to the December 2005 release, as well as submitted to another round of spell check and build surveys. References to page numbers were replaced with hyperlinks. All topics were edited to ensure that terms are defined at first use. As with any document of this magnitude, typographical errors likely remain, but these are easily handled through a process of ongoing correction and support.