
COMP/C-3/37792 Microsoft

Prepared By:
Prof. Anthony Finkelstein

With:
Prof. Jeff Magee
Prof. Jeff Kramer
Prof. Wolfgang Emmerich
Dr. Holger Schwichtenberg

Imperial College Consultants
First Floor, 47 Prince's Gate
Exhibition Road, London SW7 2QA, UK

Signed

Anthony Finkelstein

Date: February 13th, 2006
Contents

0 Executive Summary .................................................................................................................3

1 Key Concepts ..........................................................................................................................5
This section presents the key concepts that are required in order to understand the decision, what Microsoft has provided and the objections to it. It explains how the WSPP documentation goes significantly beyond what is ‘on-the-wire’.

2 Protocol-centric specification .................................................................................................10
This section explains how the interoperability information provided in the WSPP documentation has been specified. It points to where this has been misunderstood by the Trustee.

3 Interface-clarity......................................................................................................................12
This section explains the engineering principle of ‘information-hiding’, which is applied as a means of controlling complexity in software development. It shows how this sound principle has been used systematically in the WSPP documentation. It points to the problems of achieving clean and simple interfaces when being required to open a software system to interoperability in a manner not anticipated at design-time. The section highlights misunderstandings about information hiding and abstraction on the part of the Trustee.

4 Work Group Server Operating System Products.................................................................15
This section explains the challenges of building work group server operating system products and outlines the requirements that are placed on those who would wish to develop services in this area. It shows that the competitors and the Trustee have taken no account of these requirements when evaluating the WSPP documentation and have made demands that go well beyond any reasonable interpretation of interoperability.

5 The Free-Standing Documentation Myth................................................................................16
This section explains why the suggestion that the documentation should be “free-standing” makes no sense and violates basic software engineering good practice. It analyses how the documentation might be used and the support available both within and outside the WSPP.

6 Specification Quality .............................................................................................................18
This section considers the quality of the WSPP documentation and concludes that it is unreasonable to expect that the specification should be error free.

7 Documentation release process .............................................................................................18
This section discusses the process by which documentation is released and how such documentation is subject to a continuing process of improvement. It highlights the unreasonableness of assuming that you can get a complex set of specifications totally right the first time. We also discuss why the criticisms of the Trustee and the competitors on the matter of versioning are unjustified.

8 Audience ................................................................................................................................20
This section considers the suggestion that the WSPP documentation should be usable without prior knowledge of the Microsoft environment. It explains why this is wholly unreasonable.
9 Effort

This section reviews the effort that might be required to perform a sound evaluation of the WSPP
documentation. It concludes that neither the Trustee nor the competitors devoted sufficient or indeed
proportionate effort to this task.

10 Microsoft Process

This section examines the process deployed by Microsoft to develop the WSPP documentation. It
establishes that Microsoft devoted significant skilled and dedicated effort to this task. It concludes
that this process gives one confidence in the completeness and accuracy of that documentation.

11 Comparison with Open Specifications

This section summarises the comparison with an equivalent set of open specifications, OMG
CORBA, acknowledged as supporting interoperability. The comparison is detailed in Appendix A.
This comparison clearly shows that the WSPP documentation is similar in all significant respects to
that of CORBA and that on this basis the criticisms levelled at the WSPP documentation are
unreasonable.

12 Sufficiency Test

This section considers the “Sufficiency Test” performed by the Trustee and the comments made in
respect of that test. In Appendix B we go through the Sufficiency Test narrative in some detail and
expose the set of elementary misunderstandings that account in full for the difficulties that the
Trustee encountered. In Appendix C we show, on the basis of our analysis, how, in fact, the ‘add-
user’ task could be accomplished using the WSPP documentation.

13 Minor Omissions and Misunderstandings

This section considers the treatment of minor omissions and misunderstandings and explains how
they might be resolved by recourse to the technical support offered to WSPP licensees.

14 Server Source Code

This section analyses the significance of the offer by Microsoft to make available source code for
reference and concludes that, while not required it nevertheless constitutes a significant complement
to the WSPP documentation.

15 Conclusion

This section presents our conclusions. We determine that the WSPP documentation has been
reasonably calculated to provide the technical basis for a market in interoperable server operating
system products and that it is in these terms complete and accurate.

Appendix A: Comparison with CORBA

Appendix B: Sufficiency Test Analysis

Appendix C: Implementing Add User

Appendix D: Qualifications and Experience of Team

Appendix D: References
0 Executive Summary

0.1 This report presents an expert assessment of the interoperability information provided by Microsoft as part of the Work Group Server Protocol Programme (WSPP).

0.2 This assessment is given in the context of the European Commission’s Statement of Objections of 21st December 2005, stating why it believes that Microsoft has failed to comply with its Decision of 24th March 2004. This decision required Microsoft to make available Interoperability Information in respect of its Work Group Server operating system products. The assessment is further directed to a consideration of the Article 24(1) Decision of 10th November 2005 in which the Commission expresses the intention to impose periodic penalty payments on Microsoft if, in its judgement, Microsoft fails to comply with the obligations set out in the original decision. In the course of presenting the assessment this report considers the various evaluations that have been provided by competitors and more significantly, by the Trustee appointed to oversee compliance.

0.3 We conclude that the interoperability information as provided by Microsoft meets current industry standards, particularly in such a complex domain. We believe that it has provided complete and accurate information, to the extent that this can be reasonably achieved, covering protocols, dependencies and implicit knowledge. We observe that the WSPP documentation goes significantly beyond what is 'on-the-wire' so as to assist skilled and knowledgeable software developers to develop a Work Group Server operating system that can work on an equal basis with Microsoft’s own products within the Windows domain architecture. There are of course further improvements and clarifications that can be made; it is standard practice that these are performed in response to user queries.

0.4 Below we set out a quick overview of the detailed assessment given in the report. It assumes some familiarity with the European Commission’s Decisions and Statement of Objections in respect of Microsoft. For a full technical account see the relevant sections which are cross-referenced.

0.5 In an assessment of this type, definitions play an important role. Many of the concepts used are complex and have a specialised interpretation in relation to distributed computing systems. We review these concepts: interoperability, protocol, specification, dependency, and so on. We consider their use and place them in the context of the WSPP documentation [Sect 1]. On the basis of this analysis we conclude that the WSPP documentation goes significantly beyond what is ‘on-the-wire’ and provides a good basis for skilled and determined software developers to produce a Work Group Server operating system product that can work on an equal basis with Microsoft’s own products within the Windows domain architecture.
0.6 The WSPP documentation, in compliance with Decision, is given in a ‘protocol-centric’ style. This is the accepted way in which such interoperability information is presented. We explain how such specifications are constructed and point to some important misunderstandings on the part of the Trustee [Sect 2]. Based on this we explain the engineering principle of ‘information-hiding’, which is applied as a means of controlling complexity in software development. We show how this sound principle has been used systematically in the WSPP documentation. In particular we point to the problems of achieving clean and simple ‘interfaces’ between system components when required to open a software system to interoperation, in a manner not anticipated at design-time [Sect 3]. We again highlight misunderstandings about information hiding and abstraction on the part of the Trustee.

0.7 Work Group Server Operating System Products are generally large and complex. We explain the challenges of building work group server operating system products and outline the requirements that are placed on those who would wish to develop services in this area. We show that the competitors and the Trustee have taken no account of these requirements when evaluating the WSPP documentation and have made demands that go well beyond any reasonable interpretation of interoperability [Sect 4].

0.8 The Trustee bases his criticisms of the WSPP documentation in significant part on the suggestion that the WSPP documentation should be “free-standing”. We argue that this makes no sense and, indeed, violates basic software engineering good practice [Sect 5]. We analyse how the documentation might be used and the support available both within and outside the WSPP. We then consider the quality of the WSPP documentation conclude that it is unreasonable to expect that the specification should be error free [Sect 6]. We consider the evidence presented in respect of the residual errors in the WSPP documentation. A further assumption made by the Trustee is that the WSPP documentation should be usable without prior knowledge of the Microsoft environment. We explain why this is wholly unreasonable [Sect 8].

0.9 The Report considers the process by which complex documentation is made available and remarks on the established practice by which such documentation is subject to a continuing process of improvement. We highlight the unreasonable assumption that you can get a complex set of specifications totally right the first time [Sect 7]. The matter of versioning, both within the documentation and of the underlying operating systems to which the documentation makes reference, is considered. It is concluded that the strategy adopted complies with the Decision in respect of timeliness.

0.10 We consider the effort that might be required to perform a sound evaluation of the WSPP documentation. We conclude that neither the Trustee nor the competitors devoted sufficient effort to this task and in particular that this effort was not
proportionate to the effort that would be required to develop Work Group Server Operating System services [Sect 9]. We go on to examines the process deployed by Microsoft to develop the WSPP documentation. Our report establishes that Microsoft devoted significant skilled and dedicated effort to this task and concludes that this process is such as to give one confidence in the completeness and accuracy of that documentation [Sect 10].

0.11 We have performed a detailed comparison of the WSPP documentation with an equivalent set of open specifications, OMG CORBA, that support interoperability and constitute an established industry standard [Sect 11 & Appendix A]. This comparison clearly shows that the WSPP documentation is similar in all significant respects to that of CORBA and that on this basis the criticisms levelled at the WSPP documentation are unreasonable.

0.12 We have also considered the “Sufficiency Test” performed by the Trustee and the comments made in respect of that test [Sect 12]. This test has been relied upon by the Trustee and forms an important component of the Statement of Objections. We go through the Sufficiency Test narrative in some detail and expose the set of elementary misunderstandings that account in full for the difficulties that the Trustee encountered [Appendix B]. We are able to show, on the basis of our analysis, how, in fact, the ‘add-user’ task could be accomplished using the WSPP documentation [Appendix C].

0.13 Our report briefly considers the treatment of minor omissions and misunderstandings and explains how they might be resolved by recourse to the technical support offered to WSPP licensees [Sect 13]. Finally we analyse the significance of the offer by Microsoft to make available source code for reference and conclude that, while not required it nevertheless constitutes a significant complement to the WSPP documentation.

1 Key Concepts

This section presents the key concepts that are required in order to understand the decision, what Microsoft has provided and the objections to it. It explains how the WSPP documentation goes significantly beyond what is ‘on-the-wire’.

1.1 Any assessment conducted in this area must of necessity be based on an understanding of the concepts in the original Decision and in the Statement of Objections. In summary form the original Decision required that:

Microsoft Corporation shall […] make the Interoperability Information available to any undertaking having an interest in developing and distributing work group server operating system products and shall […] allow the use of
the Interoperability Information by such undertakings for the purpose of
developing and distributing work group server operating system products.

The key terms were defined as follows:

**Interoperability Information** means the complete and accurate specifications
for all the **Protocols** implemented in Windows Work Group Server Operating
Systems and that are used by Windows Work Group Servers to deliver file and
print services and group and user administration services, including the
Windows Domain Controller services, Active Directory services and Group
Policy services, to Windows Work Group Networks.

**Protocol** means a set of rules of interconnection and interaction between
various instances of Windows Work Group Server Operating Systems and
Windows Client PC Operating Systems running on different computers in a
Windows Work Group Network;

1.2 It is, in the matter of how these terms have been defined, and in the manner in which
they have been used, that much of the dispute that has arisen rests. Rather than
venturing an interpretation of the Decision, a matter that fortunately rests with the
lawyers, let us instead review some key concepts in the area of interoperability. We
present some examples to help illustrate the concepts and relevant issues that arise.

1.3 Interoperable systems are systems composed from components that are required to
cooperate to provide complex services. Such systems are generally distributed and
have significant non-functional constraints, such as reliability and performance, on
their operation. Interoperability is that property of components and of the means by
which the components are composed which ensures that they jointly meet the service
requirements placed on them.
1.4 As an example Figure 1 presents a setting, consisting of three components, in which such interoperability arises. A Microsoft Work Group Server is required to interact with a Brand X Work Group Server and with a simple client; in our example this client is resident on a Microsoft platform.

1.5 Figure 2 shows the same configuration of components as in Figure 1. In this simple example the client is requesting a service from a server (S?) and receiving a response (S!). This request is being made over a client-side interface, for a service made available through the server-side interface. The request for a service and the response
manifest themselves in the exchange of data across the network infrastructure. It should of course be understood that client and server are roles assumed by components, and that a component can assume different roles with respect to different components at different times.

1.6 The set of rules that govern the exchange between client and server (and between servers assuming these roles) and the precise formats that describe how the data is exchanged across the network infrastructure is the ‘on-the-wire’ protocol.

1.7 The Commission’s Article 24(1) Decision notes in paragraph 3.1.5 (94) in relation to “Microsoft’s position on its obligations with regard to the completeness and accuracy of the Technical Documentation”:

(94) In Appendix 2 to the WSPP Agreements, Microsoft describes the content of the Technical Documentation as follows: “This specification covers the documentation requirements and styles for Microsoft proprietary protocols and extensions to published/industry standard protocols used on the wire in networks.”

1.8 Providing that the server-side interface makes available the service that the client requests, and that the implementation of that interface interprets the protocol correctly, the Microsoft server can be replaced by the Brand X server, transparently. That is, without the client having to be aware of the change. This is, of course a slightly idealistic picture. It is in the simplifications that this picture entails that some of the difficulties on the part of the Trustee and the competitor evaluators have arisen.

*Figure 3: server-server interaction*
1.9 In Figure 3 we show a Brand X server that in the process of providing a service S that it is making available to its clients. It does so using algorithms and data structures that are specific, perhaps propriety, to Brand X. In the process of providing S it requires a service (x) to be provided to it. Using the protocol it has implemented and in line with the server rules of which it has been notified, it requests and receives x (this may of course require a complex sequence of exchanges). All should be well!

![Figure 4: service dependency](image)

1.10 Figure 4 shows how a problem could potentially arise. In the Microsoft implementation of S information produced in the course of the provision of another service, Q, is used in the computation of S. A dependency has arisen between S and Q. It is not visible ‘on-the-wire’ and in the worst case is not visible at the interface either. The Brand X server cannot now provide S in a comparable manner to that of the Microsoft server. It must be stressed that while creating such dependencies is not good software engineering practice they are inevitable in any large and complex software product. These dependencies can be subtle and may arise without the developers or technical architects being aware of them.

![Figure 5: implicit knowledge](image)
1.11 Figure 5 shows another potential problem. In this case the implementers of S, Microsoft for the purposes of illustration, know the algorithms that implement Q. They thus have implicit knowledge of the state of Q and of the results that the invocation of Q might yield. They can rely on this knowledge in their implementation of S. Again, nothing is obvious at the interface or ‘on-the-wire’, there is no protocol, and yet it would be difficult for an alternative implementor of S, say Brand X.

1.12 The term specification has received some attention, both in the original Decision and in the Statement of Objections. It is a term of very broad application in computer science. It simply denotes an abstract description of a set of computational objects or phenomena. An interface definition describes abstractly an implementation, it is said to specify it. A rigorous but nevertheless abstract description of the rules and formats governing the exchange of data over a network infrastructure constitutes a protocol specification. The term specification is often used in relation to the software development process. In this use it denotes the set of a documents produced prior to implementation and intended to describe the architecture and design of a software system to be implemented. This sense of the word has little relevance in our context as we are not in an a priori development setting and the attempt to use definitions drawn from this usage (see footnote 48: Statement of Objections) can only mislead. Thus, for example, in the Statement of Objections (58) it is suggested that practices clearly appropriate to a requirements specification can be applied to the post hoc disclosure of interoperability information.

1.13 As will become clear in the assessment that follows, we believe that Microsoft have supplied complete and accurate information, to the extent that this can be reasonably achieved, covering protocols, dependencies and implicit knowledge. The WSPP documentation goes significantly beyond what is ‘on-the-wire’ and in our judgement should permit skilled and determined software developers to produce a Work Group Server operating system product that can work on an equal basis with Microsofts own products within the Windows domain architecture.

2 Protocol-centric specification

This section explains how the interoperability information provided in the WSPP documentation has been specified. It points to where this has been misunderstood by the Trustee.

2.1 The Decision requires Microsoft to provide the interoperability information the Commission deems necessary to permit competition in the Work Group server operating system market in the form of a specification of protocols. The WSPP documentation is thus protocol-centric. Protocol-centric specifications, while the
accepted method for specifying interoperability in complex distributed systems middleware, do not always make for easy reading.

2.2 Essentially, a protocol specification consists of two parts; firstly, the set of messages and secondly, the rules by which these messages are exchanged between entities. Messages can be specified at different levels of abstraction – at the logical level, the logical components (usually termed fields) of a message are identified symbolically and the meaning of each component defined (usually by means of natural language). At the physical level, the bit representation of each field is defined and the way the fields are packed into a message is also defined. The rules by which messages are exchanged may be specified in a number of ways: by a state machine, by message sequence diagrams, by an algorithm or by a set of rules.

2.3 The Microsoft WSPP specification documents protocols in two principal ways. Firstly, protocols such as the File and Print Server protocols are largely specified using Microsoft RPC and secondly protocols such as the Network Time Protocol are specified directly by messages and rules. Where this latter approach has been adopted, the WSPP contains, usually, in diagrammatic form, a specification of the format of the messages as they are transmitted. These protocols are usually simple request-response protocols.

2.4 Microsoft RPC is a remote procedure call protocol that uses a standard way of encoding messages expressed as datatypes in a programming language such as C or C++ (or in a special purpose Interface Definition Language) into packets that are transmitted over the network infrastructure. A Remote Procedure Call also specifies standard ways for messages to be exchanged. The effect is to relieve a programmer using an interface from the details of how messages are encoded for transmission and from the details of how messages are exchanged to achieve the execution of a remote function. The WSP documentation follows good practice in seeking to present the information at the highest level that would allow a skilled programmer to implement the protocol.

2.5 The WSPP documentation, together with the documents that it refers to, aims to give sufficient information to allow the Microsoft RPC protocols to be implemented on a non-Windows platform. The aim has been to give a user of the specification sufficient information to both generate and interpret the messages sent over the physical communication medium between servers. This on-the-wire approach supports interoperability between services running on different platforms. A programmer implementing an interface to a remote service from a non-Windows platform could not of course use the Microsoft implementation of RPC and would have to provide an alternative means to implement the WSPP protocols. The WSPP documentation contains descriptions of the Microsoft RPC protocol and reference to the Open Group specification of the protocol. RPC permits the interface to a remote service to be
specified in the same way as an interface to a local service or component is specified – essentially by a list of function or procedure calls. Each call to a function in the interface is translated into a protocol that packs the parameters of the request into a message that is transmitted over the wire to the remote end, where the message is decoded and the results of executing the remote function are packed into a response packet that is transmitted back to the callers.

2.6 Under the heading “Missing behaviours and dependencies” the Trustee (in his 2nd Report of 15th December 2005) reveals a basic misunderstanding of the use of RPC as a means of interoperability. This, and the errors consequent upon it, means that he has been unable to understand the protocol specifications. The Trustee uses the **samr** function as an illustration. However, **samr** is in fact an interface that defines a set of functions or methods that can be invoked on the remote Security Account Manager. The Trustee then refers to [___]. The “connect” that the Trustee is looking for, is binding the **samr** interface to a Security Access Manager. This is part of the standard Microsoft RPC protocol. The **SamrCreateUserInDomain** returns a status code. The meaning of the value of this code is clearly documented. The Trustee objects that these are not classified into fatal or recoverable, however, this is implementation dependent. For example, if **STATUS_NO_MEMORY** is returned it is clear that action is needed to allocate more memory or free existing memory in the server before retrying. This is implementation dependent, takes place within the server operating system and goes well beyond protocol information.

2.7 Competitor evaluators, notably IBM, have commented that, had Microsoft provided the full Interface Definition Language files used in generating the code to realise the protocol, it would ensure accuracy and ease the effort required for them to implement interoperable services. Microsoft, of course, has been required to disclose interoperability information, and this they have done. All the information required to build the relevant IDL files is present in the documentation.

3 Interface-clarity

*This section explains the engineering principle of ‘information-hiding’, which is applied as a means of controlling complexity in software development. It shows how this sound principle has been used systematically in the WSPP documentation. It points to the problems of achieving clean and simple interfaces when being required to open a software system to interoperability in a manner not anticipated at design-time. The section highlights misunderstandings about information hiding and abstraction on the part of the Trustee.*
3.1 Good software engineering practice places considerable emphasis on the principle of information hiding (Parnas, 1972). By dint of careful architecture, interfaces are presented that allow high-level access to the services that the software affords. Considerable attention is paid to ensuring that these interfaces are appropriate, that the components are loosely coupled and that information on application state is managed behind the interface. Good design of this form allows the software to be changed while minimising the non-local effects of those changes, a key challenge in large-scale software development. Technologies that embed information hiding through mechanisms such as encapsulation are widely used. It is of course possible that information hiding is breached. Such breaches occur, even in the best-regulated software development settings, because of errors, programming shortcuts and performance or other optimisations.

3.2 The interfaces made available in a piece of software represent the developers understanding of the design of that software. In the case of the Microsoft Work Group Server operating system products and the client operating systems with which they interact, these appear to have been conceived of as a coherent whole and with seamless interaction as a guiding design objective. The consequences are that any attempt, a posteriori, to open these products to interoperation of a type not envisaged at design time exposes an interface that is, with the best will in the world, messy, difficult to understand and use.

3.3 Let us consider a simple analogy – a domestic HiFi system. You can replace amplifier, speakers, CD Player using the simple component interfaces provided by the designers. These interfaces are well understood and documented and relatively uncluttered. Let us however imagine you are asked instead to retrofit BrandX buttons and switches. This is going to be difficult, you will have to struggle with special purpose connectors and jumper leads. You may well discover that the manufacturer, who is well disposed towards you, does not actually have a clear understanding themselves of how the buttons and switches have been connected. You will have to access complex documentation and wiring diagrams. You might have to do large-scale disassembly in order to fit things together. And, of course, when you have finished your HiFi will never look quite as good as the original.

3.4 The WSPP specification presents the interface to services such as File and Print as a set of remote procedure calls. It should be noted that there is no industry standard method for specifying the behaviour of interfaces such as these other than using natural language to describe the permitted order (or sequence) of calls as is done in the WSPP specification. Indeed, it is very much an open research question as to the best way of precisely and formally specifying the behaviour at the interface to a service while maintaining a clear separation between the specification of an interface and its implementation. It should be noted that an interface specifying a set of
operations together with the rules constraining the order in which these operations can be used is also a way of specifying a protocol.

3.5 Abstraction and information hiding should not be confused with obfuscation. This confusion is however strongly evident in the Trustee’s reports. For example, one of the techniques used in WSPP implementation is stateless servers in which the server does not maintain information on the state of clients using the server. Instead, when a client first contacts a server, the server returns a piece of context information that the client includes in any further interactions with the server. The idea here is that rather than the server storing state for many clients, each client stores its own context or server related state. This information is simply stored by the client and not interpreted. It is referenced by a context handle (which may be of type void* in C to avoid compilation dependencies). The client should not know the structure of this context information since it relates to how the server is implemented and not to the client server interaction. This is a sound and well known distributed software engineering technique. It is therefore surprising that the Trustee refers to this as a serious problem with the specification. The same basic approach can be used to allow stateful servers in a session-less protocol. Instead of sending all client related information to server in every transaction the client sends this information once and receives a session handle from the server. In consecutive calls the client just inserts this session handle and the server uses the handle to refer to the context that was established in the initial call. This is a technique widely used by Microsoft. Much is made of the typedef void* SAMPR_HANDLE example that, in fact uses context handles. The void* type is good software engineering practice and means that the client does not depend on the server implementation. Context handles are also used in respect to binding information for remote procedure calls and their use is well documented in, among other places, MSDN.
This section explains the challenges of building work group server operating system products and outlines the requirements that are placed on those who would wish to develop services in this area. It shows that the competitors and the Trustee have taken no account of these requirements when evaluating the WSPP documentation and have made demands that go well beyond any reasonable interpretation of interoperability.

4.1 Work Group Server operating system products are complex pieces of distributed software providing critical services such as user administration. They are required to be highly available and fault tolerant, they must be scaleable to cope with the requirements of large, complex and rapidly changing organisations. The Microsoft products are sophisticated and their development has, we can reasonably surmise, required a considerable investment of time and effort. A similar effort would be required of any competitor seeking to develop a product to provide the same sorts of services in this environment. Assuming that interoperability, of the type envisaged by the Commission, is an important requirement for such products we would expect that the effort devoted to interpreting and implementing the relevant protocols would be proportionate with respect to the overall effort required. In other words the interoperability information will not be the subjects of brief scrutiny, as is clearly and perhaps unavoidably, the case for the competitor evaluators and the Trustee, but rather the focus for significant and sustained effort, months not hours.

4.2 The context for a particular evaluation is, of course, important. In looking at the competitor evaluations and at the comments of the Trustee it is particularly striking that they appear to be presented without an actual development context. Neither the competitor evaluators nor the Trustee viewed the WSPP documentation as part of a development effort that would have its own underlying platform and architecture. In other words they are speculations about the completeness and accuracy of the information unrelated to a proposed product, even in vague outline. Without a sketch, or any indication, of the structure of the directory and authentication services on which a Brand X product might be based and against which the documentation is to be tested it is difficult to attach much credibility to the criticisms.

4.3 In reading the comments of the competitor evaluators many seem to have little relation to any reasonable interpretation of the scope of the decision. Thus Sun, pushing at the very limits of what might even conceivably be regarded as interoperability information state: “Microsoft should have provided a detailed specification of the full functionality of Active Directory including all of its functions, features and externally visible behaviours and dependencies.” The OTR also appear to have an idealised view of what developing a competing product might entail, expressing surprise that a: “competitor would have to perform a considerable number of experiments using functioning combinations of Windows work group servers and
PCs”. The Commission in the Article 24(1) Decision (93) observes critically that the documentation “required a laborious, forensic-like examination in order for a reader to try and work out what was going on”. We would expect nothing less. Such an examination is the norm in software engineering and an entirely reasonable expectation to have of the WSPP licensees.

4.4 Some further critical remarks indicate unreasonable expectations. Under the heading “Insufficient description” the Trustee, in his report of 15th December 2005, criticises the documentation for assuming as a context a Windows platform for both client and server and suggests instead that a “comprehensive list of the features that the client expects the server to provide is necessary (for example, an API called ‘samr’ capable of performing the required operations and returning the expected values; an account database structured with certain expected fields, and so on)”. This goes far beyond the specification of interoperability information and would require Microsoft to provide the outline design of a replacement for the workgroup services supplied by Windows server operating systems.

5 The Free-Standing Documentation Myth

This section explains why the suggestion that the documentation should be “free-standing” makes no sense and violates basic software engineering good practice. It analyses how the documentation might be used and the support available both within and outside the WSPP.

5.1 A basic assumption of the Trustee is that the interoperability information provided by Microsoft should be “free-standing” (Annex to Trustee letter, 16th December 2005 and subsequently WSPP Documentation Sufficiency Test). This is not a realistic requirement and indeed, runs counter to good software engineering practice. It is a basic error to confuse completeness with the notion that a specification should be useable on a free-standing basis. Indeed, were it even possible to provide documentation that could be wholly free-standing, that documentation would probably as a result be so large as to be completely unusable. Clearly the documentation is not required to explain elementary software development terms nor incorporate specifications of very well established technologies for which there are commercial guides. At a basic level it may not be necessary to reference these. External standards upon which a Microsoft implementation relies should clearly be referenced, and indeed they are. It would make no sense to copy such information into the documentation in all but the most limited cases. To do so would risk the infamous ‘double maintenance problem’: an error is detected in the standard, or additional clarification information, such as notes on superseded features, is appended and there would be no guarantee that these would be carried across to the WSPP documentation. Interestingly where Microsoft has deemed it necessary to copy information from other sources, competitor evaluators have complained.
5.2 More importantly, the Microsoft product set is a large and complex one. The services within the Work Group Server operating system product set, such as Active Directory and the Microsoft implementation of Kerberos, are feature rich. This provides the context in which a WSPP licensee will be working and it is wholly unreasonable to expect this context to be described within the interoperability information. There is a significant body of development resources that aid the developer working within the Microsoft environment to access the full facilities of these services. In addition there is a significant range of interoperability information, outside the WSPP Programme, that describes the baseline interoperability features of the Microsoft products: LDAP compliance and the like. All this information is available through MSDN (the Microsoft Developer Network) and the vast support community that has grown up around it. Microsoft devotes a significant effort to ensuring that this base of information is up-to-date and easy to use. Basic matters such as the nature of a ‘context handle’ (see the Trustee’s 2nd Review of 15th December 2005) are fully described and explained, accessible by way of a simple Google search. To understand the organisation of the services, their architecture and rationale it is not even necessary to use MSDN, simply check the on-line bookshop Amazon to find the books for the developer on Active Directory, LDAP, Kerberos.

5.3 It is well known that no developer is ever wholly satisfied with the documentation they receive. There are matters of interpretation, understanding and straight out bugs. For this reason all significant bodies of interoperability information are provided in the context of an offer of help and assistance. In providing help, and answering the questions of developers using the specification, the provider of the interoperability information, Microsoft in this case, will understand how to improve their documentation and address the ‘Frequently Asked Questions’. The WSPP License Agreement (19th October 2005, provided to us) contains significant provision for such technical support. Licensees are able to obtain support within, for example, the Microsoft Premier programme that includes designated account management, 24/7 problem resolution support, critical situation management and onsite support. The Microsoft support team will, of course have access to developers. The support agreement clearly specifies that the technical support includes assistance with debugging and verifying the actual operation of the WSPP protocols.

5.4 The extended offer of support announced by Microsoft on 25th January 2006 and in the Microsoft Server Source Code Fact Sheet 29th January 2006 is clearly germane to this. The offer includes: introductory training, including an on-site presentation and tutorial on how to use the technical documentation and source code and the ways to make use of the engineering support; and, existing manuals on its server software. It reaffirms the offer to provide 500 hours of additional engineering support from a Technical Assistance Manager to WSPP licensees, made by Microsoft.
6 Specification Quality

This section considers the quality of the WSPP documentation and concludes that it is unreasonable to expect that the specification should be error free.

6.1 Nobody would expect a large body of documentation ever to be bug-free. There are undoubtedly errors contained in the WSPP documentation, a few may even have been identified during the Trustee and competitor evaluation. If Microsoft were to warrant that the information was free of errors nobody would believe them, in accordance with the well-known maxim that you can only prove the presence of bugs, not their absence. Thus the question becomes something like: ‘is the documentation of a reasonable quality and are there ‘obvious’ bugs that would impede the use of the information to support the implementation of an interoperable service’.

6.2 There is little by way of a scientific basis for saying what a reasonable volume of residual bugs in a document of this type might be. The best indications are obtained by error seeding studies in which known errors are introduced into a document and the proportion of those detected by techniques such as inspection are then identified. A standard rule of thumb (see for example the Human Error website which provides a comprehensive survey of the literature) is that errors in code run out at around 5% of lines post testing and module inspection. This is a very crude measure and excludes a higher level of design and requirements error. Comprehensive inspection might be expected to find only half these faults (see for instance So et al., 1995). Thus optimistically a document of this type, which is any event much more difficult to build and check than code, which has a formal structure and for which we have compilers and debuggers that can leverage checking, might still contain 2.5% of lines containing residual errors (316 complete pages of errors in a document of 12624 pages). It would be necessary to show that the residual error rate is unreasonable and it is evident that the OTR competitor evaluators and the Trustee have not done this, nor has anything other than an informal, ad-hoc approach been taken in this regard. Consequently, no weight can be placed on this.

6.3 A further matter must be broached. It is likely, perhaps inevitable, that the Microsoft Work Group Server operating system products themselves have bugs that are reflected in the protocol specifications. In other words, that the developers do not know the complete behaviour of the Work Group Server operating system products, and that as competitors use the interoperability information, these bugs, not attributable to the WSPP documentation, will emerge.

7 Documentation release process

This section discusses the process by which documentation is released and how such documentation is subject to a continuing process of improvement. It highlights the unreasonableness of assuming that you can get a complex set of specifications totally
right the first time. We also discuss why the criticisms of the Trustee and the competitors on the matter of versioning are unjustified.

7.1 In the process of seeking to satisfy the Commission and the Trustee with respect to its compliance to the Decision Microsoft have issued preliminary versions of the documentation (11\textsuperscript{th} November 2005 and 22\textsuperscript{nd} November 2005). From the standpoint of the matter under consideration however, only the interoperability information made available to the Commission on the 15\textsuperscript{th} December 2005 is relevant. This ‘build’ of the documentation aggregates the significant improvements made in the earlier releases. The integration of the documentation as part of the release also aids usability.

7.2 Points raised by the Trustee in his report of 15\textsuperscript{th} December 2005 have, as you would expect, also been addressed. For example the \texttt{NETLOGON\_AUTHENTICATOR} type. The type is clearly defined as a structure consisting of a \texttt{NETLOGON\_CREDENTIAL} and a timestamp. The Timestamp is clearly defined as an integer containing seconds elapsed since 1st January 1970 and the credential is an encrypted 8-character string. The documentation clearly explains that the type of the credential is an alias for a \texttt{CYPHER\_BLOCK}. This type defined with an explanation that gives clear references to the encryption.

7.3 Setting aside the demands of the Decision and the Statement of Objections, ‘real-world’ documentation is never produced in a ‘big bang’ sudden release. It is well known that even to attempt to do so is to court disaster. Documents are best released as part of an iterative process of improvement in which a ‘best effort’ initial release gives rise to feedback from users that in turn leads to a systematic improvement in quality. It is likely in any event that changes within the Microsoft Work Group Server operating system product set will drive changes and improvements in the documentation. A thorny issue that arises from the situation in which Microsoft have been placed is how exactly the evolution of the documentation is to be controlled. If Microsoft release too much of the detail of the internal implementation behaviour in interfaces, they shackle themselves with respect to future interoperability. A subsequent benign change can affect a competitor who has relied on that part of the interface for their implementation. The chances are high that the competitor would complain.

7.4 The issue of versioning is highlighted as a problem in the Statement of Objections (at 65) reflecting comments from both the competitors and the OTR. We are of the view that these comments are unjustified. The comments are rendered slightly difficult to interpret by eliding two issues: the versioning of the protocol information and the versioning of the platforms to which they make reference. The WSPP license agreement specifies precisely the policy with respect to updates. There can be no doubt that this policy is entirely consistent with the timeliness provisions of the
Decision. In our discussions with the Documentation Team at Microsoft we have ascertained that it is an underlying assumption that the WSPP documentation will be issued monthly (for as long as required by the Decision) in a current version with a ‘What’s New’ to highlight recent changes. This is accepted industry practice and common to many widely used Microsoft specifications. It hardly makes sense in this context to dot the documentation with surplus version identifiers. As far as we can determine all the definitions and RPC methods are associated with a Requirements section that directly references those operating systems that support the protocol. Where there have been changes the version of the operating system in which the change was made is indicated. The WSPP license agreement makes clear the policy with respect to Microsoft Windows Server 2003 successor releases.

8 Audience

This section considers the suggestion that the WSPP documentation should be usable without prior knowledge of the Microsoft environment. It explains why this is wholly unreasonable.

8.1 One of the most surprising of the criticisms advanced by the Trustee (WSPP Document Review 30th November 2005), implicit in the work plan he advances and reflected in the Statement of Objections (64), is the suggestion that the documentation should make no assumptions as to prior knowledge of the Microsoft environment. This can only be regarded as naïve and hardly a good working basis for the tasks that the documentation envisages. On the contrary, we would expect that any competitor seriously approaching the task of interoperation and of providing services on an equal basis within the Windows domain architecture would possess a deep familiarity with that architecture and of the challenges of programming against it. Such skills are not rare or exotic, highly skilled developers with knowledge of Active Directory, Kerberos etc. are readily available and Microsoft provides training and certification. Many developers possess knowledge of both the world of Unix and java and of the Microsoft environment. We would be most surprised if IBM, Sun, Novell and Oracle did not have the relevant familiarity in-house. The SAMBA team and its commercial offshoots are evidence of the depth of this expertise within the open-source community.

8.2 To be more explicit, it is reasonable to expect that anybody using the documentation would: understand the architecture and use of the relevant Microsoft services; be a confident user of existing Microsoft documentation and information resources; be familiar with programming these services from within the Microsoft environment and with the use of Microsoft’s Software Development Kits (SDKs); have substantial experience of protocol implementation achieved through developing analogous distributed and interoperating systems; have clearly in mind the structures and
implementation that they will have to put in place that constitute the backdrop for interoperation.

8.3 It is unreasonable to expect, and the Decision neither requires nor implies, that Microsoft should provide tutorial materials, though they have in fact offered to do so. The documentation is also not a “textbook”, though again Microsoft has volunteered to supply these. There is, in each case, sufficient introductory content within the documentation for the intended audience to be able to understand the relevant part of the documentation. The structure of the document reflects the commercial structure of the WSPP. It presents the protocols grouped into ‘scenarios’, a strategy that is surely well suited to the way in which the documentation will actually be used (as distinguished from evaluated).

9 Effort

This section reviews the effort that might be required to perform a sound evaluation of the WSPP documentation. It concludes that neither the Trustee nor the competitors devoted sufficient or indeed proportionate effort to this task.

9.1 A brief look at the timeline for the process in which we are now engaged throws into sharp relief the effort that might reasonably be required to assess the completeness and accuracy of the WSPP documentation (12,624 pages stand-alone). The competitors were given an evaluation period that while sufficient to reach a conclusion as to the commercial value of a license is patently inadequate to form an authoritative opinion as to the completeness and accuracy of the documentation, even supposing they possessed the technical tools. In fact all the competitors state that they are unable to perform a proper evaluation, before going on to opine on the matter of completeness and accuracy.

9.2 The Trustee can barely be said to have possessed sufficient time either, a fact he explicitly acknowledges. He reports on 30th November 2005 on documentation provided to him on 23rd November 2005, it is clear from reading this report that it is principally based upon the impressions gained in scrutinising earlier document releases. The 15th December report gives a precise account of the time spent on the ‘add new user’ sufficiency test, of which more below, on which heavy reliance is placed. This amounts to 42 hours. It might be germane at this point to consider again the issue of proportionate effort. Let us assume that in order to implement ‘add new user’ you need to implement an LDAP compliant directory service and a suitably enhanced Kerberos KDC (for a discussion see below) then 42 hours looking at the documentation, from a standing start and with little by way of substantive technical background, looks disproportionately low.
10 Microsoft Process

This section examines the process deployed by Microsoft to develop the WSPP documentation. It establishes that Microsoft devoted significant skilled and dedicated effort to this task. It concludes that this process is such as to give one confidence in the completeness and accuracy of that documentation.

10.1 Assessing the quality of any software development artefact, establishing in this instance the completeness and accuracy of the WSPP documentation, is notoriously difficult. An alternative or at any rate complementary approach is to look instead at the process that gave rise to it. It is notable that the Trustee has not thought to take this approach. The equation is simple – good process leads to good product. We may thus ask ourselves, was the process that Microsoft used in order to prepare the documentation a sound one that might lend confidence to our judgement that it is, for the purposes required, complete and accurate?

10.2 Microsoft has a mature documentation process and established documentation practices. Most developers have some familiarity with this. Over time Microsoft has built up a significant body of expertise in how to write documentation. Microsoft has also been compelled by legal action in the USA to document other interoperability information, the task is thus one with which it has had to become familiar. We have met the Senior Engineers assigned by Microsoft to this task. They are, insofar as we can ascertain, well qualified and highly experienced. The significant depth of technical input that Microsoft has deployed in the assembly of the interoperability information can be clearly seen from an account of the resources spent on protocol documentation. This shows not only that large core technical and documentation teams devoted significant time but that these teams reached deeply into the product and engineering teams: clear evidence of the attempt to ensure that the information was accurate.

10.3 The documentation standard used by Microsoft for this particular programme corresponds closely to that of industry standard practice. This will be clear from the comparative analysis set out below. To the extent that the internal standard is ‘bent’, it is to accommodate the additional information (such as the handling of topology) required to achieve interoperability as required by the Commission, but that does not fit neatly into the account of the protocols. Competitors and the OTR have both pointed to specifications provided by Microsoft (CIFS and XPS) that they claim to be of a higher standard. This may be, but does not, of course, imply that the standard of the WSPP documentation is inadequate. Further, neither are true comparables. XPS is essentially a document format. CIFS (Common Internet File System) is a very mature and well-established backbone specification used extensively by application developers and which has been submitted to the IETF (Internet Engineering Task Force), it addresses a different community, of a vastly different scale and in a
different context. The Trustee comments that the documentation ‘fails to conform to a minimum quality standard one could expect from a Company like Microsoft’. It seems unfair that Microsoft should in this case be penalised for the high expectations the Trustee has of their technical standards, however gratifying they may find the sentiment.

10.4 Implicit in the criticisms of the documentation supplied by Microsoft is the suggestion that Microsoft have sought to confuse or mislead. During our meetings with the senior engineers concerned with preparing the documentation we have been consistently impressed by their express determination to deliver the information required by the Decision and thus to enable interoperation. Far from seeking to confuse or mislead they appear to have sought to ensure that the fullest possible documentation was, and will continue to be provided.

11 Comparison with Open Specifications

This section summarises the comparison with an equivalent set of open specifications, OMG CORBA, acknowledged as supporting interoperability. The comparison is detailed in Appendix A. This comparison clearly shows that the WSPP documentation is similar in all significant respects to that of CORBA and that on this basis the criticisms levelled at the WSPP documentation are unreasonable.

11.1 A key plank of our analysis is an examination of the extent to which the WSPP technical documentation represents what a professional software engineer would regard as industry standard practice in providing interoperability information. We have been interested in ascertaining whether the overall quality of the documentation in respect of completeness and accuracy is reasonable. To this end we have undertaken a detailed comparison of the WSPP documentation with the OMG (Object Management Group, an industry consortium providing standards for object technologies) CORBA (Common Object Request Broker Architecture) documentation that provides a language-independent object model and specification for a distributed applications development environment.

11.2 This analysis is set out in detail in Appendix A and we commend it to the attention of the reader. We are confident that the CORBA documentation is a good comparable. Indeed, Wirsing et al in their paper Specification and Implementation of Interoperable Systems (31st October 2003) explicitly hold CORBA up to “demonstrate that specifications for full interoperability in the subject areas of this case are possible” and further that CORBA is a good analogy with “proprietary Microsoft technologies”. They point out that CORBA has created a market of competing commercial and open-source products that are interoperable. We believe it is instructive therefore to review the CORBA specification in order to understand whether the WSPP documentation ‘measures up’.
11.3 Our review shows that the CORBA specifications and the WSPP documentation: apply the same principles of information hiding; are described at the same level of abstraction; use the same on-the-wire style; provide interface descriptions in a similar manner; treat external references similarly; treat dependencies similarly; are built on the same assumption that implementers will need to devote significant effort to realising interoperability; are similar in respect of errors and defects; require user feedback in a similar manner to eliminate accidental under specification.

11.4 Overall, our consideration of CORBA demonstrates the clear similarities between it and the WSPP documentation. This supports the contention that the competitors, OTR and the Trustee have unreasonable expectations of the documentation and have not made a reasonable determination of its compliance with the Commission’s decision.

12 Sufficiency Test

This section considers the “Sufficiency Test” performed by the Trustee and the comments made in respect of that test. In Appendix B we go through the Sufficiency Test narrative in some detail and expose the set of elementary misunderstandings that account in full for the difficulties that the Trustee encountered. In Appendix C we show, on the basis of our analysis, how, in fact, the ‘add-user’ task could be accomplished using the WSPP documentation.

12.1 The “Sufficiency Test”, for which we have an account communicated to Microsoft on 17th January 2006, is an important component of the evaluation of the WSPP documentation undertaken by the Trustee, the OTR and the competitor evaluators. The Trustee has relied heavily upon it to support his assessment. For this reason we have paid special attention to this test.

12.2 Any consideration of the sufficiency test must however be immediately prefaced by a fundamental reservation. The Trustee states “I have taken as the objective one of the simplest possible operations in a distributed system environment: to receive and process a request to add a new user”. The test may at first blush seem simple because client-side ‘add-user’ is indeed very straightforward; just use the LDAP (Lightweight Directory Access Protocol, an IETF standard) interfaces or the many other routes to solving the same problem. But this reflects a misunderstanding of the selected test, and our view speaks directly to the contrary; the test chosen is one of the most difficult that could have been selected. In a distributed system, security and access rest on users being correctly added to the system. It is difficult to understand why when there were very many other, genuinely simple, tests that could have easily been used (say, the extensions to the Network Time Protocol) and that could perhaps have been completed within the time the Trustee allowed himself, even with a limited knowledge of distributed systems programming, this test was chosen. Unless, of
course, the Trustee has himself partially confused client-side ‘add-user’ with the task that the documentation was actually supposed to support. The opening sections of the sufficiency test do in fact suggest this.

12.3 There is inadequate information provided in the document of 17th January 2006 for us to comment on the second sufficiency test “to implement the directory replication services so as to propagate a password change throughout a network”. No evidence is presented in respect to this test and thus it is difficult for us attach any credence to the Trustee’s conclusion that it can be “said equally to have failed”.

12.4 In Appendix B below we go through the Sufficiency Test narrative in some detail. We expose the set of basic misunderstandings that account in full for the difficulties that the Trustee encountered. In Appendix C we show, on the basis of our analysis, how, in fact, the ‘add-user’ task could be accomplished. We note that this would, naturally, require significant effort on the part of the implementer.

13 Minor Omissions and Misunderstandings

This section considers the treatment of minor omissions and misunderstandings and explains how they might be resolved by recourse to the technical support offered to WSPP licensees.

13.1 The OTR, the competitor evaluators and to a limited extent the Trustee have identified areas in the documentation where they believe there are omissions. These are generally minor and relate in part to misunderstandings by the evaluators, often about where information can be found (see for example the OTR comments on the absent description of the backup key information which is, in fact, described in some detail). We have decided not to go through these individually as it will not serve the matter under consideration. These are precisely the sort of matters that we would anticipate being resolved rapidly by recourse to technical support. Omissions, of which there are some, (the Domain Authority Function of Passport is a possible example) could be handled when the documentation is updated and this is provided for in the license.

14 Server Source Code

This section analyses the significance of the offer by Microsoft to make available source code for reference and concludes that, while not required it nevertheless constitutes a significant complement to the WSPP documentation.

14.1 The original decision was clear in not requiring Microsoft to release source code. The Statement of Objections, to the extent that it is consistent with Article 5, also does not explicitly require the release of source code. This being said, the announcement by Bradford Smith ‘Windows Server Source Code Announcement’ and the associated press release ‘Microsoft Goes Beyond EU Decision by Offering Windows Source Code’ 25th January 2006 constitute a significant development which must be taken
into account in relation to any assessment of the WSPP documentation. The essence of Microsoft’s offer is to allow reference access by WSPP licensees to the relevant Windows server source code. While such access is not a necessary requirement in order use the WSPP documentation there is no doubt that it helps to see how Microsoft have implemented to their own specifications. In our assessment of the ‘add user’ sufficiency test we confirmed our understanding using, as appropriate, the open-source SAMBA code, a validation step that we are confident adds weight to our assessment. Clearly this would have been somewhat easier had we had reference access to the source code in the manner envisaged. Viewed overall this must be regarded as a significant complement to the WSPP documentation.

15 Conclusion

This section presents our conclusions. We determine that the WSPP documentation has been reasonably calculated to provide the technical basis for a market in interoperable server operating system products and that it is in these terms complete and accurate.

15.1 Perfection is not a reasonable requirement for the WSPP documentation nor is it necessary, in our view, to allow a competitor to Microsoft to deliver an interoperable product. The WSPP documentation undoubtedly contains bugs; there are minor usability issues that need to be addressed; the balance between what is included and what externally referenced is not perhaps made correctly in every instance; there may well be minor omissions. Software engineering is a pessimistic discipline; the optimistic programmer who believes that his or her code is error-free is usually bound for early disappointment. The critical stance that this necessitates, and which has been adopted by the Trustee, should not however blind one to what is reasonable to achieve in terms of software documentation. Microsoft have produced documentation that closely accords with standard industry practice, this is conclusively demonstrated by our comparative analysis.

15.2 We have been struck in reviewing the comments of the competitors, the OTR and the Trustee with the extent to which they assume omniscience on the part of Microsoft: the idea that Microsoft should know, exactly, the dependencies between components and should be able to prescribe, exactly, all the combinations of circumstances that might give rise to particular errors. This is simply not realistic. Furthermore, Microsoft never anticipated that it would be required to disclose any of this information.

15.3 It seems clear to us that the documentation supplied by Microsoft goes some way beyond the specification of protocols and on-the-wire observability. There are many examples, Server Rules for Directory Service Topology to pick just one, where
dependencies and implicit knowledge are treated in detail. This seems to us to reflect a concern not just with the letter of interoperability but also with its spirit.

15.4 Our analysis of the sufficiency test, which is set out above, and the very basic misunderstandings in respect of RPCs and protocol specification, that we have identified in the Trustee’s reports suggest that little reliance can be placed upon them. The Trustee has correctly identified some usability flaws but we do not believe that these are fundamental, nor that they are of any significance in respect of the overall completeness and accuracy of the documentation. Many of the points raised by competitor evaluators have been covered above. Others, such as the request for a test-suite, while reasonable in their own terms are substantially outside the scope of the decision. None of the competitor evaluators have framed their comments in the context of a potential interoperable product and we must therefore suspect that they are in large part speculative.

15.5 In conclusion, and based on our independent technical assessment, it is our opinion that the WSPP documentation has been reasonably calculated to provide the technical basis for a market in interoperable server operating system products; that it is in these terms complete and accurate.
Appendix A: Comparison with CORBA

In this Appendix we present an analysis of comparable open documentation: OMG CORBA. Our analysis demonstrates the clear similarities between this and the WSPP documentation thus supporting the contention that the competitors, OTR and the Trustee have unreasonable expectations of the WSPP documentation and have not made a reasonable determination of its compliance with the Commission’s decision.

A1 CORBA was defined by OMG in a process that commenced in 1989 and is continuing to date. CORBA’s aim is to achieve the ability to invoke operations implemented by network objects, as defined by (Birrell, et al 1993) across machine, programming language and operating system boundaries. CORBA is not just one specification; it is (like the WSPP documentation) a family of related, in this case openly available, specifications. The specifications fall into three categories the ORB specifications, the specifications of CORBA services and the specification of domains-specific CORBA facilities. On Microsoft’s platforms, COM and DCOM fulfil similar functions to the ORB in CORBA. The knowledge required to use COM and DCOM is in the public domain, in part due to a submission by Microsoft of COM as an open standard to the OpenGroup.

A2 The WSPP specification is closest in nature to the OMG’s CORBA services specifications. These are some 15 specifications that determine higher-level services, such as security, transactions, concurrency control, naming, event and notification etc.

A3 Unlike the WSPP documentation that was constructed a posteriori for the purpose of the required disclosure, all OMG specifications were defined in a public and open process during which all parties that registered an interest in a particular effort defined the specification through drafting and public review in a strictly consensus-based manner. The total size of the CORBA specifications is of the same order of magnitude as that of WSPP (approximately 10,000 pages). CORBA services specifications were only adopted once there was an implementation of the product available.

A4 Because all CORBA specifications were defined by competing organisations, it is not very surprising that each task force ensured that there was ample room for vendors to differentiate their products from those of their competitors. A guiding principle of all CORBA specification efforts was therefore to avoid over-specification and to only define the minimal interfaces, protocols and algorithms required for different implementations to remain interoperable. In doing so, the CORBA specifications apply the principle of information hiding. They strictly distinguish interfaces from implementations and in their specifications focus on interfaces and protocols rather than algorithms and implementations. The following examples from the CORBA security service specification serve to illustrate these points:
Implementor’s View of Secure Object Creation

When an object is created in a secure environment, it is associated with Security Policy, Environment, and Technology domains […] The way it is associated with Environment and Technology domains is ORB implementation-specific, and therefore not described here. [page 15-85]

The SecurityLevel2::AuditChannel Interface

audit_write

This operation writes an audit record to the Audit Channel object, and hence the audit trail. The audit trail is implementation-specific and outside the scope of this chapter. [page 15-115]

This [Security service] specification contains only the management interfaces, which are essential for security policy management. It specifies how to obtain and use security policy objects. However, it does not contain:

- Specification of facilities for handling domains, or policies other than those required for security policy administration.
- Specification of facilities for the management of some aspects of security. For example, it does not specify how to create and install permanent keys, as this is implementation-specific. [15-361]

A5 The criticisms that the Trustee has levelled at the WSPP documentation could thus be equally levelled at the CORBA specifications. It might be instructive to compare the level of abstraction with which the CORBA services specifications are defined to those in the WSPP documentation. To perform this comparison, we refer to the ISO/OSI reference model shown in Figure 6.

![ISO/OSI Reference Model](image)

Figure 6: ISO/OSI Reference Model

A6 CORBA services and WSPP both rely on TCP or UDP Transport layer implementations and would in all but very rare circumstances use the IP Network protocol. The differences arise in how Presentation and Session layers are implemented. An Object Adapter handles the session layer implementation in
CORBA, while in WSPP it would be handled by COM’s Service Control Module. The presentation layer implementation defines the on-the-wire representation. In CORBA’s case this is the External Data Representation (XDR), defined as part of the IIOP protocol into which application specific data structures are marshalled while for WSPP this would be the transfer syntax NDR defined by the Open Group as part of DCE and extended by Microsoft. The interface definitions of all CORBA services are defined in CORBA IDL, while the interfaces in the WSPP documentation are defined in Microsoft’s IDL, a variant of the OSF/DCE IDL. A CORBA/IDL compiler generates the CORBA stubs and skeletons that perform the transformation into IIOP. Interface proxies for calling WSPP services would be generated by the MIDL compiler that is part of any Microsoft SDK. If Microsoft’s competitors wanted to host these servers in a Linux or Unix environment (for which no Microsoft SDK is available), they could use CORBA in their servers and build on the COM/CORBA interworking specification defined by the OMG. Implementations of that specification are available from Iona, Expersoft and Borland. It is, of course also possible to buy the source code of a COM/DCOM/MS RPC-implementation Unix from the Open Group (http://www.opengroup.org/comsource/). This means it is both feasible and practical to invoke interfaces defined in the WSPP documentation from non-Microsoft platforms.

The level of detail at which interfaces are specified in the CORBA services specifications and the WSPP specification are strikingly similar. For each operation, parameters, failure conditions and return values and/or output parameters are explained. Even though there are dependencies between CORBA services (e.g. between persistence & externalization, between concurrency & persistence, between transactions & concurrency), these dependencies are not explained or highlighted in the CORBA services specifications and there is also no hint in the CORBA specification as to how these specifications interplay with each other. The same may be true for some of the protocols defined in the WSPP disclosure and for the same reason as for the CORBA specifications, they are not known.

Both the CORBA specification and the WSPP documentation provide normative references that are specified to be as precise and concise as possible. The trade-off is that such specifications are often not easily understandable. Nobody who wants to understand CORBA, build a CORBA-based distributed system, or build interoperability products based on CORBA would exclusively rely on, or even start by, consulting the CORBA specifications. They would instead use informative references. In case of CORBA these are provided outside the OMG specification in a series of professional books published by John Wiley and Sons (for example, Duddy & Vogel 1997); engineers also use training courses, reference examples provided by organizations other than OMG and academic text books for that purpose.
A9 A key concern underlying the WSPP specification and for the OMG in defining CORBA is interoperability across heterogeneous products produced by competing organizations. The ORBs that implemented the first public specification (CORBA 1.1) were not interoperable at all, even though CORBA was an open standard agreed through a consensus based standardization process. The reason is that the standard neglected to define the transport layer protocol and left ORB vendors to freely determine representations of object references. It took the industry until 1994 when CORBA 2.0 was adopted to define the required interoperability protocol (IIOP). In other words it was the on-the-wire protocol specification, precisely what Microsoft have disclosed, that was a key enabler of interoperability.

A10 Even with the publication of CORBA 2.0 however interoperability was not really working. To get this protocol to work it took a demonstrator implementation, CORBANet developed by DSTC in Brisbane, Australia. The Minutes of the OMG Test Working Group Meeting held in Tampa on January 14th 1997 give an interesting insight into which ambiguities the CORBANet effort found in the IIOP specification that were subsequently resolved. The industry defined a compliance test suite developed by the OpenGroup (http://www.opengroup.org/testing/testsuites), an open source reference implementation in MICO (http://www.mico.org), and revised the IIOP protocol itself. IIOP implementations were basically unusable before 1998; that is nine years after OMG got started! IIOP alongside SOAP is now of course one of the most important interoperability protocols in the industry as it is used as transport protocol in Remote Method Invocation (RMI), Enterprise JavaBeans (EJBs) and of course the remaining CORBA ORBs.

A11 The principal concerns raised with respect to the WSPP documentation are completeness and accuracy. These were, and still are, a great concern for the OMG. Despite many implementers there are a large number of defects in the CORBA specification. The issues database of OMG is publicly available at http://www.omg.org/issues and it contains in excess of 10,000 issues, many of which are unresolved to date. For example, the CORBA Transaction Service Revision Task Force that is working on a new release of the Object Transaction Service is currently addressing 102 specification defects of varying severity for just a single CORBA service. Assuming that specifications for technology that is as complex as CORBA or WSPP can be specified in such a way as to genuinely support interoperability, completely and accurately in a single shot is naïve in the extreme. We note also that even in the presence of all these defects, the CORBA specifications are extremely valuable and widely used. OTS forms the basis of the Java Transaction Service, which is a cornerstone of the Enterprise Java Bean specification on which interoperable application servers are built: a market with a size of $1bn in 2002 according to Gartner.
It is also worthwhile observing that CORBA specifications occasionally had to be revised to include details that were deliberately or accidentally underspecified earlier. A famous example is the specification of the Basic Object Adapter in CORBA 1.1. CORBA’s object adapters play the same role as the Service Control Module in COM. They dispatch requests and control object-activation and de-activation. Both are server-side interfaces. The initial BOA specification did not specify precisely enough how object activation, reference and identifier management should be handled: this was initially seen as an implementation detail. This however led to incompatible implementations. It was rectified with the introduction of the Portable Object Adapter in CORBA 1.4.
Appendix B: Sufficiency Test Analysis

In this Appendix we go through the Sufficiency Test narrative presented by the Trustee in some detail and step-by-step. We expose the set of basic misunderstandings that account in full for the difficulties that the Trustee encountered.

B1 Network Object
Page 5 WSPP Documentation Sufficiency Test

“it is by no means obvious what a ‘network object’ might be”

‘Network object’ is a commonly used term for an object-oriented representation of hard- or software elements in a computer network. In particular, the term is used for elements in directory services based on the ISO X.500 standard, such as Novell eDirectory and Microsoft Active Directory.

“The real eDirectory tree is made up of logical network objects”
(http://www.examcram2.com/articles/article.asp?p=169521)

“Active Directory will store information about organizations, sites, systems, users, shares, and just about any other network object that you can imagine”
(http://www.netscum.dk/technet/prodtechnol/windows2000serv/plan/int2ksrv/intro11.mspx)

The term ‘network object’ is used in many software architectures, it has a clear meaning in the given context of a directory services.

B2 Context Handle
Page 5 WSPP Documentation Sufficiency Test

The Trustee states, that he has “no idea what a 'context handle' might be” and that – after reading the document – he “remains completely unable to explain”, what a 'context handle' might be.

A context handle in a technical term in Open Groups Distributed Computing Environment (DCE) Version 1.1 Remote Procedure Call (RPC): “In many interfaces, a called procedure needs to maintain state between calls. This is done by means of a context handle. A context handle is a void* with the context_handle attribute.”
(http://www.opengroup.org/onlinelpubs/009629399/chap4.htm#tagcjh_08_02_16_06)
This public, well-known document is referenced frequently in the Samr chapter of the WSPP documentation (e.g. 9820, 9828, 9841, 9870, 9910).
Page 9820:

For more information about the structure and sequence of data on the wire, see the Transfer Syntax NDR topics in the CÆ Specification, DCE 1.1: Remote Procedure Call, Document Number C706 at http://go.microsoft.com/fwlink=3497717

A public Microsoft website, the very first hit at Google for “Context Handle”, provides the definition for a Context Handle in the Microsoft Interface Definition Language (MIDL), which is part of Microsoft RPC:
“The [context_handle] attribute identifies a binding handle that maintains context, or state information, on the server between remote procedure calls.”

The WSPP documentation refers in the description of its “Common Data Types” (a chapter below “Networking Protocols / General Information”) to this definition (Page 9076).

Page 9076:
handle_t
A context handle. Its implementation is server-specific, so its exact size and composition fall outside the scope of these documents.

Searching for “Context Handle”, Google will reveal many other explanations of this topic, related to Windows (e.g.: http://www.aspfree.com/c/a/.NET/Introduction-to-RPC-on-Windows-Part-II/1/) as well as UNIX (http://bama.ua.edu/cgi-bin/man-cgi?rpc_gss_svc_max_data_length+3NSL).

B3 SAMPR_HANDLE
Pages 7-8 and 16-17 of WSPP Documentation Sufficiency Test

The Trustee assumes, that a specification for the structure SAMPR_HANDLE is needed, because it is a “fundamental requirement given the operation of samr and related functions”.

This is a totally incorrect conclusion. A specification of the SAMPR_HANDLE is not necessary for any kind of communication or interoperability.

According to page 9775 (Introduction to the samr interface) and page 9828 (description of SamrConnect5) a SAMPR_HANDLE and is an RPC context handle.

Page 9775:
The SAM interactions are provided through a remote procedure call (RPC) protocol interface. The version for this interface is 1.0. The interface uses context handles to maintain client state information. The server returns a context handle when a client method call requires one.

Page 9828:
ServerHandle
A SAMPR_HANDLE pointer that points to the handle to use in future calls. This handle represents both the handle to the SAM server object and the Remote Procedure Call (RPC) context handle for the connection to the SAM subsystem.

An RPC context handle is a standard, well-known programming construct (e.g.
http://www.aspfree.com/c/a/.NET/Introduction-to-RPC-on-Windows-Part-II/1/) to maintain state between several related function calls.

An RPC handle is issued by the RPC server (here: the implementation of SamR) and received by the caller; therefore SAMPR_HANDLE is an outgoing parameter of the function SamrConnect5 that starts the communication between the parties. The caller provides a pointer that will be filled by the server with an address to data the server needs. In the next function call (e.g. SamOpenDomain), the caller will transfer this pointer back to the server. The server might change the handle during the function call and return the same or another
pointer to the caller who can use the pointer for the next function call (e.g. SamrCreateUserInDomain) and so on.

There is absolutely no need for a user of the SamR interface, neither for a client nor a server, to understand the actual content of the SAMPR function or even to know its structure. A context handle can be absolutely implementation specific and any developer of an operating system can use any kind of data structure without interfering with interoperability on the part of any other party. The caller’s obligation is just one very simple thing: send back the received handles to the server with any upcoming function call. Microsoft could change the implementation of the SAMPR_HANDLE between different versions of Windows, different service packs or even between two server restarts without interfering with any client.

This use of Context Handles is a common software engineering concept. The same concept is also used in state-of-the-art Internet and Intranet applications using the stateless HTTP protocol and cookies for maintaining state: Any web browser (HTTP client) can communicate with any web server (HTTP server) and use its web applications without any knowledge about the actual structure or the internals of the cookies issued by the server.

The WSPP documentation refers to this concept of context handles in chapter “General Information / Common Data Types” of the WSPP documentation (Page 9076).

Page 9076:

| handle_t | A context handle. Its implementation is server-specific, so its exact size and composition fall outside the scope of these documents. |

B4 Function Call Order
Page 7 of WSPP Documentation Sufficiency Test

The Trustee makes his own assumptions about the order of necessary method calls in order to perform the Add-User-operation with the justification that the “preliminary text for the list of operations ... does not make this” clear. As part of his assumptions he decided to start looking at the connection operation before considering the Add-User-operation.

It is the nature of most modern application programming interfaces and protocols that calls can be issued in a lot of different combinations and sequences. To describe all possibilities would be a tough job. In such cases it is reasonable to start looking at the function that performs the actual task instead of infrastructure requirements around this task. If a developer chose this approach with the WSPP documentation he or she could detect the method call order for the Add-User-operation within a few minutes.

The documentation of the SamrCreateUserInDomain function states that a SAMPR_HANDLE is needed as a parameter and this can be obtained from the SamrOpenDomain function (“ServerHandle: A SAMPR_HANDLE structure which store the domain handle, which is optained from a previous SamrOpenDomain”, page 9834).

Looking at the SamrOpenDomain one would see that the SamrOpenDomain function requires a SAMPR_HANDLE as well, but this time the handle has to be obtained from the function named SamrConnect5. (“ServerHandle: A SAMPR_HANDLE structure that stores the server handle, which is obtained from a previous SamrConnect5 call”, page 9866).
The SamrConnect5 function, however, contains SAMPR_HANDLE as an outgoing parameter, i.e. SamrConnect5 is the source of the context handle and the beginning of the call chain.

The resulting function call order for the Add-User task is:
SamrConnect5 → SamOpenDomain → SamrCreateUserinDomain

It cannot be difficult to find this out, because a few of the possible method call orders for the Samr interface are described on a public website:
http://www.hsc.fr/ressources/articles/win_net_srv/ch04s07s03.html

A public resource at Microsoft.com defines the general use of Samr: “The Security Account Manager Remote Procedure Call (RPC) protocol (SAMR) is an integral subsystem that is used to perform remote Service Account Manager operations, such as user account management and manipulation. The SAMR interface defines the remote Security Account Manager (SAM) methods that are called by the client. SamConnect is the function that is used to connect to the SAM database.” In: Security Account Manager (http://technet2.microsoft.com/WindowsServer/en/Library/0f79c448-0dba-497a-b1bf-2fe5b09ff56c1033.mspx)

Also, SAMBA, an open source toolkit for Unix-Windows interoperability, is successfully using the Samr interface (http://websvn.samba.org/cgi-bin/viewcvs.cgi/branches/SAMBA_4_0/source/librpc/idl/samr.idl?view=markup and http://www.hands.com/~lkcl/samr/samr.idl)

### B5 Void* SAMPR_HANDLE

The Trustee states, that the SAMPR_HANDLE is “explicitly hidden from view”, “not defined anywhere in the documentation but rather deliberately hidden”. He argues, that this is the reason, why he got stuck.

The use of void* for a context handler such as SAMPR_HANDLE is absolutely correct according to Open Group DCE 1.1 RPC:

“In many interfaces, a called procedure needs to maintain state between calls. This is done by means of a context handle. A context handle is a void* with the context_handle attribute.”
(http://www.opengroup.org/onlinepubs/009629399/chap4.htm#tagcjh_08_02_16_06)

“The void keyword is valid only in an operation or pointer declaration. In an operation declaration, it may be used to indicate an operation that does not return a function result value. In a pointer declaration, it must be used in conjunction with the context_handle attribute.”
(http://www.opengroup.org/onlinepubs/9629399/chap4.htm)

Microsoft explicitly states in the WSPP documentation, that their RPC implementation is compatible to Open Group DCE (Page 9016)
The Microsoft Implementation of RPC is not DCE-compliant, some extensions have been made to the original DCE RPC specification. However, it is compatible with the Open Software Foundation (OSF) Distributed Computing Environment (DCE) standard RPC protocol. The changes are largely confined to the respective areas of transfer syntax support.

The WSPP documentation explicitly explains, why the structure is hidden (Page 9076).

**Page 9076:**

```c
handle_t
```

A context handle. Its implementation is server-specific, so its exact size and composition fall outside the scope of these documents.

This basic RPC concept of declaring a context handle as void* is well understood and presented on public websites:

“Most of the context handles that you see while programming Windows are more or less opaque, but in the RPC world the context handles are 100 percent opaque, since they are defined to be pointers of type void*.


There is no need to, and no sense in, disclosing any internals of a context handler structure.

**B6 Ntstatus.h**

Page 10f of WSPP Documentation Sufficiency Test

The Trustee comments that he cannot use the file “ntstatus.h” in its printed form with the documentation and there is “no explanation” where to find the file.

This is wrong. Ntstatus.h is part of Microsoft SDKs and also available on other websites.

“For a complete list of exception codes, see the Ntstatus.h file located in the Inc directory of the Windows XP Professional Driver Development Kit (DDK). For more information about the DDK, see the Driver Development Kits link on the Web Resources page at http://www.microsoft.com/windows/reskits/webresources.”


Another external source for this file is:

http://cvs.sourceforge.net/viewcvs.py/mingw/w32api/include/ddk/ntstatus.h?rev=1.2

Page 9108 of the documentation, a part of the General Information chapter, explains where to obtain header files such as nhstatus.h.

Page 9108:
Obtaining Header Files

The Microsoft Platform Software Development Kit (SDK) contains header files and other information that you need to develop applications for Microsoft Windows. The Platform SDK is available as part of Microsoft Visual C++, a Microsoft Developer Network (MSDN) Professional subscription, an MSDN Universal subscription, or as a download from the Platform SDK Update site:


The Platform SDK is released more frequently than Visual C++. Use the Platform SDK Update site to ensure that you have the latest documentation, samples, and Platform SDK build environment (header files, libraries, and tools).

For more information about Platform SDK content included with Visual C++, see Installing the Platform SDK with Visual Studio.

A low-cost Platform SDK CD-ROM is available from the following location:

http://www.omegaj.ca/saurchiopdf.htm

The MSDN download site contains additional development kits, such as the .NET Enterprise Server SDKs, the .NET Framework SDK, the Microsoft Speech SDK, the Pocket PC SDK, the Mobile Information Server SDK, and the Driver Development Kit (DDK).

B7 Necessary Security Information
Page 7-9 and 12 of WSPP Documentation Sufficiency Test

The Trustee describes difficulties in finding information about the 'necessary security information' and the ACCESS_MASK.

Every description of SamR interface functions that require special access rights, clearly defines this requirement its explanatory header text (e.g. page 9834 and 9828):

Page 9828:

[____]

Page 9834:

[____]

Access Masks are required parameters when calling SamrConnect5 (9828), SamrOpenDomain (9866) and CreateUserInDomain (9834). As explained on these pages, the access mask is the kind of access the client desires.

All of the function pages contain a certain link to the chapter “Security Account Manager Remote Protocol Data Types” either in the explanatory header text (Page 9828) or the description of the ACCESS_MASK parameter (Page 9866 and 9834). For example, the description of the function SamrConnect 5 (Page 9828) contains a link to the sub-chapter Server Access Values, that specifies the different types of access rights to a SAM server object (page 9800). Page 9828 also declares that the caller must have the access type SAM_SERVER_CONNECT, which is implicitly included in the value.
The SamrConnect6 method establishes a session with a Security Account Manager (SAM) subsystem and opens the SAM server object of that subsystem. The caller must have SAM_SERVER_CONNECT access to the SAM server object. See Server Access Values for definitions and values of access types.

... DesiredAccess

An ACCESS_MASK structure that stores an integer indicating the desired access to the SAM server. These access types are reconciled with the SAM server's discretionary access control list (DACL) to determine whether the accesses will be granted or denied. The access type of SAM_SERVER_CONNECT is always implicitly included in the value.

However, the SamCreateUserInDomain function requires the DOMAIN_CREATE_USER right. There is a link to “Domain Access Values” on the first page of the description of SamCreateUserInDomain.

The SamCreateUserInDomain method adds a new user to the account database. The caller of this method becomes the owner of the user object upon creation. This method requires DOMAIN_CREATE_USER access.

... DesiredAccess

An ACCESS_MASK structure that stores an integer that indicates the desired accesses to the domain. These access types are reconciled with the domain's discretionary access control list (DACL) to determine whether the accesses will be granted or denied. See Domain Access Values for definitions and values of access types.

Following the links to “Server Access Values” and “Domain Access Values” the user of the documentation will be navigated to the chapter “Security Account Manager Remote Protocol Data Types”.

The sub-chapter “Server Access Values” defines and explains the value for SAM_SERVER_CONNECT (Page 9800) whereas the sub-chapter “Domain Access Values” defines and explains the value for DOMAIN_CREATE_USER (Page 9792).

The chapter “Security Account Manager Remote Protocol Data Types” also contains a link to a chapter called “Access Mask” within the “Local Security Authority (Domain Policy) Remote Protocol”. This states, that Access Mask is a bitmask. A bitmask is a well-defined term in computer programming. “The most common mask used, also known as a bitmask, extracts the status of certain bits in a binary string or number…” (http://en.wikipedia.org/wiki/Bitmask).
ACCESS MASK

The ACCESS_MASK data type is a bitmask that defines the access rights to grant an object. The bitmask can be any combination of the following values. Access types are reconciled with the discretionary access control list (DACL) of the object to determine whether the access requested is granted or denied.

The description of the Samr functions clearly states that that a caller of a function has to specify an ACCESS_MASK for the desired access (pages 9828 und 9834). ACCESS_MASK is defined as a combination of values (here: the Server Access Values or the Domain Access Values).

The descriptions of access values contain different predefined combinations of access masks, e.g. DOMAIN_WRITE (9793), including DOMAIN_CREATE_USER. There is no doubt, that DOMAIN_WRITE will be sufficient to perform the Add-User task.

Page 9793:

The caller can easily pass the constant MAXIMUM_ALLOWED_ACCESS. This method is for example used in the SAMBA project source code:

```
#define MAXIMUM_ALLOWED_ACCESS 0x02000000 (smb.h)

uint32 access_mask = MAXIMUM_ALLOWED_ACCESS; (Cmd_samr.c)
```

The Trustee states that the formal structure of a Discretionary Access Control List is not apparent.

Again, an Access Control List is a common term in operating systems:

“The access control list (ACL) is a concept in computer security used to enforce privilege separation. It is a means of determining the appropriate access rights to a given object depending on certain aspects of the process that is making the request. … The list is a data structure, usually a table, containing entries that specify individual user or group rights to specific system objects, such as a program, a process, or a file. These entries are known as access control entries (ACE) in the Microsoft Windows and OpenVMS operating systems. Each accessible object contains an identifier to its ACL. The privileges or permissions determine specific access rights, such as whether a user can read from, write to or execute an object. In some implementations an ACE can control whether or not a user, or group of users, may alter the ACL on an object.” (http://en.wikipedia.org/wiki/Access_control_list).
A Discretionary Access Control List (DACL) is a well-known type of ACL used in the Windows operating system. Searching for “Discretionary Access Control List” in Google reveals about 40,400 web pages containing this term.

Public documents on the Microsoft web pages for administrators (TechNet) as well as developers (MSDN) explain the term DACL:

“A discretionary access control list (DACL) identifies the trustees that are allowed or denied access to a securable object. When a process tries to access a securable object, the system checks the ACEs in the object’s DACL to determine whether to grant access to it. If the object does not have a DACL, the system grants full access to everyone.” In: Access Control List (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/secauthz/security/access_control_lists.asp)


“An object’s security descriptor can contain two types of ACLs: A discretionary access control list (DACL) that identifies the users and groups who are allowed or denied access. A system access control list (SACL) that controls how access is audited” In: What Are Security Descriptors and Access Control Lists? (http://technet2.microsoft.com/WindowsServer/en/Library/d4f08d96-f360-451f-bed3-61a60bc2acde1033.mspx).

These web pages also explain:

the relationship between ACLs and other Authorization and Access Control Components in Windows (e.g. Security Descriptors, Security Identifiers, Access Control Entries) (http://technet2.microsoft.com/WindowsServer/en/Library/d4f08d96-f360-451f-bed3-61a60bc2acde1033.mspx)

the use of access masks with the ACLs: “Instead of hard-coding access types such as “the right to open, close, read from, or write to an object,” access in Windows NT is defined as a collection of bits in a mask. The security subsystem matches the bits in the user's access mask with the bits in the object's access mask” (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsecure/html/msdn_seccpp.asp)

the reconciliation between the callers access masks with the access mask of the desired object, that happened within the internal function AccessCheck. (http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnsecure/html/msdn_seccpp.asp)
Also, the public MSDN web pages deliver numerous code examples in different languages of how to create a Discretionary Access Control List (DACL) e.g.

for C/C++: “Creating a proper discretionary access control list (DACL) is a necessary and important part of application development. Because a NULL DACL permits all types of access to all users, do not use NULL DACLs.” in: “Creating a DACL”

for Visual Basic: “The following code example defines a function that adds an Access Control Entry (ACE) to the Discretionary Access Control List (DACL) of the security descriptor of a specified Active Directory object.” In: Example Code for Setting an ACE on a Directory Object”

The documentation of the function AccessCheck is not included in the WSPP documentation. There is documentation of a homonymous function in the Windows Management Instrumentation Remote Protocol (11065 ff.). However, an explicit call to AccessCheck is not necessary for any caller (client or server) as any function that accesses protected resource will implicitly call AccessCheck. The logic of AccessCheck is obvious and also described in public documentation. Other details about AccessCheck can be considered as part of the specific implementation of a server that a caller does not need to know.

B9 Authentication
Page 13-16 of WSPP Documentation Sufficiency Test

The Trustee researches the Net Logon Remote Protocol for the Authentication of the Add-User tasks.

Authentication is done as part of Microsoft’s implementation of the Server Message Block Protocol (SMB) see http://en.wikipedia.org/wiki/Server_message_block, this implementation is available at ftp://ftp.microsoft.com/developr/drg/CIFS/. The SamR interface does not have to deal with authentication.

Page 2211:

For authentication, SMB2 uses the standard procedures of RFC 2478 (GSS-API), which allow a client or server to call for authentication independently of the final choice of authentication method. For more information, see RFC 2478.

For SMB2, the selected authentication method is either Kerberos or NT LAN Manager (NTLM). By default, networked Windows 2000 or Windows XP platforms call for authentication using Kerberos. For a Windows 2000 or Windows XP platform, both Kerberos and NTLM Security Support Provider (SSP) authentication components are loaded at startup.

Microsoft applications do not authenticate inline, but make a Security Support Provider Interface (SSPI) Negotiate call to request authentication. A Negotiate call selects the appropriate SSP component to handle the request. As a result, networked Windows 2000 and Windows XP platforms attempt to authenticate using the Kerberos SSP; stand-alone and older
For authentication, SMB2 uses the standard procedures of RFC 2478 (GSS-API), which allow a client or server to call for authentication independently of the final choice of authentication method. For more information, see RFC 2478.

For SMB2, the selected authentication method is either Kerberos or NT LAN Manager (NTLM). By default, networked Windows 2000 or Windows XP platforms call for authentication using Kerberos. For a Windows 2000 or Windows XP platform, both Kerberos and NTLM Security Support Provider (SSP) authentication components are loaded at startup.

Microsoft applications do not authenticate inline, but make a Security Support Provider Interface (SSPI) Negotiate call to request authentication. A Negotiate call selects the appropriate SSP component to handle the request. As a result, networked Windows 2000 and Windows XP platforms attempt to authenticate using the Kerberos SSP; stand-alone and older
In this Appendix we show that, in fact, the Add User task can be accomplished using the information in the WSPP documentation. We outline two methods, ascertainable from the documentation, one not considered by the Trustee, that allow the task to be performed.

C1 Through our analysis of the WSPP documentation and additional public material we have established that it is in fact possible to use it to realise a Windows-conformant implementation of the SamR interface so as to provide the functionality of adding a user to the server’s user database. The tasks of adding a user require the use of the functions SamrConnect5, SamOpenDomain, SamrCreateUserinDomain and SamRCloseHandle. All parameters of these functions are well described by the WSPP documentation. The handling of state between these functions using SAMPR_HANDLE context handlers is obvious.

C2 Transportation and authentication for SamR calls are provided by the underlying Microsoft RPC- and SMB-protocols in connection with the NT LAN Manager Protocol. These protocols are described within the WSPP documentation (SMB: Page 1686, MS RPC: 9005, NTLM: Page 10660). Additional information is available on the Internet:

http://www.opengroup.org/onlinepubs/009629399/

C3 We validated our approach by looking at the source code of the SAMBA Unix-Windows-interoperability toolkit that contains a working implementation of SamR. Of course any WSPP licensee would have reference access to the relevant Windows source code should they lack confidence.

C4 We can see no reason why any potential competitor to Microsoft could not implement a SamR interface with the information provided. There is, of course, no guarantee that an implementation effort would not be very time consuming, though not because of the SamR interface itself, but because of the need to implement the underlying transport and authentication protocols.

C5 An alternative is possible: LDAP based User Management. With Windows 2000, Microsoft introduced the Active Directory. Active Directory uses an alternative protocol stack based on LDAP and Kerberos. As can be seen from http://support.microsoft.com/kb/179442/en-us RPC and SMB are still used for some of the Active Directory functions. An Active Directory can work in a “native mode” (that is, user management totally relies on LDAP) or a “mixed mode” (that is, supporting LDAP und still the SamR interface). The alternative approach to user
management based on LDAP was not looked at, or even mentioned, in the WSPP Documentation Sufficiency Test conducted by the Trustee.

C6 In this case implementing the server side of native mode "Add User" requires: (1) implementing an LDAP server; (2) implementing a Kerberos Key Distribution Centre (KDC); (3) assigning a Security Identifier (SID) to the user account in accordance with the Active Directory Security Account Manager logic [the SID must be from a range reserved from the Domain Controller that holds the Relative Identifier Master Flexible Single-Master Operation Role (RID-FSMO)]; (4) assigning a Global Unique Identifier (GUID).

C7 The implementation of steps 1 and 2 could be achieved using the public specification of LDAP in RFC 3377 (http://www.ietf.org/rfc/rfc3377.txt); the detailed description of the Active Directory Schema on page 3331 ff. in the WSPP documentation; the public specification of Kerberos in RFC 4129 (http://www.ietf.org/rfc/rfc4120.txt); the description of the Kerberos Interactive Logon Protocol on page 9206 ff. in the WSPP documentation. The implementation of step 4 appears straightforward as a Global Unique Identifier (GUID) is a pseudo-random identifier that can be generated in any software without interacting with other systems or services. For step 3 the observable functions of the RID-FSMO, the structure of a SID and the process of RID Allocation are documented on the public Microsoft website:


The relevant protocol information can be found in the WSPP documentation “Single-Master Role Operations” in the “Server Rules for Directory Management” section of “Active Directory Further Explanatory Material and Server Rules”.

C8 As with the SamR approach some caution is required. To implement this may, depending on the implementation strategy selected, entail a significant effort on the part of the implementer and the relative simplicity of the approach can belie the overall complexity of actually achieving it.

C9 It may be useful in concluding to note that determining both methods for accomplishing the task were completed by a single team member in substantially less time than the Trustee’s failed attempt.
Appendix D: Qualifications and Experience of Team

Prof. Anthony Finkelstein

Anthony Finkelstein a graduate in systems engineering holding a BEng, MSC and PhD. He is Professor of Software Systems Engineering at University College London (UCL), a leading UK research university, where he works in the broad field of software systems engineering. He has published more than 180 scientific papers and secured more than £15m of research funding. He is a Fellow of both the Institution of Electrical Engineers and the British Computer Society and has served on their senior committees. He has been recognised for his contributions to the field of requirements engineering and for his professional service by the IEEE. In 2003 he was a winner of the prestigious International Conference on Software Engineering 'most influential paper' prize for work on 'viewpoints' and in 2004 was winner of the Requirements Engineering 'most influential paper' prize for work on traceability. He has served on numerous editorial boards including that of ACM TOSEM and IEEE TSE, and was founder editor of Automated Software Engineering. The 'state-of-the-art' review he edited remains the publication with the highest impact factor in software engineering. He also chaired numerous international meetings and was General Chair of the International Conference on Software Engineering 2004. He was keynote speaker at Automated Software Engineering 2003 in Montreal, Canada. He is outgoing Chair of IFIP WG 2.9 (Software Requirements Engineering) an international research society. He established a leading research group in software systems engineering at UCL and played a key role in the foundation of London Software Systems, a major new research institute. He is now Head of the Department of Computer Science at UCL. He serves on the UK Research Assessment Exercise panel for Computer Science and Informatics and is a member of the Committee of Visitors for the US National Science Foundation. He has provided consultancy advice to a large number of high profile companies and organisations internationally. He is Founder and Director of Systemwire Ltd. a UCL spinout technology company.

Professor Wolfgang Emmerich

Wolfgang Emmerich graduated from the Universitaet Dortmund and obtained his doctorate from the Universitaet Paderborn. He is Professor of Distributed Computing at University College London (UCL), Vice Dean (Research) of Engineering Sciences, Programme Co-Chair of International Conference on Software Engineering 2007, Member of the Editorial Board of IEEE Transactions of Software Engineering, Member of the IEE and Chartered Engineer. Aside from a wide range of research publications in distributed systems and software engineering, he is author of “Engineering Distributed Objects” a major textbook published by Wiley. Wolfgang Emmerich is a Partner of Zuhlke Technology Group and Chairman of Zuhlke Engineering Ltd. Zuhlke is a medium-sized pan-European professional services company delivering software engineering services to financial services,
telecommunication, transport and life science clients. Zuhlke are IBM Premier partner, Microsoft Gold Partner, BEA Gold Star partner and run a Sun Education Partner. Prior to joining Zuhlke, worked as a Principal Consultant and Course Author at the Central European OMG office in Germany. Both Zuhlke and UCL are OMG members and have actively contributed to various OMG specifications.

Professor Jeff Magee

Jeff Magee is a graduate in Electrical Engineering and holds M.Sc. and Ph.D. degrees in Computing Science. He is currently Head of the Department of Computing at Imperial College. His research is primarily concerned with the software engineering of distributed systems, including design methods, analysis techniques, operating systems, languages and program support environments for these systems. His work on Software Architecture led to the commercial use by Phillips of an Architecture Description language based on Darwin in their current generation of consumer television products. He is the author of over 100 refereed publications and has co-authored a book on concurrent programming entitled “Concurrency - State models and Java programs”. He was co-editor of the IEE Proceedings on Software Engineering and is currently a TOSEM Associate Editor. He was program co-chair of the 24th International Conference on Software Engineering to be held in Buenos Aires, Argentina and chaired the ICSE Steering Committee from 2002-2004. He was a member-at-large of the ACM SIGSOFT committee from 2002-2005. He was awarded the BCS 1999 Brendan Murphy prize for the best paper in Distributed systems and the IEE Informatics Premium prize for 1998/99 for a paper jointly authored with Professor Kramer on Software Architecture. He is the co-recipient of the 2005 ACM SIGSOFT Outstanding Research Award for his work in Distributed Software Engineering. He has also worked with many industries, including BP, BT, NATS, Fujitsu, Barclays Capital, QinetiQ, Kodak and Philips, in both research collaborations and in a consultancy role.

Professor Jeffrey Kramer

Jeff Kramer is a graduate in Electrical Engineering (BSc Eng, Natal) and has a MSc and PhD in Computing from Imperial College. He is Professor of Distributed Computing and Head of the Distributed Software Engineering Section in the Department of Computing at Imperial College London. He was Head of the Department from 1999 to 2004. His research work is on behaviour analysis, the use of models in requirements elaboration and architectural approaches to self-organising software systems. He was a principal investigator in the various research projects that led to the development of the CONIC and DARWIN environments for distributed programming and the associated research into software architectures and their analysis. The work on the Darwin Software Architecture led to its commercial use by Philips in their new generation of consumer television products. Jeff Kramer is a Chartered Engineer, Fellow of the IEE, Fellow of the BCS and Fellow of the ACM. In
1999, he was program co-chair of the 21st ICSE, the flagship conference in the field, and from 2000 to 2002 he was Chair of the Steering Committee. He was associate editor of ACM TOSEM (Transactions on Software Engineering and Methodology) from 1995 to 2001 and of IEEE TSE (Transactions on Software Engineering) from 2003 to 2005. He has been appointed as the Editor in Chief of the IEEE TSE from 2006, the foremost journal in the field. He has been on over 50 international conference committees in the last 10 years and has given numerous invited keynote talks at international conferences. He is co-author of a recent book on Concurrency, co-author of a previous book on Distributed Systems and Computer Networks, and the author of over 150 journal and conference publications. He has also worked with many industries, including BP, BT, NATS, Fujitsu, Barclays Capital, QinetiQ, Kodak and Philips, in research collaboration and/or as a consultant.

Dr. Holger Schwichtenberg

Dr. Holger Schwichtenberg holds a master's degree and a Ph.D. in business informatics, both from the Universitaet Duisburg-Essen. He works as an independent software architect and technology consultant for medium and large sized companies in Germany, focusing on software development on the Microsoft Windows platform. Dr. Holger Schwichtenberg is also a lecturer in computer science at the private University of Applied Sciences "FOM" in Essen, Neuss and Siegen. He is a founder and director of the Professional Developer College, a training company specialized on the further education of graduate software developers. He has published 17 books for Addison Wesley Germany, Microsoft Press Germany and WEKA Media Germany and in addition more than 250 journal articles, notably for the German IT journals "iX", "Computer Zeitung", "DOTNETpro" and "Windows IT Pro". Dr. Holger Schwichtenberg regularly speaks at professional and scientific conferences (e.g. TechEd, IT Forum, Advanced Developers Conference, OOP, Net.Object Days, Wirtschaftsinformatik, Online, BASTA, DOTNET Conference).
Appendix D: References


