**[MS-SAMR]:**

**Security Account Manager (SAM) Remote Protocol (Client-to-Server)**

Intellectual Property Rights Notice for Open Specifications Documentation

* **Technical Documentation.** Microsoft publishes Open Specifications documentation (“this documentation”) for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
* **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
* **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
* **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft [Open Specifications Promise](https://go.microsoft.com/fwlink/?LinkId=214445) or the [Microsoft Community Promise](https://go.microsoft.com/fwlink/?LinkId=214448). If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
* **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the [Patent Map](https://aka.ms/AA9ufj8).
* **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit [www.microsoft.com/trademarks](https://www.microsoft.com/trademarks).
* **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

**Reservation of Rights**. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

**Tools**. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

**Support.** For questions and support, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

**Revision Summary**

| Date | Revision History | Revision Class | Comments |
| --- | --- | --- | --- |
| 2/22/2007 | 0.01 | New | Version 0.01 release |
| 6/1/2007 | 1.0 | Major | Updated and revised the technical content. |
| 7/3/2007 | 2.0 | Major | Added example. |
| 7/20/2007 | 3.0 | Major | Rewrite of keying algorithms; clarification of user account enabling. |
| 8/10/2007 | 3.0.1 | Editorial | Changed language and formatting in the technical content. |
| 9/28/2007 | 3.0.2 | Editorial | Changed language and formatting in the technical content. |
| 10/23/2007 | 3.1 | Minor | Clarified the meaning of the technical content. |
| 11/30/2007 | 3.2 | Minor | Clarified the meaning of the technical content. |
| 1/25/2008 | 4.0 | Major | Updated and revised the technical content. |
| 3/14/2008 | 4.1 | Minor | Clarified the meaning of the technical content. |
| 5/16/2008 | 4.1.1 | Editorial | Changed language and formatting in the technical content. |
| 6/20/2008 | 5.0 | Major | Updated and revised the technical content. |
| 7/25/2008 | 6.0 | Major | Updated and revised the technical content. |
| 8/29/2008 | 7.0 | Major | Updated and revised the technical content. |
| 10/24/2008 | 8.0 | Major | Updated and revised the technical content. |
| 12/5/2008 | 9.0 | Major | Updated and revised the technical content. |
| 1/16/2009 | 10.0 | Major | Updated and revised the technical content. |
| 2/27/2009 | 11.0 | Major | Updated and revised the technical content. |
| 4/10/2009 | 12.0 | Major | Updated and revised the technical content. |
| 5/22/2009 | 13.0 | Major | Updated and revised the technical content. |
| 7/2/2009 | 14.0 | Major | Updated and revised the technical content. |
| 8/14/2009 | 15.0 | Major | Updated and revised the technical content. |
| 9/25/2009 | 16.0 | Major | Updated and revised the technical content. |
| 11/6/2009 | 17.0 | Major | Updated and revised the technical content. |
| 12/18/2009 | 18.0 | Major | Updated and revised the technical content. |
| 1/29/2010 | 19.0 | Major | Updated and revised the technical content. |
| 3/12/2010 | 20.0 | Major | Updated and revised the technical content. |
| 4/23/2010 | 21.0 | Major | Updated and revised the technical content. |
| 6/4/2010 | 22.0 | Major | Updated and revised the technical content. |
| 7/16/2010 | 23.0 | Major | Updated and revised the technical content. |
| 8/27/2010 | 23.1 | Minor | Clarified the meaning of the technical content. |
| 10/8/2010 | 24.0 | Major | Updated and revised the technical content. |
| 11/19/2010 | 25.0 | Major | Updated and revised the technical content. |
| 1/7/2011 | 26.0 | Major | Updated and revised the technical content. |
| 2/11/2011 | 27.0 | Major | Updated and revised the technical content. |
| 3/25/2011 | 28.0 | Major | Updated and revised the technical content. |
| 5/6/2011 | 29.0 | Major | Updated and revised the technical content. |
| 6/17/2011 | 29.1 | Minor | Clarified the meaning of the technical content. |
| 9/23/2011 | 30.0 | Major | Updated and revised the technical content. |
| 12/16/2011 | 31.0 | Major | Updated and revised the technical content. |
| 3/30/2012 | 31.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 7/12/2012 | 31.1 | Minor | Clarified the meaning of the technical content. |
| 10/25/2012 | 32.0 | Major | Updated and revised the technical content. |
| 1/31/2013 | 33.0 | Major | Updated and revised the technical content. |
| 8/8/2013 | 34.0 | Major | Updated and revised the technical content. |
| 11/14/2013 | 34.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 2/13/2014 | 34.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 5/15/2014 | 34.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 6/30/2015 | 35.0 | Major | Significantly changed the technical content. |
| 10/16/2015 | 36.0 | Major | Significantly changed the technical content. |
| 7/14/2016 | 37.0 | Major | Significantly changed the technical content. |
| 6/1/2017 | 38.0 | Major | Significantly changed the technical content. |
| 9/15/2017 | 39.0 | Major | Significantly changed the technical content. |
| 12/1/2017 | 39.0 | None | No changes to the meaning, language, or formatting of the technical content. |
| 9/12/2018 | 40.0 | Major | Significantly changed the technical content. |
| 4/7/2021 | 41.0 | Major | Significantly changed the technical content. |
| 6/2/2021 | 42.0 | Major | Significantly changed the technical content. |

Table of Contents

[1 Introduction 11](#_Toc72927162)

[1.1 Glossary 11](#_Toc72927163)

[1.2 References 15](#_Toc72927164)

[1.2.1 Normative References 15](#_Toc72927165)

[1.2.2 Informative References 17](#_Toc72927166)

[1.3 Overview 18](#_Toc72927167)

[1.3.1 Object-Based Perspective 18](#_Toc72927168)

[1.3.2 Method-Based Perspective 21](#_Toc72927169)

[1.4 Relationship to Other Protocols 25](#_Toc72927170)

[1.5 Prerequisites/Preconditions 26](#_Toc72927171)

[1.6 Applicability Statement 26](#_Toc72927172)

[1.7 Versioning and Capability Negotiation 27](#_Toc72927173)

[1.7.1 Method Introduction 27](#_Toc72927174)

[1.7.2 Method Versioning 27](#_Toc72927175)

[1.7.3 Introduction to Information Levels 27](#_Toc72927176)

[1.8 Vendor-Extensible Fields 27](#_Toc72927177)

[1.9 Standards Assignments 27](#_Toc72927178)

[2 Messages 28](#_Toc72927179)

[2.1 Transport 28](#_Toc72927180)

[2.2 Common Data Types 29](#_Toc72927181)

[2.2.1 Constant Value Definitions 29](#_Toc72927182)

[2.2.1.1 Common ACCESS\_MASK Values 29](#_Toc72927183)

[2.2.1.2 Generic ACCESS\_MASK Values 29](#_Toc72927184)

[2.2.1.3 Server ACCESS\_MASK Values 30](#_Toc72927185)

[2.2.1.4 Domain ACCESS\_MASK Values 30](#_Toc72927186)

[2.2.1.5 Group ACCESS\_MASK Values 31](#_Toc72927187)

[2.2.1.6 Alias ACCESS\_MASK Values 32](#_Toc72927188)

[2.2.1.7 User ACCESS\_MASK Values 33](#_Toc72927189)

[2.2.1.8 USER\_ALL Values 34](#_Toc72927190)

[2.2.1.9 ACCOUNT\_TYPE Values 36](#_Toc72927191)

[2.2.1.10 SE\_GROUP Attributes 36](#_Toc72927192)

[2.2.1.11 GROUP\_TYPE Codes 37](#_Toc72927193)

[2.2.1.12 USER\_ACCOUNT Codes 37](#_Toc72927194)

[2.2.1.13 UF\_FLAG Codes 39](#_Toc72927195)

[2.2.1.14 Predefined RIDs 40](#_Toc72927196)

[2.2.1.15 STATUS\_ Codes 41](#_Toc72927197)

[2.2.1.16 Transport Error Code 41](#_Toc72927198)

[2.2.1.17 AD ACCESS\_MASK 42](#_Toc72927199)

[2.2.1.18 AEAD-AES-256-CBC-HMAC-SHA512 Constants 42](#_Toc72927200)

[2.2.2 Basic Data Types 42](#_Toc72927201)

[2.2.2.1 RPC\_STRING, PRPC\_STRING 43](#_Toc72927202)

[2.2.2.2 OLD\_LARGE\_INTEGER 43](#_Toc72927203)

[2.2.2.3 SID\_NAME\_USE 43](#_Toc72927204)

[2.2.2.4 RPC\_SHORT\_BLOB 44](#_Toc72927205)

[2.2.3 Domain Query/Set Data Types 44](#_Toc72927206)

[2.2.3.1 Domain Fields 44](#_Toc72927207)

[2.2.3.2 DOMAIN\_SERVER\_ENABLE\_STATE 46](#_Toc72927208)

[2.2.3.3 DOMAIN\_STATE\_INFORMATION 46](#_Toc72927209)

[2.2.3.4 DOMAIN\_SERVER\_ROLE 47](#_Toc72927210)

[2.2.3.5 DOMAIN\_PASSWORD\_INFORMATION 47](#_Toc72927211)

[2.2.3.6 DOMAIN\_LOGOFF\_INFORMATION 47](#_Toc72927212)

[2.2.3.7 DOMAIN\_SERVER\_ROLE\_INFORMATION 47](#_Toc72927213)

[2.2.3.8 DOMAIN\_MODIFIED\_INFORMATION 47](#_Toc72927214)

[2.2.3.9 DOMAIN\_MODIFIED\_INFORMATION2 48](#_Toc72927215)

[2.2.3.10 SAMPR\_DOMAIN\_GENERAL\_INFORMATION 48](#_Toc72927216)

[2.2.3.11 SAMPR\_DOMAIN\_GENERAL\_INFORMATION2 49](#_Toc72927217)

[2.2.3.12 SAMPR\_DOMAIN\_OEM\_INFORMATION 49](#_Toc72927218)

[2.2.3.13 SAMPR\_DOMAIN\_NAME\_INFORMATION 49](#_Toc72927219)

[2.2.3.14 SAMPR\_DOMAIN\_REPLICATION\_INFORMATION 49](#_Toc72927220)

[2.2.3.15 SAMPR\_DOMAIN\_LOCKOUT\_INFORMATION 50](#_Toc72927221)

[2.2.3.16 DOMAIN\_INFORMATION\_CLASS 50](#_Toc72927222)

[2.2.3.17 SAMPR\_DOMAIN\_INFO\_BUFFER 51](#_Toc72927223)

[2.2.4 Group Query/Set Data Types 51](#_Toc72927224)

[2.2.4.1 Common Group Fields 52](#_Toc72927225)

[2.2.4.2 GROUP\_ATTRIBUTE\_INFORMATION 52](#_Toc72927226)

[2.2.4.3 SAMPR\_GROUP\_GENERAL\_INFORMATION 52](#_Toc72927227)

[2.2.4.4 SAMPR\_GROUP\_NAME\_INFORMATION 52](#_Toc72927228)

[2.2.4.5 SAMPR\_GROUP\_ADM\_COMMENT\_INFORMATION 53](#_Toc72927229)

[2.2.4.6 GROUP\_INFORMATION\_CLASS 53](#_Toc72927230)

[2.2.4.7 SAMPR\_GROUP\_INFO\_BUFFER 53](#_Toc72927231)

[2.2.5 Alias Query/Set Data Types 54](#_Toc72927232)

[2.2.5.1 Common Alias Fields 54](#_Toc72927233)

[2.2.5.2 SAMPR\_ALIAS\_GENERAL\_INFORMATION 54](#_Toc72927234)

[2.2.5.3 SAMPR\_ALIAS\_NAME\_INFORMATION 55](#_Toc72927235)

[2.2.5.4 SAMPR\_ALIAS\_ADM\_COMMENT\_INFORMATION 55](#_Toc72927236)

[2.2.5.5 ALIAS\_INFORMATION\_CLASS 55](#_Toc72927237)

[2.2.5.6 SAMPR\_ALIAS\_INFO\_BUFFER 55](#_Toc72927238)

[2.2.6 User Query/Set Data Types 56](#_Toc72927239)

[2.2.6.1 Common User Fields 56](#_Toc72927240)

[2.2.6.2 USER\_PRIMARY\_GROUP\_INFORMATION 58](#_Toc72927241)

[2.2.6.3 USER\_CONTROL\_INFORMATION 58](#_Toc72927242)

[2.2.6.4 USER\_EXPIRES\_INFORMATION 58](#_Toc72927243)

[2.2.6.5 SAMPR\_LOGON\_HOURS 58](#_Toc72927244)

[2.2.6.6 SAMPR\_USER\_ALL\_INFORMATION 59](#_Toc72927245)

[2.2.6.7 SAMPR\_USER\_GENERAL\_INFORMATION 60](#_Toc72927246)

[2.2.6.8 SAMPR\_USER\_PREFERENCES\_INFORMATION 60](#_Toc72927247)

[2.2.6.9 SAMPR\_USER\_PARAMETERS\_INFORMATION 60](#_Toc72927248)

[2.2.6.10 SAMPR\_USER\_LOGON\_INFORMATION 61](#_Toc72927249)

[2.2.6.11 SAMPR\_USER\_ACCOUNT\_INFORMATION 61](#_Toc72927250)

[2.2.6.12 SAMPR\_USER\_A\_NAME\_INFORMATION 62](#_Toc72927251)

[2.2.6.13 SAMPR\_USER\_F\_NAME\_INFORMATION 62](#_Toc72927252)

[2.2.6.14 SAMPR\_USER\_NAME\_INFORMATION 62](#_Toc72927253)

[2.2.6.15 SAMPR\_USER\_HOME\_INFORMATION 62](#_Toc72927254)

[2.2.6.16 SAMPR\_USER\_SCRIPT\_INFORMATION 62](#_Toc72927255)

[2.2.6.17 SAMPR\_USER\_PROFILE\_INFORMATION 63](#_Toc72927256)

[2.2.6.18 SAMPR\_USER\_ADMIN\_COMMENT\_INFORMATION 63](#_Toc72927257)

[2.2.6.19 SAMPR\_USER\_WORKSTATIONS\_INFORMATION 63](#_Toc72927258)

[2.2.6.20 SAMPR\_USER\_LOGON\_HOURS\_INFORMATION 63](#_Toc72927259)

[2.2.6.21 SAMPR\_ENCRYPTED\_USER\_PASSWORD 63](#_Toc72927260)

[2.2.6.22 SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW 64](#_Toc72927261)

[2.2.6.23 SAMPR\_USER\_INTERNAL1\_INFORMATION 65](#_Toc72927262)

[2.2.6.24 SAMPR\_USER\_INTERNAL4\_INFORMATION 65](#_Toc72927263)

[2.2.6.25 SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW 66](#_Toc72927264)

[2.2.6.26 SAMPR\_USER\_INTERNAL5\_INFORMATION 66](#_Toc72927265)

[2.2.6.27 SAMPR\_USER\_INTERNAL5\_INFORMATION\_NEW 66](#_Toc72927266)

[2.2.6.28 USER\_INFORMATION\_CLASS 67](#_Toc72927267)

[2.2.6.29 SAMPR\_USER\_INFO\_BUFFER 68](#_Toc72927268)

[2.2.6.30 SAMPR\_USER\_INTERNAL7\_INFORMATION 69](#_Toc72927269)

[2.2.6.31 SAMPR\_USER\_INTERNAL8\_INFORMATION 70](#_Toc72927270)

[2.2.6.32 SAMPR\_ENCRYPTED\_PASSWORD\_AES 70](#_Toc72927271)

[2.2.7 Miscellaneous Protocol-Specific Types 71](#_Toc72927272)

[2.2.7.1 PSAMPR\_SERVER\_NAME 71](#_Toc72927273)

[2.2.7.2 SAMPR\_HANDLE 71](#_Toc72927274)

[2.2.7.3 ENCRYPTED\_LM\_OWF\_PASSWORD, ENCRYPTED\_NT\_OWF\_PASSWORD 71](#_Toc72927275)

[2.2.7.4 SAMPR\_ULONG\_ARRAY 72](#_Toc72927276)

[2.2.7.5 SAMPR\_PSID\_ARRAY 72](#_Toc72927277)

[2.2.7.6 SAMPR\_SID\_INFORMATION 72](#_Toc72927278)

[2.2.7.7 SAMPR\_PSID\_ARRAY\_OUT 72](#_Toc72927279)

[2.2.7.8 SAMPR\_RETURNED\_USTRING\_ARRAY 73](#_Toc72927280)

[2.2.7.9 SAMPR\_RID\_ENUMERATION 73](#_Toc72927281)

[2.2.7.10 SAMPR\_ENUMERATION\_BUFFER 73](#_Toc72927282)

[2.2.7.11 SAMPR\_SR\_SECURITY\_DESCRIPTOR 74](#_Toc72927283)

[2.2.7.12 GROUP\_MEMBERSHIP 74](#_Toc72927284)

[2.2.7.13 SAMPR\_GET\_GROUPS\_BUFFER 74](#_Toc72927285)

[2.2.7.14 SAMPR\_GET\_MEMBERS\_BUFFER 74](#_Toc72927286)

[2.2.7.15 SAMPR\_REVISION\_INFO\_V1 75](#_Toc72927287)

[2.2.7.16 SAMPR\_REVISION\_INFO 75](#_Toc72927288)

[2.2.7.17 USER\_DOMAIN\_PASSWORD\_INFORMATION 76](#_Toc72927289)

[2.2.8 Selective Enumerate Associated Structures 76](#_Toc72927290)

[2.2.8.1 Common Selective Enumerate Fields 76](#_Toc72927291)

[2.2.8.2 SAMPR\_DOMAIN\_DISPLAY\_USER 77](#_Toc72927292)

[2.2.8.3 SAMPR\_DOMAIN\_DISPLAY\_MACHINE 77](#_Toc72927293)

[2.2.8.4 SAMPR\_DOMAIN\_DISPLAY\_GROUP 77](#_Toc72927294)

[2.2.8.5 SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER 78](#_Toc72927295)

[2.2.8.6 SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP 78](#_Toc72927296)

[2.2.8.7 SAMPR\_DOMAIN\_DISPLAY\_USER\_BUFFER 78](#_Toc72927297)

[2.2.8.8 SAMPR\_DOMAIN\_DISPLAY\_MACHINE\_BUFFER 78](#_Toc72927298)

[2.2.8.9 SAMPR\_DOMAIN\_DISPLAY\_GROUP\_BUFFER 79](#_Toc72927299)

[2.2.8.10 SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER\_BUFFER 79](#_Toc72927300)

[2.2.8.11 SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP\_BUFFER 79](#_Toc72927301)

[2.2.8.12 DOMAIN\_DISPLAY\_INFORMATION 80](#_Toc72927302)

[2.2.8.13 SAMPR\_DISPLAY\_INFO\_BUFFER 80](#_Toc72927303)

[2.2.9 SamrValidatePassword Data Types 81](#_Toc72927304)

[2.2.9.1 SAM\_VALIDATE\_PASSWORD\_HASH 81](#_Toc72927305)

[2.2.9.2 SAM\_VALIDATE\_PERSISTED\_FIELDS 81](#_Toc72927306)

[2.2.9.3 SAM\_VALIDATE\_VALIDATION\_STATUS 82](#_Toc72927307)

[2.2.9.4 SAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG 83](#_Toc72927308)

[2.2.9.5 SAM\_VALIDATE\_AUTHENTICATION\_INPUT\_ARG 83](#_Toc72927309)

[2.2.9.6 SAM\_VALIDATE\_PASSWORD\_CHANGE\_INPUT\_ARG 83](#_Toc72927310)

[2.2.9.7 SAM\_VALIDATE\_PASSWORD\_RESET\_INPUT\_ARG 84](#_Toc72927311)

[2.2.9.8 PASSWORD\_POLICY\_VALIDATION\_TYPE 85](#_Toc72927312)

[2.2.9.9 SAM\_VALIDATE\_INPUT\_ARG 85](#_Toc72927313)

[2.2.9.10 SAM\_VALIDATE\_OUTPUT\_ARG 85](#_Toc72927314)

[2.2.10 Supplemental Credentials Structures 86](#_Toc72927315)

[2.2.10.1 USER\_PROPERTIES 86](#_Toc72927316)

[2.2.10.2 USER\_PROPERTY 87](#_Toc72927317)

[2.2.10.3 Primary:WDigest - WDIGEST\_CREDENTIALS 87](#_Toc72927318)

[2.2.10.4 Primary:Kerberos - KERB\_STORED\_CREDENTIAL 91](#_Toc72927319)

[2.2.10.5 KERB\_KEY\_DATA 92](#_Toc72927320)

[2.2.10.6 Primary:Kerberos-Newer-Keys - KERB\_STORED\_CREDENTIAL\_NEW 93](#_Toc72927321)

[2.2.10.7 KERB\_KEY\_DATA\_NEW 95](#_Toc72927322)

[2.2.10.8 Kerberos Encryption Algorithm Identifiers 96](#_Toc72927323)

[2.2.10.9 NTLM-Strong-NTOWF 96](#_Toc72927324)

[2.2.11 Common Algorithms 96](#_Toc72927325)

[2.2.11.1 DES-ECB-LM 96](#_Toc72927326)

[2.2.11.1.1 Encrypting an NT or LM Hash Value with a Specified Key 97](#_Toc72927327)

[2.2.11.1.2 Encrypting a 64-Bit Block with a 7-Byte Key 97](#_Toc72927328)

[2.2.11.1.3 Deriving Key1 and Key2 from a Little-Endian, Unsigned Integer Key 98](#_Toc72927329)

[2.2.11.1.4 Deriving Key1 and Key2 from a 16-Byte Key 98](#_Toc72927330)

[2.3 Directory Service Schema Elements 98](#_Toc72927331)

[3 Protocol Details 99](#_Toc72927332)

[3.1 Server Details 99](#_Toc72927333)

[3.1.1 Abstract Data Model 99](#_Toc72927334)

[3.1.1.1 String Handling 100](#_Toc72927335)

[3.1.1.2 String Matching 100](#_Toc72927336)

[3.1.1.3 Attribute Listing 101](#_Toc72927337)

[3.1.1.4 Object Class List 103](#_Toc72927338)

[3.1.1.5 Password Settings Attributes for Originating Update Constraints 103](#_Toc72927339)

[3.1.1.6 Attribute Constraints for Originating Updates 104](#_Toc72927340)

[3.1.1.7 Additional Update Constraints 108](#_Toc72927341)

[3.1.1.7.1 General Password Policy 108](#_Toc72927342)

[3.1.1.7.2 Cleartext Password Policy 109](#_Toc72927343)

[3.1.1.8 Attribute Triggers for Originating Updates 112](#_Toc72927344)

[3.1.1.8.1 objectClass 112](#_Toc72927345)

[3.1.1.8.2 primaryGroupID 113](#_Toc72927346)

[3.1.1.8.3 lockoutTime 114](#_Toc72927347)

[3.1.1.8.4 sAMAccountName 114](#_Toc72927348)

[3.1.1.8.5 clearTextPassword 114](#_Toc72927349)

[3.1.1.8.6 dBCSPwd 115](#_Toc72927350)

[3.1.1.8.7 unicodePwd 115](#_Toc72927351)

[3.1.1.8.8 pwdLastSet 115](#_Toc72927352)

[3.1.1.8.9 member 115](#_Toc72927353)

[3.1.1.8.10 userAccountControl 116](#_Toc72927354)

[3.1.1.8.11 supplementalCredentials 118](#_Toc72927355)

[3.1.1.8.11.1 Processing 119](#_Toc72927356)

[3.1.1.8.11.1.1 USER\_PROPERTIES Processing 119](#_Toc72927357)

[3.1.1.8.11.1.2 USER\_PROPERTY Processing 119](#_Toc72927358)

[3.1.1.8.11.2 Packages Property 119](#_Toc72927359)

[3.1.1.8.11.3 Primary:WDigest Property 120](#_Toc72927360)

[3.1.1.8.11.3.1 WDIGEST\_CREDENTIALS Construction 120](#_Toc72927361)

[3.1.1.8.11.4 Primary:Kerberos Property 121](#_Toc72927362)

[3.1.1.8.11.5 Primary:CLEARTEXT Property 122](#_Toc72927363)

[3.1.1.8.11.6 Primary:Kerberos-Newer-Keys Property 122](#_Toc72927364)

[3.1.1.8.11.7 Primary:NTLM-Strong-NTOWF Property 123](#_Toc72927365)

[3.1.1.9 Additional Update Triggers 123](#_Toc72927366)

[3.1.1.9.1 Password History Update 123](#_Toc72927367)

[3.1.1.9.2 objectSid Value Generation 123](#_Toc72927368)

[3.1.1.9.2.1 DC Configuration 124](#_Toc72927369)

[3.1.1.9.2.2 Non-DC Configuration 124](#_Toc72927370)

[3.1.1.10 SamContextHandle Data Model 125](#_Toc72927371)

[3.1.1.11 Server Access Control List 125](#_Toc72927372)

[3.1.2 Security Model 125](#_Toc72927373)

[3.1.2.1 Server-wide Access Check 126](#_Toc72927374)

[3.1.2.2 Standard Handle-Based Access Checks 126](#_Toc72927375)

[3.1.2.3 AD Access Checks in DC Configuration 131](#_Toc72927376)

[3.1.2.4 Acquiring an SMB Session Key 131](#_Toc72927377)

[3.1.3 Timers 131](#_Toc72927378)

[3.1.4 Initialization 131](#_Toc72927379)

[3.1.4.1 Default Access 132](#_Toc72927380)

[3.1.4.2 Default Accounts 132](#_Toc72927381)

[3.1.5 Message Processing Events and Sequencing Rules 134](#_Toc72927382)

[3.1.5.1 Open Pattern 140](#_Toc72927383)

[3.1.5.1.1 SamrConnect5 (Opnum 64) 140](#_Toc72927384)

[3.1.5.1.2 SamrConnect4 (Opnum 62) 142](#_Toc72927385)

[3.1.5.1.3 SamrConnect2 (Opnum 57) 142](#_Toc72927386)

[3.1.5.1.4 SamrConnect (Opnum 0) 143](#_Toc72927387)

[3.1.5.1.5 SamrOpenDomain (Opnum 7) 144](#_Toc72927388)

[3.1.5.1.6 Common Processing for Group, Alias, and User 146](#_Toc72927389)

[3.1.5.1.7 SamrOpenGroup (Opnum 19) 147](#_Toc72927390)

[3.1.5.1.8 SamrOpenAlias (Opnum 27) 148](#_Toc72927391)

[3.1.5.1.9 SamrOpenUser (Opnum 34) 149](#_Toc72927392)

[3.1.5.2 Enumerate Pattern 151](#_Toc72927393)

[3.1.5.2.1 SamrEnumerateDomainsInSamServer (Opnum 6) 151](#_Toc72927394)

[3.1.5.2.2 Common Processing for Enumeration of Users, Groups, and Aliases 152](#_Toc72927395)

[3.1.5.2.3 SamrEnumerateGroupsInDomain (Opnum 11) 153](#_Toc72927396)

[3.1.5.2.4 SamrEnumerateAliasesInDomain (Opnum 15) 154](#_Toc72927397)

[3.1.5.2.5 SamrEnumerateUsersInDomain (Opnum 13) 155](#_Toc72927398)

[3.1.5.3 Selective Enumerate Pattern 156](#_Toc72927399)

[3.1.5.3.1 SamrQueryDisplayInformation3 (Opnum 51) 156](#_Toc72927400)

[3.1.5.3.2 SamrQueryDisplayInformation2 (Opnum 48) 158](#_Toc72927401)

[3.1.5.3.3 SamrQueryDisplayInformation (Opnum 40) 159](#_Toc72927402)

[3.1.5.3.4 SamrGetDisplayEnumerationIndex2 (Opnum 49) 159](#_Toc72927403)

[3.1.5.3.5 SamrGetDisplayEnumerationIndex (Opnum 41) 161](#_Toc72927404)

[3.1.5.4 Create Pattern 161](#_Toc72927405)

[3.1.5.4.1 Common Processing for Group and Alias Creation 161](#_Toc72927406)

[3.1.5.4.2 SamrCreateGroupInDomain (Opnum 10) 162](#_Toc72927407)

[3.1.5.4.3 SamrCreateAliasInDomain (Opnum 14) 163](#_Toc72927408)

[3.1.5.4.4 SamrCreateUser2InDomain (Opnum 50) 163](#_Toc72927409)

[3.1.5.4.5 SamrCreateUserInDomain (Opnum 12) 166](#_Toc72927410)

[3.1.5.5 Query Pattern 167](#_Toc72927411)

[3.1.5.5.1 SamrQueryInformationDomain2 (Opnum 46) 167](#_Toc72927412)

[3.1.5.5.1.1 DomainGeneralInformation 168](#_Toc72927413)

[3.1.5.5.1.2 DomainServerRoleInformation 168](#_Toc72927414)

[3.1.5.5.1.3 DomainStateInformation 169](#_Toc72927415)

[3.1.5.5.1.4 DomainGeneralInformation2 169](#_Toc72927416)

[3.1.5.5.2 SamrQueryInformationDomain (Opnum 8) 169](#_Toc72927417)

[3.1.5.5.3 SamrQueryInformationGroup (Opnum 20) 170](#_Toc72927418)

[3.1.5.5.3.1 GroupReplicationInformation 170](#_Toc72927419)

[3.1.5.5.4 SamrQueryInformationAlias (Opnum 28) 171](#_Toc72927420)

[3.1.5.5.5 SamrQueryInformationUser2 (Opnum 47) 172](#_Toc72927421)

[3.1.5.5.5.1 Common Processing 172](#_Toc72927422)

[3.1.5.5.5.2 UserAllInformation 174](#_Toc72927423)

[3.1.5.5.6 SamrQueryInformationUser (Opnum 36) 174](#_Toc72927424)

[3.1.5.6 Set Pattern 175](#_Toc72927425)

[3.1.5.6.1 SamrSetInformationDomain (Opnum 9) 175](#_Toc72927426)

[3.1.5.6.1.1 DomainServerRoleInformation 176](#_Toc72927427)

[3.1.5.6.1.2 DomainStateInformation 176](#_Toc72927428)

[3.1.5.6.1.3 DomainPasswordInformation 176](#_Toc72927429)

[3.1.5.6.2 SamrSetInformationGroup (Opnum 21) 177](#_Toc72927430)

[3.1.5.6.3 SamrSetInformationAlias (Opnum 29) 177](#_Toc72927431)

[3.1.5.6.4 SamrSetInformationUser2 (Opnum 58) 178](#_Toc72927432)

[3.1.5.6.4.1 Common Processing 179](#_Toc72927433)

[3.1.5.6.4.2 UserAllInformation (Common) 182](#_Toc72927434)

[3.1.5.6.4.3 UserAllInformation 183](#_Toc72927435)

[3.1.5.6.4.4 UserInternal4Information 184](#_Toc72927436)

[3.1.5.6.4.5 UserInternal4InformationNew 184](#_Toc72927437)

[3.1.5.6.4.6 UserInternal8Information 184](#_Toc72927438)

[3.1.5.6.5 SamrSetInformationUser (Opnum 37) 185](#_Toc72927439)

[3.1.5.7 Delete Pattern 185](#_Toc72927440)

[3.1.5.7.1 SamrDeleteGroup (Opnum 23) 185](#_Toc72927441)

[3.1.5.7.2 SamrDeleteAlias (Opnum 30) 186](#_Toc72927442)

[3.1.5.7.3 SamrDeleteUser (Opnum 35) 186](#_Toc72927443)

[3.1.5.8 Membership Pattern 187](#_Toc72927444)

[3.1.5.8.1 SamrAddMemberToGroup (Opnum 22) 187](#_Toc72927445)

[3.1.5.8.2 SamrRemoveMemberFromGroup (Opnum 24) 188](#_Toc72927446)

[3.1.5.8.3 SamrGetMembersInGroup (Opnum 25) 189](#_Toc72927447)

[3.1.5.8.4 SamrAddMemberToAlias (Opnum 31) 189](#_Toc72927448)

[3.1.5.8.5 SamrRemoveMemberFromAlias (Opnum 32) 190](#_Toc72927449)

[3.1.5.8.6 SamrGetMembersInAlias (Opnum 33) 191](#_Toc72927450)

[3.1.5.8.7 SamrRemoveMemberFromForeignDomain (Opnum 45) 191](#_Toc72927451)

[3.1.5.8.8 SamrAddMultipleMembersToAlias (Opnum 52) 192](#_Toc72927452)

[3.1.5.8.9 SamrRemoveMultipleMembersFromAlias (Opnum 53) 192](#_Toc72927453)

[3.1.5.9 Membership-Of Pattern 193](#_Toc72927454)

[3.1.5.9.1 SamrGetGroupsForUser (Opnum 39) 193](#_Toc72927455)

[3.1.5.9.2 SamrGetAliasMembership (Opnum 16) 194](#_Toc72927456)

[3.1.5.10 Change Password Pattern 194](#_Toc72927457)

[3.1.5.10.1 SamrChangePasswordUser (Opnum 38) 195](#_Toc72927458)

[3.1.5.10.2 SamrOemChangePasswordUser2 (Opnum 54) 198](#_Toc72927459)

[3.1.5.10.3 SamrUnicodeChangePasswordUser2 (Opnum 55) 199](#_Toc72927460)

[3.1.5.10.4 SamrUnicodeChangePasswordUser4 (Opnum 73) 201](#_Toc72927461)

[3.1.5.11 Lookup Pattern 202](#_Toc72927462)

[3.1.5.11.1 SamrLookupDomainInSamServer (Opnum 5) 202](#_Toc72927463)

[3.1.5.11.2 SamrLookupNamesInDomain (Opnum 17) 203](#_Toc72927464)

[3.1.5.11.3 SamrLookupIdsInDomain (Opnum 18) 204](#_Toc72927465)

[3.1.5.12 Security Pattern 206](#_Toc72927466)

[3.1.5.12.1 SamrSetSecurityObject (Opnum 2) 206](#_Toc72927467)

[3.1.5.12.1.1 SamrSetSecurityObject (DC Configuration) 207](#_Toc72927468)

[3.1.5.12.1.2 SamrSetSecurityObject (Non-DC Configuration) 208](#_Toc72927469)

[3.1.5.12.2 SamrQuerySecurityObject (Opnum 3) 210](#_Toc72927470)

[3.1.5.12.2.1 SamrQuerySecurityObject (DC Configuration) 211](#_Toc72927471)

[3.1.5.12.2.2 SamrQuerySecurityObject (Non-DC Configuration) 213](#_Toc72927472)

[3.1.5.13 Miscellaneous 214](#_Toc72927473)

[3.1.5.13.1 SamrCloseHandle (Opnum 1) 214](#_Toc72927474)

[3.1.5.13.2 SamrSetMemberAttributesOfGroup (Opnum 26) 214](#_Toc72927475)

[3.1.5.13.3 SamrGetUserDomainPasswordInformation (Opnum 44) 215](#_Toc72927476)

[3.1.5.13.4 SamrGetDomainPasswordInformation (Opnum 56) 215](#_Toc72927477)

[3.1.5.13.5 SamrRidToSid (Opnum 65) 216](#_Toc72927478)

[3.1.5.13.6 SamrSetDSRMPassword (Opnum 66) 217](#_Toc72927479)

[3.1.5.13.7 SamrValidatePassword (Opnum 67) 217](#_Toc72927480)

[3.1.5.13.7.1 SamValidateAuthentication 218](#_Toc72927481)

[3.1.5.13.7.2 SamValidatePasswordChange 220](#_Toc72927482)

[3.1.5.13.7.3 SamValidatePasswordReset 221](#_Toc72927483)

[3.1.5.14 Supplemental Message Processing 223](#_Toc72927484)

[3.1.5.14.1 distinguishedName Generation 223](#_Toc72927485)

[3.1.5.14.2 userAccountControl Mapping Table 223](#_Toc72927486)

[3.1.5.14.3 PasswordCanChange Generation 224](#_Toc72927487)

[3.1.5.14.4 PasswordMustChange Generation 224](#_Toc72927488)

[3.1.5.14.5 Account Lockout Enforcement and Reset 225](#_Toc72927489)

[3.1.5.14.6 Account Lockout State Maintenance 225](#_Toc72927490)

[3.1.5.14.7 Attributes Field Handling 225](#_Toc72927491)

[3.1.5.14.8 Domain Field to Attribute Name Mapping 226](#_Toc72927492)

[3.1.5.14.9 Group Field to Attribute Name Mapping 226](#_Toc72927493)

[3.1.5.14.10 Alias Field to Attribute Name Mapping 227](#_Toc72927494)

[3.1.5.14.11 User Field to Attribute Name Mapping 227](#_Toc72927495)

[3.1.6 Timer Events 228](#_Toc72927496)

[3.1.7 Other Local Events 229](#_Toc72927497)

[3.1.7.1 Domain Join Processing 229](#_Toc72927498)

[3.1.7.2 Domain Unjoin Processing 229](#_Toc72927499)

[3.2 Client Details 229](#_Toc72927500)

[3.2.1 Abstract Data Model 229](#_Toc72927501)

[3.2.2 Security Model 230](#_Toc72927502)

[3.2.2.1 RC4 Cipher Usage 230](#_Toc72927503)

[3.2.2.2 MD5 Usage 230](#_Toc72927504)

[3.2.2.3 Acquiring an SMB Session Key 230](#_Toc72927505)

[3.2.2.4 AES Cipher Usage 230](#_Toc72927506)

[3.2.2.5 Deriving an Encryption Key from a Plaintext Password 231](#_Toc72927507)

[3.2.3 Timers 231](#_Toc72927508)

[3.2.4 Initialization 231](#_Toc72927509)

[3.2.5 Message Processing Events and Sequencing Rules 231](#_Toc72927510)

[3.2.6 Timer Events 231](#_Toc72927511)

[3.2.7 Other Local Events 231](#_Toc72927512)

[4 Protocol Examples 232](#_Toc72927513)

[4.1 Creating a User Account 232](#_Toc72927514)

[4.2 Enabling a User Account 234](#_Toc72927515)

[4.3 Encrypting an NT or LM Hash 236](#_Toc72927516)

[5 Security 239](#_Toc72927517)

[5.1 Security Considerations for Implementers 239](#_Toc72927518)

[5.2 Index of Security Parameters 239](#_Toc72927519)

[6 Appendix A: Full IDL 240](#_Toc72927520)

[7 Appendix B: Product Behavior 261](#_Toc72927521)

[8 Change Tracking 275](#_Toc72927522)

[9 Index 278](#_Toc72927523)

# Introduction

The Security Account Manager (SAM) Remote Protocol (Client-to-Server) provides management functionality for an account store or directory containing users and groups. Users should familiarize themselves with the following documents: Windows System Overview [[MS-SYS-ARCHIVE]](%5bMS-SYS-ARCHIVE%5d.pdf), Windows Protocols Overview [[MS-WPO]](%5bMS-WPO%5d.pdf#Section_c5f54a7765be40a0bb829e4181d8ab67), and Active Directory Technical Specification [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a).

This protocol exposes the "account database" referred to in [[MS-AUTHSOD]](%5bMS-AUTHSOD%5d.pdf#Section_953d700a57cb4cf7b0c3a64f34581cc9) section 1.1.1.5, both for local and remote [**domains**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca). This document specifies the behavior for local and remote domains by having a common data model for both scenarios: the [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) data model, as specified in [MS-ADTS]. In addition, this document specifies the differences in behavior between these scenarios when necessary.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

## Glossary

This document uses the following terms:

**64-bit Network Data Representation (NDR64)**: A specific instance of a [**remote procedure call (RPC) transfer syntax**](#gt_01216ea7-ac8a-4cc8-9d19-b901bc424c09). For more information about [**RPC transfer syntax**](#gt_01216ea7-ac8a-4cc8-9d19-b901bc424c09), see [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824) section 14.

**access check**: A verification to determine whether a specific access type is allowed by checking a security context against a [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350).

**access control entry (ACE)**: An entry in an access control list (ACL) that contains a set of user rights and a [**security identifier (SID)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) that identifies a principal for whom the rights are allowed, denied, or audited.

**access mask**: A 32-bit value present in an [**access control entry (ACE)**](#gt_b581857f-39aa-4979-876b-daba67a40f15) that specifies the allowed or denied rights to manipulate an object.

**account**: A user (including machine account), group, or alias object. Also a synonym for security principal or principal.

**account domain object (account domain)**: A [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1) that represents an issuing authority in which [**user objects**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17) can be created. For more information about the concept of an issuing authority, see [[MS-AUTHSOD]](%5bMS-AUTHSOD%5d.pdf#Section_953d700a57cb4cf7b0c3a64f34581cc9) section 1.1.1.5.

**account domain security identifier**: The [**security identifier (SID)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) of the [**account domain object**](#gt_3ff707e8-0783-4b16-9370-2899f1078c87).

**account group**: A [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) whose members always include the [**security identifier (SID)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) of the group in the [**authorization context**](#gt_360f5f23-e078-4dee-971b-d5ff6018ea55).

**AccountOperatorsSid**: A [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) with the specific value of S-1-5-32-548.

**ACID**: A term that refers to the four properties that any database system must achieve in order to be considered transactional: Atomicity, Consistency, Isolation, and Durability [GRAY].

**Active Directory**: The Windows implementation of a general-purpose directory service, which uses LDAP as its primary access protocol. [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) stores information about a variety of objects in the network such as user accounts, computer accounts, groups, and all related credential information used by Kerberos [[MS-KILE]](%5bMS-KILE%5d.pdf#Section_2a32282edd484ad9a542609804b02cc9). [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) is either deployed as Active Directory Domain Services (AD DS) or Active Directory Lightweight Directory Services (AD LDS), which are both described in [[MS-ADOD]](%5bMS-ADOD%5d.pdf#Section_5ff67bf4c14548cb89cd4f5482d94664): Active Directory Protocols Overview.

**administrator**: A user who has complete and unrestricted access to the computer or [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca).

**AdministratorSid**: A [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) with the specific value of S-1-5-32-544.

**alias object**: See [**resource group**](#gt_04c2dc43-6f4a-432d-bca6-ceeef2d5f2ae).

**authorization context**: The set of identities for groups and the identity of the user made available to a server for the purpose of determining authorization to a resource.

**built-in domain**: The [**security identifier (SID)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) namespace defined by the fixed [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) S-1-5-32. Contains groups that define roles on a local machine such as Backup Operators.

**control access right**: An extended access right that can be granted or denied on an access control list (ACL).

**database object**: A representation of a named set of attribute value pairs that a protocol exposes.

**delta time**: A negative FILETIME. It represents a period of time, expressed in a negative number of 100-nanosecond time slices. For example, a period of 20 minutes is represented as -12000000000.

**discretionary access control list (DACL)**: An access control list (ACL) that is controlled by the owner of an object and that specifies the access particular users or groups can have to the object.

**domain**: A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a [**domain controller (DC)**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) and host a member list that identifies all members of the domain, as well as optionally hosting the [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) service. The domain controller provides authentication of members, creating a unit of trust for its members. Each domain has an identifier that is shared among its members. For more information, see [MS-AUTHSOD] section 1.1.1.5 and [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a).

**domain admins**: A group with a [**security identifier (SID)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) with the relative ID value of 512 in the account domain.

**domain controller (DC)**: The service, running on a server, that implements [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90), or the server hosting this service. The service hosts the data store for objects and interoperates with other [**DCs**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) to ensure that a local change to an object replicates correctly across all [**DCs**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd). When [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) is operating as Active Directory Domain Services (AD DS), the [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) contains full NC replicas of the configuration naming context (config NC), schema naming context (schema NC), and one of the domain NCs in its [**forest**](#gt_fd104241-4fb3-457c-b2c4-e0c18bb20b62). If the AD DS [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) is a global catalog server (GC server), it contains partial NC replicas of the remaining domain NCs in its [**forest**](#gt_fd104241-4fb3-457c-b2c4-e0c18bb20b62). For more information, see [MS-AUTHSOD] section 1.1.1.5.2 and [MS-ADTS]. When [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) is operating as Active Directory Lightweight Directory Services (AD LDS), several AD LDS [**DCs**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) can run on one server. When [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) is operating as AD DS, only one AD DS [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) can run on one server. However, several AD LDS [**DCs**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) can coexist with one AD DS [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) on one server. The AD LDS [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) contains full NC replicas of the config NC and the schema NC in its [**forest**](#gt_fd104241-4fb3-457c-b2c4-e0c18bb20b62). The domain controller is the server side of Authentication Protocol Domain Support [[MS-APDS]](%5bMS-APDS%5d.pdf#Section_dd444344fd7e430eb3137e95ab9c338e).

**domain functional level**: A specification of functionality available in a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca). Must be less than or equal to the DC functional level of every [**domain controller (DC)**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) that hosts a replica of the [**domain's**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) naming context (NC). For information on defined levels, corresponding features, information on how the [**domain functional level**](#gt_f9b5a663-12d5-46d4-b23a-147704f026fc) is determined, and supported [**domain controllers**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd), see [MS-ADTS] sections 6.1.4.2 and 6.1.4.3. When [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) is operating as Active Directory Lightweight Directory Services (AD LDS), [**domain functional level**](#gt_f9b5a663-12d5-46d4-b23a-147704f026fc) does not exist.

**domain name**: A domain name or a NetBIOS name that identifies a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca).

**domain object**: A unit of data storage in a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) that is maintained and made available to [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) members by a [**domain controller (DC)**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd).

**domain prefix**: A [**security identifier (SID)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) of a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) without the [**relative identifier (RID)**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) portion. The domain prefix refers to the issuing authority [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d). For example, the domain prefix of S-1-5-21-397955417-626881126-188441444-1010 is S-1-5-21-397955417-626881126-188441444.

**dsname**: A tuple that contains between one and three identifiers for an object. The term [**dsname**](#gt_4d5e1f08-aa00-4dde-9411-7dd6e09ed85a) does not stand for anything. The possible identifiers are the object's [**GUID**](#gt_f49694cc-c350-462d-ab8e-816f0103c6c1) (attribute objectGuid), [**security identifier (SID)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) (attribute objectSid), and distinguished name (DN) (attribute distinguishedName). A [**dsname**](#gt_4d5e1f08-aa00-4dde-9411-7dd6e09ed85a) can appear in a protocol message and as an attribute value (for example, a value of an attribute with syntax Object(DS-DN)). Given a [**DSName**](#gt_4d5e1f08-aa00-4dde-9411-7dd6e09ed85a), an object can be identified within a set of NC replicas according to the matching rules defined in [MS-DRSR] section 5.49.

**forest**: In the [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) directory service, a [**forest**](#gt_fd104241-4fb3-457c-b2c4-e0c18bb20b62) is a set of naming contexts (NCs) consisting of one schema NC, one config NC, and one or more domain NCs. Because a set of NCs can be arranged into a tree structure, a [**forest**](#gt_fd104241-4fb3-457c-b2c4-e0c18bb20b62) is also a set of one or several trees of NCs.

**fully qualified domain name (FQDN)**: In [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90), a fully qualified domain name (FQDN) that identifies a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca).

**globally unique identifier (GUID)**: A term used interchangeably with [**universally unique identifier (UUID)**](#gt_c4813fc3-b2e5-4aa3-bde7-421d950d68d3) in Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the value. Specifically, the use of this term does not imply or require that the algorithms described in [[RFC4122]](https://go.microsoft.com/fwlink/?LinkId=90460) or [C706] must be used for generating the [**GUID**](#gt_f49694cc-c350-462d-ab8e-816f0103c6c1). See also [**universally unique identifier (UUID)**](#gt_c4813fc3-b2e5-4aa3-bde7-421d950d68d3).

**group object**: In [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90), a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) has an object class group. A group has a forward link attribute member; the values of this attribute either represent elements of the group (for example, objects of class user or computer) or subsets of the group (objects of class group). The representation of group subsets is called "nested group membership". The back link attribute memberOf enables navigation from group members to the groups containing them. Some groups represent groups of [**security principals**](#gt_f3ef2572-95cf-4c5c-b3c9-551fd648f409) and some do not and are, for instance, used to represent email distribution lists.

**LM hash**: A DES-based cryptographic hash of a cleartext password. See LMOWFv1, as specified in [[MS-NLMP]](%5bMS-NLMP%5d.pdf#Section_b38c36ed28044868a9ff8dd3182128e4) section 3.3.1 (NTLM v1 Authentication), for a normative definition.

**machine account**: An [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) that is associated with individual client or server machines in an [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) domain.

**mixed mode**: A state of an [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) that supports [**domain controllers (DCs)**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) running Windows NT Server 4.0 operating system. Mixed mode does not allow organizations to take advantage of new [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) features such as universal groups, nested group membership, and interdomain group membership. See also [**native mode**](#gt_8adcb510-b25d-459e-9832-90079803d3c0).

**native mode**: A state of an [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) domain in which all current and future [**domain controllers (DCs)**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) use AD style domains. [**Native mode**](#gt_8adcb510-b25d-459e-9832-90079803d3c0) allows organizations to take advantage of the new [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) features such as universal groups, nested group membership, and interdomain group membership.

**Network Data Representation (NDR)**: A specification that defines a mapping from Interface Definition Language (IDL) data types onto octet streams. [**NDR**](#gt_9ebf9540-2c31-43bc-bc56-4a932faabf2d) also refers to the runtime environment that implements the mapping facilities (for example, data provided to [**NDR**](#gt_9ebf9540-2c31-43bc-bc56-4a932faabf2d)). For more information, see [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) and [C706] section 14.

**NT hash**: An MD4- or MD5-based cryptographic hash of a clear text password. For more information, see [MS-NLMP] section 3.3.1 (NTOWFv1, NTLM v1 Authentication), for a normative definition.

**original equipment manufacturer (OEM) code page**: A code page used to translate between non-Unicode encoded strings and UTF-16 encoded strings.

**primary domain controller (PDC)**: A [**domain controller (DC)**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) designated to track changes made to the accounts of all computers on a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca). It is the only computer to receive these changes directly, and is specialized so as to ensure consistency and to eliminate the potential for conflicting entries in the [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) database. A [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) has only one [**PDC**](#gt_663cb13a-8b75-477f-b6e1-bea8f2fba64d).

**RC4**: A variable key-length symmetric encryption algorithm. For more information, see [[SCHNEIER]](https://go.microsoft.com/fwlink/?LinkId=817338) section 17.1.

**read-only domain controller (RODC)**: A [**domain controller (DC)**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) that does not accept originating updates. Additionally, an [**RODC**](#gt_8b0a073b-3099-4efe-8b81-c2886b66a870) does not perform outbound replication. An RODC cannot be the primary domain controller (PDC) for its domain.

**relative distinguished name (RDN)**: The name of an object relative to its parent. This is the leftmost attribute-value pair in the distinguished name (DN) of an object. For example, in the DN "cn=Peter Houston, ou=NTDEV, dc=microsoft, dc=com", the [**RDN**](#gt_22198321-b40b-4c24-b8a2-29e44d9d92b9) is "cn=Peter Houston". For more information, see [[RFC2251]](https://go.microsoft.com/fwlink/?LinkId=90325).

**relative identifier (RID)**: The last item in the series of SubAuthority values in a [**security identifier (SID)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) [[SIDD]](https://go.microsoft.com/fwlink/?LinkId=90516). It distinguishes one account or group from all other accounts and groups in the domain. No two accounts or groups in any domain share the same RID.

**resource group**: A group object whose membership is added to the authorization context only if the server receiving the context is a member of the same domain as the resource group.

**RPC transfer syntax**: A method for encoding messages defined in an Interface Definition Language (IDL) file. Remote procedure call (RPC) can support different encoding methods or transfer syntaxes. For more information, see [C706].

**salt**: A value consisting of random bits used to increase the complexity of dictionary attacks against secret data that is protected through cryptographic means. For details, see [MENEZES] section 10.2.1.

**security descriptor**: A data structure containing the security information associated with a securable object. A [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) identifies an object's owner by its [**security identifier (SID)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d). If access control is configured for the object, its [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) contains a [**discretionary access control list (DACL)**](#gt_d727f612-7a45-48e4-9d87-71735d62b321) with [**SIDs**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) for the [**security principals**](#gt_f3ef2572-95cf-4c5c-b3c9-551fd648f409) who are allowed or denied access. Applications use this structure to set and query an object's security status. The [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) is used to guard access to an object as well as to control which type of auditing takes place when the object is accessed. The [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) format is specified in [[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2) section 2.4.6; a string representation of [**security descriptors**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350), called SDDL, is specified in [MS-DTYP] section 2.5.1.

**security identifier (SID)**: An identifier for [**security principals**](#gt_f3ef2572-95cf-4c5c-b3c9-551fd648f409) that is used to identify an account or a group. Conceptually, the [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) is composed of an account authority portion (typically a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca)) and a smaller integer representing an identity relative to the account authority, termed the [**relative identifier (RID)**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c). The [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) format is specified in [MS-DTYP] section 2.4.2; a string representation of [**SIDs**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) is specified in [MS-DTYP] section 2.4.2 and [[MS-AZOD]](%5bMS-AZOD%5d.pdf#Section_5a0a0a3ec7a742e1b5f2cc8d8bd9739e) section 1.1.1.2.

**security principal**: A unique entity, also referred to as a principal, that can be authenticated by [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90). It frequently corresponds to a human user, but also can be a service that offers a resource to other security principals. Other security principals might be a group, which is a set of principals. Groups are supported by [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90).

**server object**: The database object in the account domain with an object class of samServer.

**system access control list (SACL)**: An access control list (ACL) that controls the generation of audit messages for attempts to access a securable object. The ability to get or set an object's [**SACL**](#gt_c189801e-3752-4715-88f4-17804dad5782) is controlled by a privilege typically held only by system administrators.

**token**: A set of rights and privileges for a given user.

**UAS Compatibility**: A configuration mode that affects protocol behavior constraints specified in this document. "UAS" is an acronym for "User Account Security (Database)" and refers to products that are no longer supported, such as Microsoft NT LAN Manager. The default setting in Windows is "on".

**universal group**: An [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) group that allows [**user objects**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17), global groups, and [**universal groups**](#gt_f46053d6-0708-4094-ac63-57c1bcb73d32) from anywhere in the [**forest**](#gt_fd104241-4fb3-457c-b2c4-e0c18bb20b62) as members. A [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) g is a [**universal group**](#gt_f46053d6-0708-4094-ac63-57c1bcb73d32) if and only if GROUP\_TYPE\_UNIVERSAL\_GROUP is present in g! groupType. A security-enabled universal group is valid for inclusion within ACLs anywhere in the [**forest**](#gt_fd104241-4fb3-457c-b2c4-e0c18bb20b62). If a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) is in [**mixed mode**](#gt_06c1c70e-f2c6-4efd-bff8-474409e69660), then a [**universal group**](#gt_f46053d6-0708-4094-ac63-57c1bcb73d32) cannot be created in that [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca). See also domain local group, security-enabled group.

**universally unique identifier (UUID)**: A 128-bit value. UUIDs can be used for multiple purposes, from tagging objects with an extremely short lifetime, to reliably identifying very persistent objects in cross-process communication such as client and server interfaces, manager entry-point vectors, and RPC objects. UUIDs are highly likely to be unique. UUIDs are also known as [**globally unique identifiers (GUIDs)**](#gt_f49694cc-c350-462d-ab8e-816f0103c6c1) and these terms are used interchangeably in the Microsoft protocol technical documents (TDs). Interchanging the usage of these terms does not imply or require a specific algorithm or mechanism to generate the UUID. Specifically, the use of this term does not imply or require that the algorithms described in [RFC4122] or [C706] must be used for generating the UUID.

**user object**: An object of class user. A user object is a security principal object; the principal is a person or service entity running on the computer. The shared secret allows the person or service entity to authenticate itself, as described in ([MS-AUTHSOD] section 1.1.1.1).

**user profile**: A collection of attributes on a user object used to customize an end-user experience.

**WorldSid**: A [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) with the specific value of S-1-1-0.

**MAY, SHOULD, MUST, SHOULD NOT, MUST NOT:** These terms (in all caps) are used as defined in [[RFC2119]](https://go.microsoft.com/fwlink/?LinkId=90317). All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

## References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the [Errata](https://go.microsoft.com/fwlink/?linkid=850906).

### Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com). We will assist you in finding the relevant information.

[AES-CBC] McGrew, D. and Foley, J., "Authenticated Encryption with AES-CBC and HMAC-SHA", [https://tools.ietf.org/id/draft-mcgrew-aead-aes-cbc-hmac-sha2-03.html](https://go.microsoft.com/fwlink/?linkid=2162731)

[C706] The Open Group, "DCE 1.1: Remote Procedure Call", C706, August 1997, [https://publications.opengroup.org/c706](https://go.microsoft.com/fwlink/?LinkId=89824)

**Note** Registration is required to download the document.

[E164] ITU-T, "The International Public Telecommunication Numbering Plan", Recommendation E.164, February 2005, [http://www.itu.int/rec/T-REC-E.164/e](https://go.microsoft.com/fwlink/?LinkId=89855)

**Note** There is a charge to download the specification.

[FIPS46-2] FIPS PUBS, "Data Encryption Standard (DES)", FIPS PUB 46-2, December 1993, [https://csrc.nist.gov/publications/detail/fips/46/2/archive/1993-12-30](https://go.microsoft.com/fwlink/?LinkId=89871)

[FIPS81] FIPS PUBS, "DES Modes of Operation", December 1980, [https://csrc.nist.gov/csrc/media/publications/fips/81/archive/1980-12-02/documents/fips81.pdf](https://go.microsoft.com/fwlink/?LinkId=89874)

[GRAY] Gray, J., and Reuter, A., "Transaction Processing: Concepts and Techniques", The Morgan Kaufmann Series in Data Management Systems, San Francisco: Morgan Kaufmann Publishers, 1992, Hardcover ISBN: 9781558601901.

[MS-ADA1] Microsoft Corporation, "[Active Directory Schema Attributes A-L](%5bMS-ADA1%5d.pdf#Section_19528560f41e4623a406dabcfff0660f)".

[MS-ADA2] Microsoft Corporation, "[Active Directory Schema Attributes M](%5bMS-ADA2%5d.pdf#Section_e20ebc4e528540bab3bdffcb81c2783e)".

[MS-ADA3] Microsoft Corporation, "[Active Directory Schema Attributes N-Z](%5bMS-ADA3%5d.pdf#Section_4517e8353ee644d4bb95a94b6966bfb0)".

[MS-ADSC] Microsoft Corporation, "[Active Directory Schema Classes](%5bMS-ADSC%5d.pdf#Section_9abb5e97123d4da99557b353ab79b830)".

[MS-ADTS] Microsoft Corporation, "[Active Directory Technical Specification](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a)".

[MS-CIFS] Microsoft Corporation, "[Common Internet File System (CIFS) Protocol](%5bMS-CIFS%5d.pdf#Section_d416ff7cc536406ea9514f04b2fd1d2b)".

[MS-DRSR] Microsoft Corporation, "[Directory Replication Service (DRS) Remote Protocol](%5bMS-DRSR%5d.pdf#Section_f977faaa673e4f66b9bf48c640241d47)".

[MS-DTYP] Microsoft Corporation, "[Windows Data Types](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2)".

[MS-ERREF] Microsoft Corporation, "[Windows Error Codes](%5bMS-ERREF%5d.pdf#Section_1bc92ddfb79e413cbbaa99a5281a6c90)".

[MS-KILE] Microsoft Corporation, "[Kerberos Protocol Extensions](%5bMS-KILE%5d.pdf#Section_2a32282edd484ad9a542609804b02cc9)".

[MS-LSAD] Microsoft Corporation, "[Local Security Authority (Domain Policy) Remote Protocol](%5bMS-LSAD%5d.pdf#Section_1b5471ef4c334a91b079dfcbb82f05cc)".

[MS-LSAT] Microsoft Corporation, "[Local Security Authority (Translation Methods) Remote Protocol](%5bMS-LSAT%5d.pdf#Section_1ba21e6fd8a9462c91534375f2020894)".

[MS-NLMP] Microsoft Corporation, "[NT LAN Manager (NTLM) Authentication Protocol](%5bMS-NLMP%5d.pdf#Section_b38c36ed28044868a9ff8dd3182128e4)".

[MS-NRPC] Microsoft Corporation, "[Netlogon Remote Protocol](%5bMS-NRPC%5d.pdf#Section_ff8f970f3e3740f7bd4baf7336e4792f)".

[MS-PAC] Microsoft Corporation, "[Privilege Attribute Certificate Data Structure](%5bMS-PAC%5d.pdf#Section_166d8064c86341e19c23edaaa5f36962)".

[MS-RPCE] Microsoft Corporation, "[Remote Procedure Call Protocol Extensions](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15)".

[MS-SMB2] Microsoft Corporation, "[Server Message Block (SMB) Protocol Versions 2 and 3](%5bMS-SMB2%5d.pdf#Section_5606ad475ee0437a817e70c366052962)".

[MS-SMB] Microsoft Corporation, "[Server Message Block (SMB) Protocol](%5bMS-SMB%5d.pdf#Section_f210069c70864dc2885e861d837df688)".

[MSKB-3072595] Microsoft Corporation, "Vulnerability in Active Directory service could allow denial of service, September 2015", [https://support.microsoft.com/en-us/kb/3072595](https://go.microsoft.com/fwlink/?LinkId=626613)

[MSKB-3149090] Microsoft Corporation, "MS16-047: Description of the security update for SAM and LSAD remote protocols", April 2016, [https://support.microsoft.com/en-us/kb/3149090](https://go.microsoft.com/fwlink/?LinkId=786963)

[MSKB-4012218] Microsoft Corporation, "March 2017 Preview of Monthly Quality Rollup for Windows 8.1 and Windows Server 2012 R2", March 2017, <https://support.microsoft.com/en-us/topic/march-2017-preview-of-monthly-quality-rollup-for-windows-8-1-and-windows-server-2012-r2-3f3d4166-4b8c-6f88-17c9-64cb0378acd4>

**Note** https://go.microsoft.com/fwlink/?linkid=2163444

[MSKB-4012220] Microsoft Corporation, "March 2017 Preview of Monthly Quality Rollup for Windows Server 2012", March 2017, [https://support.microsoft.com/kb/4012220](https://go.microsoft.com/fwlink/?linkid=2163626)

[MSKB-4012606] Microsoft Corporation, "March 14, 2017—KB4012606 (OS Build 10240.17319)", March 2017, [https://support.microsoft.com/kb/4012606](https://go.microsoft.com/fwlink/?linkid=2163542)

[MSKB-4013198] Microsoft Corporation, "March 14, 2017—KB4013198 (OS Build 10586.839)", March 2017, [https://support.microsoft.com/kb/4013198](https://go.microsoft.com/fwlink/?linkid=2163543)

[MSKB-4102219] Microsoft Corporation, "March 2017 Preview of Monthly Quality Rollup for Windows 8.1 and Windows Server 2012 R2", March 2017, [https://support.microsoft.com/topic/march-2017-preview-of-monthly-quality-rollup-for-windows-8-1-and-windows-server-2012-r2-3f3d4166-4b8c-6f88-17c9-64cb0378acd4](https://go.microsoft.com/fwlink/?linkid=2163445)

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", RFC 1123, October 1989, [http://www.ietf.org/rfc/rfc1123.txt](https://go.microsoft.com/fwlink/?LinkId=90268)

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, April 1992, [http://www.ietf.org/rfc/rfc1321.txt](https://go.microsoft.com/fwlink/?LinkId=90275)

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, [http://www.rfc-editor.org/rfc/rfc2119.txt](https://go.microsoft.com/fwlink/?LinkId=90317)

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., et al., "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999, [http://www.rfc-editor.org/rfc/rfc2617.txt](https://go.microsoft.com/fwlink/?LinkId=90373)

[RFC3961] Raeburn, K., "Encryption and Checksum Specifications for Kerberos 5", RFC 3961, February 2005, [http://www.ietf.org/rfc/rfc3961.txt](https://go.microsoft.com/fwlink/?LinkId=90450)

[RFC3962] Raeburn, K., "Advanced Encryption Standard (AES) Encryption for Kerberos 5", RFC 3962, February 2005, [http://www.ietf.org/rfc/rfc3962.txt](https://go.microsoft.com/fwlink/?LinkId=90451)

[RFC4120] Neuman, C., Yu, T., Hartman, S., and Raeburn, K., "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005, [https://www.rfc-editor.org/rfc/rfc4120.txt](https://go.microsoft.com/fwlink/?LinkId=90458)

[RFC4122] Leach, P., Mealling, M., and Salz, R., "A Universally Unique Identifier (UUID) URN Namespace", RFC 4122, July 2005, [http://www.rfc-editor.org/rfc/rfc4122.txt](https://go.microsoft.com/fwlink/?LinkId=90460)

[UNICODE3.1] The Unicode Consortium, "Unicode Data 3.1.0", February 2001, [http://www.unicode.org/Public/3.1-Update/UnicodeData-3.1.0.txt](https://go.microsoft.com/fwlink/?LinkId=90551)

[X501] ITU-T, "Information Technology - Open Systems Interconnection - The Directory: The Models", Recommendation X.501, August 2005, [http://www.itu.int/rec/T-REC-X.501-200508-S/en](https://go.microsoft.com/fwlink/?LinkId=98847)

### Informative References

[LAMPORT] Lamport, L., "Time, Clocks, and the Ordering of Events in a Distributed System", July 1978, [http://portal.acm.org/citation.cfm?id=359563](https://go.microsoft.com/fwlink/?LinkId=89928)

[MS-ADOD] Microsoft Corporation, "[Active Directory Protocols Overview](%5bMS-ADOD%5d.pdf#Section_5ff67bf4c14548cb89cd4f5482d94664)".

[MS-AUTHSOD] Microsoft Corporation, "[Authentication Services Protocols Overview](%5bMS-AUTHSOD%5d.pdf#Section_953d700a57cb4cf7b0c3a64f34581cc9)".

[MS-AZOD] Microsoft Corporation, "[Authorization Protocols Overview](%5bMS-AZOD%5d.pdf#Section_5a0a0a3ec7a742e1b5f2cc8d8bd9739e)".

[MS-SYS-ARCHIVE] Microsoft Corporation, "Windows System Overview", [https://docs.microsoft.com/en-us/openspecs/windows\_protocols/MS-WINPROTLP/df36f95e-6a6b-48d6-a3ae-35a17674f546](https://go.microsoft.com/fwlink/?LinkId=311687)

[MS-UCODEREF] Microsoft Corporation, "[Windows Protocols Unicode Reference](%5bMS-UCODEREF%5d.pdf#Section_4a045e08fc294f22baf416f38c2825fb)".

[MS-WPO] Microsoft Corporation, "[Windows Protocols Overview](%5bMS-WPO%5d.pdf#Section_c5f54a7765be40a0bb829e4181d8ab67)".

[MSDN-CP] Microsoft Corporation, "Code Page Identifiers", [https://docs.microsoft.com/en-us/windows/desktop/Intl/code-page-identifiers](https://go.microsoft.com/fwlink/?LinkId=89981)

[MSDN-NMF] Microsoft Corporation, "Network Management Functions", [http://msdn.microsoft.com/en-us/library/aa370675.aspx](https://go.microsoft.com/fwlink/?LinkId=90052)

[MSDOCS-RESTRICTRMTSAM] Microsoft Corporation, "Network access: Restrict clients allowed to make remote calls to SAM", [https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/network-access-restrict-clients-allowed-to-make-remote-sam-calls](https://go.microsoft.com/fwlink/?linkid=2163448)

[SCHNEIER] Schneier, B., "Applied Cryptography, Second Edition", John Wiley and Sons, 1996, ISBN: 0471117099, [http://www.wiley.com/WileyCDA/WileyTitle/productCd-0471117099.html](https://go.microsoft.com/fwlink/?LinkId=817338)

## Overview

The goal of this protocol is to enable IT administrators and end users to manage users, groups, and computers. IT administrators and their delegates generally have full access control to these entities, and consequently can manage the entities' life cycles. End users are allowed to make changes to their own data (in most cases, limited to just their passwords).

This protocol achieves its goal by enabling the creation, reading, updating, and deleting of [**security principal**](#gt_f3ef2572-95cf-4c5c-b3c9-551fd648f409) information. These security principals could be in any account store. Windows implements this protocol, for example, in a directory service ([**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90)) and in a computer-local security account database. In this specification, normative differences in the protocol between these two cases are indicated by referring to the configuration of the server as a "DC" or "non-DC" configuration, respectively, where "DC" stands for [**domain controller (DC)**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd).

It is helpful to consider the following two perspectives when understanding and implementing this protocol:

* Object-based perspective (see section [1.3.1](#Section_8aaff2f71edd41a0ab584807ac6124c5))
* Method-based perspective (see section [1.3.2](#Section_d7b625964a46455692dc3aba6d517907))

### Object-Based Perspective

The object-based perspective shows that the protocol exposes five main object abstractions: a [**server object**](#gt_62a8c543-5998-480b-8fa7-41a8f04a18e5), a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1), a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a), an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0) (an "alias" being a type of group), and a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17). A client obtains a "handle" (an RPC context handle) to one of these objects and then performs one or more actions on the object.

The following is a brief listing of methods that operate on each of the respective object types.

Server Object:

* [SamrSetSecurityObject](#Section_6666a06658cf4118bf4bdd54ed55ecf0)
* [SamrQuerySecurityObject](#Section_0ecf8fecd17e4a88b7f1e0f0f66790db)
* [SamrEnumerateDomainsInSamServer](#Section_2142fd2d085442c1a9fb2fe964e381ce)
* [SamrOpenDomain](#Section_ba710c905b1242f89e5ad4aacc1329fa)
* [SamrLookupDomainInSamServer](#Section_47492d59e0954398b03e8a062b989123)
* [SamrCloseHandle](#Section_55d134dfe25748ad8afacb2ca45cd3cc)

Domain Object:

* SamrSetSecurityObject
* SamrQuerySecurityObject
* [SamrLookupNamesInDomain](#Section_d91271c67b2e419499278fabfa429f90)
* [SamrLookupIdsInDomain](#Section_c870951c74b347149857224595ffc61a)
* [SamrEnumerateGroupsInDomain](#Section_e0b7a4b7ecfc405f9d7d32b3cd2cd6c8)
* [SamrEnumerateUsersInDomain](#Section_6bdc92c0c6924ffb9de765858b68da75)
* [SamrEnumerateAliasesInDomain](#Section_ce340cffedef4356ace033d2874d306b)
* [SamrOpenGroup](#Section_d396e6c9d04a4729b0d8f50f2748f3c8)
* [SamrOpenAlias](#Section_c3696b7620054eba820d91fdded3422d)
* [SamrOpenUser](#Section_0aee1c31ec404633bb560cf8429093c0)
* [SamrQueryInformationDomain](#Section_5d6a2817caa941caa269fd13ecbb4fa8)
* [SamrQueryInformationDomain2](#Section_fbdbbd7401064ca2941e36ded0be0c62)
* [SamrCreateGroupInDomain](#Section_175c1cf94fa248379e5bbb1f0f950bee)
* [SamrCreateAliasInDomain](#Section_7e505875e44f4f9d922b7ff3871c752d)
* [SamrCreateUserInDomain](#Section_cd0efce013fb410687db4bef73bb4b1b)
* [SamrCreateUser2InDomain](#Section_a98d7fbb17354fbfb41aef363c899002)
* [SamrSetInformationDomain](#Section_9b7ae0b0bd1141339c62fba7095aee12)
* [SamrGetAliasMembership](#Section_0318404522084c02b38bef955d6dc3ef)
* [SamrGetDisplayEnumerationIndex](#Section_af95abf82e1142c394aaba72f8f96c09)
* [SamrGetDisplayEnumerationIndex2](#Section_bd429624f2d547178aa2659952c3e209)
* [SamrQueryDisplayInformation](#Section_c1458942f2d54317a888abd27abad504)
* [SamrQueryDisplayInformation2](#Section_9aa37aa4f77e49b199935c92ba97cd62)
* [SamrQueryDisplayInformation3](#Section_7cd1c6d088194a5897215be588c1d4ad)
* SamrCloseHandle
* [SamrRemoveMemberFromForeignDomain](#Section_03afc843584d473b834a3f5a1ac86cce)
* [SamrRidToSid](#Section_00ff8192a4f645ba9f65917e46b6a693)

Group Object:

* SamrSetSecurityObject
* SamrQuerySecurityObject
* [SamrQueryInformationGroup](#Section_8fd0003f807640ae872f9cdb6576b3c7)
* [SamrSetInformationGroup](#Section_e66db19f600a481bbc4e23953433255d)
* [SamrDeleteGroup](#Section_664ed55f198a4775b9d4398131dba577)
* [SamrAddMemberToGroup](#Section_3c70fec36a2e48efbd4eaa2b3a1cd96a)
* [SamrRemoveMemberFromGroup](#Section_613ff6635863443089238f1edc05310b)
* [SamrGetMembersInGroup](#Section_3ed5030d88a342caa6e08c12aa2fdfbd)
* SamrCloseHandle
* [SamrSetMemberAttributesOfGroup](#Section_a4db0c2755404d2fb11108db1ee0b5a4)
* SamrRidToSid

Alias Object:

* SamrSetSecurityObject
* SamrQuerySecurityObject
* [SamrQueryInformationAlias](#Section_599d6624d93b46de827a58eb0e7527bd)
* [SamrSetInformationAlias](#Section_ba787e4e3a4b47a2aca3c3ac2d2c511e)
* [SamrDeleteAlias](#Section_ecb88df184d148c8b49cd3f444943100)
* [SamrAddMemberToAlias](#Section_9a5d2c35e84b4e59b7b096c6fa0fc8d7)
* [SamrRemoveMemberFromAlias](#Section_7e99402967a244fd84397640fe5a376b)
* [SamrGetMembersInAlias](#Section_5909af7f858b4cc7a2e4f852070d479d)
* [SamrAddMultipleMembersToAlias](#Section_6d981290a203408094b04a26de22d8c5)
* [SamrRemoveMultipleMembersFromAlias](#Section_7b2455d23f1344a7996950c240562c42)
* SamrRidToSid

User Object:

* SamrSetSecurityObject
* SamrQuerySecurityObject
* [SamrQueryInformationUser](#Section_4ad8d54c0d5a4d5a9e9a1bc9ee008d47)
* [SamrQueryInformationUser2](#Section_29ab27f661da4c7d863ce228ee798f4d)
* [SamrSetInformationUser](#Section_538222f71b894811949a0eac62e38dce)
* [SamrSetInformationUser2](#Section_99ee9f3943e84bbaac3a82e0c0e0699e)
* [SamrDeleteUser](#Section_4643a57956ec4c66a1ef4ab78dd21d73)
* [SamrGetGroupsForUser](#Section_a4adbf20040f4416a960e5b7917fdae7)
* [SamrChangePasswordUser](#Section_9699d8cae1a4433ca8c3d7bebeb01476)
* [SamrGetUserDomainPasswordInformation](#Section_db4bedfe465147439381d74c3ad8c41c)
* SamrCloseHandle
* SamrRidToSid

For example, to set a policy that limits the minimum length of passwords to eight characters for all users, a client opens a handle to a domain object and updates the minimum length password policy setting via a parameter field called *MinPasswordLength*. The call sequence from the client appears as follows (with the parameter information removed for brevity):

(a) Send a [SamrConnect5](#Section_c842a8970a424ca5a6072afd05271dae) request; receive the SamrConnect5 reply.

(b) Send a SamrOpenDomain request; receive the SamrOpenDomain reply.

(c) Send a SamrSetInformationDomain request; receive the SamrSetInformationDomain reply.

(d) Send a SamrCloseHandle request; receive the SamrCloseHandle reply.

(e) Send a SamrCloseHandle request; receive the SamrCloseHandle reply.

This sequence is expanded in the following brief explanation:

Step (a): Using the network address of a server that implements this protocol, a client makes a SamrConnect5 request to obtain a handle to a server object. This server handle is necessary to obtain a subsequent handle to a domain object.

Step (b): Using the handle returned from SamrConnect5, the client makes a SamrOpenDomain request to obtain a handle to a domain object.

Step (c): Using the handle returned from SamrOpenDomain, the client makes a SamrSetInformationDomain request, setting the *MinPasswordLength* parameter field to eight.

Steps (d) and (e): The client closes the handles returned from SamrOpenDomain and SamrConnect5 by using SamrCloseHandle. These steps release server resources associated with the handle; the order in which the handles are released is not important.

Section [4.1](#Section_3d8e23d8d9df481f83b39175f980294c) provides an additional example.

### Method-Based Perspective

The method-based perspective is used to show a common set of operations for each object type. The operations fall into patterns. A list of the patterns and associated methods, along with a description of each pattern, is shown below.

* Open Pattern

This pattern returns an RPC context handle that references a specific object type. A client uses this pattern by specifying a specific access for the handle in the request, and using the returned handle to call other methods that require the returned handle along with the associated access. For example, calling the method [SamrSetInformationDomain](#Section_9b7ae0b0bd1141339c62fba7095aee12) requires a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) handle that has been opened with DOMAIN\_WRITE\_PASSWORD\_PARAMS. For more information on the range of accesses for a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1), see section [2.2.1.4](#Section_aef23495f6aa48e9aebc22e022a2b4eb).

[SamrConnect2](#Section_1076eb2a4f514c5aa7c7a78323b06198), [SamrConnect4](#Section_c1b14e670f4647a69fa70689d284afc9), and [SamrConnect5](#Section_c842a8970a424ca5a6072afd05271dae) are distinguished from the other methods in this pattern in that they are the first methods that a client calls prior to a calling any other handle-based methods.

The methods that follow the open pattern are as follows:

* + SamrConnect5
  + SamrConnect4
  + SamrConnect2
  + [SamrOpenDomain](#Section_ba710c905b1242f89e5ad4aacc1329fa)
  + [SamrOpenGroup](#Section_d396e6c9d04a4729b0d8f50f2748f3c8)
  + [SamrOpenAlias](#Section_c3696b7620054eba820d91fdded3422d)
  + [SamrOpenUser](#Section_0aee1c31ec404633bb560cf8429093c0)
* Enumerate Pattern

This pattern allows a client to obtain a complete list of all objects of a certain type (domain, group, alias, or user).

The methods that follow the enumerate pattern are as follows:

* + [SamrEnumerateDomainsInSamServer](#Section_2142fd2d085442c1a9fb2fe964e381ce)
  + [SamrEnumerateGroupsInDomain](#Section_e0b7a4b7ecfc405f9d7d32b3cd2cd6c8)
  + [SamrEnumerateAliasesInDomain](#Section_ce340cffedef4356ace033d2874d306b)
  + [SamrEnumerateUsersInDomain](#Section_6bdc92c0c6924ffb9de765858b68da75)
* Selective Enumerate Pattern

This pattern allows a client to obtain a partial list of objects based on the name of the objects. These methods, for example, allow a client to obtain a bounded number of objects from a virtual list of objects sorted alphabetically by name starting with a client-specified prefix, such as "Chr". User interface programs use these methods to allow the end user to quickly find an object, given partial knowledge of the object's name.

The methods that follow the selective enumerate pattern are as follows:

* + [SamrQueryDisplayInformation3](#Section_7cd1c6d088194a5897215be588c1d4ad)
  + [SamrQueryDisplayInformation2](#Section_9aa37aa4f77e49b199935c92ba97cd62)
  + [SamrQueryDisplayInformation](#Section_c1458942f2d54317a888abd27abad504)
  + [SamrGetDisplayEnumerationIndex2](#Section_bd429624f2d547178aa2659952c3e209)
  + [SamrGetDisplayEnumerationIndex](#Section_af95abf82e1142c394aaba72f8f96c09)
* Create Pattern

This pattern allows specified objects to be created. A handle to the newly created object is returned.

The methods that follow the create pattern are as follows:

* + [SamrCreateGroupInDomain](#Section_175c1cf94fa248379e5bbb1f0f950bee)
  + [SamrCreateAliasInDomain](#Section_7e505875e44f4f9d922b7ff3871c752d)
  + [SamrCreateUser2InDomain](#Section_a98d7fbb17354fbfb41aef363c899002)
  + [SamrCreateUserInDomain](#Section_cd0efce013fb410687db4bef73bb4b1b)
* Query Pattern

This pattern allows specified attributes of an object to be returned. The client specifies which attributes to return by using an "information level". The information level is an enumeration that the server understands and translates into a specific structure to return; the structure contains the attributes indicated by the information level.

To retrieve the name of a user, for example, a client specifies the "UserAccountNameInformation" information level in the [SamrQueryInformationUser](#Section_4ad8d54c0d5a4d5a9e9a1bc9ee008d47) method.

The methods that follow the query pattern are as follows:

* + [SamrQueryInformationDomain2](#Section_fbdbbd7401064ca2941e36ded0be0c62)
  + [SamrQueryInformationDomain](#Section_5d6a2817caa941caa269fd13ecbb4fa8)
  + [SamrQueryInformationGroup](#Section_8fd0003f807640ae872f9cdb6576b3c7)
  + [SamrQueryInformationAlias](#Section_599d6624d93b46de827a58eb0e7527bd)
  + [SamrQueryInformationUser2](#Section_29ab27f661da4c7d863ce228ee798f4d)
  + SamrQueryInformationUser
* Set Pattern

This pattern allows specified object attributes to be set. The client indicates the attributes that are to be updated by specifying an "information level". Similar to the query pattern of methods, the information level specifies the attributes that are being sent in the request.

The methods that follow the set pattern are as follows:

* + SamrSetInformationDomain
  + [SamrSetInformationGroup](#Section_e66db19f600a481bbc4e23953433255d)
  + [SamrSetInformationAlias](#Section_ba787e4e3a4b47a2aca3c3ac2d2c511e)
  + [SamrSetInformationUser2](#Section_99ee9f3943e84bbaac3a82e0c0e0699e)
  + [SamrSetInformationUser](#Section_538222f71b894811949a0eac62e38dce)
* Delete Pattern

This pattern allows a client to delete a specified object.

The methods that follow the delete pattern are as follows:

* + [SamrDeleteGroup](#Section_664ed55f198a4775b9d4398131dba577)
  + [SamrDeleteAlias](#Section_ecb88df184d148c8b49cd3f444943100)
  + [SamrDeleteUser](#Section_4643a57956ec4c66a1ef4ab78dd21d73)
* Membership Pattern

This pattern allows a client to add to, remove from, or query the membership list for either a group or an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0).

The methods that follow the membership pattern are as follows:

* + [SamrAddMemberToGroup](#Section_3c70fec36a2e48efbd4eaa2b3a1cd96a)
  + [SamrRemoveMemberFromGroup](#Section_613ff6635863443089238f1edc05310b)
  + [SamrAddMemberToAlias](#Section_9a5d2c35e84b4e59b7b096c6fa0fc8d7)
  + [SamrRemoveMemberFromAlias](#Section_7e99402967a244fd84397640fe5a376b)
  + [SamrRemoveMemberFromForeignDomain](#Section_03afc843584d473b834a3f5a1ac86cce)
  + [SamrGetMembersInGroup](#Section_3ed5030d88a342caa6e08c12aa2fdfbd)
  + [SamrGetMembersInAlias](#Section_5909af7f858b4cc7a2e4f852070d479d)
  + [SamrAddMultipleMembersToAlias](#Section_6d981290a203408094b04a26de22d8c5)
  + [SamrRemoveMultipleMembersFromAlias](#Section_7b2455d23f1344a7996950c240562c42)
* Membership-Of Pattern

This pattern allows a client to obtain the groups or aliases that a user or collection of [**security identifiers (SIDs)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) is a member of.

The methods that follow the membership-of pattern are as follows:

* + [SamrGetGroupsForUser](#Section_a4adbf20040f4416a960e5b7917fdae7)
  + [SamrGetAliasMembership](#Section_0318404522084c02b38bef955d6dc3ef)
* Change Password Pattern

This pattern allows a client to change a password on a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17). The client provides the current password and new password, and the server verifies that the client-presented current password matches the server-persisted current password for the user. If there is a match, the new password is persisted.

The methods that follow the change password pattern are as follows:

* + [SamrChangePasswordUser](#Section_9699d8cae1a4433ca8c3d7bebeb01476)
  + [SamrOemChangePasswordUser2](#Section_8d0bf63efa5f4c75be22558c52075842)
  + [SamrUnicodeChangePasswordUser2](#Section_acb3204ada8b478e91391ea589edb880)
* Lookup Pattern

This pattern allows a client to translate between a [**relative identifier (RID)**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) or SID, and a user-friendly display name (the name of the object).

The methods that follow the lookup pattern are as follows:

* + [SamrLookupDomainInSamServer](#Section_47492d59e0954398b03e8a062b989123)
  + [SamrLookupNamesInDomain](#Section_d91271c67b2e419499278fabfa429f90)
  + [SamrLookupIdsInDomain](#Section_c870951c74b347149857224595ffc61a)
* Security Pattern

This pattern allows a client to specify or query access control with a granularity of individual objects.

The methods that follow the security pattern are as follows:

* + [SamrSetSecurityObject](#Section_6666a06658cf4118bf4bdd54ed55ecf0)
  + [SamrQuerySecurityObject](#Section_0ecf8fecd17e4a88b7f1e0f0f66790db)
* Miscellaneous

The following methods do not fall into a general pattern; see the message processing sections for details about each one. A brief description of each method follows:

* + [SamrGetUserDomainPasswordInformation](#Section_db4bedfe465147439381d74c3ad8c41c): This method obtains information about the password policy on the account domain, given a user handle. Applications that allow end users to change their passwords can use this method to display policy information to an end user.
  + [SamrGetDomainPasswordInformation](#Section_1a8841487e904a93b4c47a0006f59026): This method is similar to the SamrGetUserDomainPasswordInformation method, except that the server does not enforce any security, and a user handle is not needed.
  + [SamrRidToSid](#Section_00ff8192a4f645ba9f65917e46b6a693): This method returns a SID given a RID returned by any of the methods in this interface.[<1>](#Appendix_A_1" \o "Product behavior note 1)
  + [SamrSetDSRMPassword](#Section_9bcad7d2b8e14d28a03386040b7b3ce9): This method allows a client to set the password on a local account (an account not stored in [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90)) on a [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd). This is useful for recovery scenarios where Active Directory does not start.
  + [SamrValidatePassword](#Section_c78a7239f8fc4a42bb71321e897dc046): This method allows applications that store passwords to validate the strength of the passwords against the account domain policy.
  + [SamrSetMemberAttributesOfGroup](#Section_a4db0c2755404d2fb11108db1ee0b5a4): This method allows a server to configure extra authorization information associated with a group membership. This method is ignored in DC scenarios.
  + [SamrCloseHandle](#Section_55d134dfe25748ad8afacb2ca45cd3cc): This method releases server resources associated with the RPC context handle that is passed as a parameter.

## Relationship to Other Protocols

This protocol depends on the RPC protocol because it uses RPC as a transport.

The server-side protocol relationships for non-[**domain controller**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) and domain controller configurations are illustrated in the following diagrams:

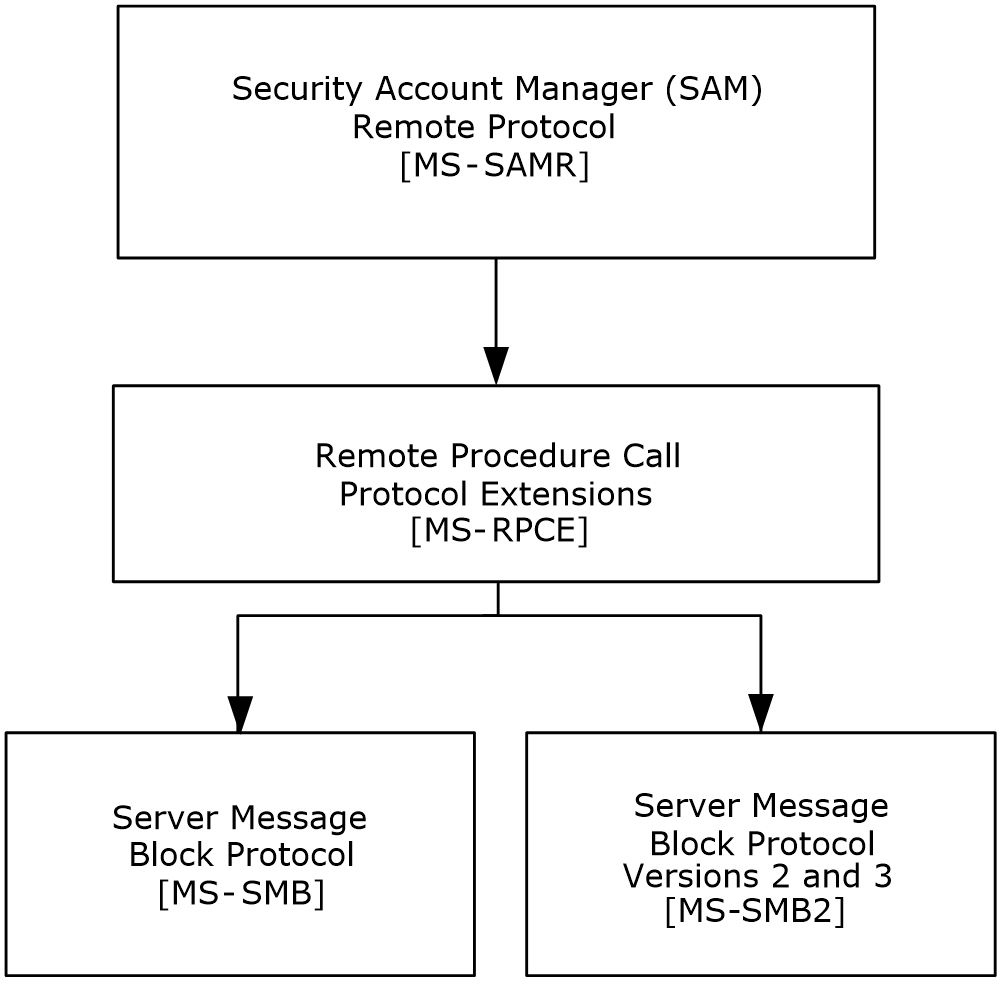


Figure 1: Server-side protocol relationships for a non-domain controller configuration

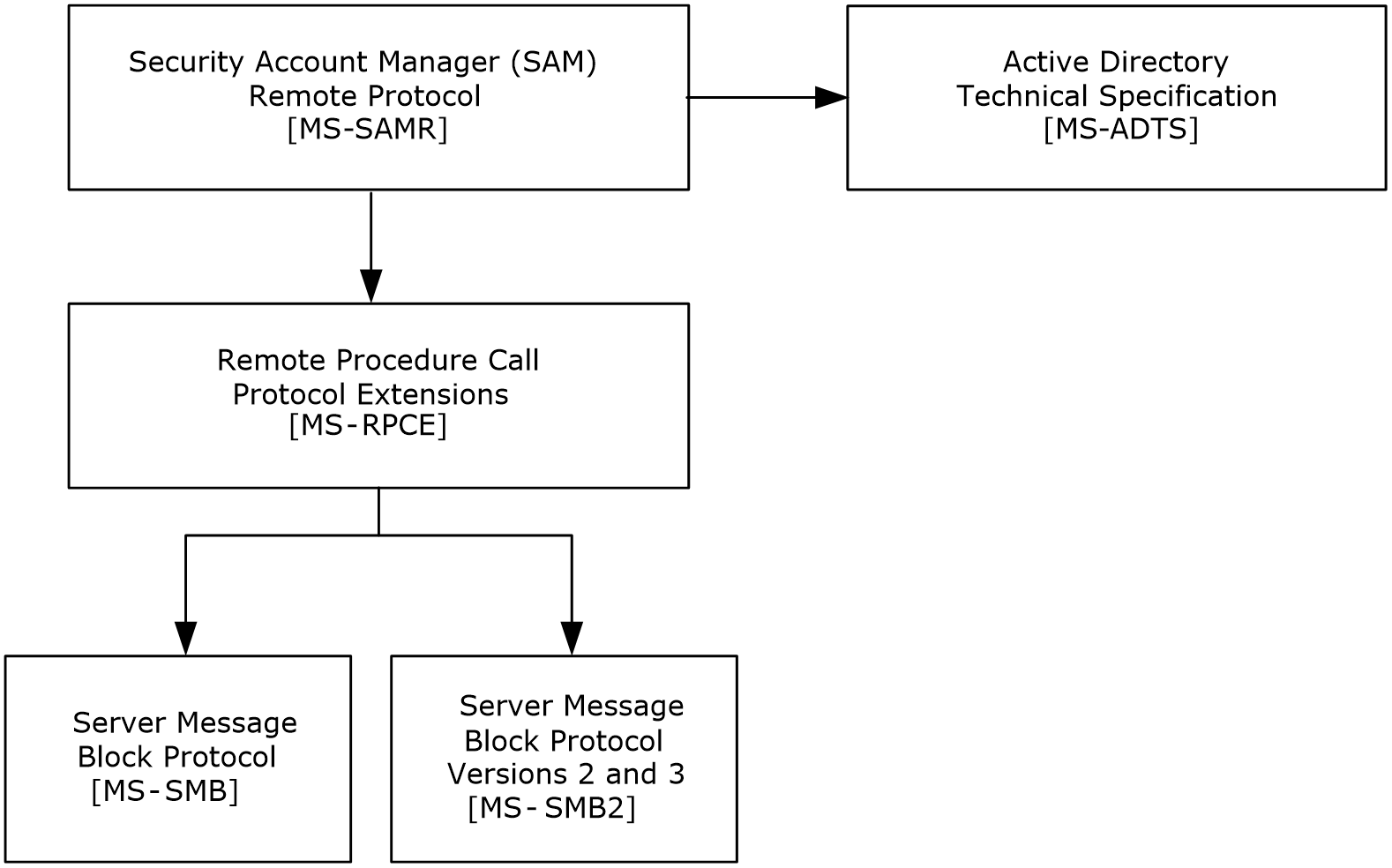


Figure 2: Server-side protocol relationships for a domain controller configuration

In the DC configuration, the data manipulated by the server of this protocol is stored in [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) and is therefore replicated by the replication protocol (described in [[MS-DRSR]](%5bMS-DRSR%5d.pdf#Section_f977faaa673e4f66b9bf48c640241d47)), made available through the LDAP interface (see [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 3.1.1.3), and replicated by the NETLOGON replication interface (as specified in [[MS-NRPC]](%5bMS-NRPC%5d.pdf#Section_ff8f970f3e3740f7bd4baf7336e4792f)). The data manipulated by the server of this protocol is used as a [**security principal**](#gt_f3ef2572-95cf-4c5c-b3c9-551fd648f409) database for authentication protocols such as NTLM [[MS-NLMP]](%5bMS-NLMP%5d.pdf#Section_b38c36ed28044868a9ff8dd3182128e4) and Kerberos [[MS-KILE]](%5bMS-KILE%5d.pdf#Section_2a32282edd484ad9a542609804b02cc9).

## Prerequisites/Preconditions

An [**original equipment manufacturer (OEM) code page**](#gt_442ab13f-d2c1-4128-b1db-f3bea4b8224e) has to be configured in the server implementation. This requirement enables the server to accept data that is encoded in an OEM code page, as well as to return select results that are encoded in an OEM code page.

The client implementation must know the network address of the server. The network address must satisfy the requirements of a network address for the underlying transport of RPC. When using RPC over SMB, for example, the network address must be a network address that is compatible with the Server Message Block (SMB) Protocol ([[MS-SMB]](%5bMS-SMB%5d.pdf#Section_f210069c70864dc2885e861d837df688) or [[MS-SMB2]](%5bMS-SMB2%5d.pdf#Section_5606ad475ee0437a817e70c366052962)), such as a NETBIOS name.

## Applicability Statement

This protocol is useful for manipulating an account database consisting of users, groups, and other [**security principals**](#gt_f3ef2572-95cf-4c5c-b3c9-551fd648f409). This protocol can be used equally well for a database that is backed by a distributed, replicated system, as well as a small, single-instance scenario, such as a single machine.[<2>](#Appendix_A_2" \o "Product behavior note 2)[<3>](#Appendix_A_3" \o "Product behavior note 3)

## Versioning and Capability Negotiation

### Method Introduction

See the following product-behavior citation for a timeline of when each method was introduced.[<4>](#Appendix_A_4" \o "Product behavior note 4)

### Method Versioning

Clients determine whether a method is supported by attempting to invoke the method. If the transport, RPC, returns the error RPC\_S\_PROCNUM\_OUT\_OF\_RANGE (defined in section [2.2.1.16](#Section_775e581d03f34d2dafa45e9ad3604fee)), the client tries the deprecated equivalent of the invoked method if there is one. The following table describes the deprecated method to invoke if the current method is not supported.[<5>](#Appendix_A_5" \o "Product behavior note 5)

| Current method | Old method (in order of preference) |
| --- | --- |
| [SamrQueryInformationDomain2](#Section_fbdbbd7401064ca2941e36ded0be0c62) | [SamrQueryInformationDomain](#Section_5d6a2817caa941caa269fd13ecbb4fa8) |
| [SamrCreateUser2InDomain](#Section_a98d7fbb17354fbfb41aef363c899002) | [SamrCreateUserInDomain](#Section_cd0efce013fb410687db4bef73bb4b1b) |
| [SamrQueryDisplayInformation3](#Section_7cd1c6d088194a5897215be588c1d4ad) | [SamrQueryDisplayInformation2](#Section_9aa37aa4f77e49b199935c92ba97cd62)  [SamrQueryDisplayInformation](#Section_c1458942f2d54317a888abd27abad504) |
| [SamrGetDisplayEnumerationIndex2](#Section_bd429624f2d547178aa2659952c3e209) | [SamrGetDisplayEnumerationIndex](#Section_af95abf82e1142c394aaba72f8f96c09) |
| [SamrSetInformationUser2](#Section_99ee9f3943e84bbaac3a82e0c0e0699e) | [SamrSetInformationUser](#Section_538222f71b894811949a0eac62e38dce) |
| [SamrConnect5](#Section_c842a8970a424ca5a6072afd05271dae) | [SamrConnect4](#Section_c1b14e670f4647a69fa70689d284afc9)  [SamrConnect2](#Section_1076eb2a4f514c5aa7c7a78323b06198) |

### Introduction to Information Levels

The set, query, and selective enumerate patterns of methods use information levels to communicate the set of object attributes that are to be set or queried in the method request. Information levels are enumerations (that is, numerical values).

It is possible that future versions of the protocol will introduce new information levels, creating a situation in which a client can specify an information level that is not supported by the server. This situation can occur, for example, when a later client communicates with an earlier server.[<6>](#Appendix_A_6" \o "Product behavior note 6)

## Vendor-Extensible Fields

None.

## Standards Assignments

None.

# Messages

## Transport

This protocol configures the RPC runtime to perform a strict [**Network Data Representation (NDR)**](#gt_9ebf9540-2c31-43bc-bc56-4a932faabf2d) data consistency check at target level 5.0, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

This protocol uses [**UUID**](#gt_c4813fc3-b2e5-4aa3-bde7-421d950d68d3) 12345778-1234-ABCD-EF00-0123456789AC to identify the RPC interface.

This protocol enables the ms\_union extension that is specified in [MS-RPCE] section 2.2.4.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles that are created by a method of a different RPC interface than this one, as specified in [MS-RPCE] section 3.

This protocol uses the following RPC protocol sequences:[<7>](#Appendix_A_7" \o "Product behavior note 7)

* RPC over SMB, as specified in [MS-RPCE] section 2.1.1.2.[<8>](#Appendix_A_8" \o "Product behavior note 8)

This protocol uses the pipe name "\PIPE\samr" for the endpoint name.[<9>](#Appendix_A_9" \o "Product behavior note 9)

* RPC over TCP.[<10>](#Appendix_A_10" \o "Product behavior note 10)

This protocol uses RPC dynamic endpoints, as specified in [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824) section 6.

This protocol MUST indicate to the RPC runtime that it is to support both the Network Data Representation (NDR) and [**64-bit Network Data Representation (NDR64)**](#gt_29385c51-3799-4da6-8291-32fc46a81970) [**transfer syntaxes**](#gt_01216ea7-ac8a-4cc8-9d19-b901bc424c09) and provide a negotiation mechanism for determining which RPC transfer syntax will be used, as specified in [MS-RPCE] section 3.

This protocol MUST use the UUID as specified previously. The RPC version number is 1.0.

The protocol uses the underlying RPC protocol to retrieve the identity of the client that made the method call, as specified in [MS-RPCE] section 3.3.3.4.3. The server SHOULD use this identity to perform method-specific [**access checks**](#gt_d7906f17-bb2c-4193-a3f0-848bcc351dec), as specified in the message processing section of each method.[<11>](#Appendix_A_11" \o "Product behavior note 11)

RPC clients for this protocol MUST use the authentication level RPC\_C\_AUTHN\_LEVEL\_NONE when invoking RPC over SMB methods.

The server SHOULD[<12>](#Appendix_A_12" \o "Product behavior note 12) reject calls that do not use an authentication level of either RPC\_C\_AUTHN\_LEVEL\_NONE or RPC\_C\_AUTHN\_LEVEL\_PKT\_PRIVACY (see [MS-RPCE] section 2.2.1.1.8).

RPC clients for this protocol MUST use RPC over TCP/IP for the [SamrValidatePassword](#Section_c78a7239f8fc4a42bb71321e897dc046) method and MUST use RPC over SMB for the [SamrSetDSRMPassword](#Section_9bcad7d2b8e14d28a03386040b7b3ce9) method.

RPC clients MUST use only RPC over SMB for the [SamrSetInformationUser](#Section_538222f71b894811949a0eac62e38dce) and [SamrSetInformationUser2](#Section_99ee9f3943e84bbaac3a82e0c0e0699e) methods when UserInformationClass is UserAllInformation, UserInternal1Information, UserInternal4Information, UserInternal4InformationNew, UserInternal5Information, UserInternal5InformationNew, UserInternal7Information, or UserInternal8Information.

For the SamrValidatePassword method, the client SHOULD use transport security to encrypt the message because the message contents contain cleartext password data. That is, the client SHOULD use an SPNEGO security provider, as specified in [MS-RPCE] section 2.2.1.1.7, and SHOULD use the packet authentication level, as specified in [MS-RPCE] section 3.3.1.5.2.[<13>](#Appendix_A_13" \o "Product behavior note 13)

## Common Data Types

In addition to RPC base types and definitions specified in [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824) and [[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2), additional data types are defined in the following subsections.

This protocol MUST indicate to the RPC runtime that it is to support both the [**NDR**](#gt_9ebf9540-2c31-43bc-bc56-4a932faabf2d) and [**NDR64**](#gt_29385c51-3799-4da6-8291-32fc46a81970) transfer syntaxes, and provide a negotiation mechanism for determining which transfer syntax will be used, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

### Constant Value Definitions

This section is used as a reference from one or more message syntax and message processing sections.

#### Common ACCESS\_MASK Values

These values specify an access control that is applicable to all object types exposed by this protocol. These values can appear in the **Mask** field of an [**access control entry (ACE)**](#gt_b581857f-39aa-4979-876b-daba67a40f15) or in methods to obtain a handle (for example, [SamrConnect5](#Section_c842a8970a424ca5a6072afd05271dae)).

| Constant/value | Description |
| --- | --- |
| DELETE  0x00010000 | Specifies the ability to delete the object. |
| READ\_CONTROL  0x00020000 | Specifies the ability to read the [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350). |
| WRITE\_DAC  0x00040000 | Specifies the ability to update the [**discretionary access control list (DACL)**](#gt_d727f612-7a45-48e4-9d87-71735d62b321) of the security descriptor. |
| WRITE\_OWNER  0x00080000 | Specifies the ability to update the **Owner** field of the security descriptor. |
| ACCESS\_SYSTEM\_SECURITY  0x01000000 | Specifies access to the system security portion of the security descriptor. |
| MAXIMUM\_ALLOWED  0x02000000 | Indicates that the caller is requesting the most access possible to the object. |

For more information, see [[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2) section 2.4.3. Values that are not listed have no meaning in this protocol.

#### Generic ACCESS\_MASK Values

These values appear in methods that are used to obtain a handle (for example, [SamrConnect5](#Section_c842a8970a424ca5a6072afd05271dae)). They are translated by the server into specific ACCESS\_MASK values. For more information on object-specific semantics, see sections [2.2.1.3](#Section_e8afb15ec0534984b84b66877236e141), [2.2.1.4](#Section_aef23495f6aa48e9aebc22e022a2b4eb), [2.2.1.5](#Section_f24f9fa8798d4e7da110a5eda6900f41), [2.2.1.6](#Section_2da21c6c5b1546c8bd4e6a8443216e1a), and [2.2.1.7](#Section_c0be3f43bcf943eeb0273d02ab372c53).

| Constant/value | Description |
| --- | --- |
| GENERIC\_READ  0x80000000 | Specifies access control suitable for reading the object. |
| GENERIC\_WRITE  0x40000000 | Specifies access control suitable for updating attributes on the object. |
| GENERIC\_EXECUTE  0x20000000 | Specifies access control suitable for executing an action on the object. |
| GENERIC\_ALL  0x10000000 | Specifies all defined access control on the object. |

#### Server ACCESS\_MASK Values

These are the specific values available to describe the access control on a [**server object**](#gt_62a8c543-5998-480b-8fa7-41a8f04a18e5). A bitwise OR operation can be performed on these values, along with values from section [2.2.1.1](#Section_15b9ebf7161d4c83a672dceb2ac8c448). For more information on the message processing of these values, see section [3.1.5.1.1](#Section_c842a8970a424ca5a6072afd05271dae).

| Constant/value | Description |
| --- | --- |
| SAM\_SERVER\_CONNECT  0x00000001 | Specifies access control to obtain a server handle. |
| SAM\_SERVER\_SHUTDOWN  0x00000002 | Does not specify any access control. |
| SAM\_SERVER\_INITIALIZE  0x00000004 | Does not specify any access control. |
| SAM\_SERVER\_CREATE\_DOMAIN  0x00000008 | Does not specify any access control. |
| SAM\_SERVER\_ENUMERATE\_DOMAINS  0x00000010 | Specifies access control to view domain objects. |
| SAM\_SERVER\_LOOKUP\_DOMAIN  0x00000020 | Specifies access control to perform [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d)-to-name translation. |
| SAM\_SERVER\_ALL\_ACCESS  0x000F003F | The specified accesses for a GENERIC\_ALL request. |
| SAM\_SERVER\_READ  0x00020010 | The specified accesses for a GENERIC\_READ request. |
| SAM\_SERVER\_WRITE  0x0002000E | The specified accesses for a GENERIC\_WRITE request. |
| SAM\_SERVER\_EXECUTE  0x00020021 | The specified accesses for a GENERIC\_EXECUTE request. |

#### Domain ACCESS\_MASK Values

These are the specific values available to describe the access control on a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1). A bitwise OR operation can be performed on these values, along with values from section [2.2.1.1](#Section_15b9ebf7161d4c83a672dceb2ac8c448). For more information on the message processing of these values, see section [3.1.5.1.2](#Section_c1b14e670f4647a69fa70689d284afc9).

| Constant/value | Description |
| --- | --- |
| DOMAIN\_READ\_PASSWORD\_PARAMETERS  0x00000001 | Specifies access control to read password policy. |
| DOMAIN\_WRITE\_PASSWORD\_PARAMS  0x00000002 | Specifies access control to write password policy. |
| DOMAIN\_READ\_OTHER\_PARAMETERS  0x00000004 | Specifies access control to read attributes not related to password policy. |
| DOMAIN\_WRITE\_OTHER\_PARAMETERS  0x00000008 | Specifies access control to write attributes not related to password policy. |
| DOMAIN\_CREATE\_USER  0x00000010 | Specifies access control to create a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17). |
| DOMAIN\_CREATE\_GROUP  0x00000020 | Specifies access control to create a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a). |
| DOMAIN\_CREATE\_ALIAS  0x00000040 | Specifies access control to create an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0). |
| DOMAIN\_GET\_ALIAS\_MEMBERSHIP  0x00000080 | Specifies access control to read the alias membership of a set of [**SIDs**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d). |
| DOMAIN\_LIST\_ACCOUNTS  0x00000100 | Specifies access control to enumerate objects. |
| DOMAIN\_LOOKUP  0x00000200 | Specifies access control to look up objects by name and SID. |
| DOMAIN\_ADMINISTER\_SERVER  0x00000400 | Specifies access control to various administrative operations on the server. |
| DOMAIN\_ALL\_ACCESS  0x000F07FF | The specified accesses for a GENERIC\_ALL request. |
| DOMAIN\_READ  0x00020084 | The specified accesses for a GENERIC\_READ request. |
| DOMAIN\_WRITE  0x0002047A | The specified accesses for a GENERIC\_WRITE request. |
| DOMAIN\_EXECUTE  0x00020301 | The specified accesses for a GENERIC\_EXECUTE request. |

#### Group ACCESS\_MASK Values

These are the specific values available to describe the access control on a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a). A bitwise OR operation can be performed on these values, along with values from section [2.2.1.1](#Section_15b9ebf7161d4c83a672dceb2ac8c448). For more information on the message processing of these values, see section [3.1.5.1.6](#Section_fb25440b269d422099f28b29841cb1ce).

| Constant/value | Description |
| --- | --- |
| GROUP\_READ\_INFORMATION  0x00000001 | Specifies the ability to read various attributes. |
| GROUP\_WRITE\_ACCOUNT  0x00000002 | Specifies the ability to write various attributes, not including the **member** attribute. |
| GROUP\_ADD\_MEMBER  0x00000004 | Specifies the ability to add a value to the **member** attribute. |
| GROUP\_REMOVE\_MEMBER  0x00000008 | Specifies the ability to remove a value from the **member** attribute. |
| GROUP\_LIST\_MEMBERS  0x00000010 | Specifies the ability to read the values of the **member** attribute. |
| GROUP\_ALL\_ACCESS  0x000F001F | The specified accesses for a GENERIC\_ALL request. |
| GROUP\_READ  0x00020010 | The specified accesses for a GENERIC\_READ request. |
| GROUP\_WRITE  0x0002000E | The specified accesses for a GENERIC\_WRITE request. |
| GROUP\_EXECUTE  0x00020001 | The specified accesses for a GENERIC\_EXECUTE request. |

#### Alias ACCESS\_MASK Values

These are the specific values available to describe the access control on an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0). A bitwise OR operation can be performed on these values, along with values from section [2.2.1.1](#Section_15b9ebf7161d4c83a672dceb2ac8c448). For more information on the message processing of these values, see section [3.1.5.1.8](#Section_c3696b7620054eba820d91fdded3422d).

| Constant/value | Description |
| --- | --- |
| ALIAS\_ADD\_MEMBER  0x00000001 | Specifies the ability to add a value to the **member** attribute. |
| ALIAS\_REMOVE\_MEMBER  0x00000002 | Specifies the ability to remove a value from the **member** attribute. |
| ALIAS\_LIST\_MEMBERS  0x00000004 | Specifies the ability to read the **member** attribute. |
| ALIAS\_READ\_INFORMATION  0x00000008 | Specifies the ability to read various attributes, not including the **member** attribute. |
| ALIAS\_WRITE\_ACCOUNT  0x00000010 | Specifies the ability to write various attributes, not including the **member** attribute. |
| ALIAS\_ALL\_ACCESS  0x000F001F | The specified accesses for a GENERIC\_ALL request. |
| ALIAS\_READ  0x00020004 | The specified accesses for a GENERIC\_READ request. |
| ALIAS\_WRITE  0x00020013 | The specified accesses for a GENERIC\_WRITE request. |
| ALIAS\_EXECUTE  0x00020008 | The specified accesses for a GENERIC\_EXECUTE request. |

#### User ACCESS\_MASK Values

These are the specific values available to describe the access control on a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17). A bitwise OR operation can be performed on these values, along with values from section [2.2.1.1](#Section_15b9ebf7161d4c83a672dceb2ac8c448). For more information on the message processing of these values, see section [3.1.5.1.9](#Section_0aee1c31ec404633bb560cf8429093c0).

| Constant/value | Description |
| --- | --- |
| USER\_READ\_GENERAL  0x00000001 | Specifies the ability to read sundry attributes. |
| USER\_READ\_PREFERENCES  0x00000002 | Specifies the ability to read general information attributes. |
| USER\_WRITE\_PREFERENCES  0x00000004 | Specifies the ability to write general information attributes. |
| USER\_READ\_LOGON  0x00000008 | Specifies the ability to read attributes related to logon statistics. |
| USER\_READ\_ACCOUNT  0x00000010 | Specifies the ability to read attributes related to the administration of the user object. |
| USER\_WRITE\_ACCOUNT  0x00000020 | Specifies the ability to write attributes related to the administration of the user object. |
| USER\_CHANGE\_PASSWORD  0x00000040 | Specifies the ability to change the user's password. |
| USER\_FORCE\_PASSWORD\_CHANGE  0x00000080 | Specifies the ability to set the user's password. |
| USER\_LIST\_GROUPS  0x00000100 | Specifies the ability to query the membership of the user object. |
| USER\_READ\_GROUP\_INFORMATION  0x00000200 | Does not specify any access control. |
| USER\_WRITE\_GROUP\_INFORMATION  0x00000400 | Does not specify any access control. |
| USER\_ALL\_ACCESS  0x000F07FF | The specified accesses for a GENERIC\_ALL request. |
| USER\_READ  0x0002031A | The specified accesses for a GENERIC\_READ request. |
| USER\_WRITE  0x00020044 | The specified accesses for a GENERIC\_WRITE request. |
| USER\_EXECUTE  0x00020041 | The specified accesses for a GENERIC\_EXECUTE request. |

#### USER\_ALL Values

USER\_ALL values are used in the WhichFields bit field in the [SAMPR\_USER\_ALL\_INFORMATION](#Section_dc966b81da274daea28cec16534f1cb9) structure. All bits can be combined with a logical OR in any combination that is in accordance with the processing instructions specified in sections [3.1.5.6.5](#Section_538222f71b894811949a0eac62e38dce), [3.1.5.6.4](#Section_99ee9f3943e84bbaac3a82e0c0e0699e), [3.1.5.5.6](#Section_4ad8d54c0d5a4d5a9e9a1bc9ee008d47) and [3.1.5.5.5](#Section_29ab27f661da4c7d863ce228ee798f4d). If a bit is set, the associated field of [SAMPR\_USER\_ALL\_INFORMATION](#Section_dc966b81da274daea28cec16534f1cb9) MUST be processed by the server. If a bit is not set, the server MUST ignore the associated field. The last column of the following table indicates the bit-to-field association.

| Constant/value | Description |
| --- | --- |
| USER\_ALL\_USERNAME  0x00000001 | UserName |
| USER\_ALL\_FULLNAME  0x00000002 | FullName |
| USER\_ALL\_USERID  0x00000004 | UserId |
| USER\_ALL\_PRIMARYGROUPID  0x00000008 | PrimaryGroupId |
| USER\_ALL\_ADMINCOMMENT  0x00000010 | AdminComment |
| USER\_ALL\_USERCOMMENT  0x00000020 | UserComment |
| USER\_ALL\_HOMEDIRECTORY  0x00000040 | HomeDirectory |
| USER\_ALL\_HOMEDIRECTORYDRIVE  0x00000080 | HomeDirectoryDrive |
| USER\_ALL\_SCRIPTPATH  0x00000100 | ScriptPath |
| USER\_ALL\_PROFILEPATH  0x00000200 | ProfilePath |
| USER\_ALL\_WORKSTATIONS  0x00000400 | WorkStations |
| USER\_ALL\_LASTLOGON  0x00000800 | LastLogon |
| USER\_ALL\_LASTLOGOFF  0x00001000 | LastLogoff |
| USER\_ALL\_LOGONHOURS  0x00002000 | LogonHours |
| USER\_ALL\_BADPASSWORDCOUNT  0x00004000 | BadPasswordCount |
| USER\_ALL\_LOGONCOUNT  0x00008000 | LogonCount |
| USER\_ALL\_PASSWORDCANCHANGE  0x00010000 | PasswordCanChange |
| USER\_ALL\_PASSWORDMUSTCHANGE  0x00020000 | PasswordMustChange |
| USER\_ALL\_PASSWORDLASTSET  0x00040000 | PasswordLastSet |
| USER\_ALL\_ACCOUNTEXPIRES  0x00080000 | AccountExpires |
| USER\_ALL\_USERACCOUNTCONTROL  0x00100000 | UserAccountControl |
| USER\_ALL\_PARAMETERS  0x00200000 | Parameters |
| USER\_ALL\_COUNTRYCODE  0x00400000 | CountryCode |
| USER\_ALL\_CODEPAGE  0x00800000 | CodePage |
| USER\_ALL\_NTPASSWORDPRESENT  0x01000000 | NtPasswordPresent |
| USER\_ALL\_LMPASSWORDPRESENT  0x02000000 | LmPasswordPresent |
| USER\_ALL\_PRIVATEDATA  0x04000000 | PrivateData |
| USER\_ALL\_PASSWORDEXPIRED  0x08000000 | PasswordExpired |
| USER\_ALL\_SECURITYDESCRIPTOR  0x10000000 | SecurityDescriptor |
| USER\_ALL\_UNDEFINED\_MASK  0xC0000000 | Undefined mask. |

#### ACCOUNT\_TYPE Values

Account type values are associated with [**accounts**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) and indicate the type of account. These values are not to be combined through logical operations.

| Constant/value | Description |
| --- | --- |
| SAM\_DOMAIN\_OBJECT  0x00000000 | Represents a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1). |
| SAM\_GROUP\_OBJECT  0x10000000 | Represents a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a). |
| SAM\_NON\_SECURITY\_GROUP\_OBJECT  0x10000001 | Represents a group object that is not used for [**authorization context**](#gt_360f5f23-e078-4dee-971b-d5ff6018ea55) generation. |
| SAM\_ALIAS\_OBJECT  0x20000000 | Represents an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0). |
| SAM\_NON\_SECURITY\_ALIAS\_OBJECT  0x20000001 | Represents an alias object that is not used for authorization context generation. |
| SAM\_USER\_OBJECT  0x30000000 | Represents a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17). |
| SAM\_MACHINE\_ACCOUNT  0x30000001 | Represents a computer object. |
| SAM\_TRUST\_ACCOUNT  0x30000002 | Represents a user object that is used for [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) trusts. |
| SAM\_APP\_BASIC\_GROUP  0x40000000 | Represents an application-defined group. |
| SAM\_APP\_QUERY\_GROUP  0x40000001 | Represents an application-defined group whose members are determined by the results of a query. |

#### SE\_GROUP Attributes

These values are attributes of a security group membership and can be combined by using the bitwise OR operation. They are used by an [**access check**](#gt_d7906f17-bb2c-4193-a3f0-848bcc351dec) mechanism to specify whether the membership is to be used in an access check decision. The values can be set by using the [SamrSetMemberAttributesOfGroup](#Section_a4db0c2755404d2fb11108db1ee0b5a4) method.

| Constant/value | Description |
| --- | --- |
| SE\_GROUP\_MANDATORY  0x00000001 | The [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) cannot have the **SE\_GROUP\_ENABLED** attribute removed. |
| SE\_GROUP\_ENABLED\_BY\_DEFAULT  0x00000002 | The SID is enabled by default (rather than being added by an application). |
| SE\_GROUP\_ENABLED  0x00000004 | The SID is enabled for access checks. |

#### GROUP\_TYPE Codes

These values specify the type of a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a). They are used in the **groupType** attribute. The values are mutually exclusive, except for the GROUP\_TYPE\_SECURITY\_ENABLED bit, which can be combined using a logical OR with any other value.

| Constant/value | Description |
| --- | --- |
| GROUP\_TYPE\_ACCOUNT\_GROUP  0x00000002 | Specifies that the group is an [**account group**](#gt_c6113cf2-b171-436b-a199-05e62686b8e9). |
| GROUP\_TYPE\_RESOURCE\_GROUP  0x00000004 | Specifies that the group is a [**resource group**](#gt_04c2dc43-6f4a-432d-bca6-ceeef2d5f2ae). |
| GROUP\_TYPE\_UNIVERSAL\_GROUP  0x00000008 | Specifies that the group is a [**universal group**](#gt_f46053d6-0708-4094-ac63-57c1bcb73d32). |
| GROUP\_TYPE\_SECURITY\_ENABLED  0x80000000 | Specifies that the group's membership is to be included in an [**authorization context**](#gt_360f5f23-e078-4dee-971b-d5ff6018ea55). |
| GROUP\_TYPE\_SECURITY\_ACCOUNT  0x80000002 | A combination of two of the bits shown above for the purposes of this specification. |
| GROUP\_TYPE\_SECURITY\_RESOURCE  0x80000004 | A combination of two of the bits shown above for the purposes of this specification. |
| GROUP\_TYPE\_SECURITY\_UNIVERSAL  0x80000008 | A combination of two of the bits shown above for the purposes of this specification. |

#### USER\_ACCOUNT Codes

These values are attributes of a user account and can be combined by using a bitwise OR operation. They are used in the UserAccountControl field for [**user objects**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17). For more information, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

| Constant/value | Description |
| --- | --- |
| USER\_ACCOUNT\_DISABLED  0x00000001 | Specifies that the account is not enabled for authentication. |
| USER\_HOME\_DIRECTORY\_REQUIRED  0x00000002 | Specifies that the **homeDirectory** attribute is required. |
| USER\_PASSWORD\_NOT\_REQUIRED  0x00000004 | Specifies that the password-length policy does not apply to this user. |
| USER\_TEMP\_DUPLICATE\_ACCOUNT  0x00000008 | This bit is ignored by clients and servers. |
| USER\_NORMAL\_ACCOUNT  0x00000010 | Specifies that the user is not a computer object. |
| USER\_MNS\_LOGON\_ACCOUNT  0x00000020 | This bit is ignored by clients and servers. |
| USER\_INTERDOMAIN\_TRUST\_ACCOUNT  0x00000040 | Specifies that the object represents a trust object. For more information about trust objects, see [[MS-LSAD]](%5bMS-LSAD%5d.pdf#Section_1b5471ef4c334a91b079dfcbb82f05cc). |
| USER\_WORKSTATION\_TRUST\_ACCOUNT  0x00000080 | Specifies that the object is a member workstation or server. |
| USER\_SERVER\_TRUST\_ACCOUNT  0x00000100 | Specifies that the object is a [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd). |
| USER\_DONT\_EXPIRE\_PASSWORD  0x00000200 | Specifies that the maximum-password-age policy does not apply to this user. |
| USER\_ACCOUNT\_AUTO\_LOCKED  0x00000400 | Specifies that the account has been locked out. |
| USER\_ENCRYPTED\_TEXT\_PASSWORD\_ALLOWED  0x00000800 | Specifies that the cleartext password is to be persisted. |
| USER\_SMARTCARD\_REQUIRED  0x00001000 | Specifies that the user can authenticate only with a smart card. |
| USER\_TRUSTED\_FOR\_DELEGATION  0x00002000 | This bit is used by the Kerberos protocol. It indicates that the "OK as Delegate" ticket flag (described in [[RFC4120]](https://go.microsoft.com/fwlink/?LinkId=90458) section 2.8) is to be set. |
| USER\_NOT\_DELEGATED  0x00004000 | This bit is used by the Kerberos protocol. It indicates that the ticket-granting tickets (TGTs) of this account and the service tickets obtained by this account are not marked as forwardable or proxiable when the forwardable or proxiable ticket flags are requested. For more information, see [RFC4120]. |
| USER\_USE\_DES\_KEY\_ONLY  0x00008000 | This bit is used by the Kerberos protocol. It indicates that only des-cbc-md5 or des-cbc-crc keys (as defined in [[RFC3961]](https://go.microsoft.com/fwlink/?LinkId=90450)) are used in the Kerberos protocol for this account. |
| USER\_DONT\_REQUIRE\_PREAUTH  0x00010000 | This bit is used by the Kerberos protocol. It indicates that the account is not required to present valid pre-authentication data, as described in [RFC4120] section 7.5.2. |
| USER\_PASSWORD\_EXPIRED  0x00020000 | Specifies that the password age on the user has exceeded the maximum password age policy. |
| USER\_TRUSTED\_TO\_AUTHENTICATE\_FOR\_DELEGATION  0x00040000 | This bit is used by the Kerberos protocol, as specified in [[MS-KILE]](%5bMS-KILE%5d.pdf#Section_2a32282edd484ad9a542609804b02cc9) section 3.3.1.1. |
| USER\_NO\_AUTH\_DATA\_REQUIRED  0x00080000 | This bit is used by the Kerberos protocol. It indicates that when the key distribution center (KDC) is issuing a service ticket for this account, the privilege attribute certificate (PAC) is not to be included. For more information, see [RFC4120]. |
| USER\_PARTIAL\_SECRETS\_ACCOUNT  0x00100000 | Specifies that the object is a [**read-only domain controller (RODC)**](#gt_8b0a073b-3099-4efe-8b81-c2886b66a870). |
| USER\_USE\_AES\_KEYS  0x00200000 | This bit is ignored by clients and servers. |

#### UF\_FLAG Codes

These values are attributes of a user account, as expressed at the data model level (see section [3.1.1](#Section_814e12f610374b648d288f1b899dc57f) for the data model). Unless otherwise specified in the table, see section [3.1.5.14.2](#Section_8a193181a7a249dfa8b1f689aaa6987c) to map these values to USER\_ACCOUNT values, and then see section [2.2.1.12](#Section_b10cfda1f24f441b8f4380cb93e786ec) for a description.

| Constant/value | Description |
| --- | --- |
| UF\_SCRIPT  0x00000001 | This bit is ignored by clients and servers. |
| UF\_ACCOUNTDISABLE  0x00000002 | See description in introductory paragraph. |
| UF\_HOMEDIR\_REQUIRED  0x00000008 | See description in introductory paragraph. |
| UF\_LOCKOUT  0x00000010 | See description in introductory paragraph. |
| UF\_PASSWD\_NOTREQD  0x00000020 | See description in introductory paragraph. |
| UF\_PASSWD\_CANT\_CHANGE  0x00000040 | This bit is ignored by clients and servers. |
| UF\_ENCRYPTED\_TEXT\_PASSWORD\_ALLOWED  0x00000080 | See description in introductory paragraph. |
| UF\_TEMP\_DUPLICATE\_ACCOUNT  0x00000100 | See description in introductory paragraph. |
| UF\_NORMAL\_ACCOUNT  0x00000200 | See description in introductory paragraph. |
| UF\_INTERDOMAIN\_TRUST\_ACCOUNT  0x00000800 | See description in introductory paragraph. |
| UF\_WORKSTATION\_TRUST\_ACCOUNT  0x00001000 | See description in introductory paragraph. |
| UF\_SERVER\_TRUST\_ACCOUNT  0x00002000 | See description in introductory paragraph. |
| UF\_DONT\_EXPIRE\_PASSWD  0x00010000 | See description in introductory paragraph. |
| UF\_MNS\_LOGON\_ACCOUNT  0x00020000 | See description in introductory paragraph. |
| UF\_SMARTCARD\_REQUIRED  0x00040000 | See description in introductory paragraph. |
| UF\_TRUSTED\_FOR\_DELEGATION  0x00080000 | See description in introductory paragraph. |
| UF\_NOT\_DELEGATED  0x00100000 | See description in introductory paragraph. |
| UF\_USE\_DES\_KEY\_ONLY  0x00200000 | See description in introductory paragraph. |
| UF\_DONT\_REQUIRE\_PREAUTH  0x00400000 | See description in introductory paragraph. |
| UF\_PASSWORD\_EXPIRED  0x00800000 | See description in introductory paragraph. |
| UF\_TRUSTED\_TO\_AUTHENTICATE\_FOR\_DELEGATION  0x01000000 | See description in introductory paragraph. |
| UF\_NO\_AUTH\_DATA\_REQUIRED  0x02000000 | See description in introductory paragraph. |
| UF\_PARTIAL\_SECRETS\_ACCOUNT  0x04000000 | See description in introductory paragraph. |
| UF\_USE\_AES\_KEYS  0x08000000 | See description in introductory paragraph. |

#### Predefined RIDs

These are predefined [**RIDs**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of users and groups. The description column briefly describes what the user or group is used for.

| Constant/value | Description |
| --- | --- |
| DOMAIN\_USER\_RID\_ADMIN  0x000001F4 | Name: Administrator  User for administering the computer or [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca). |
| DOMAIN\_USER\_RID\_GUEST  0x000001F5 | Name: Guest  User for guest access to the computer or domain. |
| DOMAIN\_USER\_RID\_KRBTGT  0x000001F6 | Name: krbtgt  User for Key Distribution Center Service. |
| DOMAIN\_GROUP\_RID\_USERS  0x00000201 | Name: Domain Users  A group that represents all domain users. |
| DOMAIN\_GROUP\_RID\_COMPUTERS  0x00000203 | Name: Domain Computers  A group that represents all workstations and servers joined to the domain. |
| DOMAIN\_GROUP\_RID\_CONTROLLERS  0x00000204 | Name: Domain Controllers  A group that represents all [**DCs**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) in the domain. |
| DOMAIN\_ALIAS\_RID\_ADMINS  0x00000220 | Name: Administrators  A group that has complete and unrestricted access to the computer or domain. |
| DOMAIN\_GROUP\_RID\_READONLY\_CONTROLLERS  0x00000209 | Name: Read-only Domain Controllers  A group that represents all [**RODCs**](#gt_8b0a073b-3099-4efe-8b81-c2886b66a870) in the domain. |

#### STATUS\_ Codes

These values are return status codes from the server. This section is provided as a reference for the message processing sections in section [3.1.5](#Section_3f15814e46004647abfd3890f5f3570c).

| Constant/value | Description |
| --- | --- |
| STATUS\_ACCESS\_DENIED  0xC0000022 | Returned when a client has requested access to an object but has not been granted those access rights. |
| STATUS\_MORE\_ENTRIES  0x00000105 | Returned by enumeration methods to indicate that more information is available. |
| STATUS\_NO\_MORE\_ENTRIES  0x8000001A | Returned by enumeration methods to indicate that no more information is available. |
| STATUS\_SOME\_NOT\_MAPPED  0x00000107 | Returned when some of the information to be translated has not been translated. |
| STATUS\_NONE\_MAPPED  0xC0000073 | Returned when none of the information to be translated has been translated. |
| STATUS\_WRONG\_PASSWORD  0xC000006A | Returned when trying to update a password and the value provided as the current password is not correct. |
| STATUS\_ACCOUNT\_LOCKED\_OUT  0xC0000234 | Returned when the user account has been automatically locked because too many invalid logon attempts or password change attempts have been requested. |
| STATUS\_GROUP\_EXISTS  0xC0000065 | Returned when the specified group already exists. |
| STATUS\_USER\_EXISTS  0xC0000063 | Returned when the specified account already exists. |
| STATUS\_LM\_CROSS\_ENCRYPTION\_REQUIRED  0xC000017F | Returned when the client is to retry the request using the current password [**LM hash**](#gt_58a72159-b1e9-4770-9b06-d7a6b16279a8) as an encryption key. See section [3.1.5.10.1](#Section_9699d8cae1a4433ca8c3d7bebeb01476) for details. |
| STATUS\_NT\_CROSS\_ENCRYPTION\_REQUIRED  0xC000015D | Returned when the client is to retry the request using the current password [**NT hash**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f) as an encryption key. See section 3.1.5.10.1 for details. |

#### Transport Error Code

| Constant/value | Description |
| --- | --- |
| RPC\_S\_PROCNUM\_OUT\_OF\_RANGE  0x6D1 | The server does not implement the requested method. |

#### AD ACCESS\_MASK

These [**access mask**](#gt_462f2aa7-2cfc-404a-b479-30f127c512b1) values are specific to [**ACEs**](#gt_b581857f-39aa-4979-876b-daba67a40f15) that apply to [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) objects. More information about these values is specified in [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 5.1.3.

| Constant/value | Description |
| --- | --- |
| ACTRL\_DS\_LIST  0x00000004 | Indicates the ability to read the children of an object in Active Directory. |
| ACTRL\_DS\_READ\_PROP  0x00000010 | Indicates the access control to read a property in Active Directory. |
| ACTRL\_DS\_WRITE\_PROP  0x00000020 | Indicates the access control to write a property in Active Directory. |
| ACTRL\_DS\_DELETE\_TREE  0x00000040 | Indicates the ability to delete a tree of objects. |
| ACTRL\_DS\_CONTROL\_ACCESS  0x00000100 | Indicates the ability to perform an operation on an object as indicated by the **ObjectGuid** field in the ACE. |

#### AEAD-AES-256-CBC-HMAC-SHA512 Constants

The following constants are used for wire encryption of sensitive data with the AEAD-AES-256-CBC-HMAC-SHA512 cipher, as specified in [[AES-CBC]](https://go.microsoft.com/fwlink/?linkid=2162731) and in section [3.2.2.4](#Section_0fc25352662b40e79deef10c6244c4c9).

| Constant Name | Value |
| --- | --- |
| versionbyte | 0x01 |
| versionbyte\_length | 1 |
| SAM\_AES\_256\_ALG | "AEAD-AES-256-CBC-HMAC-SHA512" |
| SAM\_AES256\_ENC\_KEY\_STRING | "Microsoft SAM encryption key AEAD-AES-256-CBC-HMAC-SHA512 16" |
| SAM\_AES256\_MAC\_KEY\_STRING | "Microsoft SAM MAC key AEAD-AES-256-CBC-HMAC-SHA512 16" |
| SAM\_AES256\_ENC\_KEY\_STRING\_LENGTH | sizeof(SAM\_AES256\_ENC\_KEY\_STRING) |
| SAM\_AES256\_MAC\_KEY\_STRING\_LENGTH | sizeof(SAM\_AES256\_MAC\_KEY\_STRING) |

### Basic Data Types

The following basic types are elementary to the SAM Remote Protocol (Client-to-Server) and are used in many methods. These types also appear in other protocols.

#### RPC\_STRING, PRPC\_STRING

The RPC\_STRING structure holds a counted string encoded in the [**OEM code page**](#gt_442ab13f-d2c1-4128-b1db-f3bea4b8224e).

1. typedef struct \_RPC\_STRING {
2. unsigned short Length;
3. unsigned short MaximumLength;
4. [size\_is(MaximumLength), length\_is(Length)]
5. char\* Buffer;
6. } RPC\_STRING,
7. \*PRPC\_STRING;

**Length:**  The size, in bytes, not including a terminating null character, of the string contained in **Buffer**.

**MaximumLength:**  The size, in bytes, of the **Buffer** member.

**Buffer:**  A buffer containing a string encoded in the OEM code page. The string is counted (by the **Length** member), and therefore is not null-terminated.

#### OLD\_LARGE\_INTEGER

The OLD\_LARGE\_INTEGER structure defines a 64-bit value that is accessible in two 4-byte chunks.

1. typedef struct \_OLD\_LARGE\_INTEGER {
2. unsigned long LowPart;
3. long HighPart;
4. } OLD\_LARGE\_INTEGER,
5. \*POLD\_LARGE\_INTEGER;

**LowPart:**  The least-significant portion of a 64-bit value.

**HighPart:**  The most-significant portion of a 64-bit value.

#### SID\_NAME\_USE

The SID\_NAME\_USE enumeration specifies the type of [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) that a [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) references.

1. typedef enum \_SID\_NAME\_USE
2. {
3. SidTypeUser = 1,
4. SidTypeGroup,
5. SidTypeDomain,
6. SidTypeAlias,
7. SidTypeWellKnownGroup,
8. SidTypeDeletedAccount,
9. SidTypeInvalid,
10. SidTypeUnknown,
11. SidTypeComputer,
12. SidTypeLabel
13. } SID\_NAME\_USE,
14. \*PSID\_NAME\_USE;

**SidTypeUser:** Indicates a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).

**SidTypeGroup:** Indicates a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a).

**SidTypeDomain:** Indicates a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**SidTypeAlias:** Indicates an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0).

**SidTypeWellKnownGroup:** Indicates an object whose SID is invariant.

**SidTypeDeletedAccount:** Indicates an object that has been deleted.

**SidTypeInvalid:** This member is not used.

**SidTypeUnknown:** Indicates that the type of object could not be determined. For example, no object with that SID exists.

**SidTypeComputer:** This member is not used.

**SidTypeLabel:** This member is not used.

#### RPC\_SHORT\_BLOB

The RPC\_SHORT\_BLOB structure holds a counted array of unsigned short values.

1. typedef struct \_RPC\_SHORT\_BLOB {
2. unsigned short Length;
3. unsigned short MaximumLength;
4. [size\_is(MaximumLength/2), length\_is(Length/2)]
5. unsigned short\* Buffer;
6. } RPC\_SHORT\_BLOB,
7. \*PRPC\_SHORT\_BLOB;

**Length:**  The number of bytes of data contained in the **Buffer** member.

**MaximumLength:**  The length, in bytes, of the **Buffer** member.

**Buffer:**  A buffer containing **Length**/2 unsigned short values.

### Domain Query/Set Data Types

The structures in this section relate to the following methods:

* [SamrQueryInformationDomain](#Section_5d6a2817caa941caa269fd13ecbb4fa8)
* [SamrQueryInformationDomain2](#Section_fbdbbd7401064ca2941e36ded0be0c62)
* [SamrSetInformationDomain](#Section_9b7ae0b0bd1141339c62fba7095aee12)

The model of the methods is for the client to specify an enumeration that indicates the attributes to be either set or queried. There is duplication among the structures that contain the attributes. For a description of each attribute that is common among structures, see section [2.2.3.1](#Section_d275ab1910b040e094bb45b7fc130025).

#### Domain Fields

There are a number of domain-related structures that use the same fields, as denoted by their field names. This section specifies all such fields. The structures group the available set of [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) attributes in different ways to allow the client to control which attributes are queried or set. Although each structure can have a different subset of these attributes, they all draw from this same set of attributes, detailed as follows.

**AliasCount**: A 32-bit unsigned integer indicating the number of [**alias objects**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0) in the domain. This field is read-only.

**CreationTime**: A 64-bit time stamp, equivalent to a FILETIME, indicating the time of creation for the domain in 100-nanosecond intervals from 12:00 A.M., January 1, 1601 (UTC). This field is read-only.

**DomainModifiedCount**: A 64-bit update sequence number representing the number of database updates relevant to the Windows NT 4.0 operating system replication protocol. This field is read-only. On the server, the value to return for this field corresponds to the SamNT4ReplicationUSN and BuiltinNT4ReplicationUSN values specified in [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 3.1.1.7.1.1.

**DomainName**: A counted Unicode string of type RPC\_UNICODE\_STRING, containing the NetBIOS name of the domain. This field is read-only.

**DomainServerRole**: An enumerated value (see [DOMAIN\_SERVER\_ROLE](#Section_abb07f2eb9e648ec91187caeb03cca35)) indicating the role of the server in the domain. Possible values are Primary Domain Controller (DomainServerRolePrimary) or Backup Domain Controller (DomainServerRoleBackup).

**DomainServerState**: An enumerated value (see [DOMAIN\_SERVER\_ENABLE\_STATE](#Section_5b1a45c9937e4129bec4593f64c56d48)) indicating whether the server is enabled. Possible values are enabled (DomainServerEnabled) or disabled (DomainServerDisabled). This field SHOULD be set to DomainServerEnabled and implementations SHOULD ignore any input to this field.

**ForceLogoff**: A 64-bit value, with [**delta time**](#gt_8a861b7f-5224-44ab-8de7-18ab8f3e800d) syntax, indicating the policy setting for the amount of time that an interactive logon session is allowed to continue.

**GroupCount**: A 32-bit unsigned integer indicating the number of group accounts. This field is read-only.

**LockoutDuration**: A 64-bit value, with delta time syntax, indicating the duration for which an [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) is locked out before being automatically reset to an unlocked state.

**LockoutObservationWindow**: A 64-bit value, with delta time syntax, indicating the time period in which failed password attempts are counted without resetting the count to zero.

**LockoutThreshold**: A 16-bit unsigned integer indicating the number of bad password attempts within a LockoutObservationWindow that will cause an account to be locked out.

**MaxPasswordAge**: A 64-bit value, with delta time syntax, indicating the policy setting for the maximum time allowed before a password reset or change is required.

**MinPasswordAge**: A 64-bit value, with delta time syntax, indicating the policy setting for the minimum time allowed before a password change operation is allowed.

**MinPasswordLength**: A 16-bit unsigned integer indicating the minimum password length policy setting.

**ModifiedCountAtLastPromotion**: A 64-bit update sequence number representing the number of database updates relevant to the Windows NT 4.0 replication protocol that had occurred at the time when the current server obtained the [**PDC**](#gt_663cb13a-8b75-477f-b6e1-bea8f2fba64d) role (see [MS-ADTS] section 6.1.5.4 for more information on the PDC role). This field is read-only.

**OemInformation**: A counted Unicode string of type RPC\_UNICODE\_STRING that clients can set to any value. There are no known scenarios that use this field.

**PasswordHistoryLength**: A 16-bit unsigned integer indicating the policy setting for the password history length.

**PasswordProperties**: A 32-bit bit field indicating the password properties policy setting. The defined bits are shown in the following table. All bits can be combined using a logical OR in any combination. Undefined bits SHOULD be persisted by the server (that is, stored in its database) and returned to future queries. Clients SHOULD ignore undefined bits.

| Name/value | Description |
| --- | --- |
| DOMAIN\_PASSWORD\_COMPLEX  0x00000001 | The server enforces password complexity policy. See section [3.1.1.7.2](#Section_cb1f380a0a51470584cdeea1b85953e0) for details of the password policy. |
| DOMAIN\_PASSWORD\_NO\_ANON\_CHANGE  0x00000002 | Reserved. No effect on password policy. |
| DOMAIN\_PASSWORD\_NO\_CLEAR\_CHANGE  0x00000004 | Change-password methods that provide the cleartext password are disabled by the server. |
| DOMAIN\_LOCKOUT\_ADMINS  0x00000008 | Reserved. No effect on password policy. |
| DOMAIN\_PASSWORD\_STORE\_CLEARTEXT  0x00000010 | The server MUST store the cleartext password, not just the computed hashes. |
| DOMAIN\_REFUSE\_PASSWORD\_CHANGE  0x00000020 | Reserved. No effect on password policy. |

**ReplicaSourceNodeName**: A counted Unicode string of type RPC\_UNICODE\_STRING that contains the NetBIOS name of the primary domain controller (PDC) at the time of upgrade from Windows NT 4.0. The default value is the empty string.

**UasCompatibilityRequired**: A 1-byte value that, if nonzero, indicates that [**UAS Compatibility**](#gt_795ab0c8-e256-4394-bfd4-df445cb48c72) mode is effective; if zero, UAS Compatibility mode is not effective. This field is read-only and the default value is nonzero.

**UserCount**: A 32-bit unsigned integer indicating the number of user accounts. This field is read-only.

#### DOMAIN\_SERVER\_ENABLE\_STATE

The DOMAIN\_SERVER\_ENABLE\_STATE enumeration describes the enabled or disabled state of a server.

1. typedef enum \_DOMAIN\_SERVER\_ENABLE\_STATE
2. {
3. DomainServerEnabled = 1,
4. DomainServerDisabled
5. } DOMAIN\_SERVER\_ENABLE\_STATE,
6. \*PDOMAIN\_SERVER\_ENABLE\_STATE;

**DomainServerEnabled:** The server is considered "enabled" to the client.

**DomainServerDisabled:** This field is not used.

#### DOMAIN\_STATE\_INFORMATION

The DOMAIN\_STATE\_INFORMATION structure holds the enabled/disabled state of the server.

1. typedef struct \_DOMAIN\_STATE\_INFORMATION {
2. DOMAIN\_SERVER\_ENABLE\_STATE DomainServerState;
3. } DOMAIN\_STATE\_INFORMATION,
4. \*PDOMAIN\_STATE\_INFORMATION;

For information on each field, see section [2.2.3.1](#Section_d275ab1910b040e094bb45b7fc130025).

#### DOMAIN\_SERVER\_ROLE

The DOMAIN\_SERVER\_ROLE enumeration indicates whether a server is a [**PDC**](#gt_663cb13a-8b75-477f-b6e1-bea8f2fba64d).

1. typedef enum \_DOMAIN\_SERVER\_ROLE
2. {
3. DomainServerRoleBackup = 2,
4. DomainServerRolePrimary = 3
5. } DOMAIN\_SERVER\_ROLE,
6. \*PDOMAIN\_SERVER\_ROLE;

**DomainServerRoleBackup:** The [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) is not the PDC.

**DomainServerRolePrimary:** The DC is the PDC.

#### DOMAIN\_PASSWORD\_INFORMATION

The DOMAIN\_PASSWORD\_INFORMATION structure contains [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) fields.

1. typedef struct \_DOMAIN\_PASSWORD\_INFORMATION {
2. unsigned short MinPasswordLength;
3. unsigned short PasswordHistoryLength;
4. unsigned long PasswordProperties;
5. OLD\_LARGE\_INTEGER MaxPasswordAge;
6. OLD\_LARGE\_INTEGER MinPasswordAge;
7. } DOMAIN\_PASSWORD\_INFORMATION,
8. \*PDOMAIN\_PASSWORD\_INFORMATION;

For information on each field, see section [2.2.3.1](#Section_d275ab1910b040e094bb45b7fc130025).

#### DOMAIN\_LOGOFF\_INFORMATION

The DOMAIN\_LOGOFF\_INFORMATION structure contains [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) fields.

1. typedef struct \_DOMAIN\_LOGOFF\_INFORMATION {
2. OLD\_LARGE\_INTEGER ForceLogoff;
3. } DOMAIN\_LOGOFF\_INFORMATION,
4. \*PDOMAIN\_LOGOFF\_INFORMATION;

For information on each field, see section [2.2.3.1](#Section_d275ab1910b040e094bb45b7fc130025).

#### DOMAIN\_SERVER\_ROLE\_INFORMATION

The DOMAIN\_SERVER\_ROLE\_INFORMATION structure contains [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) fields.

1. typedef struct \_DOMAIN\_SERVER\_ROLE\_INFORMATION {
2. DOMAIN\_SERVER\_ROLE DomainServerRole;
3. } DOMAIN\_SERVER\_ROLE\_INFORMATION,
4. \*PDOMAIN\_SERVER\_ROLE\_INFORMATION;

For information on each field, see section [2.2.3.1](#Section_d275ab1910b040e094bb45b7fc130025).

#### DOMAIN\_MODIFIED\_INFORMATION

The DOMAIN\_MODIFIED\_INFORMATION structure contains [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) fields.

1. typedef struct \_DOMAIN\_MODIFIED\_INFORMATION {
2. OLD\_LARGE\_INTEGER DomainModifiedCount;
3. OLD\_LARGE\_INTEGER CreationTime;
4. } DOMAIN\_MODIFIED\_INFORMATION,
5. \*PDOMAIN\_MODIFIED\_INFORMATION;

For information on each field, see section [2.2.3.1](#Section_d275ab1910b040e094bb45b7fc130025).

#### DOMAIN\_MODIFIED\_INFORMATION2

The DOMAIN\_MODIFIED\_INFORMATION2 structure contains [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) fields.

1. typedef struct \_DOMAIN\_MODIFIED\_INFORMATION2 {
2. OLD\_LARGE\_INTEGER DomainModifiedCount;
3. OLD\_LARGE\_INTEGER CreationTime;
4. OLD\_LARGE\_INTEGER ModifiedCountAtLastPromotion;
5. } DOMAIN\_MODIFIED\_INFORMATION2,
6. \*PDOMAIN\_MODIFIED\_INFORMATION2;

For information on each field, see section [2.2.3.1](#Section_d275ab1910b040e094bb45b7fc130025).

#### SAMPR\_DOMAIN\_GENERAL\_INFORMATION

The SAMPR\_DOMAIN\_GENERAL\_INFORMATION structure contains [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) fields.

1. typedef struct \_SAMPR\_DOMAIN\_GENERAL\_INFORMATION {
2. OLD\_LARGE\_INTEGER ForceLogoff;
3. RPC\_UNICODE\_STRING OemInformation;
4. RPC\_UNICODE\_STRING DomainName;
5. RPC\_UNICODE\_STRING ReplicaSourceNodeName;
6. OLD\_LARGE\_INTEGER DomainModifiedCount;
7. unsigned long DomainServerState;
8. unsigned long DomainServerRole;
9. unsigned char UasCompatibilityRequired;
10. unsigned long UserCount;
11. unsigned long GroupCount;
12. unsigned long AliasCount;
13. } SAMPR\_DOMAIN\_GENERAL\_INFORMATION,
14. \*PSAMPR\_DOMAIN\_GENERAL\_INFORMATION;

For information on each field, see section [2.2.3.1](#Section_d275ab1910b040e094bb45b7fc130025).

**Note**  In section 2.2.3.1, the types for the **DomainServerState** and **DomainServerRole** members are the [DOMAIN\_SERVER\_ENABLE\_STATE](#Section_5b1a45c9937e4129bec4593f64c56d48) and [DOMAIN\_SERVER\_ROLE](#Section_abb07f2eb9e648ec91187caeb03cca35) enumerations, respectively. These fields have the same purpose as the enumeration values, but the data types are different. The following tables show the corresponding mappings.

For **DomainServerState**:

| Enumeration DOMAIN\_SERVER\_ENABLE\_STATE value | unsigned long value |
| --- | --- |
| DomainServerEnabled | 1 |
| DomainServerDisabled | 2 |

For **DomainServerRole**:

| Enumeration DOMAIN\_SERVER\_ROLE value | unsigned long value |
| --- | --- |
| DomainServerRoleBackup | 2 |
| DomainServerRolePrimary | 3 |

#### SAMPR\_DOMAIN\_GENERAL\_INFORMATION2

The SAMPR\_DOMAIN\_GENERAL\_INFORMATION2 structure contains [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) fields.

1. typedef struct \_SAMPR\_DOMAIN\_GENERAL\_INFORMATION2 {
2. SAMPR\_DOMAIN\_GENERAL\_INFORMATION I1;
3. LARGE\_INTEGER LockoutDuration;
4. LARGE\_INTEGER LockoutObservationWindow;
5. unsigned short LockoutThreshold;
6. } SAMPR\_DOMAIN\_GENERAL\_INFORMATION2,
7. \*PSAMPR\_DOMAIN\_GENERAL\_INFORMATION2;

For information on each field, see section [2.2.3.1](#Section_d275ab1910b040e094bb45b7fc130025), except for **I1**, which is specified in section [2.2.3.10](#Section_85973e1c96f24c808135b24d74ad7794).

#### SAMPR\_DOMAIN\_OEM\_INFORMATION

The SAMPR\_DOMAIN\_OEM\_INFORMATION structure contains [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) fields.

1. typedef struct \_SAMPR\_DOMAIN\_OEM\_INFORMATION {
2. RPC\_UNICODE\_STRING OemInformation;
3. } SAMPR\_DOMAIN\_OEM\_INFORMATION,
4. \*PSAMPR\_DOMAIN\_OEM\_INFORMATION;

For information on each field, see section [2.2.3.1](#Section_d275ab1910b040e094bb45b7fc130025).

#### SAMPR\_DOMAIN\_NAME\_INFORMATION

The SAMPR\_DOMAIN\_NAME\_INFORMATION structure contains [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) fields.

1. typedef struct \_SAMPR\_DOMAIN\_NAME\_INFORMATION {
2. RPC\_UNICODE\_STRING DomainName;
3. } SAMPR\_DOMAIN\_NAME\_INFORMATION,
4. \*PSAMPR\_DOMAIN\_NAME\_INFORMATION;

For information on each field, see section [2.2.3.1](#Section_d275ab1910b040e094bb45b7fc130025).

#### SAMPR\_DOMAIN\_REPLICATION\_INFORMATION

The SAMPR\_DOMAIN\_REPLICATION\_INFORMATION structure contains [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) fields.

1. typedef struct SAMPR\_DOMAIN\_REPLICATION\_INFORMATION {
2. RPC\_UNICODE\_STRING ReplicaSourceNodeName;
3. } SAMPR\_DOMAIN\_REPLICATION\_INFORMATION,
4. \*PSAMPR\_DOMAIN\_REPLICATION\_INFORMATION;

For information on each field, see section [2.2.3.1](#Section_d275ab1910b040e094bb45b7fc130025).

#### SAMPR\_DOMAIN\_LOCKOUT\_INFORMATION

The SAMPR\_DOMAIN\_LOCKOUT\_INFORMATION structure contains [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) fields.

1. typedef struct \_SAMPR\_DOMAIN\_LOCKOUT\_INFORMATION {
2. LARGE\_INTEGER LockoutDuration;
3. LARGE\_INTEGER LockoutObservationWindow;
4. unsigned short LockoutThreshold;
5. } SAMPR\_DOMAIN\_LOCKOUT\_INFORMATION,
6. \*PSAMPR\_DOMAIN\_LOCKOUT\_INFORMATION;

For information on each field, see section [2.2.3.1](#Section_d275ab1910b040e094bb45b7fc130025).

#### DOMAIN\_INFORMATION\_CLASS

The DOMAIN\_INFORMATION\_CLASS enumeration indicates how to interpret the Buffer parameter for [SamrSetInformationDomain](#Section_9b7ae0b0bd1141339c62fba7095aee12) and [SamrQueryInformationDomain](#Section_5d6a2817caa941caa269fd13ecbb4fa8). For a list of associated structures, see section [2.2.3.17](#Section_1adc2142dbb84554aa24010c713698bf).

1. typedef enum \_DOMAIN\_INFORMATION\_CLASS
2. {
3. DomainPasswordInformation = 1,
4. DomainGeneralInformation = 2,
5. DomainLogoffInformation = 3,
6. DomainOemInformation = 4,
7. DomainNameInformation = 5,
8. DomainReplicationInformation = 6,
9. DomainServerRoleInformation = 7,
10. DomainModifiedInformation = 8,
11. DomainStateInformation = 9,
12. DomainGeneralInformation2 = 11,
13. DomainLockoutInformation = 12,
14. DomainModifiedInformation2 = 13
15. } DOMAIN\_INFORMATION\_CLASS;

**DomainPasswordInformation:** Indicates the *Buffer* parameter is to be interpreted as a **DOMAIN\_PASSWORD\_INFORMATION** structure (see section [2.2.3.5](#Section_0ae356d8c2204706846eebbdc6fabdcb)).

**DomainGeneralInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_DOMAIN\_GENERAL\_INFORMATION** structure (see section [2.2.3.10](#Section_85973e1c96f24c808135b24d74ad7794)).

**DomainLogoffInformation:** Indicates the *Buffer* parameter is to be interpreted as a **DOMAIN\_LOGOFF\_INFORMATION** structure (see section [2.2.3.6](#Section_6fb0bbea888c4353b5f875e7862344be)).

**DomainOemInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_DOMAIN\_OEM\_INFORMATION** structure (see section [2.2.3.12](#Section_7cbb7ff0e593440d8341a3435195cdf1)).

**DomainNameInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_DOMAIN\_NAME\_INFORMATION** structure (see section [2.2.3.13](#Section_5131d2c004c74c1b8fd50b0b6cfa6c24)).

**DomainReplicationInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_DOMAIN\_REPLICATION\_INFORMATION** structure (see section [2.2.3.14](#Section_c9293797e11d4098be12bf9e1de91f20)).

**DomainServerRoleInformation:** Indicates the *Buffer* parameter is to be interpreted as a **DOMAIN\_SERVER\_ROLE\_INFORMATION** structure (see section [2.2.3.7](#Section_cb0e586a29c849b28cedc273a7476c22)).

**DomainModifiedInformation:** Indicates the *Buffer* parameter is to be interpreted as a **DOMAIN\_MODIFIED\_INFORMATION** structure (see section [2.2.3.8](#Section_e1da96808968423b98c0fbdcf1535ef9)).

**DomainStateInformation:** Indicates the *Buffer* parameter is to be interpreted as a **DOMAIN\_STATE\_INFORMATION** structure (see section [2.2.3.3](#Section_f224edcf8d4e4294b0c3b0eda384c402)).

**DomainGeneralInformation2:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_DOMAIN\_GENERAL\_INFORMATION2** structure (see section [2.2.3.11](#Section_9a663cf209234959b2c52e25c19735ff)).

**DomainLockoutInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_DOMAIN\_LOCKOUT\_INFORMATION** structure (see section [2.2.3.15](#Section_c9d789edc54a4450be56251e627e1f52)).

**DomainModifiedInformation2:** Indicates the *Buffer* parameter is to be interpreted as a **DOMAIN\_MODIFIED\_INFORMATION2** structure (see section [2.2.3.9](#Section_47eea81b5fee4925b5c1fc594dcc8dff)).

#### SAMPR\_DOMAIN\_INFO\_BUFFER

The SAMPR\_DOMAIN\_INFO\_BUFFER union combines all possible structures used in the [SamrSetInformationDomain](#Section_9b7ae0b0bd1141339c62fba7095aee12) and [SamrQueryInformationDomain](#Section_5d6a2817caa941caa269fd13ecbb4fa8) methods. For details on each field, see the associated section for each field structure.

1. typedef
2. [switch\_type(DOMAIN\_INFORMATION\_CLASS)]
3. union \_SAMPR\_DOMAIN\_INFO\_BUFFER {
4. [case(DomainPasswordInformation)]
5. DOMAIN\_PASSWORD\_INFORMATION Password;
6. [case(DomainGeneralInformation)]
7. SAMPR\_DOMAIN\_GENERAL\_INFORMATION General;
8. [case(DomainLogoffInformation)]
9. DOMAIN\_LOGOFF\_INFORMATION Logoff;
10. [case(DomainOemInformation)]
11. SAMPR\_DOMAIN\_OEM\_INFORMATION Oem;
12. [case(DomainNameInformation)]
13. SAMPR\_DOMAIN\_NAME\_INFORMATION Name;
14. [case(DomainServerRoleInformation)]
15. DOMAIN\_SERVER\_ROLE\_INFORMATION Role;
16. [case(DomainReplicationInformation)]
17. SAMPR\_DOMAIN\_REPLICATION\_INFORMATION Replication;
18. [case(DomainModifiedInformation)]
19. DOMAIN\_MODIFIED\_INFORMATION Modified;
20. [case(DomainStateInformation)]
21. DOMAIN\_STATE\_INFORMATION State;
22. [case(DomainGeneralInformation2)]
23. SAMPR\_DOMAIN\_GENERAL\_INFORMATION2 General2;
24. [case(DomainLockoutInformation)]
25. SAMPR\_DOMAIN\_LOCKOUT\_INFORMATION Lockout;
26. [case(DomainModifiedInformation2)]
27. DOMAIN\_MODIFIED\_INFORMATION2 Modified2;
28. } SAMPR\_DOMAIN\_INFO\_BUFFER,
29. \*PSAMPR\_DOMAIN\_INFO\_BUFFER;

### Group Query/Set Data Types

The structures and fields in this section relate to the following methods:

* [SamrQueryInformationGroup](#Section_8fd0003f807640ae872f9cdb6576b3c7)
* [SamrSetInformationGroup](#Section_e66db19f600a481bbc4e23953433255d)

The model of the methods is for the client to specify an enumeration that indicates the attributes to be either set or queried. There is duplication among the structures that contain the attributes. For a description of each attribute that is common among structures, see section [2.2.4.1](#Section_2c93edfeae774f49b849b6435ff63444).

#### Common Group Fields

There are a number of group-related structures that use the same fields, as denoted by their field names. This section specifies all such fields.

The structures group the available set of group attributes in different ways to allow the client to control which attributes are queried or set. While each structure might have a different subset of these attributes, they all draw from this same set of attributes, detailed as follows.

**AdminComment**: A counted Unicode string of type RPC\_UNICODE\_STRING, indicating the description of the [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a).

**Attributes**: A 32-bit bit field containing characteristics about a group; for possible values, see section [2.2.1.10](#Section_9e093bd2e4514dd5970097b977d7ebb2).

**MemberCount**: A 32-bit unsigned integer indicating the number of members in the group object. This field is read-only.

**Name**: A counted Unicode string of type RPC\_UNICODE\_STRING, indicating the name of the group object.

#### GROUP\_ATTRIBUTE\_INFORMATION

The GROUP\_ATTRIBUTE\_INFORMATION structure contains group fields.

1. typedef struct \_GROUP\_ATTRIBUTE\_INFORMATION {
2. unsigned long Attributes;
3. } GROUP\_ATTRIBUTE\_INFORMATION,
4. \*PGROUP\_ATTRIBUTE\_INFORMATION;

For information on each field, see section [2.2.4.1](#Section_2c93edfeae774f49b849b6435ff63444).

#### SAMPR\_GROUP\_GENERAL\_INFORMATION

The SAMPR\_GROUP\_GENERAL\_INFORMATION structure contains group fields.

1. typedef struct \_SAMPR\_GROUP\_GENERAL\_INFORMATION {
2. RPC\_UNICODE\_STRING Name;
3. unsigned long Attributes;
4. unsigned long MemberCount;
5. RPC\_UNICODE\_STRING AdminComment;
6. } SAMPR\_GROUP\_GENERAL\_INFORMATION,
7. \*PSAMPR\_GROUP\_GENERAL\_INFORMATION;

For information on each field, see section [2.2.4.1](#Section_2c93edfeae774f49b849b6435ff63444).

#### SAMPR\_GROUP\_NAME\_INFORMATION

The SAMPR\_GROUP\_NAME\_INFORMATION structure contains group fields.

1. typedef struct \_SAMPR\_GROUP\_NAME\_INFORMATION {
2. RPC\_UNICODE\_STRING Name;
3. } SAMPR\_GROUP\_NAME\_INFORMATION,
4. \*PSAMPR\_GROUP\_NAME\_INFORMATION;

For information on each field, see section [2.2.4.1](#Section_2c93edfeae774f49b849b6435ff63444).

#### SAMPR\_GROUP\_ADM\_COMMENT\_INFORMATION

The SAMPR\_GROUP\_ADM\_COMMENT\_INFORMATION structure contains group fields.

1. typedef struct \_SAMPR\_GROUP\_ADM\_COMMENT\_INFORMATION {
2. RPC\_UNICODE\_STRING AdminComment;
3. } SAMPR\_GROUP\_ADM\_COMMENT\_INFORMATION,
4. \*PSAMPR\_GROUP\_ADM\_COMMENT\_INFORMATION;

For information on each field, see section [2.2.4.1](#Section_2c93edfeae774f49b849b6435ff63444).

#### GROUP\_INFORMATION\_CLASS

The GROUP\_INFORMATION\_CLASS enumeration indicates how to interpret the *Buffer* parameter for [SamrSetInformationGroup](#Section_e66db19f600a481bbc4e23953433255d) and [SamrQueryInformationGroup](#Section_8fd0003f807640ae872f9cdb6576b3c7). For a list of associated structures, see section [2.2.4.7](#Section_a37ef7399b2147a789b4d237dd6491f3).

1. typedef enum \_GROUP\_INFORMATION\_CLASS
2. {
3. GroupGeneralInformation = 1,
4. GroupNameInformation,
5. GroupAttributeInformation,
6. GroupAdminCommentInformation,
7. GroupReplicationInformation
8. } GROUP\_INFORMATION\_CLASS;

**GroupGeneralInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_GROUP\_GENERAL\_INFORMATION** structure (see section [2.2.4.3](#Section_fe821071d3334eef862af45408343e4e)).

**GroupNameInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_GROUP\_NAME\_INFORMATION** structure (see section [2.2.4.4](#Section_75b2f39b46fd44d980bda78123dddbdb)).

**GroupAttributeInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_GROUP\_ATTRIBUTE\_INFORMATION** structure (see section [2.2.4.2](#Section_cb80061b78014082bbe720d88b118eaa)).

**GroupAdminCommentInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_GROUP\_ADM\_COMMENT\_INFORMATION** structure (see section [2.2.4.5](#Section_84b322dd06214502ac311f50af943abb)).

**GroupReplicationInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_GROUP\_GENERAL\_INFORMATION** structure (see section 2.2.4.3).

#### SAMPR\_GROUP\_INFO\_BUFFER

The SAMPR\_GROUP\_INFO\_BUFFER union combines all possible structures used in the [SamrSetInformationGroup](#Section_e66db19f600a481bbc4e23953433255d) and [SamrQueryInformationGroup](#Section_8fd0003f807640ae872f9cdb6576b3c7) methods. For information on each field, with the exception of the **DoNotUse** field, see the associated section for the field structure.

1. typedef
2. [switch\_type(GROUP\_INFORMATION\_CLASS)]
3. union \_SAMPR\_GROUP\_INFO\_BUFFER {
4. [case(GroupGeneralInformation)]
5. SAMPR\_GROUP\_GENERAL\_INFORMATION General;
6. [case(GroupNameInformation)]
7. SAMPR\_GROUP\_NAME\_INFORMATION Name;
8. [case(GroupAttributeInformation)]
9. GROUP\_ATTRIBUTE\_INFORMATION Attribute;
10. [case(GroupAdminCommentInformation)]
11. SAMPR\_GROUP\_ADM\_COMMENT\_INFORMATION AdminComment;
12. [case(GroupReplicationInformation)]
13. SAMPR\_GROUP\_GENERAL\_INFORMATION DoNotUse;
14. } SAMPR\_GROUP\_INFO\_BUFFER,
15. \*PSAMPR\_GROUP\_INFO\_BUFFER;

**DoNotUse:**  This field exists to allow the [GroupReplicationInformation](#Section_951d1311bbfa453bb3a610f2e7bf6a4f) enumeration to be specified by the client.

As specified in section 3.1.5.5.3.1, the **General** field (instead of **DoNotUse**) MUST be used by the server when GroupReplicationInformation is received. GroupReplicationInformation is not valid for a set operation.

### Alias Query/Set Data Types

The structures and fields in this section relate to the following methods:

* [SamrQueryInformationAlias](#Section_599d6624d93b46de827a58eb0e7527bd)
* [SamrSetInformationAlias](#Section_ba787e4e3a4b47a2aca3c3ac2d2c511e)

The model of the methods is for the client to specify an enumeration that indicates the attributes to be either set or queried. There is duplication among the structures that contain the attributes. For a description of each attribute that is common among structures, see section [2.2.5.1](#Section_e60b8ed7aeb64627a19a7ac37a74bc10).

#### Common Alias Fields

There are a number of alias-related structures that use the same fields, as denoted by their field names. This section specifies all such fields.

The structures group the available set of alias attributes in different ways to allow the client to control which attributes are queried or set. While each structure might have a different subset of these attributes, they all draw from this same set of attributes, detailed as follows.

**AdminComment**: A counted Unicode string of type RPC\_UNICODE\_STRING, indicating the description of the [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0).

**MemberCount**: A 32-bit unsigned integer indicating the number of members in the alias object. This field is read-only.

**Name**: A counted Unicode string of type RPC\_UNICODE\_STRING, indicating the name of the alias object.

#### SAMPR\_ALIAS\_GENERAL\_INFORMATION

The SAMPR\_ALIAS\_GENERAL\_INFORMATION structure contains alias fields.

1. typedef struct \_SAMPR\_ALIAS\_GENERAL\_INFORMATION {
2. RPC\_UNICODE\_STRING Name;
3. unsigned long MemberCount;
4. RPC\_UNICODE\_STRING AdminComment;
5. } SAMPR\_ALIAS\_GENERAL\_INFORMATION,
6. \*PSAMPR\_ALIAS\_GENERAL\_INFORMATION;

For information on each field, see section [2.2.5.1](#Section_e60b8ed7aeb64627a19a7ac37a74bc10).

#### SAMPR\_ALIAS\_NAME\_INFORMATION

The SAMPR\_ALIAS\_NAME\_INFORMATION structure contains alias fields.

1. typedef struct \_SAMPR\_ALIAS\_NAME\_INFORMATION {
2. RPC\_UNICODE\_STRING Name;
3. } SAMPR\_ALIAS\_NAME\_INFORMATION,
4. \*PSAMPR\_ALIAS\_NAME\_INFORMATION;

For information on each field, see section [2.2.5.1](#Section_e60b8ed7aeb64627a19a7ac37a74bc10).

#### SAMPR\_ALIAS\_ADM\_COMMENT\_INFORMATION

The SAMPR\_ALIAS\_ADM\_COMMENT\_INFORMATION structure contains alias fields.

1. typedef struct \_SAMPR\_ALIAS\_ADM\_COMMENT\_INFORMATION {
2. RPC\_UNICODE\_STRING AdminComment;
3. } SAMPR\_ALIAS\_ADM\_COMMENT\_INFORMATION,
4. \*PSAMPR\_ALIAS\_ADM\_COMMENT\_INFORMATION;

For information on each field, see section [2.2.5.1](#Section_e60b8ed7aeb64627a19a7ac37a74bc10).

#### ALIAS\_INFORMATION\_CLASS

The ALIAS\_INFORMATION\_CLASS enumeration indicates how to interpret the *Buffer* parameter for [SamrQueryInformationAlias](#Section_599d6624d93b46de827a58eb0e7527bd) and [SamrSetInformationAlias](#Section_ba787e4e3a4b47a2aca3c3ac2d2c511e). For a list of the structures associated with each enumeration, see section [2.2.5.6](#Section_aaf25a5522c3463996c3ef5ae3362cf4).

1. typedef enum \_ALIAS\_INFORMATION\_CLASS
2. {
3. AliasGeneralInformation = 1,
4. AliasNameInformation,
5. AliasAdminCommentInformation
6. } ALIAS\_INFORMATION\_CLASS;

**AliasGeneralInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_ALIAS\_GENERAL\_INFORMATION** structure (see section [2.2.5.2](#Section_025890e9c48347e08761632caf30be3f)).

**AliasNameInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_ALIAS\_NAME\_INFORMATION** structure (see section [2.2.5.3](#Section_e32831e5429e493d8cd2297236f0792c)).

**AliasAdminCommentInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_ALIAS\_ADM\_COMMENT\_INFORMATION** structure (see section [2.2.5.4](#Section_0968f7a5d87c4fec85c45287a7be6cf4)).

#### SAMPR\_ALIAS\_INFO\_BUFFER

The SAMPR\_ALIAS\_INFO\_BUFFER union combines all possible structures used in the [SamrSetInformationAlias](#Section_ba787e4e3a4b47a2aca3c3ac2d2c511e) and [SamrQueryInformationAlias](#Section_599d6624d93b46de827a58eb0e7527bd) methods. For information on each field, see the associated section for the field structure.

1. typedef
2. [switch\_type(ALIAS\_INFORMATION\_CLASS)]
3. union \_SAMPR\_ALIAS\_INFO\_BUFFER {
4. [case(AliasGeneralInformation)]
5. SAMPR\_ALIAS\_GENERAL\_INFORMATION General;
6. [case(AliasNameInformation)]
7. SAMPR\_ALIAS\_NAME\_INFORMATION Name;
8. [case(AliasAdminCommentInformation)]
9. SAMPR\_ALIAS\_ADM\_COMMENT\_INFORMATION AdminComment;
10. } SAMPR\_ALIAS\_INFO\_BUFFER,
11. \*PSAMPR\_ALIAS\_INFO\_BUFFER;

### User Query/Set Data Types

The structures and fields in this section relate to the following methods:

* [SamrQueryInformationUser](#Section_4ad8d54c0d5a4d5a9e9a1bc9ee008d47)
* [SamrQueryInformationUser2](#Section_29ab27f661da4c7d863ce228ee798f4d)
* [SamrSetInformationUser](#Section_538222f71b894811949a0eac62e38dce)
* [SamrSetInformationUser2](#Section_99ee9f3943e84bbaac3a82e0c0e0699e)

The model of the methods is for the client to specify an enumeration that indicates the attributes to be either set or queried. There is duplication among the structures that contain the attributes. For a description of each attribute that is common among structures, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### Common User Fields

There are a number of user-related structures that use the same fields, as denoted by their field names. This section specifies all such fields.

These structures group the available set of user attributes in different ways to allow the client greater control over which attributes are queried or set. While each structure might have a different subset of these attributes, they all draw from this same set of attributes, detailed as follows.

There are a number of fields that are of type "user profile information" (as indicated in their descriptions). The server does not enforce any format restrictions on these values during an update. These values are used by authentication protocols—Kerberos, for example, as specified in [[MS-PAC]](%5bMS-PAC%5d.pdf#Section_166d8064c86341e19c23edaaa5f36962) section 2.5—to communicate end-user environment values to an interactive-logon application running on a member workstation or server. For clarity, Windows behavior is cited in this section to describe the expectations of such Windows interactive-logon applications with respect to these values. If no Windows behavior is cited, there is no expectation of a specific format.

The mapping between the fields described below and the actual attributes in the database is defined in section [3.1.5.14.11](#Section_66CF69CD731847DA914F24F4960B2C36).

**AccountExpires**: A 64-bit value, equivalent to a FILETIME, indicating the time at which an [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) is no longer permitted to log on.

**AdminComment**: A counted Unicode string of type RPC\_UNICODE\_STRING, indicating the description of the [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).

**BadPasswordCount**: A 16-bit unsigned integer indicating the number of bad password attempts. This field is read-only.

**CodePage**: A 16-bit unsigned integer indicating a code page preference specific to this user object. The space of values is the Microsoft code page designation. For more information, see [[MSDN-CP]](https://go.microsoft.com/fwlink/?LinkId=89981).

**CountryCode**: A 16-bit unsigned integer indicating a country preference specific to this user. The space of values is the international country calling code, as specified in [[E164]](https://go.microsoft.com/fwlink/?LinkId=89855). For example, the country code of the United Kingdom, in decimal notation, is 44.

**FullName**: A counted Unicode string of type RPC\_UNICODE\_STRING, indicating a free format string for any name type (for example, "Akers, Kim").

**HomeDirectory**: A counted Unicode string of type RPC\_UNICODE\_STRING, indicating a directory for use by an end-user interactive-logon application. This is [**user profile**](#gt_dcec4633-c6dd-4dc0-a43f-b57b009e96ee) information.[<14>](#Appendix_A_14" \o "Product behavior note 14)

**HomeDirectoryDrive**: A counted Unicode string of type RPC\_UNICODE\_STRING, indicating the disk drive to which HomeDirectory is relative. This is user profile information.[<15>](#Appendix_A_15" \o "Product behavior note 15)

**LastLogoff**: A 64-bit value, equivalent to a FILETIME, indicating the time at which the account last logged off. This field is read-only.[<16>](#Appendix_A_16" \o "Product behavior note 16)

**LastLogon**: A 64-bit value, equivalent to a FILETIME, indicating the time at which the account last logged on. This field is read-only.[<17>](#Appendix_A_17" \o "Product behavior note 17)

**LogonCount**: A 16-bit unsigned integer indicating the number of times that the user account has been authenticated. This field is read-only.[<18>](#Appendix_A_18" \o "Product behavior note 18)

**LogonHours**: A binary value with the structure [SAMPR\_LOGON\_HOURS](#Section_d83c356b7dda409682705c581f84a4d9), indicating a logon policy describing the time periods during which the user can authenticate. This policy is specified in detail in section 2.2.6.5.

**Parameters**: A binary value stored in the **Buffer** field of a RPC\_UNICODE\_STRING for per-user application state. Per-user application state is any binary data that an application associates with a user. However, because there is no requirement for the server of this protocol to enforce any format, application developers are discouraged from using this mechanism in order to avoid the chance of one application overwriting another application's data.

**PasswordCanChange**: A 64-bit value, equivalent to a FILETIME, indicating the time at which a password change request will be accepted by the server. This field is read-only.

**PasswordExpired**: A 1-byte value. On receipt at the server, a nonzero value for this field indicates that the password MUST be expired immediately (see [SamrSetInformationUser2 (section 3.1.5.6.4)](#Section_99ee9f3943e84bbaac3a82e0c0e0699e) for details). On receipt at the client, a nonzero value for this field indicates that the password has expired; a value of zero indicates that the password has not expired.

**PasswordLastSet**: A 64-bit value, equivalent to a FILETIME, indicating the time at which a password was last updated. This field is read-only.

**PasswordMustChange**: A 64-bit value, equivalent to a FILETIME, indicating the time at which authentications will fail unless a password reset or change occurs. This field is read-only.

**PrimaryGroupId**: A 32-bit unsigned integer indicating the primary group ID of the user.

**ProfilePath**: A counted Unicode string of type RPC\_UNICODE\_STRING, containing a UNC path to a network-based user profile. This is user profile information.

**ScriptPath**: A counted Unicode string of type RPC\_UNICODE\_STRING, containing a UNC path to a network-based script or executable file that is executed during an interactive logon. This is user profile information.

**UserAccountControl**: A 32-bit bit field specifying characteristics of the account. See section [2.2.1.12](#Section_b10cfda1f24f441b8f4380cb93e786ec) for possible values.

**UserComment**: A counted Unicode string of type RPC\_UNICODE\_STRING containing an end-user–writable comment about the user. This is distinguished from **AdminComment** by the fact that, by default, end users can update this value on their own accounts.

**UserId**: A 32-bit unsigned integer representing the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of the account. This field is read-only.

**UserName**: A counted Unicode string of type RPC\_UNICODE\_STRING containing the name of the account.

**WorkStations**: A binary value stored in an RPC\_UNICODE\_STRING structure containing the list of workstations from which the account can interactively log on. For information on the required format of the binary value, see section [3.1.1.6](#Section_e270cd0a529541be9e892c3dc3a39536).

#### USER\_PRIMARY\_GROUP\_INFORMATION

The USER\_PRIMARY\_GROUP\_INFORMATION structure contains user fields.

1. typedef struct \_USER\_PRIMARY\_GROUP\_INFORMATION {
2. unsigned long PrimaryGroupId;
3. } USER\_PRIMARY\_GROUP\_INFORMATION,
4. \*PUSER\_PRIMARY\_GROUP\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### USER\_CONTROL\_INFORMATION

The USER\_CONTROL\_INFORMATION structure contains user fields.

1. typedef struct \_USER\_CONTROL\_INFORMATION {
2. unsigned long UserAccountControl;
3. } USER\_CONTROL\_INFORMATION,
4. \*PUSER\_CONTROL\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### USER\_EXPIRES\_INFORMATION

The USER\_EXPIRES\_INFORMATION structure contains user fields.

1. typedef struct \_USER\_EXPIRES\_INFORMATION {
2. OLD\_LARGE\_INTEGER AccountExpires;
3. } USER\_EXPIRES\_INFORMATION,
4. \*PUSER\_EXPIRES\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_LOGON\_HOURS

The SAMPR\_LOGON\_HOURS structure contains logon policy information that describes when a user account is permitted to authenticate.

1. typedef struct \_SAMPR\_LOGON\_HOURS {
2. unsigned short UnitsPerWeek;
3. [size\_is(1260), length\_is((UnitsPerWeek+7)/8)]
4. unsigned char\* LogonHours;
5. } SAMPR\_LOGON\_HOURS,
6. \*PSAMPR\_LOGON\_HOURS;

**UnitsPerWeek:**  A division of the week (7 days). For example, the value 7 means that each unit is a day; a value of (7\*24) means that the units are hours. The minimum granularity of time is one minute, where the UnitsPerWeek would be 10080; therefore, the maximum size of LogonHours is 10080/8, or 1,260 bytes.

**LogonHours:**  A pointer to a bit field containing at least **UnitsPerWeek** number of bits. The leftmost bit represents the first unit, starting at Sunday, 12 A.M. If a bit is set, authentication is allowed to occur; otherwise, authentication is not allowed to occur.

For example, if the **UnitsPerWeek** value is 168 (that is, the units per week is hours, resulting in a 21-byte bit field), and if the leftmost bit is set and the rightmost bit is set, the user is able to log on for two consecutive hours between Saturday, 11 P.M. and Sunday, 1 A.M.

#### SAMPR\_USER\_ALL\_INFORMATION

The SAMPR\_USER\_ALL\_INFORMATION structure contains user attribute information. Most fields are described in section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f). The exceptions are described below.

1. typedef struct \_SAMPR\_USER\_ALL\_INFORMATION {
2. OLD\_LARGE\_INTEGER LastLogon;
3. OLD\_LARGE\_INTEGER LastLogoff;
4. OLD\_LARGE\_INTEGER PasswordLastSet;
5. OLD\_LARGE\_INTEGER AccountExpires;
6. OLD\_LARGE\_INTEGER PasswordCanChange;
7. OLD\_LARGE\_INTEGER PasswordMustChange;
8. RPC\_UNICODE\_STRING UserName;
9. RPC\_UNICODE\_STRING FullName;
10. RPC\_UNICODE\_STRING HomeDirectory;
11. RPC\_UNICODE\_STRING HomeDirectoryDrive;
12. RPC\_UNICODE\_STRING ScriptPath;
13. RPC\_UNICODE\_STRING ProfilePath;
14. RPC\_UNICODE\_STRING AdminComment;
15. RPC\_UNICODE\_STRING WorkStations;
16. RPC\_UNICODE\_STRING UserComment;
17. RPC\_UNICODE\_STRING Parameters;
18. RPC\_SHORT\_BLOB LmOwfPassword;
19. RPC\_SHORT\_BLOB NtOwfPassword;
20. RPC\_UNICODE\_STRING PrivateData;
21. SAMPR\_SR\_SECURITY\_DESCRIPTOR SecurityDescriptor;
22. unsigned long UserId;
23. unsigned long PrimaryGroupId;
24. unsigned long UserAccountControl;
25. unsigned long WhichFields;
26. SAMPR\_LOGON\_HOURS LogonHours;
27. unsigned short BadPasswordCount;
28. unsigned short LogonCount;
29. unsigned short CountryCode;
30. unsigned short CodePage;
31. unsigned char LmPasswordPresent;
32. unsigned char NtPasswordPresent;
33. unsigned char PasswordExpired;
34. unsigned char PrivateDataSensitive;
35. } SAMPR\_USER\_ALL\_INFORMATION,
36. \*PSAMPR\_USER\_ALL\_INFORMATION;

**LmOwfPassword:**  An [RPC\_SHORT\_BLOB](#Section_77dbfdbb66274871ab125333929347dc) structure where **Length** and **MaximumLength** MUST be 16, and the **Buffer** MUST be formatted with an [ENCRYPTED\_LM\_OWF\_PASSWORD](#Section_ce061fef6d4f4802bd5d26b11f14f4a6) structure with the cleartext value being an [**LM hash**](#gt_58a72159-b1e9-4770-9b06-d7a6b16279a8), and the encryption key being the 16-byte SMB session key obtained as specified in either section [3.1.2.4](#Section_c718b655919f43a88897be080553e746) or section [3.2.2.3](#Section_a26ab2cb3e0949809f2d8d75f73005ae).

**NtOwfPassword:**  An RPC\_SHORT\_BLOB structure where **Length** and **MaximumLength** MUST be 16, and the **Buffer** MUST be formatted with an ENCRYPTED\_NT\_OWF\_PASSWORD structure with the cleartext value being an [**NT hash**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f), and the encryption key being the 16-byte SMB session key obtained as specified in either section 3.1.2.4 or section 3.2.2.3.

**PrivateData:**  Not used. Ignored on receipt at the server and client. Clients MUST set to zero when sent, and servers MUST set to zero on return.

**SecurityDescriptor:**  Not used. Ignored on receipt at the server and client. Clients MUST set to zero when sent, and servers MUST set to zero on return.

**WhichFields:**  A 32-bit bit field indicating which fields within the SAMPR\_USER\_ALL\_INFORMATION structure will be processed by the server. Section [2.2.1.8](#Section_2675c17672e04ac9ae6dcdd87b8ba520) specifies the valid bits and also specifies the structure field to which each bit corresponds.

**Note**  If a given bit is set, the associated field MUST be processed; if a given bit is not set, then the associated field MUST be ignored.

**LmPasswordPresent:**  If zero, LmOwfPassword MUST be ignored; otherwise, LmOwfPassword MUST be processed.

**NtPasswordPresent:**  If zero, NtOwfPassword MUST be ignored; otherwise, NtOwfPassword MUST be processed.

**PrivateDataSensitive:**  Not used. Ignored on receipt at the server and client.

#### SAMPR\_USER\_GENERAL\_INFORMATION

The SAMPR\_USER\_GENERAL\_INFORMATION structure contains user fields.

1. typedef struct \_SAMPR\_USER\_GENERAL\_INFORMATION {
2. RPC\_UNICODE\_STRING UserName;
3. RPC\_UNICODE\_STRING FullName;
4. unsigned long PrimaryGroupId;
5. RPC\_UNICODE\_STRING AdminComment;
6. RPC\_UNICODE\_STRING UserComment;
7. } SAMPR\_USER\_GENERAL\_INFORMATION,
8. \*PSAMPR\_USER\_GENERAL\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_PREFERENCES\_INFORMATION

The SAMPR\_USER\_PREFERENCES\_INFORMATION structure contains user fields.

1. typedef struct \_SAMPR\_USER\_PREFERENCES\_INFORMATION {
2. RPC\_UNICODE\_STRING UserComment;
3. RPC\_UNICODE\_STRING Reserved1;
4. unsigned short CountryCode;
5. unsigned short CodePage;
6. } SAMPR\_USER\_PREFERENCES\_INFORMATION,
7. \*PSAMPR\_USER\_PREFERENCES\_INFORMATION;

**Reserved1:**  Ignored by the client and server and MUST be a zero-length string when sent and returned.

For information on all other fields, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_PARAMETERS\_INFORMATION

The SAMPR\_USER\_PARAMETERS\_INFORMATION structure contains user fields.

1. typedef struct \_SAMPR\_USER\_PARAMETERS\_INFORMATION {
2. RPC\_UNICODE\_STRING Parameters;
3. } SAMPR\_USER\_PARAMETERS\_INFORMATION,
4. \*PSAMPR\_USER\_PARAMETERS\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_LOGON\_INFORMATION

The SAMPR\_USER\_LOGON\_INFORMATION structure contains user fields.

1. typedef struct \_SAMPR\_USER\_LOGON\_INFORMATION {
2. RPC\_UNICODE\_STRING UserName;
3. RPC\_UNICODE\_STRING FullName;
4. unsigned long UserId;
5. unsigned long PrimaryGroupId;
6. RPC\_UNICODE\_STRING HomeDirectory;
7. RPC\_UNICODE\_STRING HomeDirectoryDrive;
8. RPC\_UNICODE\_STRING ScriptPath;
9. RPC\_UNICODE\_STRING ProfilePath;
10. RPC\_UNICODE\_STRING WorkStations;
11. OLD\_LARGE\_INTEGER LastLogon;
12. OLD\_LARGE\_INTEGER LastLogoff;
13. OLD\_LARGE\_INTEGER PasswordLastSet;
14. OLD\_LARGE\_INTEGER PasswordCanChange;
15. OLD\_LARGE\_INTEGER PasswordMustChange;
16. SAMPR\_LOGON\_HOURS LogonHours;
17. unsigned short BadPasswordCount;
18. unsigned short LogonCount;
19. unsigned long UserAccountControl;
20. } SAMPR\_USER\_LOGON\_INFORMATION,
21. \*PSAMPR\_USER\_LOGON\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_ACCOUNT\_INFORMATION

The SAMPR\_USER\_ACCOUNT\_INFORMATION structure contains user fields.

1. typedef struct \_SAMPR\_USER\_ACCOUNT\_INFORMATION {
2. RPC\_UNICODE\_STRING UserName;
3. RPC\_UNICODE\_STRING FullName;
4. unsigned long UserId;
5. unsigned long PrimaryGroupId;
6. RPC\_UNICODE\_STRING HomeDirectory;
7. RPC\_UNICODE\_STRING HomeDirectoryDrive;
8. RPC\_UNICODE\_STRING ScriptPath;
9. RPC\_UNICODE\_STRING ProfilePath;
10. RPC\_UNICODE\_STRING AdminComment;
11. RPC\_UNICODE\_STRING WorkStations;
12. OLD\_LARGE\_INTEGER LastLogon;
13. OLD\_LARGE\_INTEGER LastLogoff;
14. SAMPR\_LOGON\_HOURS LogonHours;
15. unsigned short BadPasswordCount;
16. unsigned short LogonCount;
17. OLD\_LARGE\_INTEGER PasswordLastSet;
18. OLD\_LARGE\_INTEGER AccountExpires;
19. unsigned long UserAccountControl;
20. } SAMPR\_USER\_ACCOUNT\_INFORMATION,
21. \*PSAMPR\_USER\_ACCOUNT\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_A\_NAME\_INFORMATION

The SAMPR\_USER\_A\_NAME\_INFORMATION structure contains user fields.

1. typedef struct \_SAMPR\_USER\_A\_NAME\_INFORMATION {
2. RPC\_UNICODE\_STRING UserName;
3. } SAMPR\_USER\_A\_NAME\_INFORMATION,
4. \*PSAMPR\_USER\_A\_NAME\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_F\_NAME\_INFORMATION

The SAMPR\_USER\_F\_NAME\_INFORMATION structure contains user fields.

1. typedef struct \_SAMPR\_USER\_F\_NAME\_INFORMATION {
2. RPC\_UNICODE\_STRING FullName;
3. } SAMPR\_USER\_F\_NAME\_INFORMATION,
4. \*PSAMPR\_USER\_F\_NAME\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_NAME\_INFORMATION

The SAMPR\_USER\_NAME\_INFORMATION structure contains user fields.

1. typedef struct \_SAMPR\_USER\_NAME\_INFORMATION {
2. RPC\_UNICODE\_STRING UserName;
3. RPC\_UNICODE\_STRING FullName;
4. } SAMPR\_USER\_NAME\_INFORMATION,
5. \*PSAMPR\_USER\_NAME\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_HOME\_INFORMATION

The SAMPR\_USER\_HOME\_INFORMATION structure contains user fields.

1. typedef struct \_SAMPR\_USER\_HOME\_INFORMATION {
2. RPC\_UNICODE\_STRING HomeDirectory;
3. RPC\_UNICODE\_STRING HomeDirectoryDrive;
4. } SAMPR\_USER\_HOME\_INFORMATION,
5. \*PSAMPR\_USER\_HOME\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_SCRIPT\_INFORMATION

The SAMPR\_USER\_SCRIPT\_INFORMATION structure contains user fields.

1. typedef struct \_SAMPR\_USER\_SCRIPT\_INFORMATION {
2. RPC\_UNICODE\_STRING ScriptPath;
3. } SAMPR\_USER\_SCRIPT\_INFORMATION,
4. \*PSAMPR\_USER\_SCRIPT\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_PROFILE\_INFORMATION

The SAMPR\_USER\_PROFILE\_INFORMATION structure contains user fields.

1. typedef struct \_SAMPR\_USER\_PROFILE\_INFORMATION {
2. RPC\_UNICODE\_STRING ProfilePath;
3. } SAMPR\_USER\_PROFILE\_INFORMATION,
4. \*PSAMPR\_USER\_PROFILE\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_ADMIN\_COMMENT\_INFORMATION

The SAMPR\_USER\_ADMIN\_COMMENT\_INFORMATION structure contains user fields.

1. typedef struct \_SAMPR\_USER\_ADMIN\_COMMENT\_INFORMATION {
2. RPC\_UNICODE\_STRING AdminComment;
3. } SAMPR\_USER\_ADMIN\_COMMENT\_INFORMATION,
4. \*PSAMPR\_USER\_ADMIN\_COMMENT\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_WORKSTATIONS\_INFORMATION

The SAMPR\_USER\_WORKSTATIONS\_INFORMATION structure contains user fields.

1. typedef struct \_SAMPR\_USER\_WORKSTATIONS\_INFORMATION {
2. RPC\_UNICODE\_STRING WorkStations;
3. } SAMPR\_USER\_WORKSTATIONS\_INFORMATION,
4. \*PSAMPR\_USER\_WORKSTATIONS\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_LOGON\_HOURS\_INFORMATION

The SAMPR\_USER\_LOGON\_HOURS\_INFORMATION structure contains user fields.

1. typedef struct \_SAMPR\_USER\_LOGON\_HOURS\_INFORMATION {
2. SAMPR\_LOGON\_HOURS LogonHours;
3. } SAMPR\_USER\_LOGON\_HOURS\_INFORMATION,
4. \*PSAMPR\_USER\_LOGON\_HOURS\_INFORMATION;

For information on each field, see section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_ENCRYPTED\_USER\_PASSWORD

The SAMPR\_ENCRYPTED\_USER\_PASSWORD structure carries an encrypted string.

1. typedef struct \_SAMPR\_ENCRYPTED\_USER\_PASSWORD {
2. unsigned char Buffer[(256 \* 2) + 4];
3. } SAMPR\_ENCRYPTED\_USER\_PASSWORD,
4. \*PSAMPR\_ENCRYPTED\_USER\_PASSWORD;

**Buffer:**  An array to carry encrypted cleartext password data. The encryption key is method-specific, while the algorithm specified in section [3.2.2.1](#Section_5fe3c4c4e71b440db2fd8448bfaf6e04) is common for all methods that use this structure. See the message syntax for [SamrOemChangePasswordUser2 (section 3.1.5.10.2)](#Section_8d0bf63efa5f4c75be22558c52075842) and [SamrUnicodeChangePasswordUser2 (section 3.1.5.10.3)](#Section_acb3204ada8b478e91391ea589edb880), and the message processing for [SamrSetInformationUser2 (section 3.1.5.6.4)](#Section_99ee9f3943e84bbaac3a82e0c0e0699e), for details on the encryption key selection. The size of (256 \* 2) + 4 for **Buffer** is determined by the size of the structure that is encrypted, SAMPR\_USER\_PASSWORD; see below for more details.

For all protocol uses, the decrypted format of **Buffer** is the following structure.

1. typedef struct \_SAMPR\_USER\_PASSWORD {
2. wchar\_t Buffer[256];
3. unsigned long Length;
4. } SAMPR\_USER\_PASSWORD, \*PSAMPR\_USER\_PASSWORD;

**Buffer**: This array contains the cleartext value at the end of the buffer. The start of the string is **Length** number of bytes from the end of the buffer. The cleartext value can be no more than 512 bytes. The unused portions of SAMPR\_USER\_PASSWORD.Buffer SHOULD be filled with random bytes by the client. The value 512 is chosen because that is the longest password allowed by this protocol (and enforced by the server).

**Length**: An unsigned integer, in little-endian byte order, that indicates the number of bytes of the cleartext value located in SAMPR\_USER\_PASSWORD.Buffer.

Implementations of this protocol MUST protect the SAMPR\_ENCRYPTED\_USER\_PASSWORD structure by encrypting the 516 bytes of data referenced in its **Buffer** field on request (and reply), and decrypting on receipt. See section 3.2.2.1 for the specification of the algorithm performing encryption and decryption.

#### SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW

The SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW structure carries an encrypted string.

1. typedef struct \_SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW {
2. unsigned char Buffer[(256 \* 2) + 4 + 16];
3. } SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW,
4. \*PSAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW;

**Buffer:**  An array to carry encrypted cleartext password data.

For all protocol uses, the decrypted format of **Buffer** is the following structure.

1. typedef struct \_SAMPR\_USER\_PASSWORD\_NEW {
2. WCHAR Buffer[256];
3. ULONG Length;
4. UCHAR ClearSalt[16];
5. } SAMPR\_USER\_PASSWORD\_NEW, \*PSAMPR\_USER\_PASSWORD\_NEW;

**Buffer**: This array contains the cleartext value at the end of the buffer. The cleartext value can be no more than 512 bytes. The start of the string is **Length** number of bytes from the end of the buffer. The unused portions of SAMPR\_USER\_PASSWORD\_NEW.Buffer SHOULD be filled with random bytes by the client.

**Length**: An unsigned integer, in little-endian byte order, that indicates the number of bytes of the cleartext value (located in SAMPR\_USER\_PASSWORD\_NEW.Buffer).

**ClearSalt**: This value (a [**salt**](#gt_1672c769-f184-404a-9575-e637fd3a43ed)) MUST be filled with random bytes by the client and MUST NOT be encrypted. The length of 16 was chosen in particular because 128 bits of randomness was deemed sufficiently secure when this protocol was introduced (circa 1998).

Implementations of this protocol MUST protect the SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW structure by encrypting the first 516 bytes of data referenced in its **Buffer** field on request (and reply) and by decrypting on receipt. See section [3.2.2.1](#Section_5fe3c4c4e71b440db2fd8448bfaf6e04) for the specification of the algorithm performing encryption and decryption.

The first 516 bytes are defined as the first 516 bytes of the SAMPR\_USER\_PASSWORD\_NEW structure defined previously. The last 16 bytes of the SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW structure are defined as the last 16 bytes of the SAMPR\_USER\_PASSWORD\_NEW structure and MUST NOT be encrypted or decrypted.

#### SAMPR\_USER\_INTERNAL1\_INFORMATION

The SAMPR\_USER\_INTERNAL1\_INFORMATION structure holds the hashed form of a cleartext password.

1. typedef struct \_SAMPR\_USER\_INTERNAL1\_INFORMATION {
2. ENCRYPTED\_NT\_OWF\_PASSWORD EncryptedNtOwfPassword;
3. ENCRYPTED\_LM\_OWF\_PASSWORD EncryptedLmOwfPassword;
4. unsigned char NtPasswordPresent;
5. unsigned char LmPasswordPresent;
6. unsigned char PasswordExpired;
7. } SAMPR\_USER\_INTERNAL1\_INFORMATION,
8. \*PSAMPR\_USER\_INTERNAL1\_INFORMATION;

**EncryptedNtOwfPassword:**  An [**NT hash**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f) encrypted with the 16-byte SMB session key obtained as specified in either section [3.1.2.4](#Section_c718b655919f43a88897be080553e746) or section [3.2.2.3](#Section_a26ab2cb3e0949809f2d8d75f73005ae).

**EncryptedLmOwfPassword:**  An [**LM hash**](#gt_58a72159-b1e9-4770-9b06-d7a6b16279a8) encrypted with the 16-byte SMB session key obtained as specified in either section 3.1.2.4 or section 3.2.2.3.

**NtPasswordPresent:**  If nonzero, indicates that the **EncryptedNtOwfPassword** value is valid; otherwise, **EncryptedNtOwfPassword** MUST be ignored.

**LmPasswordPresent:**  If nonzero, indicates that the **EncryptedLmOwfPassword** value is valid; otherwise, **EncryptedLmOwfPassword** MUST be ignored.

**PasswordExpired:**  See section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_INTERNAL4\_INFORMATION

The SAMPR\_USER\_INTERNAL4\_INFORMATION structure holds all attributes of a user, along with an encrypted password.

1. typedef struct \_SAMPR\_USER\_INTERNAL4\_INFORMATION {
2. SAMPR\_USER\_ALL\_INFORMATION I1;
3. SAMPR\_ENCRYPTED\_USER\_PASSWORD UserPassword;
4. } SAMPR\_USER\_INTERNAL4\_INFORMATION,
5. \*PSAMPR\_USER\_INTERNAL4\_INFORMATION;

**I1:**  See section [2.2.6.6](#Section_dc966b81da274daea28cec16534f1cb9).

**UserPassword:**  See section [2.2.6.21](#Section_23f9ef4ccf3e43309287ea4799b03201).

#### SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW

The SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW structure holds all attributes of a user, along with an encrypted password. The encrypted password uses a [**salt**](#gt_1672c769-f184-404a-9575-e637fd3a43ed) to improve the encryption algorithm. See the specification for [SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW (section 2.2.6.22)](#Section_112ecc941cbe41cdb669377402c20786) for details on salt value selection.

1. typedef struct \_SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW {
2. SAMPR\_USER\_ALL\_INFORMATION I1;
3. SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW UserPassword;
4. } SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW,
5. \*PSAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW;

**I1:**  See section [2.2.6.6](#Section_dc966b81da274daea28cec16534f1cb9).

**UserPassword:**  See section 2.2.6.22.

#### SAMPR\_USER\_INTERNAL5\_INFORMATION

The SAMPR\_USER\_INTERNAL5\_INFORMATION structure holds an encrypted password.

This structure is used to carry a new password for a particular [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) from the client to the server, encrypted in a way that protects it from disclosure or tampering while in transit.

1. typedef struct \_SAMPR\_USER\_INTERNAL5\_INFORMATION {
2. SAMPR\_ENCRYPTED\_USER\_PASSWORD UserPassword;
3. unsigned char PasswordExpired;
4. } SAMPR\_USER\_INTERNAL5\_INFORMATION,
5. \*PSAMPR\_USER\_INTERNAL5\_INFORMATION;

**UserPassword:**  A cleartext password, encrypted according to the specification for [SAMPR\_ENCRYPTED\_USER\_PASSWORD](#Section_23f9ef4ccf3e43309287ea4799b03201), with the encryption key being the 16-byte SMB session key obtained as specified in either section [3.1.2.4](#Section_c718b655919f43a88897be080553e746) or section [3.2.2.3](#Section_a26ab2cb3e0949809f2d8d75f73005ae).

**PasswordExpired:**  See section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_INTERNAL5\_INFORMATION\_NEW

The SAMPR\_USER\_INTERNAL5\_INFORMATION\_NEW structure communicates an encrypted password. The encrypted password uses a [**salt**](#gt_1672c769-f184-404a-9575-e637fd3a43ed) to improve the encryption algorithm. See the specification for [SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW (section 2.2.6.22)](#Section_112ecc941cbe41cdb669377402c20786) for details on salt value selection.

This structure is used to carry a new password for a particular [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) from the client to the server, encrypted in a way that protects it from disclosure or tampering while in transit. A random value, a salt, is used by the client to seed the encryption routine; see section 2.2.6.22 for details.

1. typedef struct \_SAMPR\_USER\_INTERNAL5\_INFORMATION\_NEW {
2. SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW UserPassword;
3. unsigned char PasswordExpired;
4. } SAMPR\_USER\_INTERNAL5\_INFORMATION\_NEW,
5. \*PSAMPR\_USER\_INTERNAL5\_INFORMATION\_NEW;

**UserPassword:**  A password, encrypted according to the specification for SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW, with the encryption key being the 16-byte SMB session key obtained as specified in either section [3.1.2.4](#Section_c718b655919f43a88897be080553e746) or section [3.2.2.3](#Section_a26ab2cb3e0949809f2d8d75f73005ae).

**PasswordExpired:**  See section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### USER\_INFORMATION\_CLASS

The USER\_INFORMATION\_CLASS enumeration indicates how to interpret the *Buffer* parameter for [SamrSetInformationUser](#Section_538222f71b894811949a0eac62e38dce), [SamrQueryInformationUser](#Section_4ad8d54c0d5a4d5a9e9a1bc9ee008d47), [SamrSetInformationUser2](#Section_99EE9F3943E84BBAAC3A82E0C0E0699E), and [SamrQueryInformationUser2](#Section_29AB27F661DA4C7D863CE228EE798F4D). For a list of associated structures, see section [2.2.6.29](#Section_9496c26e490b4e76827f2695fc216f35).

1. typedef enum \_USER\_INFORMATION\_CLASS
2. {
3. UserGeneralInformation = 1,
4. UserPreferencesInformation = 2,
5. UserLogonInformation = 3,
6. UserLogonHoursInformation = 4,
7. UserAccountInformation = 5,
8. UserNameInformation = 6,
9. UserAccountNameInformation = 7,
10. UserFullNameInformation = 8,
11. UserPrimaryGroupInformation = 9,
12. UserHomeInformation = 10,
13. UserScriptInformation = 11,
14. UserProfileInformation = 12,
15. UserAdminCommentInformation = 13,
16. UserWorkStationsInformation = 14,
17. UserControlInformation = 16,
18. UserExpiresInformation = 17,
19. UserInternal1Information = 18,
20. UserParametersInformation = 20,
21. UserAllInformation = 21,
22. UserInternal4Information = 23,
23. UserInternal5Information = 24,
24. UserInternal4InformationNew = 25,
25. UserInternal5InformationNew = 26
26. } USER\_INFORMATION\_CLASS,
27. \*PUSER\_INFORMATION\_CLASS;

**UserGeneralInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_GENERAL\_INFORMATION** structure (see section [2.2.6.7](#Section_d74231bd81e242299e82ce6d3713cc62)).

**UserPreferencesInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_PREFERENCES\_INFORMATION** structure (see section [2.2.6.8](#Section_551365bdd06d4e0ca72a41567e008f1e)).

**UserLogonInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_LOGON\_INFORMATION** structure (see section [2.2.6.10](#Section_29b54f06896143fd8ecb4b2a8020d474)).

**UserLogonHoursInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_LOGON\_HOURS\_INFORMATION** structure (see section [2.2.6.20](#Section_094647addb3647b6956de672b13ec3e6)).

**UserAccountInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_ACCOUNT\_INFORMATION** structure (see section [2.2.6.11](#Section_7aef58f2474a46abaa2a54f7eb8625e9)).

**UserNameInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_NAME\_INFORMATION** structure (see section [2.2.6.14](#Section_400d937e66e544af929d13dfab550d46)).

**UserAccountNameInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_A\_NAME\_INFORMATION** structure (see section [2.2.6.12](#Section_89a4133172c74814af3979bdba0d822f)).

**UserFullNameInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_F\_NAME\_INFORMATION** structure (see section [2.2.6.13](#Section_22f073f55e9340098c9a8569e910f379)).

**UserPrimaryGroupInformation:** Indicates the *Buffer* parameter is to be interpreted as a **USER\_PRIMARY\_GROUP\_INFORMATION** structure (see section [2.2.6.2](#Section_3fd86f83f5ad4bb4814a6554f5797b34)).

**UserHomeInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_HOME\_INFORMATION** structure (see section [2.2.6.15](#Section_c42b2b9d301d40b7ad4c257f31904602)).

**UserScriptInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_SCRIPT\_INFORMATION** structure (see section [2.2.6.16](#Section_4e55e798c5874e0cbcfd944e7480dc22)).

**UserProfileInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_PROFILE\_INFORMATION** structure (see section [2.2.6.17](#Section_53883066b32941938a9fcbf53927804d)).

**UserAdminCommentInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_ADMIN\_COMMENT\_INFORMATION** structure (see section [2.2.6.18](#Section_fd4e8928c3704d0cb0fb5c781128cccb)).

**UserWorkStationsInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_WORKSTATIONS\_INFORMATION** structure (see section [2.2.6.19](#Section_f991ad1bf9b043ac853b4d900c7618f0)).

**UserControlInformation:** Indicates the *Buffer* parameter is to be interpreted as a **USER\_CONTROL\_INFORMATION** structure (see section [2.2.6.3](#Section_eb5f1508ede14ff1be8255f3e2ef1633)).

**UserExpiresInformation:** Indicates the *Buffer* parameter is to be interpreted as a **USER\_EXPIRES\_INFORMATION** structure (see section [2.2.6.4](#Section_8c1308e0723e4170817045a6a723989a)).

**UserInternal1Information:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_INTERNAL1\_INFORMATION** structure (see section [2.2.6.23](#Section_50d17755c6b840bd8cacbd6cfa31adf2)).

**UserParametersInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_PARAMETERS\_INFORMATION** structure (see section [2.2.6.9](#Section_05f1ac12c90f4aed8821d211a9c834b3)).

**UserAllInformation:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_ALL\_INFORMATION** structure (see section [2.2.6.6](#Section_dc966b81da274daea28cec16534f1cb9)).

**UserInternal4Information:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_INTERNAL4\_INFORMATION** structure (see section [2.2.6.24](#Section_e28bf420898944fb8b08f5a7c2f2e33c)).

**UserInternal5Information:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_INTERNAL5\_INFORMATION** structure (see section [2.2.6.26](#Section_1d2be36a754e46b18697d8aaa62bc450)).

**UserInternal4InformationNew:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW** structure (see section [2.2.6.25](#Section_b2f614b90312421aabed10ee002ef780)).

**UserInternal5InformationNew:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_USER\_INTERNAL5\_INFORMATION\_NEW** structure (see section [2.2.6.27](#Section_563cc0495e4649a1b0e086efe4bc6e19)).

#### SAMPR\_USER\_INFO\_BUFFER

The SAMPR\_USER\_INFO\_BUFFER union combines all possible structures used in the SamrSetInformationUser, SamrSetInformationUser2, SamrQueryInformationUser, and SamrQueryInformationUser2 methods (see sections [3.1.5.6.5](#Section_538222f71b894811949a0eac62e38dce), [3.1.5.6.4](#Section_99ee9f3943e84bbaac3a82e0c0e0699e), [3.1.5.5.6](#Section_4ad8d54c0d5a4d5a9e9a1bc9ee008d47), and [3.1.5.5.5](#Section_29ab27f661da4c7d863ce228ee798f4d)). For details on each field, see the associated section for the field structure.

1. typedef
2. [switch\_type(USER\_INFORMATION\_CLASS)]
3. union \_SAMPR\_USER\_INFO\_BUFFER {
4. [case(UserGeneralInformation)]
5. SAMPR\_USER\_GENERAL\_INFORMATION General;
6. [case(UserPreferencesInformation)]
7. SAMPR\_USER\_PREFERENCES\_INFORMATION Preferences;
8. [case(UserLogonInformation)]
9. SAMPR\_USER\_LOGON\_INFORMATION Logon;
10. [case(UserLogonHoursInformation)]
11. SAMPR\_USER\_LOGON\_HOURS\_INFORMATION LogonHours;
12. [case(UserAccountInformation)]
13. SAMPR\_USER\_ACCOUNT\_INFORMATION Account;
14. [case(UserNameInformation)]
15. SAMPR\_USER\_NAME\_INFORMATION Name;
16. [case(UserAccountNameInformation)]
17. SAMPR\_USER\_A\_NAME\_INFORMATION AccountName;
18. [case(UserFullNameInformation)]
19. SAMPR\_USER\_F\_NAME\_INFORMATION FullName;
20. [case(UserPrimaryGroupInformation)]
21. USER\_PRIMARY\_GROUP\_INFORMATION PrimaryGroup;
22. [case(UserHomeInformation)]
23. SAMPR\_USER\_HOME\_INFORMATION Home;
24. [case(UserScriptInformation)]
25. SAMPR\_USER\_SCRIPT\_INFORMATION Script;
26. [case(UserProfileInformation)]
27. SAMPR\_USER\_PROFILE\_INFORMATION Profile;
28. [case(UserAdminCommentInformation)]
29. SAMPR\_USER\_ADMIN\_COMMENT\_INFORMATION AdminComment;
30. [case(UserWorkStationsInformation)]
31. SAMPR\_USER\_WORKSTATIONS\_INFORMATION WorkStations;
32. [case(UserControlInformation)]
33. USER\_CONTROL\_INFORMATION Control;
34. [case(UserExpiresInformation)]
35. USER\_EXPIRES\_INFORMATION Expires;
36. [case(UserInternal1Information)]
37. SAMPR\_USER\_INTERNAL1\_INFORMATION Internal1;
38. [case(UserParametersInformation)]
39. SAMPR\_USER\_PARAMETERS\_INFORMATION Parameters;
40. [case(UserAllInformation)]
41. SAMPR\_USER\_ALL\_INFORMATION All;
42. [case(UserInternal4Information)]
43. SAMPR\_USER\_INTERNAL4\_INFORMATION Internal4;
44. [case(UserInternal5Information)]
45. SAMPR\_USER\_INTERNAL5\_INFORMATION Internal5;
46. [case(UserInternal4InformationNew)]
47. SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW Internal4New;
48. [case(UserInternal5InformationNew)]
49. SAMPR\_USER\_INTERNAL5\_INFORMATION\_NEW Internal5New;
50. [case(UserInternal7Information)]
51. SAMPR\_USER\_INTERNAL7\_INFORMATION Internal7;
52. [case(UserInternal8Information)]
53. SAMPR\_USER\_INTERNAL8\_INFORMATION Internal8;
54. } SAMPR\_USER\_INFO\_BUFFER,
55. \*PSAMPR\_USER\_INFO\_BUFFER;

#### SAMPR\_USER\_INTERNAL7\_INFORMATION

The SAMPR\_USER\_INTERNAL7\_INFORMATION structure holds an encrypted password.

This structure is used to carry a new password for a particular account from the client to the server, encrypted in a way that protects such a password from disclosure or tampering while in transit.

1. typedef struct \_SAMPR\_USER\_INTERNAL7\_INFORMATION {
2. SAMPR\_ENCRYPTED\_PASSWORD\_AES UserPassword;
3. BOOLEAN PasswordExpired;
4. } SAMPR\_USER\_INTERNAL7\_INFORMATION,
5. \*PSAMPR\_USER\_INTERNAL7\_INFORMATION;

**UserPassword**: A cleartext password, encrypted as specified in SAMPR\_ENCRYPTED\_PASSWORD\_AES (section [2.2.6.32](#Section_2cbb49a2be3844908be7b0e4951bc221)) and AES Cipher Usage (section [3.2.2.4](#Section_0fc25352662b40e79deef10c6244c4c9)), where the encryption key is the 16-byte SMB session key obtained as specified in either section [3.1.2.4](#Section_c718b655919f43a88897be080553e746) or section [3.2.2.3](#Section_a26ab2cb3e0949809f2d8d75f73005ae).

**PasswordExpired**: Handle expired passwords as specified in section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

#### SAMPR\_USER\_INTERNAL8\_INFORMATION

The SAMPR\_USER\_INTERNAL8\_INFORMATION structure holds all attributes of a user, along with an encrypted password.

1. typedef struct \_SAMPR\_USER\_INTERNAL8\_INFORMATION {
2. SAMPR\_USER\_ALL\_INFORMATION I1;
3. SAMPR\_ENCRYPTED\_PASSWORD\_AES UserPassword;
4. } SAMPR\_USER\_INTERNAL8\_INFORMATION,
5. \*PSAMPR\_USER\_INTERNAL8\_INFORMATION;

**I1**: As specified in section [2.2.6.6](#Section_dc966b81da274daea28cec16534f1cb9).

**UserPassword**: As specified in SAMPR\_ENCRYPTED\_PASSWORD\_AES (section [2.2.6.32](#Section_2cbb49a2be3844908be7b0e4951bc221)).

#### SAMPR\_ENCRYPTED\_PASSWORD\_AES

The SAMPR\_ENCRYPTED\_PASSWORD\_AES structure carries an encrypted string.

1. typedef struct \_SAMPR\_ENCRYPTED\_PASSWORD\_AES {
2. UCHAR AuthData[64];
3. UCHAR Salt[16];
4. ULONG cbCipher;
5. [size\_is(cbCipher)] PUCHAR Cipher;
6. ULONGLONG PBKDF2Iterations;
7. } SAMPR\_ENCRYPTED\_PASSWORD\_AES, \*PSAMPR\_ENCRYPTED\_PASSWORD\_AES;

**AuthData**: An authentication signature HMAC-SHA-512 hash of the value of Cipher+versionbyte+versionbyte\_length as specified in AES Cipher Usage (section [3.2.2.4](#Section_0fc25352662b40e79deef10c6244c4c9)).

**Salt**: A random number used by the client to encrypt the data stored in Cipher with AES and to derive encryption keys with PBKDF2.

**cbCipher**: The size of Cipher in bytes.

**Cipher**: A pointer to a buffer of UCHAR to carry encrypted cleartext password. The encryption key is method-specific, while the algorithm is specified in AES Cipher Usage (section 3.2.2.4) and is common for all methods that use this structure. Refer to the SamrUnicodeChangePasswordUser4 method syntax (section [3.1.5.10.4](#Section_bbc1c5e59b814038b2b9c87d3569ed38)) and the processing instructions in SamrSetInformationUser2 (section [3.1.5.6.4.6](#Section_ce4e6060629d4bedad7ec464f4b989f8)) for encryption key selection specifications.

**Note**: The SamrUnicodeChangePasswordUser4 method does not require a context handle and can be called directly, nor does it return a context handle.

**PBKDF2Iterations**: The number of PBKDF2 Iterations used by the client to derive an encryption key with PBKDF2.

For all protocol uses, the decrypted format of **Cipher** is the following structure.

1. typedef struct \_SAMPR\_USER\_PASSWORD\_AES {
2. USHORT PasswordLength;
3. WCHAR Buffer [SAM\_MAX\_PASSWORD\_LENGTH];
4. } SAMPR\_USER\_PASSWORD\_AES, \*PSAMPR\_USER\_PASSWORD\_AES;

**PasswordLength**: An unsigned short integer value that indicates the length of the cleartext password in bytes.

**Buffer**: A buffer of 512 characters that holds the cleartext value of the password that is **PasswordLength** number of bytes from the beginning of the buffer. The remaining unused portions of the buffer MUST be filled by the client with random bytes.

A decrypted cipher of less than (SAM\_MAX\_PASSWORD\_LENGTH \* sizeof(WCHAR)) + sizeof(USHORT)) bytes will be rejected by the server and a failure status of STATUS\_WRONG\_PASSWORD returned to the client.

### Miscellaneous Protocol-Specific Types

These types are specific to the SAM Remote Protocol (Client-to-Server). Many types are used by multiple methods, while others are used by only one method. This section is useful when used as a reference while reading the method syntax in section [3.1.5](#Section_3f15814e46004647abfd3890f5f3570c).

#### PSAMPR\_SERVER\_NAME

An RPC handle that is represented by a zero-terminated, UTF-16 encoded string. [[UNICODE3.1]](https://go.microsoft.com/fwlink/?LinkId=90551) describes the Unicode encoding.

The string represents the network address of the server.

This type is declared as follows:

1. typedef [handle] wchar\_t\* PSAMPR\_SERVER\_NAME;

#### SAMPR\_HANDLE

An RPC context handle, as specified in [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824) section 6, that is used to share a session between method calls.

This type is declared as follows:

1. typedef [context\_handle] void\* SAMPR\_HANDLE;

For more information on this protocol's usage of RPC context handles, see section [3.1.1.10](#Section_cc2dc6cee45f481aa8ed5d0e273336b3).

#### ENCRYPTED\_LM\_OWF\_PASSWORD, ENCRYPTED\_NT\_OWF\_PASSWORD

The ENCRYPTED\_LM\_OWF\_PASSWORD structure defines a block of encrypted data used in various methods to communicate sensitive information.

1. typedef struct \_ENCRYPTED\_LM\_OWF\_PASSWORD {
2. char data[16];
3. } ENCRYPTED\_LM\_OWF\_PASSWORD,
4. \*PENCRYPTED\_LM\_OWF\_PASSWORD,
5. ENCRYPTED\_NT\_OWF\_PASSWORD,
6. \*PENCRYPTED\_NT\_OWF\_PASSWORD;

**data:**  16 bytes of unstructured data used to hold an encrypted 16-byte hash (either an [**LM hash**](#gt_58a72159-b1e9-4770-9b06-d7a6b16279a8) or an [**NT hash**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f)). The encryption algorithm is specified in section [2.2.11.1](#Section_3f5ec79db4494ab29423c4dccbe0b184). The methods specified in sections [3.1.5.10](#Section_41d7ca60909f4d0db85ac9a35b5f2aaa) and [3.1.5.13.6](#Section_9bcad7d2b8e14d28a03386040b7b3ce9) use this structure and specify the type of hash and the encryption key.

#### SAMPR\_ULONG\_ARRAY

The SAMPR\_ULONG\_ARRAY structure holds a counted array of unsigned long values.

1. typedef struct \_SAMPR\_ULONG\_ARRAY {
2. unsigned long Count;
3. [size\_is(Count)] unsigned long\* Element;
4. } SAMPR\_ULONG\_ARRAY,
5. \*PSAMPR\_ULONG\_ARRAY;

**Count:**  The number of elements in **Element**. If zero, **Element** MUST be ignored. If nonzero, **Element** MUST point to at least Count \* sizeof(unsigned long) bytes of memory.

**Element:**  A pointer to an array of unsigned integers with **Count** elements. The semantic meaning is dependent on the method in which the structure is being used.

#### SAMPR\_PSID\_ARRAY

The SAMPR\_PSID\_ARRAY structure holds an array of [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) values.

1. typedef struct \_SAMPR\_PSID\_ARRAY {
2. [range(0,1024)] unsigned long Count;
3. [size\_is(Count)] PSAMPR\_SID\_INFORMATION Sids;
4. } SAMPR\_PSID\_ARRAY,
5. \*PSAMPR\_PSID\_ARRAY;

**Count:**  The number of elements in the **Sids** field. If zero, **Sids** MUST be ignored. If nonzero, **Sids** MUST point to at least Count \* sizeof(SAMPR\_SID\_INFORMATION) bytes of memory.

**Sids:**  An array of pointers to SID values. For more information, see section [2.2.7.6](#Section_0c138399f6634039b4e7b3c9f82bff65).

#### SAMPR\_SID\_INFORMATION

The SAMPR\_SID\_INFORMATION structure holds a [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) pointer.

1. typedef struct \_SAMPR\_SID\_INFORMATION {
2. PRPC\_SID SidPointer;
3. } SAMPR\_SID\_INFORMATION,
4. \*PSAMPR\_SID\_INFORMATION;

**SidPointer**: A pointer to a **SID** value, as described in [[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2) section 2.4.2.3.

#### SAMPR\_PSID\_ARRAY\_OUT

The SAMPR\_PSID\_ARRAY\_OUT structure holds an array of [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) values.

1. typedef struct \_SAMPR\_PSID\_ARRAY\_OUT {
2. unsigned long Count;
3. [size\_is(Count)] PSAMPR\_SID\_INFORMATION Sids;
4. } SAMPR\_PSID\_ARRAY\_OUT,
5. \*PSAMPR\_PSID\_ARRAY\_OUT;

**Count:**  The number of elements in **Sids**. If zero, **Sids** MUST be ignored. If nonzero, **Sids** MUST point to at least **Count** \* sizeof([SAMPR\_SID\_INFORMATION](#Section_1d40622e52e44aaabc77aa626089f116)) bytes of memory.

**Sids:**  An array of pointers to SID values. For more information, see section 2.2.7.5.

#### SAMPR\_RETURNED\_USTRING\_ARRAY

The SAMPR\_RETURNED\_USTRING\_ARRAY structure holds an array of counted UTF-16 encoded strings.

1. typedef struct \_SAMPR\_RETURNED\_USTRING\_ARRAY {
2. unsigned long Count;
3. [size\_is(Count)] PRPC\_UNICODE\_STRING Element;
4. } SAMPR\_RETURNED\_USTRING\_ARRAY,
5. \*PSAMPR\_RETURNED\_USTRING\_ARRAY;

**Count:**  The number of elements in **Element**. If zero, **Element** MUST be ignored. If nonzero, **Element** MUST point to at least Count \* sizeof(RPC\_UNICODE\_STRING) bytes of memory.

**Element:**  Array of counted strings (see RPC\_UNICODE\_STRING in [[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2) section 2.3.10). The semantic meaning is method-dependent.

#### SAMPR\_RID\_ENUMERATION

The SAMPR\_RID\_ENUMERATION structure holds the name and [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) information about an [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815).

1. typedef struct \_SAMPR\_RID\_ENUMERATION {
2. unsigned long RelativeId;
3. RPC\_UNICODE\_STRING Name;
4. } SAMPR\_RID\_ENUMERATION,
5. \*PSAMPR\_RID\_ENUMERATION;

**RelativeId:**  A RID.

**Name:**  The UTF-16 encoded name of the account that is associated with **RelativeId**.

#### SAMPR\_ENUMERATION\_BUFFER

The SAMPR\_ENUMERATION\_BUFFER structure holds an array of [SAMPR\_RID\_ENUMERATION](#Section_5c94a35ae7f24675af34741f5a8ee1a2) elements.

1. typedef struct \_SAMPR\_ENUMERATION\_BUFFER {
2. unsigned long EntriesRead;
3. [size\_is(EntriesRead)] PSAMPR\_RID\_ENUMERATION Buffer;
4. } SAMPR\_ENUMERATION\_BUFFER,
5. \*PSAMPR\_ENUMERATION\_BUFFER;

**EntriesRead:**  The number of elements in **Buffer**. If zero, **Buffer** MUST be ignored. If nonzero, **Buffer** MUST point to at least EntriesRead \* sizeof(SAMPR\_RID\_ENUMERATION) bytes of memory.

**Buffer:**  An array of SAMPR\_RID\_ENUMERATION elements.

#### SAMPR\_SR\_SECURITY\_DESCRIPTOR

The SAMPR\_SR\_SECURITY\_DESCRIPTOR structure holds a formatted [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350).

1. typedef struct \_SAMPR\_SR\_SECURITY\_DESCRIPTOR {
2. [range(0, 256 \* 1024)] unsigned long Length;
3. [size\_is(Length)] unsigned char\* SecurityDescriptor;
4. } SAMPR\_SR\_SECURITY\_DESCRIPTOR,
5. \*PSAMPR\_SR\_SECURITY\_DESCRIPTOR;

**Length:**  The size, in bytes, of SecurityDescriptor. If zero, SecurityDescriptor MUST be ignored. The maximum size of 256 \* 1024 is an arbitrary value chosen to limit the amount of memory a client can force the server to allocate.

**SecurityDescriptor:**  A binary format per the **SECURITY\_DESCRIPTOR** format in [[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2) section 2.4.6.

#### GROUP\_MEMBERSHIP

The GROUP\_MEMBERSHIP structure holds information on a group membership.

1. typedef struct \_GROUP\_MEMBERSHIP {
2. unsigned long RelativeId;
3. unsigned long Attributes;
4. } GROUP\_MEMBERSHIP,
5. \*PGROUP\_MEMBERSHIP;

**RelativeId:**  A [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) that represents one membership value.

**Attributes:**  Characteristics about the membership represented as a bitmask. Values are defined in section [2.2.1.10](#Section_9e093bd2e4514dd5970097b977d7ebb2).

#### SAMPR\_GET\_GROUPS\_BUFFER

The SAMPR\_GET\_GROUPS\_BUFFER structure represents the members of a group.

1. typedef struct \_SAMPR\_GET\_GROUPS\_BUFFER {
2. unsigned long MembershipCount;
3. [size\_is(MembershipCount)] PGROUP\_MEMBERSHIP Groups;
4. } SAMPR\_GET\_GROUPS\_BUFFER,
5. \*PSAMPR\_GET\_GROUPS\_BUFFER;

**MembershipCount:**  The number of elements in **Groups**. If zero, **Groups** MUST be ignored. If nonzero, **Groups** MUST point to at least MembershipCount \* sizeof(GROUP\_MEMBERSHIP) bytes of memory.

**Groups:**  An array to hold information about the members of the group.

#### SAMPR\_GET\_MEMBERS\_BUFFER

The SAMPR\_GET\_MEMBERS\_BUFFER structure represents the membership of a group.

1. typedef struct \_SAMPR\_GET\_MEMBERS\_BUFFER {
2. unsigned long MemberCount;
3. [size\_is(MemberCount)] unsigned long\* Members;
4. [size\_is(MemberCount)] unsigned long\* Attributes;
5. } SAMPR\_GET\_MEMBERS\_BUFFER,
6. \*PSAMPR\_GET\_MEMBERS\_BUFFER;

**MemberCount:**  The number of elements in **Members** and **Attributes**. If zero, **Members** and **Attributes** MUST be ignored. If nonzero, **Members** and **Attributes** MUST point to at least MemberCount \* sizeof(unsigned long) bytes of memory.

**Members:**  An array of [**RIDs**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c).

**Attributes:**  Characteristics about the membership, represented as a bitmask. Values are defined in section [2.2.1.10](#Section_9e093bd2e4514dd5970097b977d7ebb2).

#### SAMPR\_REVISION\_INFO\_V1

The SAMPR\_REVISION\_INFO\_V1 structure is used to communicate the revision and capabilities of client and server. For more information, see [SamrConnect5](#Section_c842a8970a424ca5a6072afd05271dae).

1. typedef struct \_SAMPR\_REVISION\_INFO\_V1 {
2. unsigned long Revision;
3. unsigned long SupportedFeatures;
4. } SAMPR\_REVISION\_INFO\_V1,
5. \*PSAMPR\_REVISION\_INFO\_V1;

**Revision:**  The revision of the client or server side of this protocol (depending on which side sends the structure). The value MUST be set to 3 and MUST be ignored.

**SupportedFeatures:**  A bit field. When sent from the client, this field MUST be zero and ignored on receipt by the server. When returned from the server, the following fields are handled by the client; all other bits are ignored by the client and MUST be zero when returned from the server.

| Value | Meaning |
| --- | --- |
| 0x00000001 | On receipt by the client, this value, when set, indicates that [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) values returned from the server MUST NOT be concatenated with the [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) to create the SID for the account referenced by the RID. Instead, the client MUST call [SamrRidToSid](#Section_00ff8192a4f645ba9f65917e46b6a693) to obtain the SID. This field can be combined with other bits using a logical OR.  See the product behavior citation at the end of this section for more information (about Windows implementations). |
| 0x00000002 | Reserved. See the product behavior citation at the end of this section for additional details. |
| 0x00000004 | Reserved. See the product behavior citation at the end of this section for additional details. |
| 0x00000010 | On receipt by the client, this value, when set, indicates that the client should use AES Encryption with the SAMPR\_ENCRYPTED\_PASSWORD\_AES structure to encrypt password buffers when sent over the wire. See AES Cipher Usage (section [3.2.2.4](#Section_0fc25352662b40e79deef10c6244c4c9)) and [SAMPR\_ENCRYPTED\_PASSWORD\_AES](#Section_2cbb49a2be3844908be7b0e4951bc221) (section 2.2.6.32). |

The following citation in section [7](#Section_fa61e5fcf8fb4d5b9695c724af0c3829) is relevant to the **SupportedFeatures** field.[<19>](#Appendix_A_19" \o "Product behavior note 19)

#### SAMPR\_REVISION\_INFO

The SAMPR\_REVISION\_INFO union holds revision information structures that are used in the [SamrConnect5](#Section_c842a8970a424ca5a6072afd05271dae) method.

1. typedef
2. [switch\_type(unsigned long)]
3. union {
4. [case(1)]
5. SAMPR\_REVISION\_INFO\_V1 V1;
6. } SAMPR\_REVISION\_INFO,
7. \*PSAMPR\_REVISION\_INFO;

**V1:**  Version 1 revision information, as described in [SAMPR\_REVISION\_INFO\_V1 (section 2.2.7.15)](#Section_963e60b592334669b8a885bf4f0806dc).

#### USER\_DOMAIN\_PASSWORD\_INFORMATION

The USER\_DOMAIN\_PASSWORD\_INFORMATION structure contains [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) fields.

1. typedef struct \_USER\_DOMAIN\_PASSWORD\_INFORMATION {
2. unsigned short MinPasswordLength;
3. unsigned long PasswordProperties;
4. } USER\_DOMAIN\_PASSWORD\_INFORMATION,
5. \*PUSER\_DOMAIN\_PASSWORD\_INFORMATION;

For information on each field, see section [2.2.3.1](#Section_d275ab1910b040e094bb45b7fc130025).

### Selective Enumerate Associated Structures

The structures and fields in this section relate to the following methods:

* [SamrQueryDisplayInformation3](#Section_7cd1c6d088194a5897215be588c1d4ad)
* [SamrQueryDisplayInformation2](#Section_9aa37aa4f77e49b199935c92ba97cd62)
* [SamrQueryDisplayInformation](#Section_c1458942f2d54317a888abd27abad504)
* [SamrGetDisplayEnumerationIndex2](#Section_bd429624f2d547178aa2659952c3e209)
* [SamrGetDisplayEnumerationIndex](#Section_af95abf82e1142c394aaba72f8f96c09)

The model of the methods is for the client to specify an enumeration that indicates the attributes that are to be queried. There is duplication among the structures that contain the attributes. For a description of each attribute that is common among structures, see section [2.2.8.1](#Section_4e123dbfe1014ade9ffff2247ed25a11).

#### Common Selective Enumerate Fields

There are a number of selective enumerate–related structures that use the same fields, as denoted by their field names. This section describes all such fields, and subsequent sections specify the fields in protocol structures. While each structure might have a different subset of these attributes, they all draw from this same set of attributes, detailed as follows.

When specified in a given structure, these fields all contain information about the same user or [**machine account**](#gt_ccf481e1-fb56-43d1-b6c6-0733fb52e001), or group. The selective enumerate methods return an array of structures, thereby returning information about a set of users, machines, or groups.

**AccountControl**: A 32-bit bit field representing the **UserAccountControl** field as described in section [2.2.6.1](#Section_899d68a981454334a52f2fbd92bec94f).

**AccountName**: A counted Unicode string of type RPC\_UNICODE\_STRING. When this field is used with a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a), it represents the **Name** field as described in section [2.2.4.1](#Section_2c93edfeae774f49b849b6435ff63444) (common group fields). Otherwise, this field represents the **UserName** field as described in section 2.2.6.1 (common user fields).

**AdminComment**: A counted Unicode string of type RPC\_UNICODE\_STRING. When this field is used with a group object, it represents the **AdminComment** field as described in section 2.2.4.1 (common group fields). Otherwise, this field represents the **AdminComment** field as described in section 2.2.6.1 (common user fields).

**Attributes**: A 32-bit bit field representing the **Attributes** field, as described in section 2.2.4.1 (common group fields).

**Index**: A 32-bit unsigned integer; see the message processing of [SamrQueryDisplayInformation3 (section 3.1.5.3.1)](#Section_7cd1c6d088194a5897215be588c1d4ad) for details on the semantics of this field.

**Rid**: A 32-bit unsigned integer representing the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of an [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815).

#### SAMPR\_DOMAIN\_DISPLAY\_USER

The SAMPR\_DOMAIN\_DISPLAY\_USER structure contains a subset of user account information sufficient to show a summary of the [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) for an account management application.

1. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_USER {
2. unsigned long Index;
3. unsigned long Rid;
4. unsigned long AccountControl;
5. RPC\_UNICODE\_STRING AccountName;
6. RPC\_UNICODE\_STRING AdminComment;
7. RPC\_UNICODE\_STRING FullName;
8. } SAMPR\_DOMAIN\_DISPLAY\_USER,
9. \*PSAMPR\_DOMAIN\_DISPLAY\_USER;

For information on each field, see section [2.2.8.1](#Section_4e123dbfe1014ade9ffff2247ed25a11).

#### SAMPR\_DOMAIN\_DISPLAY\_MACHINE

The SAMPR\_DOMAIN\_DISPLAY\_MACHINE structure contains a subset of [**machine account**](#gt_ccf481e1-fb56-43d1-b6c6-0733fb52e001) information sufficient to show a summary of the [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) for an account management application.

1. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_MACHINE {
2. unsigned long Index;
3. unsigned long Rid;
4. unsigned long AccountControl;
5. RPC\_UNICODE\_STRING AccountName;
6. RPC\_UNICODE\_STRING AdminComment;
7. } SAMPR\_DOMAIN\_DISPLAY\_MACHINE,
8. \*PSAMPR\_DOMAIN\_DISPLAY\_MACHINE;

For information on each field, see section [2.2.8.1](#Section_4e123dbfe1014ade9ffff2247ed25a11).

#### SAMPR\_DOMAIN\_DISPLAY\_GROUP

The SAMPR\_DOMAIN\_DISPLAY\_GROUP structure contains a subset of group information sufficient to show a summary of the [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) for an account management application.

1. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_GROUP {
2. unsigned long Index;
3. unsigned long Rid;
4. unsigned long Attributes;
5. RPC\_UNICODE\_STRING AccountName;
6. RPC\_UNICODE\_STRING AdminComment;
7. } SAMPR\_DOMAIN\_DISPLAY\_GROUP,
8. \*PSAMPR\_DOMAIN\_DISPLAY\_GROUP;

For information on each field, see section [2.2.8.1](#Section_4e123dbfe1014ade9ffff2247ed25a11).

#### SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER

The SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER structure contains a subset of user account information sufficient to show a summary of the [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) for an account management application. This structure exists to support non–Unicode-based systems.

1. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER {
2. unsigned long Index;
3. RPC\_STRING OemAccountName;
4. } SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER,
5. \*PSAMPR\_DOMAIN\_DISPLAY\_OEM\_USER;

For information on each field, see section [2.2.8.1](#Section_4e123dbfe1014ade9ffff2247ed25a11).

#### SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP

The SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP structure contains a subset of group information sufficient to show a summary of the [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) for an account management application. This structure exists to support non–Unicode-based systems.

1. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP {
2. unsigned long Index;
3. RPC\_STRING OemAccountName;
4. } SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP,
5. \*PSAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP;

For information on each field, see section [2.2.8.1](#Section_4e123dbfe1014ade9ffff2247ed25a11).

#### SAMPR\_DOMAIN\_DISPLAY\_USER\_BUFFER

The SAMPR\_DOMAIN\_DISPLAY\_USER\_BUFFER structure holds an array of [SAMPR\_DOMAIN\_DISPLAY\_USER](#Section_4f5a1a08c75346ee888b39d360022745) elements used to return a list of users through the SamrQueryDisplayInformation family of methods (see section [3.1.5.3](#Section_f1970f00e3d54ec485bc2caf0bf18f6d)).

1. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_USER\_BUFFER {
2. unsigned long EntriesRead;
3. [size\_is(EntriesRead)] PSAMPR\_DOMAIN\_DISPLAY\_USER Buffer;
4. } SAMPR\_DOMAIN\_DISPLAY\_USER\_BUFFER,
5. \*PSAMPR\_DOMAIN\_DISPLAY\_USER\_BUFFER;

**EntriesRead:**  The number of elements in **Buffer**. If zero, **Buffer** MUST be ignored. If nonzero, **Buffer** MUST point to at least **EntriesRead** number of elements.

**Buffer:**  An array of SAMPR\_DOMAIN\_DISPLAY\_USER elements.

#### SAMPR\_DOMAIN\_DISPLAY\_MACHINE\_BUFFER

The SAMPR\_DOMAIN\_DISPLAY\_MACHINE\_BUFFER structure holds an array of [SAMPR\_DOMAIN\_DISPLAY\_MACHINE](#Section_353bfb6107304aee848988c249b1a9e1) elements used to return a list of machine accounts through the SamrQueryDisplayInformation family of methods (see section [3.1.5.3](#Section_f1970f00e3d54ec485bc2caf0bf18f6d)).

1. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_MACHINE\_BUFFER {
2. unsigned long EntriesRead;
3. [size\_is(EntriesRead)] PSAMPR\_DOMAIN\_DISPLAY\_MACHINE Buffer;
4. } SAMPR\_DOMAIN\_DISPLAY\_MACHINE\_BUFFER,
5. \*PSAMPR\_DOMAIN\_DISPLAY\_MACHINE\_BUFFER;

**EntriesRead:**  The number of elements in **Buffer**. If zero, **Buffer** MUST be ignored. If nonzero, **Buffer** MUST point to at least **EntriesRead** number of elements.

**Buffer:**  An array of SAMPR\_DOMAIN\_DISPLAY\_MACHINE elements.

#### SAMPR\_DOMAIN\_DISPLAY\_GROUP\_BUFFER

The SAMPR\_DOMAIN\_DISPLAY\_GROUP\_BUFFER structure holds an array of [SAMPR\_DOMAIN\_DISPLAY\_GROUP](#Section_3dd24be0ed1342f28fba8af4445b4fdc) elements used to return a list of groups through the SamrQueryDisplayInformation family of methods (see section [3.1.5.3](#Section_f1970f00e3d54ec485bc2caf0bf18f6d)).

1. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_GROUP\_BUFFER {
2. unsigned long EntriesRead;
3. [size\_is(EntriesRead)] PSAMPR\_DOMAIN\_DISPLAY\_GROUP Buffer;
4. } SAMPR\_DOMAIN\_DISPLAY\_GROUP\_BUFFER,
5. \*PSAMPR\_DOMAIN\_DISPLAY\_GROUP\_BUFFER;

**EntriesRead:**  The number of elements in **Buffer**. If zero, **Buffer** MUST be ignored. If nonzero, **Buffer** MUST point to at least **EntriesRead** number of elements.

**Buffer:**  An array of SAMPR\_DOMAIN\_DISPLAY\_GROUP elements.

#### SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER\_BUFFER

The SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER\_BUFFER structure holds an array of [SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER](#Section_c8e910bdfeb549a4939e2262fe17ad2b) elements used to return a list of users through the SamrQueryDisplayInformation family of methods (see section [3.1.5.3](#Section_f1970f00e3d54ec485bc2caf0bf18f6d)).

1. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER\_BUFFER {
2. unsigned long EntriesRead;
3. [size\_is(EntriesRead)] PSAMPR\_DOMAIN\_DISPLAY\_OEM\_USER Buffer;
4. } SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER\_BUFFER,
5. \*PSAMPR\_DOMAIN\_DISPLAY\_OEM\_USER\_BUFFER;

**EntriesRead:**  The number of elements in **Buffer**. If zero, **Buffer** MUST be ignored. If nonzero, **Buffer** MUST point to at least **EntriesRead** number of elements.

**Buffer:**  An array of SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER elements.

#### SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP\_BUFFER

The SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP\_BUFFER structure holds an array of [SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP](#Section_5392e27d9f924803b6b70a9f02497443) elements used to return a list of user accounts through the SamrQueryDisplayInformation family of methods (see section [3.1.5.3](#Section_f1970f00e3d54ec485bc2caf0bf18f6d)).

1. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP\_BUFFER {
2. unsigned long EntriesRead;
3. [size\_is(EntriesRead)] PSAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP Buffer;
4. } SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP\_BUFFER,
5. \*PSAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP\_BUFFER;

**EntriesRead:**  The number of elements in **Buffer**. If zero, **Buffer** MUST be ignored. If nonzero, **Buffer** MUST point to at least **EntriesRead** number of elements.

**Buffer:**  An array of SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP elements.

#### DOMAIN\_DISPLAY\_INFORMATION

The DOMAIN\_DISPLAY\_INFORMATION enumeration indicates how to interpret the Buffer parameter for [SamrQueryDisplayInformation](#Section_c1458942f2d54317a888abd27abad504), [SamrQueryDisplayInformation2](#Section_9aa37aa4f77e49b199935c92ba97cd62), [SamrQueryDisplayInformation3](#Section_7cd1c6d088194a5897215be588c1d4ad), [SamrGetDisplayEnumerationIndex](#Section_af95abf82e1142c394aaba72f8f96c09), and [SamrGetDisplayEnumerationIndex2](#Section_bd429624f2d547178aa2659952c3e209). See section [2.2.8.13](#Section_bd189d28f3e1466984d21ecc04340e8b) for the list of the structures that are associated with each enumeration.

1. typedef enum \_DOMAIN\_DISPLAY\_INFORMATION
2. {
3. DomainDisplayUser = 1,
4. DomainDisplayMachine,
5. DomainDisplayGroup,
6. DomainDisplayOemUser,
7. DomainDisplayOemGroup
8. } DOMAIN\_DISPLAY\_INFORMATION,

 \*PDOMAIN\_DISPLAY\_INFORMATION;

**DomainDisplayUser:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_DOMAIN\_DISPLAY\_USER\_BUFFER** structure (see section [2.2.8.7](#Section_453c355ea84d421ebdbcdd7fbfc148a4)).

**DomainDisplayMachine:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_DOMAIN\_DISPLAY\_MACHINE\_BUFFER** structure (see section [2.2.8.8](#Section_18aa8d70157e482eb2e44e6f4dbb76fe)).

**DomainDisplayGroup:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_DOMAIN\_DISPLAY\_GROUP\_BUFFER** structure (see section [2.2.8.9](#Section_ce9c4119c5e446b5a71aec16daf8f457)).

**DomainDisplayOemUser:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER\_BUFFER** structure (see section [2.2.8.10](#Section_0e9f25a9d30540b68b9fa50e58430124)).

**DomainDisplayOemGroup:** Indicates the *Buffer* parameter is to be interpreted as a **SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP\_BUFFER** structure (see section [2.2.8.11](#Section_a549f5f16b2b4eef87122d43d94410d2)).

#### SAMPR\_DISPLAY\_INFO\_BUFFER

The SAMPR\_DISPLAY\_INFO\_BUFFER union is a union of display structures returned by the SamrQueryDisplayInformation family of methods (see section [3.1.5.3](#Section_f1970f00e3d54ec485bc2caf0bf18f6d)). For details on each field, see the associated section for the field structure.

1. typedef
2. [switch\_type(DOMAIN\_DISPLAY\_INFORMATION)]
3. union \_SAMPR\_DISPLAY\_INFO\_BUFFER {
4. [case(DomainDisplayUser)]
5. SAMPR\_DOMAIN\_DISPLAY\_USER\_BUFFER UserInformation;
6. [case(DomainDisplayMachine)]
7. SAMPR\_DOMAIN\_DISPLAY\_MACHINE\_BUFFER MachineInformation;
8. [case(DomainDisplayGroup)]
9. SAMPR\_DOMAIN\_DISPLAY\_GROUP\_BUFFER GroupInformation;
10. [case(DomainDisplayOemUser)]
11. SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER\_BUFFER OemUserInformation;
12. [case(DomainDisplayOemGroup)]
13. SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP\_BUFFER OemGroupInformation;
14. } SAMPR\_DISPLAY\_INFO\_BUFFER,
15. \*PSAMPR\_DISPLAY\_INFO\_BUFFER;

### SamrValidatePassword Data Types

The following structures are used exclusively for the [SamrValidatePassword](#Section_c78a7239f8fc4a42bb71321e897dc046) method.

As stated in section [2.1](#Section_084da2e70ba044fc8f17e8a200c69eb5), all structures SHOULD be encrypted by the client using transport layer security to hide any cleartext data embedded in the structures.

The authentication, password change, and password reset structures (sections [2.2.9.5](#Section_5a666b2ccd214615819b6dbfe7a2b13f), [2.2.9.6](#Section_c2efbd2771664305a33de2519a1dc2cf), and [2.2.9.7](#Section_74ba81886b2447d9837e7210f6dd70a1)) refer to a password-related operation that occurs in an application external to this protocol. A canonical scenario is an application, such as Microsoft SQL Server, that might maintain its own account database (independent of an operating system's account data) and might require that the passwords of those accounts be subject to the same policy as the policy enforced by the server of this protocol (such as [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90)). Such an application uses the SamrValidatePassword method and these structures to accomplish this goal. Said application is also responsible for storing, in whatever manner it chooses, the [SAM\_VALIDATE\_PERSISTED\_FIELDS (section 2.2.9.2)](#Section_e0b2d21d0b1c4fc08f4a895bef6ffc4e) structure returned by SamrValidatePassword.

#### SAM\_VALIDATE\_PASSWORD\_HASH

The SAM\_VALIDATE\_PASSWORD\_HASH structure holds a binary value that represents a cryptographic hash.

1. typedef struct \_SAM\_VALIDATE\_PASSWORD\_HASH {
2. unsigned long Length;
3. [unique, size\_is(Length)] unsigned char\* Hash;
4. } SAM\_VALIDATE\_PASSWORD\_HASH,
5. \*PSAM\_VALIDATE\_PASSWORD\_HASH;

**Length:**  The size, in bytes, of **Hash**. If zero, **Hash** MUST be ignored.

**Hash:**  A binary value.

#### SAM\_VALIDATE\_PERSISTED\_FIELDS

The SAM\_VALIDATE\_PERSISTED\_FIELDS structure holds various characteristics about password state.

1. typedef struct \_SAM\_VALIDATE\_PERSISTED\_FIELDS {
2. unsigned long PresentFields;
3. LARGE\_INTEGER PasswordLastSet;
4. LARGE\_INTEGER BadPasswordTime;
5. LARGE\_INTEGER LockoutTime;
6. unsigned long BadPasswordCount;
7. unsigned long PasswordHistoryLength;
8. [unique, size\_is(PasswordHistoryLength)]
9. PSAM\_VALIDATE\_PASSWORD\_HASH PasswordHistory;
10. } SAM\_VALIDATE\_PERSISTED\_FIELDS,
11. \*PSAM\_VALIDATE\_PERSISTED\_FIELDS;

**PresentFields:**  A bitmask to indicate which of the fields are valid. The following table shows the defined values. If a bit is set, the corresponding field is valid; if a bit is not set, the field is not valid.

| Value | Meaning |
| --- | --- |
| SAM\_VALIDATE\_PASSWORD\_LAST\_SET  0x00000001 | PasswordLastSet |
| SAM\_VALIDATE\_BAD\_PASSWORD\_TIME  0x00000002 | BadPasswordTime |
| SAM\_VALIDATE\_LOCKOUT\_TIME  0x00000004 | LockoutTime |
| SAM\_VALIDATE\_BAD\_PASSWORD\_COUNT  0x00000008 | BadPasswordCount |
| SAM\_VALIDATE\_PASSWORD\_HISTORY\_LENGTH  0x00000010 | PasswordHistoryLength |
| SAM\_VALIDATE\_PASSWORD\_HISTORY  0x00000020 | PasswordHistory |

**PasswordLastSet:**  This field represents the time at which the password was last reset or changed. It uses FILETIME syntax.

**BadPasswordTime:**  This field represents the time at which an invalid password was presented to either a password change request or an authentication request. It uses FILETIME syntax.

**LockoutTime:**  This field represents the time at which the owner of the password data was locked out. It uses FILETIME syntax.

**BadPasswordCount:**  Indicates how many invalid passwords have accumulated (see message processing for details).

**PasswordHistoryLength:**  Indicates how many previous passwords are in the **PasswordHistory** field.

**PasswordHistory:**  An array of hash values representing the previous **PasswordHistoryLength** passwords.

#### SAM\_VALIDATE\_VALIDATION\_STATUS

The SAM\_VALIDATE\_VALIDATION\_STATUS enumeration defines policy evaluation outcomes.

1. typedef enum \_SAM\_VALIDATE\_VALIDATION\_STATUS
2. {
3. SamValidateSuccess = 0,
4. SamValidatePasswordMustChange,
5. SamValidateAccountLockedOut,
6. SamValidatePasswordExpired,
7. SamValidatePasswordIncorrect,
8. SamValidatePasswordIsInHistory,
9. SamValidatePasswordTooShort,
10. SamValidatePasswordTooLong,
11. SamValidatePasswordNotComplexEnough,
12. SamValidatePasswordTooRecent,
13. SamValidatePasswordFilterError
14. } SAM\_VALIDATE\_VALIDATION\_STATUS,
15. \*PSAM\_VALIDATE\_VALIDATION\_STATUS;

**SamValidateSuccess:** Password validation succeeded.

**SamValidatePasswordMustChange:** The password must be changed.

**SamValidateAccountLockedOut:** The account is locked out.

**SamValidatePasswordExpired:** The password has expired.

**SamValidatePasswordIncorrect:** The password is incorrect.

**SamValidatePasswordIsInHistory:** The password is in the password history.

**SamValidatePasswordTooShort:** The password is too short.

**SamValidatePasswordTooLong:** The password is too long.

**SamValidatePasswordNotComplexEnough:** The password is not complex enough.

**SamValidatePasswordTooRecent:** The password was changed recently.

**SamValidatePasswordFilterError:** The password filter failed to validate the password.

See the message processing of [SamrValidatePassword (section 3.1.5.13.7)](#Section_c78a7239f8fc4a42bb71321e897dc046) for the semantic meanings of the enumeration values.

#### SAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG

The SAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG structure holds the output of [SamrValidatePassword](#Section_c78a7239f8fc4a42bb71321e897dc046).

1. typedef struct \_SAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG {
2. SAM\_VALIDATE\_PERSISTED\_FIELDS ChangedPersistedFields;
3. SAM\_VALIDATE\_VALIDATION\_STATUS ValidationStatus;
4. } SAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG,
5. \*PSAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG;

**ChangedPersistedFields:**  The password state that has changed. See section [2.2.9.2](#Section_e0b2d21d0b1c4fc08f4a895bef6ffc4e).

**ValidationStatus:**  The result of the policy evaluation. See section [2.2.9.3](#Section_e0b65a04ca2a458b8e32e5dcb569c919).

#### SAM\_VALIDATE\_AUTHENTICATION\_INPUT\_ARG

The SAM\_VALIDATE\_AUTHENTICATION\_INPUT\_ARG structure holds information about an authentication request.

1. typedef struct \_SAM\_VALIDATE\_AUTHENTICATION\_INPUT\_ARG {
2. SAM\_VALIDATE\_PERSISTED\_FIELDS InputPersistedFields;
3. unsigned char PasswordMatched;
4. } SAM\_VALIDATE\_AUTHENTICATION\_INPUT\_ARG,
5. \*PSAM\_VALIDATE\_AUTHENTICATION\_INPUT\_ARG;

**InputPersistedFields:**  Password state.

**PasswordMatched:**  A nonzero value indicates that a valid password was presented to the change-password request.

#### SAM\_VALIDATE\_PASSWORD\_CHANGE\_INPUT\_ARG

The SAM\_VALIDATE\_PASSWORD\_CHANGE\_INPUT\_ARG structure holds information about a password change request.

1. typedef struct \_SAM\_VALIDATE\_PASSWORD\_CHANGE\_INPUT\_ARG {
2. SAM\_VALIDATE\_PERSISTED\_FIELDS InputPersistedFields;
3. RPC\_UNICODE\_STRING ClearPassword;
4. RPC\_UNICODE\_STRING UserAccountName;
5. SAM\_VALIDATE\_PASSWORD\_HASH HashedPassword;
6. unsigned char PasswordMatch;
7. } SAM\_VALIDATE\_PASSWORD\_CHANGE\_INPUT\_ARG,
8. \*PSAM\_VALIDATE\_PASSWORD\_CHANGE\_INPUT\_ARG;

**InputPersistedFields:**  Password state. See section [2.2.9.2](#Section_e0b2d21d0b1c4fc08f4a895bef6ffc4e).

**ClearPassword:**  The cleartext password of the change-password operation.

**UserAccountName:**  The application-specific logon name of an [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) performing the change-password operation.

**HashedPassword:**  A binary value containing a hashed form of the value contained in the **ClearPassword** field. The structure of this binary value is specified in section [2.2.9.1](#Section_295b336b27c84dfebc18216c13bb6ebc). The hash function used to generate this value is chosen by the client. An example hash function might be MD5 (as specified in [[RFC1321]](https://go.microsoft.com/fwlink/?LinkId=90275)). The server implementation is independent of that choice; that is, through this protocol, the server is exposed to a sequence of bytes formatted per section 2.2.9.1 and is, therefore, not exposed to the hash function chosen by the client. Furthermore, there is no processing by the server that requires knowledge of the specific hash function chosen. Section [2.2.9](#Section_218e3a7a042f4c8097ce3d46c4efa3fe) contains more information about a scenario in which this field is used.

**PasswordMatch:**  A nonzero value indicates that a valid password was presented to the change-password request.

#### SAM\_VALIDATE\_PASSWORD\_RESET\_INPUT\_ARG

The SAM\_VALIDATE\_PASSWORD\_RESET\_INPUT\_ARG structure holds various information about a password reset request.

1. typedef struct \_SAM\_VALIDATE\_PASSWORD\_RESET\_INPUT\_ARG {
2. SAM\_VALIDATE\_PERSISTED\_FIELDS InputPersistedFields;
3. RPC\_UNICODE\_STRING ClearPassword;
4. RPC\_UNICODE\_STRING UserAccountName;
5. SAM\_VALIDATE\_PASSWORD\_HASH HashedPassword;
6. unsigned char PasswordMustChangeAtNextLogon;
7. unsigned char ClearLockout;
8. } SAM\_VALIDATE\_PASSWORD\_RESET\_INPUT\_ARG,
9. \*PSAM\_VALIDATE\_PASSWORD\_RESET\_INPUT\_ARG;

**InputPersistedFields:**  Password state. See section [2.2.9.2](#Section_e0b2d21d0b1c4fc08f4a895bef6ffc4e).

**ClearPassword:**  The cleartext password of the reset-password operation.

**UserAccountName:**  The application-specific logon name of the [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) performing the reset-password operation.

**HashedPassword:**  See the specification for [SAM\_VALIDATE\_PASSWORD\_CHANGE\_INPUT\_ARG (section 2.2.9.6)](#Section_c2efbd2771664305a33de2519a1dc2cf) for the field with the same name.

**PasswordMustChangeAtNextLogon:**  Nonzero indicates that a password change MUST occur before an authentication request can succeed.

**ClearLockout:**  Nonzero indicates that the lockout state is to be reset.

#### PASSWORD\_POLICY\_VALIDATION\_TYPE

The PASSWORD\_POLICY\_VALIDATION\_TYPE enumeration indicates the type of policy validation that is being requested.

1. typedef enum \_PASSWORD\_POLICY\_VALIDATION\_TYPE
2. {
3. SamValidateAuthentication = 1,
4. SamValidatePasswordChange,
5. SamValidatePasswordReset
6. } PASSWORD\_POLICY\_VALIDATION\_TYPE;

**SamValidateAuthentication:** Indicates a request to execute the password policy validation performed at logon.

**SamValidatePasswordChange:** Indicates a request to execute the password policy validation performed during a password change request.

**SamValidatePasswordReset:** Indicates a request to execute the password policy validation performed during a password reset.

#### SAM\_VALIDATE\_INPUT\_ARG

The SAM\_VALIDATE\_INPUT\_ARG union holds the various input types to [SamrValidatePassword (section 3.1.5.13.7)](#Section_c78a7239f8fc4a42bb71321e897dc046).

1. typedef
2. [switch\_type(PASSWORD\_POLICY\_VALIDATION\_TYPE)]
3. union \_SAM\_VALIDATE\_INPUT\_ARG {
4. [case(SamValidateAuthentication)]
5. SAM\_VALIDATE\_AUTHENTICATION\_INPUT\_ARG ValidateAuthenticationInput;
6. [case(SamValidatePasswordChange)]
7. SAM\_VALIDATE\_PASSWORD\_CHANGE\_INPUT\_ARG ValidatePasswordChangeInput;
8. [case(SamValidatePasswordReset)]
9. SAM\_VALIDATE\_PASSWORD\_RESET\_INPUT\_ARG ValidatePasswordResetInput;
10. } SAM\_VALIDATE\_INPUT\_ARG,
11. \*PSAM\_VALIDATE\_INPUT\_ARG;

For more information, see the message processing of SamrValidatePassword.

#### SAM\_VALIDATE\_OUTPUT\_ARG

The SAM\_VALIDATE\_OUTPUT\_ARG union holds the output of [SamrValidatePassword (section 3.1.5.13.7)](#Section_c78a7239f8fc4a42bb71321e897dc046).

1. typedef
2. [switch\_type(PASSWORD\_POLICY\_VALIDATION\_TYPE)]
3. union \_SAM\_VALIDATE\_OUTPUT\_ARG {
4. [case(SamValidateAuthentication)]
5. SAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG ValidateAuthenticationOutput;
6. [case(SamValidatePasswordChange)]
7. SAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG ValidatePasswordChangeOutput;
8. [case(SamValidatePasswordReset)]
9. SAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG ValidatePasswordResetOutput;
10. } SAM\_VALIDATE\_OUTPUT\_ARG,
11. \*PSAM\_VALIDATE\_OUTPUT\_ARG;

For more information, see the message processing of SamrValidatePassword.

### Supplemental Credentials Structures

These structures define the format of the **supplementalCredentials** attribute in [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) that the server of this protocol updates in the [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration. The structures are not part of the SAM Remote Protocol (Client-to-Server) but are listed here in normative detail because the persisted value (in the **supplementalCredentials** attribute) is replicated in Active Directory. See section [3.1.1.8.11](#Section_0705f88862e14a4cbac0b4d427f396f8) for details on how this attribute is updated.

#### USER\_PROPERTIES

The USER\_PROPERTIES structure defines the format of the **supplementalCredentials** attribute.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Reserved1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Length | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Reserved2 | | | | | | | | | | | | | | | | Reserved3 | | | | | | | | | | | | | | | |
| Reserved4 (96 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PropertySignature | | | | | | | | | | | | | | | | PropertyCount (optional) | | | | | | | | | | | | | | | |
| UserProperties (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Reserved5 | | | | | | | |

**Reserved1 (4 bytes):** This value MUST be set to zero and MUST be ignored by the recipient.

**Length (4 bytes):** This value MUST be set to the length, in bytes, of the entire structure, starting from the **Reserved4** field.

**Reserved2 (2 bytes):** This value MUST be set to zero and MUST be ignored by the recipient.

**Reserved3 (2 bytes):** This value MUST be set to zero and MUST be ignored by the recipient.

**Reserved4 (96 bytes):** This value MUST be ignored by the recipient and MAY[<20>](#Appendix_A_20" \o "Product behavior note 20) contain arbitrary values.

**PropertySignature (2 bytes):** This field MUST be the value 0x50, in little-endian byte order. This is an arbitrary value used to indicate whether the structure is corrupt. That is, if this value is not 0x50 on read, the structure is considered corrupt, processing MUST be aborted, and an error code MUST be returned.

**PropertyCount (2 bytes):** The number of [USER\_PROPERTY](#Section_7c0f2eca1783450bb5a0754cf11f22c9) elements in the **UserProperties** field. When there are zero USER\_PROPERTY elements in the **UserProperties** field, this field MUST be omitted; the resultant USER\_PROPERTIES structure has a constant size of 0x6F bytes.

**UserProperties (variable):** An array of **PropertyCount** USER\_PROPERTY elements.

**Reserved5 (1 byte):** This value SHOULD[<21>](#Appendix_A_21" \o "Product behavior note 21) be set to zero and MUST be ignored by the recipient.

#### USER\_PROPERTY

The USER\_PROPERTY structure defines an array element that contains a single property name and value for the **supplementalCredentials** attribute.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| NameLength | | | | | | | | | | | | | | | | ValueLength | | | | | | | | | | | | | | | |
| Reserved | | | | | | | | | | | | | | | | PropertyName (variable) | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| PropertyValue (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**NameLength (2 bytes):** The number of bytes, in little-endian byte order, of **PropertyName**. The property name is located at an offset of zero bytes just following the **Reserved** field. For more information, see the message processing section for [supplementalCredentials (section 3.1.1.8.11)](#Section_0705f88862e14a4cbac0b4d427f396f8).

**ValueLength (2 bytes):** The number of bytes contained in **PropertyValue**.

**Reserved (2 bytes):** This value MUST be ignored by the recipient and MAY[<22>](#Appendix_A_22" \o "Product behavior note 22) be set to arbitrary values on update.

**PropertyName (variable):** The name of this property as a UTF-16 encoded string.

**PropertyValue (variable):** The value of this property. The value MUST be hexadecimal-encoded using an 8-bit character size, and the values '0' through '9' inclusive and 'a' through 'f' inclusive (the specification of 'a' through 'f' is case-sensitive).

#### Primary:WDigest - WDIGEST\_CREDENTIALS

The WDIGEST\_CREDENTIALS structure defines the format of the Primary:WDigest property within the **supplementalCredentials** attribute. This structure is stored as a property value in a [USER\_PROPERTY](#Section_7c0f2eca1783450bb5a0754cf11f22c9) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Reserved1 | | | | | | | | Reserved2 | | | | | | | | Version | | | | | | | | NumberOfHashes | | | | | | | |
| Reserved3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash1 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash2 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash3 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash4 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash5 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash6 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash7 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash8 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash9 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash10 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash11 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash12 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash13 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash14 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash15 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash16 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash17 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash18 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash19 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash20 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash21 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash22 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash23 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash24 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash25 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash26 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash27 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash28 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Hash29 (16 bytes) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Reserved1 (1 byte):** This value MUST be ignored by the recipient and MAY[<23>](#Appendix_A_23" \o "Product behavior note 23) be set to arbitrary values upon an update to the **supplementalCredentials** attribute.

**Reserved2 (1 byte):** This value MUST be ignored by the recipient and MUST be set to zero.

**Version (1 byte):** This value MUST be set to 1.

**NumberOfHashes (1 byte):** This value MUST be set to 29 because there are 29 hashes in the array.

**Reserved3 (12 bytes):** This value MUST be ignored by the recipient and MUST be set to zero.

For information on the **Hash** fields, see section [3.1.1.8.11](#Section_0705f88862e14a4cbac0b4d427f396f8).

#### Primary:Kerberos - KERB\_STORED\_CREDENTIAL

The KERB\_STORED\_CREDENTIAL structure is a variable-length structure that defines the format of the Primary:Kerberos property within the **supplementalCredentials** attribute. For information on how this structure is created, see section [3.1.1.8.11.4](#Section_573b6c4061414a41ba3ffa72cca9d861).

This structure is stored as a property value in a [USER\_PROPERTY](#Section_7c0f2eca1783450bb5a0754cf11f22c9) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Revision | | | | | | | | | | | | | | | | Flags | | | | | | | | | | | | | | | |
| CredentialCount | | | | | | | | | | | | | | | | OldCredentialCount | | | | | | | | | | | | | | | |
| DefaultSaltLength | | | | | | | | | | | | | | | | DefaultSaltMaximumLength | | | | | | | | | | | | | | | |
| DefaultSaltOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Credentials (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OldCredentials (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DefaultSalt (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| KeyValues (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Revision (2 bytes):** This value MUST be set to 3.

**Flags (2 bytes):** This value MUST be zero and ignored on read.

**CredentialCount (2 bytes):** This is the count of elements in the **Credentials** array. This value MUST be set to 2.

**OldCredentialCount (2 bytes):** This is the count of elements in the **OldCredentials** array that contain the keys for the previous password. This value MUST be set to 0 or 2.

**DefaultSaltLength (2 bytes):** The length, in bytes, of a [**salt**](#gt_1672c769-f184-404a-9575-e637fd3a43ed) value.

This value is in little-endian byte order. This value SHOULD be ignored on read.

**DefaultSaltMaximumLength (2 bytes):** The length, in bytes, of the buffer containing the salt value.

This value is in little-endian byte order. This value SHOULD be ignored on read.

**DefaultSaltOffset (4 bytes):** An offset, in little-endian byte order, from the beginning of the attribute value (that is, from the beginning of the **Revision** field of KERB\_STORED\_CREDENTIAL) to where the salt value starts. This value SHOULD be ignored on read.

**Credentials (variable):** An array of **CredentialCount** [KERB\_KEY\_DATA (section 2.2.10.5)](#Section_9921c3edf4704f369edf7afe24c468ec) elements.

**OldCredentials (variable):** An array of **OldCredentialCount** KERB\_KEY\_DATA elements.

**DefaultSalt (variable):** The default salt value.

**KeyValues (variable):** An array of **CredentialCount** + **OldCredentialCount** key values. Each key value MUST be located at the offset specified by the corresponding **KeyOffset** values specified in **Credentials** and **OldCredentials**.

#### KERB\_KEY\_DATA

The KERB\_KEY\_DATA structure holds a cryptographic key. This structure is used in conjunction with [KERB\_STORED\_CREDENTIAL](#Section_866f018086944e14be7edfd91d91694f). For more information, see section [3.1.1.8.11.4](#Section_573b6c4061414a41ba3ffa72cca9d861).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Reserved1 | | | | | | | | | | | | | | | | Reserved2 | | | | | | | | | | | | | | | |
| Reserved3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| KeyType | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| KeyLength | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| KeyOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Reserved1 (2 bytes):** This value MUST be ignored by the recipient and MUST be set to zero.

**Reserved2 (2 bytes):** This value MUST be ignored by the recipient and MUST be set to zero.

**Reserved3 (4 bytes):** This value MUST be ignored by the recipient and MUST be set to zero.

**KeyType (4 bytes):** Indicates the type of key, stored as a 32-bit unsigned integer in little-endian byte order. This MUST be set to one of the following values, which are defined in section [2.2.10.8](#Section_1355fa6bd0974ecc8d5e75b3a6533e04).

| Value | Meaning |
| --- | --- |
| 1 | dec-cbc-crc |
| 3 | des-cbc-md5 |

**KeyLength (4 bytes):** The length, in bytes, of the value beginning at **KeyOffset**. The value of this field is stored in little-endian byte order.

**KeyOffset (4 bytes):** An offset, in little-endian byte order, from the beginning of the property value (that is, from the beginning of the **Revision** field of KERB\_STORED\_CREDENTIAL) to where the key value starts. The key value is the hash value specified according to the **KeyType**.

#### Primary:Kerberos-Newer-Keys - KERB\_STORED\_CREDENTIAL\_NEW

The KERB\_STORED\_CREDENTIAL\_NEW structure is a variable-length structure that defines the format of the Primary:Kerberos-Newer-Keys property within the **supplementalCredentials** attribute. For information on how this structure is created, see section [3.1.1.8.11.6](#Section_f3fd5c777a6343dcac8d598ca374a5a3).

This structure is stored as a property value in a [USER\_PROPERTY](#Section_7c0f2eca1783450bb5a0754cf11f22c9) structure.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Revision | | | | | | | | | | | | | | | | Flags | | | | | | | | | | | | | | | |
| CredentialCount | | | | | | | | | | | | | | | | ServiceCredentialCount | | | | | | | | | | | | | | | |
| OldCredentialCount | | | | | | | | | | | | | | | | OlderCredentialCount | | | | | | | | | | | | | | | |
| DefaultSaltLength | | | | | | | | | | | | | | | | DefaultSaltMaximumLength | | | | | | | | | | | | | | | |
| DefaultSaltOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DefaultIterationCount | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Credentials (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ServiceCredentials (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OldCredentials (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| OlderCredentials (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| DefaultSalt (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| KeyValues (variable) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Revision (2 bytes):** This value MUST be set to 4.

**Flags (2 bytes):** This value MUST be zero and ignored on read.

**CredentialCount (2 bytes):** This is the count of elements in the **Credentials** field.

**ServiceCredentialCount (2 bytes):**  This is the count of elements in the **ServiceCredentials** field. It MUST be zero.

**OldCredentialCount (2 bytes):** This is the count of elements in the **OldCredentials** field that contain the keys for the previous password.

**OlderCredentialCount (2 bytes):**  This is the count of elements in the **OlderCredentials** field that contain the keys for the previous password.

**DefaultSaltLength (2 bytes):** The length, in bytes, of a [**salt**](#gt_1672c769-f184-404a-9575-e637fd3a43ed) value.

This value is in little-endian byte order. This value SHOULD be ignored on read.

**DefaultSaltMaximumLength (2 bytes):** The length, in bytes, of the buffer containing the salt value.

This value is in little-endian byte order. This value SHOULD be ignored on read.

**DefaultSaltOffset (4 bytes):** An offset, in little-endian byte order, from the beginning of the attribute value (that is, from the beginning of the **Revision** field of [KERB\_STORED\_CREDENTIAL](#Section_866f018086944e14be7edfd91d91694f)) to where **DefaultSalt** starts. This value SHOULD be ignored on read.

**DefaultIterationCount (4 bytes):** The default iteration count used to calculate the password hashes.

**Credentials (variable):** An array of **CredentialCount** [KERB\_KEY\_DATA\_NEW (section 2.2.10.7)](#Section_447520a5e1cc48cc8fdcb90db57f7eac) elements.

**ServiceCredentials (variable):** (This field is optional.) An array of **ServiceCredentialCount** KERB\_KEY\_DATA\_NEW elements.

**OldCredentials (variable):** (This field is optional.) An array of **OldCredentialCount** KERB\_KEY\_DATA\_NEW elements.

**OlderCredentials (variable):** (This field is optional.) An array of **OlderCredentialCount** KERB\_KEY\_DATA\_NEW elements.

**DefaultSalt (variable):** The default salt value.

**KeyValues (variable):** An array of **CredentialCount** + **ServiceCredentialCount** + **OldCredentialCount** + **OlderCredentialCount** key values. Each key value MUST be located at the offset specified by the corresponding **KeyOffset** values specified in **Credentials**, **ServiceCredentials**, **OldCredentials**, and **OlderCredentials**.

#### KERB\_KEY\_DATA\_NEW

The KERB\_KEY\_DATA\_NEW structure holds a cryptographic key. This structure is used in conjunction with [KERB\_STORED\_CREDENTIAL\_NEW](#Section_08cb3ca7954b45e3902e77512fe3ba8e). For more information, see section [3.1.1.8.11.6](#Section_f3fd5c777a6343dcac8d598ca374a5a3).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| Reserved1 | | | | | | | | | | | | | | | | Reserved2 | | | | | | | | | | | | | | | |
| Reserved3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| IterationCount | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| KeyType | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| KeyLength | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| KeyOffset | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**Reserved1 (2 bytes):** This value MUST be ignored by the recipient and MUST be set to zero.

**Reserved2 (2 bytes):** This value MUST be ignored by the recipient and MUST be set to zero.

**Reserved3 (4 bytes):** This value MUST be ignored by the recipient and MUST be set to zero.

**IterationCount (4 bytes):**  Indicates the iteration count used to calculate the password hashes.

**KeyType (4 bytes):** Indicates the type of key, stored as a 32-bit unsigned integer in little-endian byte order. This MUST be one of the values listed in section [2.2.10.8](#Section_1355fa6bd0974ecc8d5e75b3a6533e04).

**KeyLength (4 bytes):** The length, in bytes, of the value beginning at **KeyOffset**. The value of this field is stored in little-endian byte order.

**KeyOffset (4 bytes):** An offset, in little-endian byte order, from the beginning of the property value (that is, from the beginning of the **Revision** field of KERB\_STORED\_CREDENTIAL\_NEW) to where the key value starts.

#### Kerberos Encryption Algorithm Identifiers

The following table identifies the various algorithms that can be used in the [KERB\_KEY\_DATA](#Section_9921c3edf4704f369edf7afe24c468ec) and [KERB\_KEY\_DATA\_NEW](#Section_447520a5e1cc48cc8fdcb90db57f7eac) structures.[<24>](#Appendix_A_24" \o "Product behavior note 24)

| Value | Meaning |
| --- | --- |
| 1 | dec-cbc-crc ([[RFC3961]](https://go.microsoft.com/fwlink/?LinkId=90450) section 6.2.3) |
| 3 | des-cbc-md5 ([RFC3961] section 6.2.1) |
| 17 | aes128-cts-hmac-sha1-96 ([[RFC3962]](https://go.microsoft.com/fwlink/?LinkId=90451) section 6) |
| 18 | aes256-cts-hmac-sha1-96 ([RFC3962] section 6) |

#### NTLM-Strong-NTOWF

The NTLM-Strong-NTOWF structure holds a cryptographic key. For more information, see section [3.1.1.8.11.7](#Section_898a260ad6734777a456a181797b7aed).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 1  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 2  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 3  0 | 1 |
| NTLMStrongNTOWF  Reserved2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| … | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| … | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| … | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

**NTLMStrongNTOWF (16 bytes):** Specifies the cryptographic key.

### Common Algorithms

#### DES-ECB-LM

This section specifies an algorithm to encrypt and decrypt [**NT**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f) and [**LM hashes**](#gt_58a72159-b1e9-4770-9b06-d7a6b16279a8) that is used throughout the processing of this protocol. The structure that holds an encrypted hash value is found in section [2.2.7.3](#Section_ce061fef6d4f4802bd5d26b11f14f4a6), which contains references to the methods that use that structure, and therefore specify the encryption key to use for processing.

The base algorithm is the DES [[FIPS46-2]](https://go.microsoft.com/fwlink/?LinkId=89871) in ECB mode [[FIPS81]](https://go.microsoft.com/fwlink/?LinkId=89874). This section specifies how to generate the 64-bit data blocks and 7-byte keys necessary for [FIPS81] from the hash value and the key specified in the referring sections.

For simplicity, this section specifies just the encryption processing. The processing is the same for encryption and decryption; the only exception is when the DES algorithm is invoked in ECB mode. In this case, the implementer MUST specify whether the operation is encryption or decryption. (For more information, see [FIPS81].)

This protocol provides two types of encryption and decryption keys: an unsigned integer and an array of 16 bytes. The exact key is specified in the message processing or syntax sections that reference this section indirectly through section 2.2.7.3.

First, the way to encrypt the hash value is specified, followed by the way to generate the 7-byte keys.

##### Encrypting an NT or LM Hash Value with a Specified Key

This section specifies how to encrypt an [**NT**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f) or [**LM hash**](#gt_58a72159-b1e9-4770-9b06-d7a6b16279a8) (both 16-byte values).

Split the hash value into two blocks, Block1 and Block2. Block1 is the first 8 bytes of the hash (starting from the left); Block2 is the remaining 8 bytes.

Each block is encrypted with a different 7-byte key; call them Key1 and Key2.

If the specified key is an unsigned integer, see section [2.2.11.1.3](#Section_b1b0094f2546431fb06d582158a9f2bb) for the way to derive Key1 and Key2.

If the specified key is a 16-byte value, see section [2.2.11.1.4](#Section_ba4377867de847b4b2695a595c255327) for the way to derive Key1 and Key2.

Let EncryptedBlock1 be the result of applying the algorithm in section [2.2.11.1.2](#Section_ebdb15df8d0d43479d62082e6eccac40) over Block1 with Key1.

Let EncryptedBlock2 be the result of applying the algorithm in section 2.2.11.1.2 over Block2 with Key2.

The encrypted hash value is the concatenation of EncryptedBlock1 and EncryptedBlock2.

See section [4.3](#Section_8c5c143818074d199f4e66290f214a63) for an example.

##### Encrypting a 64-Bit Block with a 7-Byte Key

Transform the 7-byte key into an 8-byte key as follows:

1. Let InputKey be the 7-byte key, represented as a zero-base-index array.
2. Let OutputKey be an 8-byte key, represented as a zero-base-index array.
3. Let OutputKey be assigned as follows.
4. OutputKey[0] = InputKey[0] >> 0x01;
5. OutputKey[1] = ((InputKey[0]&0x01)<<6) | (InputKey[1]>>2);
6. OutputKey[2] = ((InputKey[1]&0x03)<<5) | (InputKey[2]>>3);
7. OutputKey[3] = ((InputKey[2]&0x07)<<4) | (InputKey[3]>>4);
8. OutputKey[4] = ((InputKey[3]&0x0F)<<3) | (InputKey[4]>>5);
9. OutputKey[5] = ((InputKey[4]&0x1F)<<2) | (InputKey[5]>>6);
10. OutputKey[6] = ((InputKey[5]&0x3F)<<1) | (InputKey[6]>>7);
11. OutputKey[7] = InputKey[6] & 0x7F;

The 7-byte InputKey is expanded to 8 bytes by inserting a 0-bit after every seventh bit.

1. for( int i=0; i<8; i++ )
2. {
3. OutputKey[i] = (OutputKey[i] << 1) & 0xfe;
4. }
5. Let the least-significant bit of each byte of OutputKey be a parity bit. That is, if the sum of the preceding seven bits is odd, the eighth bit is 0; otherwise, the eighth bit is 1. The processing starts at the leftmost bit of OutputKey.

Use [[FIPS81]](https://go.microsoft.com/fwlink/?LinkId=89874) to encrypt the 64-bit block using OutputKey. If the higher-level operation is decryption instead of encryption, this is the point at which an implementer MUST specify the decryption intent to [FIPS81].

##### Deriving Key1 and Key2 from a Little-Endian, Unsigned Integer Key

Let I be the little-endian, unsigned integer.

Let I[X] be the Xth byte of I, where I is interpreted as a zero-base-index array of bytes. Note that because I is in little-endian byte order, I[0] is the least significant byte.

Key1 is a concatenation of the following values: I[0], I[1], I[2], I[3], I[0], I[1], I[2].

Key2 is a concatenation of the following values: I[3], I[0], I[1], I[2], I[3], I[0], I[1].

##### Deriving Key1 and Key2 from a 16-Byte Key

Let Key1 be the first 7 bytes of the 16-byte key.

Let Key2 be the next 7 bytes of the 16-byte value. For example, consider a zero-base-index array of 16 bytes called KeyArray that contains the 16-byte key. Key2 is composed of the bytes KeyArray[7] through KeyArray[13], inclusive.

**Note**  A consequence of this derivation is that the fifteenth and sixteenth bytes are ignored.

## Directory Service Schema Elements

This protocol is part of the [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) core family of protocols. In order to be fully compliant with Active Directory, an implementation of this protocol MUST be used in conjunction with the full Active Directory schema, containing all the schema attributes and classes specified in [[MS-ADA1]](%5bMS-ADA1%5d.pdf#Section_19528560f41e4623a406dabcfff0660f), [[MS-ADA2]](%5bMS-ADA2%5d.pdf#Section_e20ebc4e528540bab3bdffcb81c2783e), [[MS-ADA3]](%5bMS-ADA3%5d.pdf#Section_4517e8353ee644d4bb95a94b6966bfb0), and [[MS-ADSC]](%5bMS-ADSC%5d.pdf#Section_9abb5e97123d4da99557b353ab79b830).

# Protocol Details

## Server Details

This protocol enables create, read, update, and delete semantics over an account domain, as described in [[MS-AUTHSOD]](%5bMS-AUTHSOD%5d.pdf#Section_953d700a57cb4cf7b0c3a64f34581cc9) section 1.1.1.5. Five abstract objects are exposed through this protocol: server, [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca), group, alias, and user. User, group, and [**alias objects**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0) can be created and deleted; all objects can be updated and read.

This specification uses the [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) data model, as specified in the entire document of [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a), for the server of this protocol. The attribute names specified in this section are normative for the [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration. Section [3.1.1](#Section_814e12f610374b648d288f1b899dc57f) contains a brief overview of that data model that is relevant to this protocol.

Because the behavior of this protocol is very similar between the DC and non-DC configurations, the Active Directory data model is also used for the non-DC configuration. However, when implementing this protocol for the non-DC scenario, the names of attributes in the data model are not normative. For example, it is conceivable that the backing store in a non-DC configuration could be a text file written and read solely by the server of this protocol.

### Abstract Data Model

In the [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration, this protocol operates over a directory database that is composed of a set of named objects. The name format is an X.501 name [[X501]](https://go.microsoft.com/fwlink/?LinkId=98847); therefore, the objects are arranged in a hierarchy by name. Each object's name MUST be unique within the directory. In a non-DC configuration, the name format of X.501 is not normative; this specification assumes that the format is X.501 for consistency between the two configurations. This protocol is based largely on the use of RPC context handles to maintain session state between the client and server. The basic context-handle programming model is described in [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824) section 6.1.6. In the Security Account Manager (SAM) Remote Protocol (Client-to-Server), for the context handles that have been returned to clients, the server MUST maintain information that maps those handles to the internal objects they represent.

Each object possesses a collection of attributes. Attributes can be multivalued. Each attribute is identified by a value called ldapDisplayName. For example, the X.501 name of the object is a single-valued attribute with the ldapDisplayName: distinguishedName. This specification describes the constraints on the attributes for behaviors relevant to this protocol. For the DC configuration, [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 3.1.1.5 contains additional constraints.

Objects are retrieved from the directory database by specifying attribute-value constraints that the object's attributes (and their values) MUST satisfy. Attribute values are updated by identifying the target object by **distinguishedName** and specifying the new set of attribute-value pairs. Section [3.1.1.3](#Section_2a79a0a80d564b5bbee5c8df393cde57) and section [3.1.1.4](#Section_a4ebabe2e00e49e7b8315a5eb345eb94) contain a list of the [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) attributes and classes relevant to this protocol.

Implementations MUST support creating, reading, updating, and deleting multiple objects, attributes, and attribute values with [**ACID**](#gt_44f0cb35-f618-456d-b7b1-25213fed1045) (atomic, consistent, isolated, and durable) properties [GRAY]. Such an update is referred to as a transaction in this specification.

A [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17) refers to a [**database object**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f) whose **objectClass** attribute is user or derived from user.

A computer object refers to a database object whose **objectClass** attribute is computer or derived from computer.

A [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) refers to a database object whose **objectClass** attribute is group or derived from group, and whose **groupType** contains GROUP\_TYPE\_ACCOUNT\_GROUP or GROUP\_TYPE\_UNIVERSAL\_GROUP.

An [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0) refers to a database object whose **objectClass** attribute is group or derived from group, and whose **groupType** contains GROUP\_TYPE\_RESOURCE\_GROUP.

Two [**domains**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) are exposed from a given server: an account domain and a [**built-in domain**](#gt_22904540-f2fb-470e-af48-ce1b0e9a02b8); this fact is true for both DC and non-DC configurations. The account domain refers to the object with **objectClass** domainDNS. The built-in domain refers to the object with the **objectClass** builtinDomain. The built-in domain has the characteristic that its **objectSid** value is invariant (S-1-5-32) through all deployments and only contains aliases. There is exactly one built-in domain for every account domain. When opening a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1) (through [SamrOpenDomain (section 3.1.5.1.5)](#Section_ba710c905b1242f89e5ad4aacc1329fa)) a client selects the domain to open based on the DomainId parameter. A domain can be in either [**mixed mode**](#gt_06c1c70e-f2c6-4efd-bff8-474409e69660) or [**native mode**](#gt_8adcb510-b25d-459e-9832-90079803d3c0), as specified in [MS-ADTS] section 6.1.4.1.

Domain object refers to either the account domain or the built-in domain.

Server object refers to the single object in the account domain with the samServer objectClass.

The following sections normatively describe the database constraints and triggers required for the message processing of this protocol.

* Constraints are relationships between attributes that MUST be satisfied for a database update to be successful. The constraints are specified in section [3.1.1.6](#Section_e270cd0a529541be9e892c3dc3a39536).
* Triggers are actions that MUST be executed for a database update to be successful. An attribute-scoped trigger is a trigger that is executed when a particular attribute is updated. The attribute-scoped triggers are specified in section [3.1.1.8](#Section_90f7a608d9f14831b2feb9a50cf26aec).

The methods that make up this RPC interface MUST all return STATUS\_SUCCESS (0x00000000) on success. Error statuses (also called error codes) generated by a failure to comply with a constraint are in the NTSTATUS space (a long data type), as specified in [[MS-ERREF]](%5bMS-ERREF%5d.pdf#Section_1bc92ddfb79e413cbbaa99a5281a6c90) section 2.3. Unless specifically called out, error codes are returned to the client of the protocol and are not handled by any special processing at the client; therefore, the exact error code is implementation-specific. Cases in which the client might handle a specific error code are called out. The set of such error codes is found in section [2.2.1.15](#Section_7440cfac6052492584e4c32e417de300).

#### String Handling

The data model for storing an attribute of syntax "string" is a UTF-16 encoded string not including the terminating null character. In this protocol, a string is represented within an RPC\_UNICODE\_STRING structure, which is a counted string.

When a string to be stored in the database arrives through this protocol, it MUST be processed such that the database attribute is updated with RPC\_UNICODE\_STRING.Length bytes of RPC\_UNICODE\_STRING.Buffer.

When a database attribute is to be returned as an RPC\_UNICODE\_STRING via this protocol, RPC\_UNICODE\_STRING.Length MUST be the count of bytes stored in the database for that attribute, and RPC\_UNICODE\_STRING.Buffer MUST contain the database value for that attribute.

In addition, when receiving an RPC\_UNICODE\_STRING or [RPC\_STRING](#Section_28330caf78c449e091c146bbfeed633b), if the **Length** field is nonzero and the **Buffer** field is NULL, an error MUST be returned.

#### String Matching

When string matching is required by the message processing (for example, when processing a [SamrCreateGroupInDomain](#Section_175c1cf94fa248379e5bbb1f0f950bee) method and the data model checks for uniqueness of the name property), the following string matching rules apply:

1. On a [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration, refer to [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 6.5 for how strings are compared.
2. When comparing two strings on a non-DC configuration, they MUST be compared in a case-insensitive manner by transforming them to uppercase, per [[UNICODE3.1]](https://go.microsoft.com/fwlink/?LinkId=90551), and then performing a byte-comparison on their values.

#### Attribute Listing

The following attributes are referenced by this protocol (listed by ldapDisplayName). For a normative description of the syntax, see [[MS-ADA1]](%5bMS-ADA1%5d.pdf#Section_19528560f41e4623a406dabcfff0660f), [[MS-ADA2]](%5bMS-ADA2%5d.pdf#Section_e20ebc4e528540bab3bdffcb81c2783e), and [[MS-ADA3]](%5bMS-ADA3%5d.pdf#Section_4517e8353ee644d4bb95a94b6966bfb0).

* accountExpires
* badPasswordTime
* badPwdCount
* codePage
* countryCode
* dBCSPwd
* description
* displayName
* domainReplica
* forceLogoff
* groupType
* homeDirectory
* homeDrive
* memberOf
* lastLogoff
* lastLogon
* lmPwdHistory
* lockOutObservationWindow
* lockoutDuration
* lockoutThreshold
* lockoutTime
* logonCount
* logonHours
* maxPwdAge
* member
* minPwdAge
* minPwdLength
* mS-DS-CreatorSID
* mS-DS-MachineAccountQuota
* msDS-LockoutObservationWindow
* msDS-LockoutDuration
* msDS-LockoutThreshold
* msDS-MaximumPasswordAge
* msDS-MinimumPasswordAge
* msDS-MinimumPasswordLength
* msDS-PasswordComplexityEnabled
* msDS-PasswordHistoryLength
* msDS-PasswordReversibleEncryptionEnabled
* ntPwdHistory
* nTSecurityDescriptor
* objectClass
* objectSid
* oEMInformation
* primaryGroupID
* profilePath
* pwdHistoryLength
* pwdLastSet
* pwdProperties
* rIDAllocationPool
* rIDPreviousAllocationPool
* rIDSetReferences
* sAMAccountName
* sAMAccountType
* scriptPath
* serverState
* supplementalCredentials
* uASCompat
* unicodePwd
* userAccountControl
* comment
* userParameters
* userWorkstations
* objectClass
* clearTextPassword\*

\*This attribute is not directly persisted. It has triggers that are applied when an update occurs that, in turn, can update other attributes. As such, it is not found in the [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) schema.

#### Object Class List

The following classes are referenced by this protocol (listed by ldapDisplayName). For a normative description of these classes, see [[MS-ADSC]](%5bMS-ADSC%5d.pdf#Section_9abb5e97123d4da99557b353ab79b830).

* user
* computer
* domainDNS
* samServer
* builtinDomain
* group

#### Password Settings Attributes for Originating Update Constraints

The following computed attributes are defined for each [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17). These attributes are read-only.

Effective-LockoutObservationWindow: A 64-bit value with [**delta time**](#gt_8a861b7f-5224-44ab-8de7-18ab8f3e800d) syntax, indicating the time period in which bad password attempts are counted without resetting the count to zero.

Effective-LockoutDuration: A 64-bit value with delta time syntax, indicating the duration for which an [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) is locked out before being automatically reset to an unlocked state.

Effective-LockoutThreshold: A 16-bit unsigned integer indicating the number of bad password attempts within an Effective-LockoutObservationWindow that will cause an account to be locked out.

Effective-MaximumPasswordAge: A 64-bit value with delta time syntax, indicating the policy setting for the maximum time allowed before a password reset or change is required.

Effective-MinimumPasswordAge: A 64-bit value with delta time syntax, indicating the policy setting for the minimum time allowed before a password change operation is allowed.

Effective-MinimumPasswordLength: A 16-bit unsigned integer indicating the policy setting for the minimum number of characters allowed in a password.

Effective-PasswordComplexityEnabled: A Boolean value indicating that password complexity rules (as defined in section [3.1.1.7.1](#Section_bbee51353db24824873a55104c3610ad)) are enabled for the user.

Effective-PasswordHistoryLength: A 16-bit unsigned integer indicating the policy setting for the password history length.

Effective-PasswordReversibleEncryptionEnabled: A Boolean value indicating that the user's cleartext password is to be stored in the **supplementalCredentials** attribute, as defined in section [3.1.1.8.11](#Section_0705f88862e14a4cbac0b4d427f396f8).

The values for these attributes on user objects are computed according to the following algorithm:

1. If the server is in a [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration and the msDS-ResultantPSO computed attribute (as specified in [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 3.1.1.4.5.36) on the user object has value O, values are calculated as follows using attribute values on object O:[<25>](#Appendix_A_25" \o "Product behavior note 25)
   1. Effective-LockoutObservationWindow = msDS-LockoutObservationWindow
   2. Effective-LockoutDuration = msDS-LockoutDuration
   3. Effective-LockoutThreshold = msDS-LockoutThreshold
   4. Effective-MaximumPasswordAge = msDS-MaximumPasswordAge
   5. Effective-MinimumPasswordAge = msDS-MinimumPasswordAge
   6. Effective-MinimumPasswordLength = msDS-MinimumPasswordLength
   7. Effective-PasswordComplexityEnabled = msDS-PasswordComplexityEnabled
   8. Effective-PasswordHistoryLength = msDS-PasswordHistoryLength
   9. Effective-PasswordReversibleEncryptionEnabled = true if either of the following is true:
      * The value of msDS-PasswordReversibleEncryptionEnabled is true.
      * pwdProperties on the [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1) contains DOMAIN\_PASSWORD\_STORE\_CLEARTEXT.

Otherwise, false.

1. Otherwise, values are calculated as follows using attribute values on the domain object:
   1. Effective-LockoutObservationWindow = lockOutObservationWindow on the domain object.
   2. Effective-LockoutDuration = lockoutDuration on the domain object.
   3. Effective-LockoutThreshold = lockoutThreshold on the domain object.
   4. Effective-MaximumPasswordAge = maxPwdAge on the domain object.
   5. Effective-MinimumPasswordAge = minPwdAge on the domain object.
   6. Effective-MinimumPasswordLength = minPwdLength on the domain object.
   7. Effective-PasswordComplexityEnabled = true if pwdProperties on the domain object contains DOMAIN\_PASSWORD\_COMPLEX; otherwise, false.
   8. Effective-PasswordHistoryLength = pwdHistoryLength on the domain object.
   9. Effective-PasswordReversibleEncryptionEnabled = true if pwdProperties on the domain object contains DOMAIN\_PASSWORD\_STORE\_CLEARTEXT; otherwise, false.

#### Attribute Constraints for Originating Updates

The following attribute constraints MUST be enforced during originating updates to the database.

The term "previous" refers to the value at the beginning of the transaction before any updates occurred. Unless otherwise specified, other attributes referenced for a particular constraint refer to the attribute on the same object as the attribute whose constraint is currently being satisfied. An exception to this rule is for [Password Settings Attributes (section 3.1.1.5)](#Section_6da47028e678475eb015a5a15ee98212).

Unless specifically called out, all failure codes are implementation-specific.

A client implementation MUST treat all failure codes as complete failures of the requested operation unless explicitly noted in this section. The possible status codes used for these explicit return codes are found in section [2.2.1.15](#Section_7440cfac6052492584e4c32e417de300).

1. lockOutObservationWindow MUST be greater than or equal to lockoutDuration; on error, return a failure code. "Greater than", in this context, means a smaller absolute value because both are negative (see the next two constraints).
2. lockOutObservationWindow MUST be less than or equal to 0; on error, return a failure code.
3. lockoutDuration MUST be less than or equal to 0; on error, return a failure code.
4. maxPwdAge MUST be less than or equal to 0; on error, return a failure code.
5. minPwdAge MUST be less than or equal to 0; on error, return a failure code.
6. minPwdLength MUST be less than or equal to 256 unless uASCompat is nonzero, in which case minPwdLength MUST be less than or equal to 14; on error, return a failure code.
7. pwdHistoryLength MUST be less than or equal to 1024; on error, return a failure code.
8. sAMAccountName MUST contain at least one non-blank character; on error, return a failure code.
9. sAMAccountName MUST NOT end with a '.' (period) character; on error, return a failure code.
10. sAMAccountName MUST NOT contain any of the following characters (shown here as the binary values of UTF-16 encoded characters):

Characters 0x0000 through 0x001F, inclusive, and the characters in the following table.

| Hexadecimal value | Character encoded |
| --- | --- |
| 0x0022 | " |
| 0x002F | / |
| 0x005C | \ |
| 0x005B | [ |
| 0x005D | ] |
| 0x003A | : |
| 0x007C | | |
| 0x003C | < |
| 0x003E | > |
| 0x002B | + |
| 0x003D | = |
| 0x003B | ; |
| 0x003F | ? |
| 0x002C | , |
| 0x002A | \* |

On error, return a failure code.

1. sAMAccountName MUST contain less than or equal to 20 characters if the object's objectClass is user; on error, return a failure code.
2. sAMAccountName MUST contain less than or equal to 256 characters if the object's objectClass is group; on error, return a failure code.
3. sAMAccountName MUST be the value "krbtgt" (UTF-16 encoded) if the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of the objectSid attribute is DOMAIN\_USER\_RID\_KRBTGT; on error, return a failure code.
4. accountExpires MUST be equal to 0 if the RID of the objectSid attribute value is DOMAIN\_USER\_RID\_ADMIN; on error, return a failure code.
5. logonHours MUST conform to the binary structure of [SAMPR\_LOGON\_HOURS (section 2.2.6.5)](#Section_d83c356b7dda409682705c581f84a4d9), and SAMPR\_LOGON\_HOURS.UnitsPerWeek MUST be less than or equal to 10080.
6. userWorkstations MUST conform to the following constraints, with the value interpreted as a UTF-16 encoded string:
   1. The string MUST be composed of substrings separated by a ',' (comma) character; therefore, a substring cannot contain a comma character. Specifically:
      1. If no comma is present, there is one substring, and it is equal to the string itself.
      2. A comma MUST NOT be the first or final character in the value.
      3. If a comma is present, the first substring MUST be the characters starting from the start of the value to the character just preceding the first comma; the final substring MUST be the characters starting just after the final comma to the final character in the string.
   2. Each substring MUST be less than or equal to 256 characters.
   3. Each substring MUST satisfy at least one of the following conditions:
      1. Satisfy the DNS naming syntax for a full DNS host name, as specified in [[RFC1123]](https://go.microsoft.com/fwlink/?LinkId=90268) section 2.1.
      2. Have a length greater than 1 character and less than or equal to 20 characters, not have a leading or trailing blank character (0x0020), and not contain any of the following characters:

Characters of the value 0x0000 through 0x001F, inclusive, and the characters in the following table.

| Hexadecimal value | Character encoded |
| --- | --- |
| 0x0022 | " |
| 0x002F | / |
| 0x005C | \ |
| 0x005B | [ |
| 0x005D | ] |
| 0x003A | : |
| 0x007C | | |
| 0x003C | < |
| 0x003E | > |
| 0x002B | + |
| 0x003D | = |
| 0x003B | ; |
| 0x003F | ? |
| 0x002C | , |
| 0x002A | \* |

* 1. Any processing error or constraint violation MUST return a failure code.

1. primaryGroupId MUST be equal to DOMAIN\_GROUP\_RID\_CONTROLLERS if userAccountControl contains the bit UF\_SERVER\_TRUST\_ACCOUNT; on error, return a failure code.
2. userAccountControl MUST contain only the following bits, as defined in section [2.2.1.13](#Section_10bf6c8e34af4cf98dff6b6330922863). Note that constraints in this section further limit the possible variations that are legal.

| Bits |
| --- |
| UF\_ACCOUNTDISABLE |
| UF\_HOMEDIR\_REQUIRED |
| UF\_PASSWD\_NOTREQD |
| UF\_ENCRYPTED\_TEXT\_PASSWORD\_ALLOWED |
| UF\_NORMAL\_ACCOUNT |
| UF\_INTERDOMAIN\_TRUST\_ACCOUNT |
| UF\_WORKSTATION\_TRUST\_ACCOUNT |
| UF\_SERVER\_TRUST\_ACCOUNT |
| UF\_DONT\_EXPIRE\_PASSWD |
| UF\_MNS\_LOGON\_ACCOUNT |
| UF\_SMARTCARD\_REQUIRED |
| UF\_TRUSTED\_FOR\_DELEGATION |
| UF\_NOT\_DELEGATED |
| UF\_USE\_DES\_KEY\_ONLY |
| UF\_DONT\_REQUIRE\_PREAUTH |
| UF\_TRUSTED\_TO\_AUTHENTICATE\_FOR\_DELEGATION |
| UF\_NO\_AUTH\_DATA\_REQUIRED |
| UF\_PARTIAL\_SECRETS\_ACCOUNT |
| UF\_USE\_AES\_KEYS |

1. userAccountControl MUST contain one and only one of the following bits, as defined in section 2.2.1.13; on error, return a failure code.

| Bits |
| --- |
| UF\_NORMAL\_ACCOUNT |
| UF\_INTERDOMAIN\_TRUST\_ACCOUNT |
| UF\_WORKSTATION\_TRUST\_ACCOUNT |
| UF\_SERVER\_TRUST\_ACCOUNT |

1. An existing userAccountControl attribute SHOULD NOT be modified such that the UF\_WORKSTATION\_TRUST\_ACCOUNT bit is removed and the UF\_NORMAL\_ACCOUNT bit is added, or vice-versa; on error, return a failure code. This modification, however, MUST be allowed if the client is a member of the Domain Administrators group.[<26>](#Appendix_A_26" \o "Product behavior note 26)
2. userAccountControl MUST NOT contain the UF\_ACCOUNTDISABLE bit if the RID of objectSid has the value DOMAIN\_USER\_RID\_ADMIN or DOMAIN\_USER\_RID\_KRBTGT; on error, return a failure code.
3. objectClass MUST be of type computer or derived from computer if userAccountControl contains the following bit: UF\_SERVER\_TRUST\_ACCOUNT.
4. unicodePwd MUST be exactly 16 bytes in length or not present.
5. dBCSPwd MUST be exactly 16 bytes in length or not present.
6. lmPwdHistory MUST have the following binary format:
   1. The length MUST be a multiple of 16 bytes.
   2. If a value is present, the first 16 bytes MUST be equal to the current value of dBCSPwd.
7. ntPwdHistory MUST have the following binary format:
   1. The length MUST be a multiple of 16 bytes.
   2. If a value is present, the first 16 bytes MUST be equal to the current value of unicodePwd.
8. groupType MUST contain only bits specified in section [2.2.1.11](#Section_1f8d7ea1fcc14833839af94d67c08fcd).
9. groupType MUST NOT contain GROUP\_TYPE\_UNIVERSAL if the account domain is in [**mixed mode**](#gt_06c1c70e-f2c6-4efd-bff8-474409e69660).
10. groupType MUST NOT be changed after it has been added if the account domain is in mixed mode.

#### Additional Update Constraints

The following constraints are referenced from other constraints or triggers.

##### General Password Policy

This policy is referenced from the dBCSPwd and unicodePwd triggers.

The following constraints MUST be satisfied; on error, the server MUST return a processing error. For more information on error codes, see section [3.1.5](#Section_3f15814e46004647abfd3890f5f3570c).

1. Minimum Password Length Constraint: If all of the following conditions are true, the following constraint MUST be satisfied:
   1. Conditions:
      1. The **userAccountControl** attribute value contains UF\_NORMAL\_ACCOUNT.
      2. The **objectSid** attribute value does not have the DOMAIN\_USER\_RID\_KRBTGT value as the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c).
      3. The **userAccountControl** attribute value does NOT contain UF\_PASSWD\_NOTREQD.
      4. The Effective-MinimumPasswordLength attribute value (see section [3.1.1.5](#Section_6da47028e678475eb015a5a15ee98212)) is greater than 0.
      5. The requesting protocol message is a password change (as compared to a password set).
   2. Constraint:
      1. At least one of dBCSPwd or unicodePwd MUST be nonzero-length and equal to a value other than the hash of a zero-length string.
2. Minimum Password Age Constraint: If all of the following conditions are true, the following constraint MUST be satisfied:
   1. Conditions:
      1. The **userAccountControl** attribute contains UF\_NORMAL\_ACCOUNT.
      2. At least one of the **dBCSPwd** or **unicodePwd** attribute values is present and not equal to a hash value of a zero-length string.
   2. Constraint:
      1. The **pwdLastSet** attribute MUST be less than the current time plus the value of the Effective-MinimumPasswordAge attribute (see section 3.1.1.5).
3. Password History Length Constraint: If all of the following conditions are true, the following constraints MUST be satisfied:
   1. Conditions:
      1. The **userAccountControl** attribute contains UF\_NORMAL\_ACCOUNT.
      2. objectSid does not have the DOMAIN\_USER\_RID\_KRBTGT value as the RID.
      3. userAccountControl does NOT contain UF\_PASSWD\_NOTREQD.
      4. minPwdHistory on the [**account domain object**](#gt_3ff707e8-0783-4b16-9370-2899f1078c87) is greater than 0.
      5. The requesting protocol message is a password change (as compared to a password set).
   2. Constraints:
      1. If the **unicodePwd** attribute is being updated, the value of the unicodePwd MUST NOT be present in the first N hashes stored in the ntPwdHistory attribute value, where N is the value of the Effective-PasswordHistoryLength attribute (see section 3.1.1.5). For details on how ntPwdHistory is maintained, see section [3.1.1.9.1](#Section_d0047e7b77bf428691a1648edb2405de).
      2. If the **dBCSPwd** attribute is being updated, the value of the **dBCSPwd** MUST NOT be present in the first N hashes stored in the lmPwdHistory attribute value, where N is the value of the Effective-PasswordHistoryLength attribute (see section 3.1.1.5). For details on how lmPwdHistory is maintained, see section 3.1.1.9.1.

##### Cleartext Password Policy

This constraint is referenced when a cleartext password is updated.

The following constraints MUST be satisfied; on error, the server MUST return a processing error. For more information on error codes, see section [3.1.5](#Section_3f15814e46004647abfd3890f5f3570c).

1. The value MUST be interpreted as a UTF-16 encoded string. If the length of the value is an odd-byte count, ignore the final byte, interpret the remaining characters as a UTF-16 encoded string, and ignore the last constraint (starting with the text "If the Effective-PasswordComplexityEnabled value...").
2. The value MUST be less than or equal to 256 characters (this constraint is called the "maximum password length constraint").
3. The value MUST satisfy all of the following constraints if all of the following conditions are met:
   1. Conditions:
      1. The userAccountControl attribute value contains UF\_NORMAL\_ACCOUNT.
      2. objectSid does not have the DOMAIN\_USER\_RID\_KRBTGT value as the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c).
      3. userAccountControl does not contain UF\_PASSWD\_NOTREQD.
   2. Constraints:
      1. The number of characters in the value MUST not be smaller than the value of the Effective-MinimumPasswordLength attribute (see section [3.1.1.5](#Section_6da47028e678475eb015a5a15ee98212)). This constraint is called the "minimum password length constraint".
      2. The value MUST NOT contain the sAMAccountName attribute value as a case-insensitive substring if that value contains more than two characters.
      3. The value MUST NOT contain any case-insensitive portion of the displayName attribute value that is greater than two characters and delimited by one or more of the following characters.

| Hexadecimal value | Character encoded |
| --- | --- |
| 0x0020 | [SP] |
| 0x002c | , |
| 0x002e | . |
| 0x0009 | [HT] |
| 0x002d | - |
| 0x005f | \_ (underscore) |
| 0x0023 | # |

* + 1. If the Effective-PasswordComplexityEnabled value (see section 3.1.1.5) is set, the password MUST contain characters from at least three of the following five classes:
       1. English uppercase letters: characters 0x41 to 0x56, inclusive.
       2. English lowercase letters: characters 0x62 to 0x7a, inclusive.
       3. Westernized Arabic numerals: characters 0x30 to 0x39, inclusive.
       4. Any character from [[UNICODE3.1]](https://go.microsoft.com/fwlink/?LinkId=90551) that is categorized as Lu, LI, Lt, Lm, Lo.
       5. The following characters.

| Hexadecimal value | Character encoded |
| --- | --- |
| 0x0028 | ( |
| 0x0060 | ` |
| 0x007e | ~ |
| 0x0021 | ! |
| 0x0040 | @ |
| 0x0023 | # |
| 0x0024 | $ |
| 0x0025 | % |
| 0x005e | ^ |
| 0x0026 | & |
| 0x002a | \* |
| 0x005f | \_ (underscore) |
| 0x002d | - |
| 0x002b | + |
| 0x003d | = |
| 0x007c | | |
| 0x005c | \ |
| 0x007b | { |
| 0x007d | } |
| 0x005b | [ |
| 0x005d | ] |
| 0x003a | : |
| 0x003b | ; |
| 0x0022 | " |
| 0x0027 | ' |
| 0x003c | < |
| 0x003e | > |
| 0x002c | , |
| 0x002e | . |
| 0x003f | ? |
| 0x0029 | ) |
| 0x002f | / |

#### Attribute Triggers for Originating Updates

The following attribute-scoped triggers MUST be executed during originating updates to the database.

The term "previous" refers to the value at the beginning of the transaction, before any updates occurred. Unless otherwise specified, other attributes referenced for a particular trigger refer to the attribute on the same object as the attribute whose trigger is currently being executed.

##### objectClass

1. If the **objectClass** attribute value is user or computer, or derived from either of these classes, all of the following constraints MUST be satisfied:
   1. The **objectSid** attribute MUST be updated according to the supplemental trigger specified in section [3.1.1.9.2](#Section_9338c3bd859d490598007a45de6d1482).
   2. The following attributes MUST be updated with the associated values if no value is present in the database.

| Attribute | Value |
| --- | --- |
| badPwdCount | 0 |
| codePage | 0 |
| countryCode | 0 |
| badPasswordTime | 0 |
| lastLogoff | 0 |
| lastLogon | 0 |
| pwdLastSet | 0 |
| accountExpires | 0x7FFFFFFF FFFFFFFF (default value) |
| logonCount | 0 |

* 1. If the value of the **userAccountControl** attribute in the database contains a bit that is specified in the following table, the **sAMAccountType** attribute MUST be updated with the corresponding value.

| userAccountControl | sAMAccountType |
| --- | --- |
| UF\_NORMAL\_ACCOUNT | SAM\_USER\_OBJECT |
| UF\_INTERDOMAIN\_TRUST\_ACCOUNT | SAM\_TRUST\_ACCOUNT |
| UF\_WORKSTATION\_TRUST\_ACCOUNT | SAM\_MACHINE\_ACCOUNT |
| UF\_SERVER\_TRUST\_ACCOUNT | SAM\_MACHINE\_ACCOUNT |

* 1. If the value of the **userAccountControl** attribute in the database contains a bit or bit combination that is specified in the following table, the **primaryGroupId** attribute MUST be updated with the corresponding value.

| userAccountControl | primaryGroupId |
| --- | --- |
| UF\_NORMAL\_ACCOUNT | DOMAIN\_GROUP\_RID\_USERS |
| UF\_INTERDOMAIN\_TRUST\_ACCOUNT | DOMAIN\_GROUP\_RID\_USERS |
| UF\_WORKSTATION\_TRUST\_ACCOUNT | DOMAIN\_GROUP\_RID\_COMPUTERS |
| UF\_SERVER\_TRUST\_ACCOUNT | DOMAIN\_GROUP\_RID\_CONTROLLERS |
| UF\_WORKSTATION\_TRUST\_ACCOUNT & UF\_PARTIAL\_SECRETS\_ACCOUNT | DOMAIN\_GROUP\_RID\_READONLY\_CONTROLLERS |

* 1. If the value of the **userAccountControl** attribute in the database contains a bit that is specified in the following table, the **userAccountControl** attribute MUST be updated with the corresponding bit(s) using a bitwise OR.

| userAccountControl | userAccountControl bits to augment existing value |
| --- | --- |
| UF\_NORMAL\_ACCOUNT | UF\_ACCOUNTDISABLE  UF\_PASSWD\_NOTREQD |

1. If the **objectClass** attribute value is group or is derived from this class, all of the following constraints MUST be satisfied:
   1. The **objectSid** attribute MUST be updated according to the supplemental trigger specified in section 3.1.1.9.2.
   2. The **groupType** attribute MUST be updated, if no value is present in the database, with the value GROUP\_TYPE\_SECURITY\_ACCOUNT.
   3. The **sAMAccountType** attribute MUST be updated with the value dictated by an exact match with the value in the groupType attribute.

| groupType | sAMAccountType |
| --- | --- |
| GROUP\_TYPE\_SECURITY\_ACCOUNT | SAM\_GROUP\_OBJECT |
| GROUP\_TYPE\_ACCOUNT\_GROUP | SAM\_NON\_SECURITY\_GROUP\_OBJECT |
| GROUP\_TYPE\_SECURITY\_RESOURCE | SAM\_ALIAS\_OBJECT |
| GROUP\_TYPE\_RESOURCE\_GROUP | SAM\_NON\_SECURITY\_ALIAS\_OBJECT |
| GROUP\_TYPE\_SECURITY\_UNIVERSAL | SAM\_GROUP\_OBJECT |
| GROUP\_TYPE\_UNIVERSAL\_GROUP | SAM\_NON\_SECURITY\_GROUP\_OBJECT |

##### primaryGroupID

Let O be the object whose **primaryGroupID** attribute is being updated.

Let G be the [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) such that the value of the **primaryGroupId** attribute of O contains the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of the **objectSid** attribute of G prior to the update.

Let G' be the group object such that the value of the **primaryGroupId** attribute of O contains the RID of the **objectSid** attribute of G' after the update.

The following MUST be true prior to the update:

1. The **groupType** of G MUST be one of the following two values: GROUP\_TYPE\_SECURITY\_ACCOUNT or GROUP\_TYPE\_SECURITY\_RESOURCE.
2. The **groupType** of G' MUST be one of the following two values: GROUP\_TYPE\_SECURITY\_ACCOUNT or GROUP\_TYPE\_SECURITY\_RESOURCE.
3. O MUST NOT be in the **member** attribute of G.
4. O MUST be in the **member** attribute of G'.

If the update to the **primaryGroupID** attribute of O is NOT a result of an internal trigger, all of the following constraints MUST be satisfied after the update:

1. O MUST be in the **member** attribute of G.
2. O MUST NOT be in the **member** attribute of G'.

##### lockoutTime

If the lockoutTime attribute value is 0, badPwdCount MUST be updated to the value of 0.[<27>](#Appendix_A_27" \o "Product behavior note 27)

##### sAMAccountName

1. If the **objectSid** attribute has a [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of DOMAIN\_USER\_RID\_KRBTGT and there is already a value present in the **sAMAccountName** attribute, the server MUST return an error status.
2. If the **sAMAccountName** attribute value is NOT unique with respect to the union of all **sAMAccountName** and **msDS-AdditionalSamAccountName** attribute values for all other objects within the scope of the account and [**built-in domain**](#gt_22904540-f2fb-470e-af48-ce1b0e9a02b8), the server MUST return an error status, according to the following conditions.

| Condition | Error status |
| --- | --- |
| The object whose **sAMAccountName** matches the **sAMAccountName** attribute of the current object is a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) as defined in section [3.1.1](#Section_814e12f610374b648d288f1b899dc57f). | STATUS\_GROUP\_EXISTS |
| The object whose **sAMAccountName** matches the **sAMAccountName** attribute of the current object is an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0) as defined in section 3.1.1. | STATUS\_ALIAS\_EXISTS |
| Otherwise: | STATUS\_USER\_EXISTS |

##### clearTextPassword

1. If the **pwdProperties** attribute value on the [**account domain object**](#gt_3ff707e8-0783-4b16-9370-2899f1078c87) contains the DOMAIN\_PASSWORD\_NO\_CLEAR\_CHANGE bit, the server MUST abort the request and return an error status.
2. If the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of the **objectSid** attribute is DOMAIN\_USER\_RID\_KRBTGT and the requesting protocol is a change-password protocol, the server MUST abort the request and return an error status.
3. If the RID of the **objectSid** attribute is DOMAIN\_USER\_RID\_KRBTGT and the requesting protocol is a set-password protocol, the value of clearTextPassword MUST be replaced with a randomly generated value that satisfies all criteria in section [3.1.1.7.2](#Section_cb1f380a0a51470584cdeea1b85953e0).
4. The constraints in section 3.1.1.7.2 MUST be satisfied.
5. The **unicodePwd** attribute MUST be updated with the [**NT hash**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f) of new value.
6. The **dBCSPwd** attribute MUST be updated with the [**LM hash**](#gt_58a72159-b1e9-4770-9b06-d7a6b16279a8) of new value.
7. On a [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration, the **supplementalCredentials** attribute MUST be updated with the cleartext value (see section [3.1.1.8.11](#Section_0705f88862e14a4cbac0b4d427f396f8) for processing details on how **supplementalCredentials** is updated).

##### dBCSPwd

1. The constraints in section [3.1.1.7.1](#Section_bbee51353db24824873a55104c3610ad) MUST be satisfied.
2. The new value MUST be encrypted before being persisted. Encryption is accomplished using the algorithm specified in section [2.2.11.1](#Section_3f5ec79db4494ab29423c4dccbe0b184), with the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) (an unsigned integer) as the encryption key.
3. If the client has access to the Unexpire-Password [**control access right**](#gt_42f6c9e0-a2b3-4bc3-9b87-fdb902e5505e) ([[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 5.1.3.2.1) on the [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1), **pwdLastSet** MUST be updated to the current time; otherwise, **pwdLastSet** MUST be updated to the value zero, which causes the new password to expire immediately.
4. If the update to this attribute is not from an internal trigger, the **supplementalCredential** attribute MUST be removed.
5. The **lmPwdHistory** attribute MUST be updated with the new **dBCSPwd** attribute value (encrypted with the RID, according to constraint 2) according to the constraints in section [3.1.1.9.1](#Section_d0047e7b77bf428691a1648edb2405de).

##### unicodePwd

1. The constraints in section [3.1.1.7.1](#Section_bbee51353db24824873a55104c3610ad) MUST be satisfied.
2. The new value MUST be encrypted before being persisted. Encryption is accomplished using the algorithm specified in section [2.2.11.1](#Section_3f5ec79db4494ab29423c4dccbe0b184), with the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) (an unsigned integer) as the encryption key.
3. If the client has access to the Unexpire-Password [**control access right**](#gt_42f6c9e0-a2b3-4bc3-9b87-fdb902e5505e) ([[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 5.1.3.2.1) on the [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1), **pwdLastSet** MUST be updated to the current time; otherwise, **pwdLastSet** MUST be updated to the value zero, which causes the new password to expire immediately.
4. If the update to this attribute is not from an internal trigger, the **supplementalCredential** attribute MUST be removed.
5. The **ntPwdHistory** attribute MUST be updated with the new **unicodePwd** attribute value (encrypted with the RID, according to constraint 2) according to the constraints in section [3.1.1.9.1](#Section_d0047e7b77bf428691a1648edb2405de).

##### pwdLastSet

See the following citation in Appendix B: Product Behavior.[<28>](#Appendix_A_28" \o "Product behavior note 28)

##### member

1. If all of the following conditions are true, the subsequent constraint MUST be satisfied:
   1. Conditions:
      1. The value contains a [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d)-only [**dsname**](#gt_4d5e1f08-aa00-4dde-9411-7dd6e09ed85a) value.
      2. The dsname value does not resolve to an existing object in the [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) NC.
      3. The server is in a [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration, and the [**domain prefix**](#gt_9066e9dc-8275-4452-9073-daab5fd427c5) of the SID value is not equal to any domain SID in the [**forest**](#gt_fd104241-4fb3-457c-b2c4-e0c18bb20b62); or the server is in a non-DC configuration, and the value is different than the [**account domain security identifier**](#gt_e7f8992c-d815-4e0e-b4c0-4d5d55668561).
   2. Constraint:
      1. A new object with the following characteristics MUST be created with the following attributes and values. The dsname value added to the member attribute MUST reference this object.

| Attribute | Value |
| --- | --- |
| objectClass | foreignSecurityPrincipal |
| objectSid | The SID value of the new dsname value. |
| distinguishedName | The parent MUST be the well-known object container for foreign principal objects. (More information about this container is specified in [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 6.1.1.4.) There is no constraint on the [**relative distinguished name (RDN)**](#gt_22198321-b40b-4c24-b8a2-29e44d9d92b9) value. |
| ntSecurityDescriptor | The default [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) for foreignSecurityPrincipal objects; the **Owner** and **Group** fields of the security descriptor value MUST be the [**Domain Admins**](#gt_d6220a40-b608-4244-8f43-c5a4a0be89ed) SID from the domain in which the object is created. |

1. If the **groupType** is GROUP\_TYPE\_SECURITY\_ACCOUNT, all of the following constraints MUST be satisfied:
   1. If the domain is in [**mixed mode**](#gt_06c1c70e-f2c6-4efd-bff8-474409e69660), the member values MUST refer to [**user objects**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17) (or objects derived from user).
   2. If the domain is in [**native mode**](#gt_8adcb510-b25d-459e-9832-90079803d3c0), the member values MUST satisfy at least one of the following criteria:
      1. The member value refers to a user account.
      2. The member value refers to a group account whose **groupType** is GROUP\_TYPE\_SECURITY\_ACCOUNT.
2. If the **groupType** is GROUP\_TYPE\_SECURITY\_RESOURCE, all of the following constraints MUST be satisfied:
   1. If the domain is in mixed mode, the member values MUST either refer to user objects (or objects derived from user) or refer to [**group objects**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) whose **groupType** is GROUP\_TYPE\_SECURITY\_ACCOUNT.
   2. If the domain is in native mode, the constraint shown above is relaxed to include member values that refer to group objects whose **groupType** is GROUP\_TYPE\_SECURITY\_RESOURCE.
3. If the **groupType** contains the GROUP\_TYPE\_UNIVERSAL\_GROUP, each member value MUST satisfy at least one of the following conditions:
   1. The value refers to a user object (or an object derived from user).
   2. The value refers to a group object (or an object derived from group) with a **groupType** attribute that contains GROUP\_TYPE\_ACCOUNT\_GROUP or GROUP\_TYPE\_UNIVERSAL\_GROUP.

##### userAccountControl

1. If the UF\_LOCKOUT bit (section [2.2.1.13](#Section_10bf6c8e34af4cf98dff6b6330922863)) is set and the **lockoutTime** attribute is nonzero, the **lockoutTime** attribute MUST be updated to a value of zero.
2. The following bits, if set, MUST be unset before committing the transaction: UF\_LOCKOUT and UF\_PASSWORD\_EXPIRED.
3. If the UF\_SERVER\_TRUST\_ACCOUNT bit is set, all of the following constraints MUST be satisfied:
   1. The **primaryGroupId** attribute MUST be updated to the value DOMAIN\_GROUP\_RID\_CONTROLLERS.
   2. If the previous **primaryGroupId** value is NOT DOMAIN\_GROUP\_RID\_COMPUTERS, let G be the group whose **objectSid** value has the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of the previous **primaryGroupId** on the current object. G's member attribute MUST be updated to add a reference to the current object if it is not already present; processing errors for this constraint MUST be ignored.
4. If either UF\_TRUSTED\_TO\_AUTHENTICATE\_FOR\_DELEGATION or UF\_TRUSTED\_FOR\_DELEGATION is set, the client's [**token**](#gt_95f17071-c8f1-403a-8a92-cf87aa7d40f5) MUST be retrieved using the method described in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.3.3.4.3. The **RpcImpersonationAccessToken.Privileges[]** field MUST have the SE\_ENABLE\_DELEGATION\_NAME privilege (defined in [[MS-LSAD]](%5bMS-LSAD%5d.pdf#Section_1b5471ef4c334a91b079dfcbb82f05cc) section 3.1.1.2.1). Otherwise, the server MUST abort processing and return STATUS\_ACCESS\_DENIED.
5. If any of the following bits are set, the client MUST have the associated [**control access right**](#gt_42f6c9e0-a2b3-4bc3-9b87-fdb902e5505e) (defined in [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 5.1.3.2.1) on the **ntSecurityDescriptor** for the [**account domain object**](#gt_3ff707e8-0783-4b16-9370-2899f1078c87), per an [**access check**](#gt_d7906f17-bb2c-4193-a3f0-848bcc351dec). (Information about the access check mechanism is specified in [MS-ADTS] section 5.1.3.3.) If this constraint fails, the server MUST abort processing and return STATUS\_ACCESS\_DENIED.

| userAccountControlBit | Required control access right |
| --- | --- |
| UF\_PASSWD\_NOTREQD | Update-Password-Not-Required-Bit |
| UF\_DONT\_EXPIRE\_PASSWD | Unexpire-Password |
| UF\_ENCRYPTED\_TEXT\_PASSWORD\_ALLOWED | Enable-Per-User-Reversibly-Encrypted-Password |
| UF\_SERVER\_TRUST\_ACCOUNT | DS-Install-Replica |
| UF\_PARTIAL\_SECRETS\_ACCOUNT | DS-Install-Replica |

1. If the UF\_SMARTCARD\_REQUIRED bit is set and is NOT present in the previous value, the **dBCSPwd** and **unicodePwd** attributes MUST be updated with 16 bytes of random bytes, and all [USER\_PROPERTY](#Section_7c0f2eca1783450bb5a0754cf11f22c9) elements MUST be removed from the **supplementalCredentials** attribute.
2. If the UF\_PASSWD\_NOTREQD bit is removed from the **userAccountControl** value, the server MUST abort processing and return an error status if all of the following conditions are true:
   1. **userAccountControl** contains UF\_NORMAL\_ACCOUNT.
   2. **userAccountControl** does not contain the UF\_ACCOUNTDISABLE.
   3. The Effective-MinimumPasswordLength attribute (see section [3.1.1.5](#Section_6da47028e678475eb015a5a15ee98212)) is nonzero.
3. If the UF\_INTERDOMAIN\_TRUST\_ACCOUNT bit is set, and the write request did not originate over the MS-LSAD protocol (see [MS-ADTS] section 6.1.6.9.7), the server MUST abort processing and return an error status.
4. If both UF\_USER\_PARTIAL\_SECRETS\_ACCOUNT and UF\_TRUSTED\_FOR\_DELEGATION are set, the server MUST abort processing and return an error status.
5. If UF\_USER\_PARTIAL\_SECRETS\_ACCOUNT is set and UF\_WORKSTATION\_TRUST\_ACCOUNT is not set, the server MUST abort processing and return an error status.
6. If more than one of the following bits are set, the server MUST abort processing and return an error status.

| userAccountControlBit |
| --- |
| UF\_NORMAL\_ACCOUNT |
| UF\_INTERDOMAIN\_TRUST\_ACCOUNT |
| UF\_WORKSTATION\_TRUST\_ACCOUNT |
| UF\_SERVER\_TRUST\_ACCOUNT |

1. If the UF\_TEMP\_DUPLICATE\_ACCOUNT is set, the server MUST abort processing and return an error status.
2. If none of the following bits are set, the server MUST set the UF\_NORMAL\_ACCOUNT bit.

| userAccountControlBit |
| --- |
| UF\_NORMAL\_ACCOUNT |
| UF\_INTERDOMAIN\_TRUST\_ACCOUNT |
| UF\_WORKSTATION\_TRUST\_ACCOUNT |
| UF\_SERVER\_TRUST\_ACCOUNT |

For more information about the UF\_SERVER\_TRUST\_ACCOUNT and UF\_WORKSTATION\_TRUST\_ACCOUNT bits, see the following citation in [Appendix B: Product Behavior](#Section_fa61e5fcf8fb4d5b9695c724af0c3829).[<29>](#Appendix_A_29" \o "Product behavior note 29)

##### supplementalCredentials

The **supplementalCredentials** attribute is a structured binary value that contains additional cryptographic forms of the cleartext password (and optionally the cleartext password itself) that are stored as property-value pairs.

The format of **supplementalCredentials** is a [USER\_PROPERTIES (section 2.2.10.1)](#Section_8263e7ababa943d28a363a9cb2dd3dad) structure.

When **supplementalCredentials** is updated with a value (which is interpreted as a UTF-16 encoded cleartext password) as a result of a trigger, this value is not stored directly; instead, it is processed and the result is stored in **supplementalCredentials** as specified in this section.

Each property name is a UTF-16 encoded string; each value has its own unique binary format. The properties that are in **supplementalCredentials** are listed in the following table.

| Property name (normative) | Property value semantic | Property value format specification section |
| --- | --- | --- |
| Packages | A list of the credential types that are stored as properties in **supplementalCredentials**. | [3.1.1.8.11.2](#Section_3bf85ff103ec4ed7878312c57e4daf81) |
| Primary:WDigest | Cryptographic hashes of the cleartext password for the Digest authentication protocol. | [3.1.1.8.11.3](#Section_f02dbe354ef941d991400b404f2794bf) |
| Primary:Kerberos | Cryptographic hashes of the cleartext password for the Kerberos authentication protocol. | [3.1.1.8.11.4](#Section_573b6c4061414a41ba3ffa72cca9d861) |
| Primary:CLEARTEXT | The cleartext password. | [3.1.1.8.11.5](#Section_e348e08f1c1942bbb1a9687843059414) |
| Primary:Kerberos-Newer-Keys | Cryptographic hashes of the cleartext password for the Kerberos authentication protocol. | [3.1.1.8.11.6](#Section_f3fd5c777a6343dcac8d598ca374a5a3) |
| Primary:NTLM-Strong-NTOWF | Cryptographic key used for the NTLM authentication protocol. This key has no relationship to the cleartext password. | [3.1.1.8.11.7](#Section_898a260ad6734777a456a181797b7aed) |

###### Processing

Section [3.1.1.8.11.1.1](#Section_79c2674dbf554083a63e587b1aaad829) describes how to update the [USER\_PROPERTIES](#Section_8263e7ababa943d28a363a9cb2dd3dad) structure when properties are added or removed.

Section [3.1.1.8.11.1.2](#Section_94aa20923aba467da4a6e1e8bdeaa95f) describes how to update a [USER\_PROPERTY](#Section_7c0f2eca1783450bb5a0754cf11f22c9) structure given a property-value pair.

USER\_PROPERTIES Processing

When a new property-value pair is added (as a result of an update, for example), the **PropertyCount** field of the [USER\_PROPERTIES](#Section_8263e7ababa943d28a363a9cb2dd3dad) structure MUST be incremented by one, and the property structure (a [USER\_PROPERTY](#Section_7c0f2eca1783450bb5a0754cf11f22c9) structure) MUST be added to the variable-length array of USER\_PROPERTY structures that follow USER\_PROPERTIES. The order of the USER\_PROPERTY entries is not important.

When a property-value pair is removed and the property-value is present in the USER\_PROPERTIES structure, the **PropertyCount** field of the USER\_PROPERTIES structure MUST be decremented by one, and the property structure (a USER\_PROPERTY structure) MUST be removed from the variable-length array of USER\_PROPERTY structures that follow USER\_PROPERTIES.

When the last property-value pair is removed, the **PropertyCount** field is no longer included in the USER\_PROPERTIES structure. In this state, the absence of any user properties MUST be inferred from the structure’s total length (0x6F bytes).

If the property-value is not present on removal, then no change to USER\_PROPERTIES is required.

USER\_PROPERTY Processing

This section describes how to structure a given property-value pair in a [USER\_PROPERTY](#Section_7c0f2eca1783450bb5a0754cf11f22c9) structure.

* The **NameLength** field MUST be set to the size, in bytes, of the property name. The property name "WDigest", for example, has a **NameLength** of 14.
* The **ValueLength** field MUST be set to the size, in bytes, of the value of the property after hexadecimal-encoding the value per the specification in section 2.2.10.2.
* The property name MUST follow the **Reserved** field of the USER\_PROPERTY structure.
* The hex-encoded value MUST follow the property name.

###### Packages Property

The property value is a UTF-16 encoded string. The string itself is composed of a set of substrings separated by a NULL Unicode character value, as defined in [[UNICODE3.1]](https://go.microsoft.com/fwlink/?LinkId=90551). The final character does not need to be a NULL Unicode character. Each substring is the name of a credential type stored as a property in the **supplementalCredentials** value.

When an update occurs, if a credential-type property (that is, a property that represents a credential type) is successfully computed, this value MUST be updated with the associated credential name. The following table shows the legal values of names to be used as strings in the property value of the "Packages" property along with their associated credential type.

| Credential type property | Name |
| --- | --- |
| Primary:WDigest | WDigest |
| Primary:Kerberos | Kerberos |
| Primary:CLEARTEXT | CLEARTEXT |
| Primary:Kerberos-Newer-Keys | Kerberos-Newer-Keys |
| Primary:NTLM-Strong-NTOWF | NTLM-Strong-NTOWF |

###### Primary:WDigest Property

The WDigest property contains pre-calculated hash forms that are used in the digest authentication protocols ([[RFC2617]](https://go.microsoft.com/fwlink/?LinkId=90373)). A normative description of the hashes used by the protocol is specified in [RFC2617] section 3.2.2.2.

When an update to **supplementalCredentials** occurs, the server MUST create a WDIGEST\_CREDENTIALS-structured value (section [2.2.10.3](#Section_830b39623a0042cf94dfc192dcfd4803)) using the hash-computation mechanisms in section [3.1.1.8.11.3.1](#Section_511f65e83dbe4377b65730b47dc4f26c). This value MUST then be placed in a [USER\_PROPERTY](#Section_7c0f2eca1783450bb5a0754cf11f22c9) structure along with the property name "Primary:WDigest". Finally, the resulting USER\_PROPERTY-structured value MUST be added to the list of properties within **supplementalCredentials** per section [3.1.1.8.11.1.1](#Section_79c2674dbf554083a63e587b1aaad829).

WDIGEST\_CREDENTIALS Construction

The following notation is used to describe how the hash values are constructed. All strings are converted from UTF-16 encoding to ISO 8859-1 Latin I code page ([[MS-UCODEREF]](%5bMS-UCODEREF%5d.pdf#Section_4a045e08fc294f22baf416f38c2825fb) section 2.2.1, Codepage ID 28591, and [MS-UCODEREF] section 2.2.2) prior to the hashing.

| Notation | Description |
| --- | --- |
| MD5(x, y, z) | An MD5 hash of the values x, y, z, in that order. |
| MD5(x, y) | An MD5 hash of the values x, y, in that order. |
| UPPER(x) | The uppercase version of the string as defined in the Unicode standard ([[UNICODE3.1]](https://go.microsoft.com/fwlink/?LinkId=90551)). |
| LOWER(x) | The lowercase version of the string as defined in the Unicode standard ([UNICODE3.1]). |
| sAMAccountName | The **sAMAccountName** attribute value. |
| NETBIOSDomainName | The name attribute of the [**account domain object**](#gt_3ff707e8-0783-4b16-9370-2899f1078c87). |
| DNSDomainName | The [**fully qualified domain name (FQDN)**](#gt_1769aec9-237e-44ed-9014-1abb3ec6de6e) of the [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca). |

Hash1: MD5(sAMAccountName, NETBIOSDomainName, password)

Hash2: MD5(LOWER(sAMAccountName), LOWER(NETBIOSDomainName), password)

Hash3: MD5(UPPER(sAMAccountName), UPPER(NETBIOSDomainName), password)

Hash4: MD5(sAMAccountName, UPPER(NETBIOSDomainName), password)

Hash5: MD5(sAMAccountName, LOWER(NETBIOSDomainName), password)

Hash6: MD5(UPPER(sAMAccountName), LOWER(NETBIOSDomainName), password)

Hash7: MD5(LOWER(sAMAccountName), UPPER(NETBIOSDomainName), password)

Hash8: MD5(sAMAccountName, DNSDomainName, password)

Hash9: MD5(LOWER(sAMAccountName), LOWER(DNSDomainName), password)

Hash10: MD5(UPPER(sAMAccountName), UPPER(DNSDomainName), password)

Hash11: MD5(sAMAccountName, UPPER(DNSDomainName), password)

Hash12: MD5(sAMAccountName, LOWER(DNSDomainName), password)

Hash13: MD5(UPPER(sAMAccountName), LOWER(DNSDomainName), password)

Hash14: MD5(LOWER(sAMAccountName), UPPER(DNSDomainName), password)

Hash15: MD5(userPrincipalName, password)

Hash16: MD5(LOWER(userPrincipalName), password)

Hash17: MD5(UPPER(userPrincipalName), password)

Hash18: MD5(NETBIOSDomainName\sAMAccountName, password)

Hash19: MD5(LOWER(NETBIOSDomainName\sAMAccountName), password)

Hash20: MD5(UPPER(NETBIOSDomainName\sAMAccountName), password)

Hash21: MD5(sAMAccountName, "Digest", password)

Hash22: MD5(LOWER(sAMAccountName), "Digest", password)

Hash23: MD5(UPPER(sAMAccountName), "Digest", password)

Hash24: MD5(userPrincipalName, "Digest", password)

Hash25: MD5(LOWER(userPrincipalName), "Digest", password)

Hash26: MD5(UPPER(userPrincipalName), "Digest", password)

Hash27: MD5(NETBIOSDomainName\sAMAccountName, "Digest", password)

Hash28: MD5(LOWER(NETBIOSDomainName\sAMAccountName), "Digest", password)

Hash29: MD5(UPPER(NETBIOSDomainName\sAMAccountName), "Digest", password)

###### Primary:Kerberos Property

When an update to **supplementalCredentials** occurs, the server MUST create a KERB\_STORED\_CREDENTIAL-structured value as specified below. This value MUST then be placed in a [USER\_PROPERTY](#Section_7c0f2eca1783450bb5a0754cf11f22c9) structure along with the property name "Primary:Kerberos". Finally, the resulting USER\_PROPERTY-structured value MUST be added to the list of properties within **supplementalCredentials** according to section [3.1.1.8.11.1.1](#Section_79c2674dbf554083a63e587b1aaad829).

[KERB\_STORED\_CREDENTIAL](#Section_866f018086944e14be7edfd91d91694f) is a variable-length structure starting with a KERB\_STORED\_CREDENTIAL structure, followed by two or four [KERB\_KEY\_DATA](#Section_9921c3edf4704f369edf7afe24c468ec) structures, followed by a [**salt**](#gt_1672c769-f184-404a-9575-e637fd3a43ed) value and two or four key values. The salt and key values are referenced from the KERB\_STORED\_CREDENTIAL and KERB\_KEY\_DATA structures.

**Revision**, **Flags**, **DefaultSaltLength**, **DefaultSaltMaximumLength**, and **DefaultSaltOffset** MUST be set as specified in section 2.2.10.4. **DefaultSaltOffset**, for example, is the offset of the "DefaultSalt value" section from the start of the **Revision** field.[<30>](#Appendix_A_30" \o "Product behavior note 30)

The server MUST calculate two hash forms of the cleartext password, as specified in [[RFC3961]](https://go.microsoft.com/fwlink/?LinkId=90450) sections 6.2.1 and 6.2.3. Call these values Key1 and Key2.

The first two KERB\_KEY\_DATA MUST be set to hold Key1 and Key2. Key1 and Key2 MUST be added to the end of the structure.

If there are existing KERB\_KEY\_DATA elements in the property prior to the current update, these elements MUST be copied into the third and fourth KERB\_KEY\_DATA elements. Call the associated key values of these KERB\_KEY\_DATA structures Key3 and Key4. Key3 and Key4 MUST be added to the end of the structure.[<31>](#Appendix_A_31" \o "Product behavior note 31)

If there are no existing KERB\_KEY\_DATA elements in the property prior to the current update, the resulting KERB\_STORED\_CREDENTIAL in the third and fourth optional KERB\_KEY\_DATA elements are excluded from the resulting value (and Key3 and Key4, from the preceding paragraph, are also excluded).

###### Primary:CLEARTEXT Property

This credential type is the cleartext password. The value format is the UTF-16 encoded cleartext password.

Storage of the cleartext password for an object is configured when the Effective-PasswordReversibleEncryptionEnabled value (section [3.1.1.5](#Section_6da47028e678475eb015a5a15ee98212)) is set or when the current object's **userAccountControl** contains the USER\_ENCRYPTED\_TEXT\_PASSWORD\_ALLOWED bit.

If during a **clearTextPassword** attribute update, there is a Primary:CLEARTEXT property present in **supplementalCredentials** and storage of the cleartext password is not configured, the Primary:CLEARTEXT property MUST be removed, and the Packages property within **supplementalCredentials** MUST be updated to not contain the "CLEARTEXT" string.

If during a password set or change operation, there is a Primary:CLEARTEXT property present in **supplementalCredentials** and storage of the cleartext password is configured, the Primary:CLEARTEXT property MUST be updated (or added if not present), and the Packages property with **supplementalCredentials** MUST be updated to contain the "CLEARTEXT" string, if it is not already present.

###### Primary:Kerberos-Newer-Keys Property

When an update to **supplementalCredentials** occurs, and the current [**domain functional level**](#gt_f9b5a663-12d5-46d4-b23a-147704f026fc) is DS\_BEHAVIOR\_WIN2008 or greater, the server MUST create a KERB\_STORED\_CREDENTIAL\_NEW-structured value as specified in section [2.2.10.6](#Section_08cb3ca7954b45e3902e77512fe3ba8e). This value MUST then be placed in a [USER\_PROPERTY](#Section_7c0f2eca1783450bb5a0754cf11f22c9) structure along with the property name "Primary:Kerberos-Newer-Keys". Finally, the resulting USER\_PROPERTY-structured value MUST be added to the list of properties within **supplementalCredentials** according to section [3.1.1.8.11.1.1](#Section_79c2674dbf554083a63e587b1aaad829).

**Revision**, **Flags**, **DefaultSaltLength**, **DefaultSaltMaximumLength**, and **DefaultSaltOffset** MUST be set as specified in section 2.2.10.6. **DefaultSaltOffset**, for example, is the offset of the "DefaultSalt value" section from the start of the **Revision** field.

The server MUST calculate four hash forms of the cleartext password, as specified in [[RFC3961]](https://go.microsoft.com/fwlink/?LinkId=90450) sections 6.2.1 and 6.2.3, and as specified in [[RFC3962]](https://go.microsoft.com/fwlink/?LinkId=90451) section 6. Call these values Key1, Key2, Key3, and Key4.

The **Credentials** field MUST be set to hold Key1, Key2, Key3, and Key4. If there are existing keys in the **Credentials** field, they MUST be moved to the **OldCredentials** field. If there are existing keys in the **OldCredentials** field, they MUST be moved to the **OlderCredentials** field. Any existing keys in the **OlderCredentials** field MUST be discarded.[<32>](#Appendix_A_32" \o "Product behavior note 32)

###### Primary:NTLM-Strong-NTOWF Property

When an update to **supplementalCredentials** occurs, the server MUST create an NTLM-Strong-NTOWF-structured value as specified in section [2.2.10.9](#Section_290820c9f52a4b85b311b61084f8c85e).[<33>](#Appendix_A_33" \o "Product behavior note 33) The **NTLMStrongNTOWF** field MUST be set to a random value with no relationship to the cleartext password used to generate the other values in the **supplementalCredentials** attribute. The NTLM-Strong-NTOWF value MUST then be placed in a USER\_PROPERTY structure (section [2.2.10.2](#Section_7c0f2eca1783450bb5a0754cf11f22c9)) along with the property name "Primary:NTLM-Strong-NTOWF". Any previously existing property with that name is discarded. Finally, the resulting USER\_PROPERTY-structured value MUST be added to the list of properties within **supplementalCredentials** according to section [3.1.1.8.11.1.1](#Section_79c2674dbf554083a63e587b1aaad829).

#### Additional Update Triggers

The following triggers are referenced from other constraints or triggers.

##### Password History Update

The following constraints MUST be satisfied for ntPwdHistory and lmPwdHistory. The term "history attribute" refers to one or the other in the following constraints, and the term "associated password" refers to dBCSPwd when the history attribute is lmPwdHistory, and unicodePwd when the history attribute is ntPwdHistory.

Let Password-History-Length be the value of the Effective-PasswordHistoryLength attribute (see section [3.1.1.5](#Section_6da47028e678475eb015a5a15ee98212)). If the target object being updated is the krbtgt account (that is, the objectSid value has the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) value of DOMAIN\_USER\_RID\_KRBTGT), and Password-History-Length is less than 3, the value of 3 MUST be used for Password-History-Length.

1. If the Password-History-Length is greater than 0 and the history attribute is zero length, the history attribute MUST be updated with the previous associated password if the old associated password's length is nonzero.
2. If the Password-History-Length is zero, the history attribute MUST be updated with a zero-length value.
3. If the Password-History-Length is nonzero, the associated password value MUST be placed at the beginning of the history attribute, and existing values MUST be shifted by 16 bytes to the right. If the size of the attribute exceeds Password-History-Length \* 16, the attribute value MUST be truncated to not exceed Password-History-Length \* 16 bytes.

##### objectSid Value Generation

This section is referenced by object creation triggers to update the **objectSid** attribute with a [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) value. The SID value is generated by first generating a 32-bit unsigned integer value (the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c)) and then concatenating that value with the [**account domain security identifier**](#gt_e7f8992c-d815-4e0e-b4c0-4d5d55668561).

The key part of this section is how the RID is generated, because it MUST be unique for all time and space for a given [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca). For all algorithms, once the RID is generated, the SID value is generated as specified in the previous sentence, and the **objectSid** attribute is updated with that value.

The simplest RID-generation algorithm is to maintain a counter and increment the counter for each RID that is issued. This algorithm is entirely sufficient for the non–[**domain controller**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) case for this protocol. In a distributed environment, where any domain controller might be creating a [**security principal**](#gt_f3ef2572-95cf-4c5c-b3c9-551fd648f409) and therefore needs to assign a RID to that principal, the algorithm becomes more complicated. Many schemes are possible, up to and including a distributed counter, as described in [[LAMPORT]](https://go.microsoft.com/fwlink/?LinkId=89928).

The RID-generation algorithm is different between a DC and non-DC configuration.

The following specifications present the constraints that MUST be satisfied when generating a RID. Generating RIDs in a monotonically increasing manner when possible (in addition to satisfying the constraints) is one implementation choice, but is not required.

###### DC Configuration

The following steps are used to generate a unique [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) on a [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration.

Let Rid-Set be the directory object referenced in the **rIDSetReferences** attribute, as stored on the configured computer object for the host server.

Let Rid-Range be the range specified by the **rIDPreviousAllocationPool** attribute of the Rid-Set object. The lower bound of the Rid-Range is the first 32-bit integer (in little-endian byte order) of the **rIDPreviousAllocationPool** attribute value. The upper bound of the Rid-Range is the second 32-bit integer (in little-endian byte order).

1. The server MUST generate a 32-bit integer value subject to all of the following constraints:
   1. The value MUST be within the Rid-Range.
   2. Any value chosen from the Rid-Range that is used for an **objectSid** value that is successfully committed in a transaction MUST NOT ever be used again for **objectSid** generation within the current [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca).
2. If the constraints in step 1 cannot be satisfied because the **rIDPreviousAllocationPool** attribute does not exist or because all possible RIDs within the Rid-Range have been consumed:
   1. If the **rIDAllocationPool** attribute of the Rid-Set object exists and has a value different from that of **rIDPreviousAllocationPool**, the server copies the value of **rIDAllocationPool** to **rIDPreviousAllocationPool**, and attempts to generate a 32-bit value according to the constraints in step 1.
   2. If the **rIDAllocationPool** attribute of the Rid-Set object does not exist or has a value identical to that of **rIDPreviousAllocationPool**, the server MUST call the IDL\_DRSGetNCChanges method (as specified in [[MS-DRSR]](%5bMS-DRSR%5d.pdf#Section_f977faaa673e4f66b9bf48c640241d47) section 4.1.10) to obtain a (new) value for **rIDAllocationPool**, copy this value to **rIDPreviousAllocationPool**, and attempt to generate a 32-bit value according to the constraints in step 1. The server MAY also return an error code if the constraints in step 1 cannot be satisfied.[<34>](#Appendix_A_34" \o "Product behavior note 34)

###### Non-DC Configuration

The following steps are used to generate a unique [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) on a non-[**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration.

1. The server MUST generate a 32-bit integer value subject to all of the following constraints:
   1. The value MUST be greater than or equal to 1000.
   2. Any value chosen by this algorithm that is successfully committed in a transaction MUST NOT ever be used again for **objectSid** generation within the current [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca).
2. If the constraints in step 1 cannot be satisfied, the server MUST abort processing and return an error status.

#### SamContextHandle Data Model

This protocol is based largely on the use of RPC context handles to maintain session state between the client and the server. The basic context-handle programming model is described in [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824) section 6.1.6.

The server MUST maintain the following data elements for each context handle that is returned to a client.

| Name | Type |
| --- | --- |
| GrantedAccess | ACCESS\_MASK |
| HandleType | HandleType MUST be one of the following:   * Server * Domain * Group * Alias * User |
| Object | A reference to an object in the database of the type specified in HandleType. |

#### Server Access Control List

The server MUST maintain an access control list for authorizing clients to make calls to the server. Authorization is based on whether or not a client is granted STANDARD\_RIGHTS\_READ. The default configuration of this access control list depends on whether or not the server is a domain controller, per the following table:

| Role | Default Server access control list |
| --- | --- |
| Non-DC | STANDARD\_RIGHTS\_READ is granted to BuiltIn\Administrators. |
| DC | STANDARD\_RIGHTS\_READ is granted to all clients. |

### Security Model

The security model has two layers. The first layer is a server-wide security check that applies to all calls. Calls that fail this check are rejected and do not proceed. The second layer is applied on a per-context handle and/or per-method basis, depending on the scenario.

For methods that accept a context handle, the security model is a handle-based security model. A client obtains a handle with a client-specified access for that handle. The handle can then be used for operations that require the granted access. The access is encoded in a 32-bit value (an [**access mask**](#gt_462f2aa7-2cfc-404a-b479-30f127c512b1)). Note that some methods MUST enforce additional security requirements based on the input.

The security model assumes that whenever a context handle is presented to a method, the identity of the client is the same as the identity of the client that originally opened the handle.[<35>](#Appendix_A_35" \o "Product behavior note 35)

#### Server-wide Access Check

The server MUST perform an authorization check[<36>](#Appendix_A_36" \o "Product behavior note 36) at the beginning of each call, before method-specific processing begins. Method-specific processing proceeds if the client is granted STANDARD\_RIGHTS\_READ via the Server Access Control List (section [3.1.1.11](#Section_1ed7ed5bd7b5444f86b8ebe8bad1403a)). If the client is not granted this right, the call is rejected with an access denied error.

#### Standard Handle-Based Access Checks

The following tables specify the required access for the RPC methods that enforce required access on a handle parameter.

* [SamrCloseHandle](#Section_55d134dfe25748ad8afacb2ca45cd3cc)

| Information level | Required access |
| --- | --- |
| N/A | None checked |

* [SamrLookupDomainInSamServer](#Section_47492d59e0954398b03e8a062b989123)

| Information level | Required access |
| --- | --- |
| N/A | SAM\_SERVER\_LOOKUP\_DOMAIN |

* [SamrEnumerateDomainsInSamServer](#Section_2142fd2d085442c1a9fb2fe964e381ce)

| Information level | Required access |
| --- | --- |
| N/A | SAM\_SERVER\_ENUMERATE\_DOMAINS |

* [SamrOpenDomain](#Section_ba710c905b1242f89e5ad4aacc1329fa)

| Information level | Required access |
| --- | --- |
| N/A | SAM\_SERVER\_LOOKUP\_DOMAIN |

* [SamrQueryInformationDomain](#Section_5d6a2817caa941caa269fd13ecbb4fa8)

[SamrQueryInformationDomain2](#Section_fbdbbd7401064ca2941e36ded0be0c62)

| Information level | Required access |
| --- | --- |
| DomainPasswordInformation | DOMAIN\_READ\_PASSWORD\_PARAMETERS |
| DomainLockoutInformation: | DOMAIN\_READ\_PASSWORD\_PARAMETERS |
| DomainGeneralInformation | DOMAIN\_READ\_OTHER\_PARAMETERS |
| DomainLogoffInformation | DOMAIN\_READ\_OTHER\_PARAMETERS |
| DomainOemInformation | DOMAIN\_READ\_OTHER\_PARAMETERS |
| DomainNameInformation | DOMAIN\_READ\_OTHER\_PARAMETERS |
| DomainServerRoleInformation | DOMAIN\_READ\_OTHER\_PARAMETERS |
| DomainReplicationInformation | DOMAIN\_READ\_OTHER\_PARAMETERS |
| DomainModifiedInformation | DOMAIN\_READ\_OTHER\_PARAMETERS |
| DomainStateInformation | DOMAIN\_READ\_OTHER\_PARAMETERS |
| DomainModifiedInformation2 | DOMAIN\_READ\_OTHER\_PARAMETERS |
| DomainGeneralInformation2 | DOMAIN\_READ\_PASSWORD\_PARAMETERS |  DOMAIN\_READ\_OTHER\_PARAMETERS |

* [SamrSetInformationDomain](#Section_9b7ae0b0bd1141339c62fba7095aee12)

| Information level | Required access |
| --- | --- |
| DomainPasswordInformation | DOMAIN\_WRITE\_PASSWORD\_PARAMS |
| DomainLockoutInformation | DOMAIN\_WRITE\_PASSWORD\_PARAMS |
| DomainLogoffInformation | DOMAIN\_WRITE\_OTHER\_PARAMETERS |
| DomainOemInformation | DOMAIN\_WRITE\_OTHER\_PARAMETERS |
| DomainReplicationInformation | DOMAIN\_ADMINISTER\_SERVER |
| DomainStateInformation | DOMAIN\_ADMINISTER\_SERVER |
| DomainServerRoleInformation | DOMAIN\_ADMINISTER\_SERVER |

* [SamrCreateGroupInDomain](#Section_175c1cf94fa248379e5bbb1f0f950bee)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_CREATE\_GROUP |

* [SamrEnumerateGroupsInDomain](#Section_e0b7a4b7ecfc405f9d7d32b3cd2cd6c8)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_LIST\_ACCOUNTS |

* [SamrCreateUserInDomain](#Section_cd0efce013fb410687db4bef73bb4b1b)

[SamrCreateUser2InDomain](#Section_a98d7fbb17354fbfb41aef363c899002)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_CREATE\_USER |

* [SamrEnumerateUsersInDomain](#Section_6bdc92c0c6924ffb9de765858b68da75)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_LIST\_ACCOUNTS |

* [SamrCreateAliasInDomain](#Section_7e505875e44f4f9d922b7ff3871c752d)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_CREATE\_ALIAS |

* [SamrEnumerateAliasesInDomain](#Section_ce340cffedef4356ace033d2874d306b)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_LIST\_ACCOUNTS |

* [SamrGetAliasMembership](#Section_0318404522084c02b38bef955d6dc3ef)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_GET\_ALIAS\_MEMBERSHIP |

* [SamrLookupNamesInDomain](#Section_d91271c67b2e419499278fabfa429f90)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_LOOKUP |

* [SamrLookupIdsInDomain](#Section_c870951c74b347149857224595ffc61a)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_LOOKUP |

* [SamrOpenGroup](#Section_d396e6c9d04a4729b0d8f50f2748f3c8)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_LOOKUP |

* [SamrQueryInformationGroup](#Section_8fd0003f807640ae872f9cdb6576b3c7)

| Information level | Required access |
| --- | --- |
| GroupGeneralInformation | GROUP\_READ\_INFORMATION |
| GroupNameInformation | GROUP\_READ\_INFORMATION |
| GroupAttributeInformation | GROUP\_READ\_INFORMATION |
| GroupAdminCommentInformation | GROUP\_READ\_INFORMATION |
| GroupReplicationInformation | GROUP\_READ\_INFORMATION |

* [SamrSetInformationGroup](#Section_e66db19f600a481bbc4e23953433255d)

| Information level | Required access |
| --- | --- |
| GroupNameInformation | GROUP\_WRITE\_ACCOUNT |
| GroupAttributeInformation | GROUP\_WRITE\_ACCOUNT |
| GroupAdminCommentInformation | GROUP\_WRITE\_ACCOUNT |

* [SamrAddMemberToGroup](#Section_3c70fec36a2e48efbd4eaa2b3a1cd96a)

| Information level | Required access |
| --- | --- |
| N/A | GROUP\_ADD\_MEMBER |

* [SamrDeleteGroup](#Section_664ed55f198a4775b9d4398131dba577)

| Information level | Required access |
| --- | --- |
| N/A | DELETE |

* [SamrRemoveMemberFromGroup](#Section_613ff6635863443089238f1edc05310b)

| Information level | Required access |
| --- | --- |
| N/A | GROUP\_REMOVE\_MEMBER |

* [SamrGetMembersInGroup](#Section_3ed5030d88a342caa6e08c12aa2fdfbd)

| Information level | Required access |
| --- | --- |
| N/A | GROUP\_LIST\_MEMBERS |

* [SamrSetMemberAttributesOfGroup](#Section_a4db0c2755404d2fb11108db1ee0b5a4)

| Information level | Required access |
| --- | --- |
| N/A | GROUP\_ADD\_MEMBER |

* [SamrOpenAlias](#Section_c3696b7620054eba820d91fdded3422d)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_LOOKUP |

* [SamrQueryInformationAlias](#Section_599d6624d93b46de827a58eb0e7527bd)

| Information level | Required access |
| --- | --- |
| AliasGeneralInformation | ALIAS\_READ\_INFORMATION |
| AliasNameInformation | ALIAS\_READ\_INFORMATION |
| AliasAdminCommentInformation | ALIAS\_READ\_INFORMATION |
| AliasReplicationInformation | ALIAS\_READ\_INFORMATION |

* [SamrSetInformationAlias](#Section_ba787e4e3a4b47a2aca3c3ac2d2c511e)

| Information level | Required access |
| --- | --- |
| AliasNameInformation | ALIAS\_WRITE\_ACCOUNT |
| AliasAdminCommentInformation | ALIAS\_WRITE\_ACCOUNT |

* [SamrDeleteAlias](#Section_ecb88df184d148c8b49cd3f444943100)

| Information level | Required access |
| --- | --- |
| N/A | DELETE |

* [SamrAddMemberToAlias](#Section_9a5d2c35e84b4e59b7b096c6fa0fc8d7)

| Information level | Required access |
| --- | --- |
| N/A | ALIAS\_ADD\_MEMBER |

* [SamrRemoveMemberFromAlias](#Section_7e99402967a244fd84397640fe5a376b)

| Information level | Required access |
| --- | --- |
| N/A | ALIAS\_REMOVE\_MEMBER |

* [SamrGetMembersInAlias](#Section_5909af7f858b4cc7a2e4f852070d479d)

| Information level | Required access |
| --- | --- |
| N/A | ALIAS\_LIST\_MEMBERS |

* [SamrOpenUser](#Section_0aee1c31ec404633bb560cf8429093c0)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_LOOKUP |

* [SamrDeleteUser](#Section_4643a57956ec4c66a1ef4ab78dd21d73)

| Information level | Required access |
| --- | --- |
| N/A | DELETE |

* [SamrChangePasswordUser](#Section_9699d8cae1a4433ca8c3d7bebeb01476)

| Information level | Required access |
| --- | --- |
| N/A | None checked |

* [SamrGetGroupsForUser](#Section_a4adbf20040f4416a960e5b7917fdae7)

| Information level | Required access |
| --- | --- |
| N/A | USER\_LIST\_GROUPS |

* [SamrQueryDisplayInformation](#Section_c1458942f2d54317a888abd27abad504)

[SamrQueryDisplayInformation2](#Section_9aa37aa4f77e49b199935c92ba97cd62)

[SamrQueryDisplayInformation3](#Section_7cd1c6d088194a5897215be588c1d4ad)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_LIST\_ACCOUNTS |

* [SamrGetDisplayEnumerationIndex](#Section_af95abf82e1142c394aaba72f8f96c09)

[SamrGetDisplayEnumerationIndex2](#Section_bd429624f2d547178aa2659952c3e209)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_LIST\_ACCOUNTS |

* [SamrRemoveMemberFromForeignDomain](#Section_03afc843584d473b834a3f5a1ac86cce)

| Information level | Required access |
| --- | --- |
| N/A | DOMAIN\_LOOKUP |

* [SamrAddMultipleMembersToAlias](#Section_6d981290a203408094b04a26de22d8c5)

| Information level | Required access |
| --- | --- |
| N/A | ALIAS\_ADD\_MEMBER |

* [SamrRemoveMultipleMembersFromAlias](#Section_7b2455d23f1344a7996950c240562c42)

| Information level | Required access |
| --- | --- |
| N/A | ALIAS\_REMOVE\_MEMBER |

* [SamrRidToSid](#Section_00ff8192a4f645ba9f65917e46b6a693)

| Information level | Required access |
| --- | --- |
| N/A | None checked |

#### AD Access Checks in DC Configuration

Unless otherwise specified, the create, update, delete, and read [**access checks**](#gt_d7906f17-bb2c-4193-a3f0-848bcc351dec) enforced by the MS-ADTS data model (specified in [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 5.1.3) are not enforced during the message processing of this protocol.

#### Acquiring an SMB Session Key

The server MUST retrieve the SMB session key as specified in [[MS-CIFS]](%5bMS-CIFS%5d.pdf#Section_d416ff7cc536406ea9514f04b2fd1d2b) section 3.5.4.4.

### Timers

This protocol does not introduce any timers. Information about any transport-level timers is specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15).

### Initialization

This section covers the default users and groups that the server MUST have and the default access control on the data manipulated by this protocol.

#### Default Access

Information about the default access control (expressed in the default [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350)) on user, group, alias, [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca), and [**server objects**](#gt_62a8c543-5998-480b-8fa7-41a8f04a18e5) is specified in [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 3.1.1.2. This is significant because this server MUST use the security descriptor from the [MS-ADTS] data model to determine whether the client has access to perform the requested operation. If, for example, a client opens a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1) with [SamrOpenDomain (section 3.1.5.1.5)](#Section_ba710c905b1242f89e5ad4aacc1329fa) requesting DOMAIN\_READ\_PASSWORD\_PROPERTIES, SamrOpenDomain uses the [MS-ADTS] data model security descriptor to determine whether the client has access to read password-related properties. For more information related to this example, see the message processing section of SamrOpenDomain.

#### Default Accounts

The following accounts MUST be present in a server's database.[<37>](#Appendix_A_37" \o "Product behavior note 37)

Non-[**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration, user accounts.

| Name | Domain | Rid | userAccountControl |
| --- | --- | --- | --- |
| Administrator | Account | 500 | UF\_NORMAL\_ACCOUNT |  UF\_DONT\_EXPIRE\_PASSWORD |
| Guest | Account | 501 | UF\_NORMAL\_ACCOUNT |  UF\_ACCOUNTDISABLE |  UF\_DONT\_EXPIRE\_PASSWORD |

Non-DC configuration, alias accounts.

| Name | Domain | Rid | Member |
| --- | --- | --- | --- |
| Administrators | Built-in | 544 | Administrator |
| Users | Built-in | 545 |  |
| Guests | Built-in | 546 | Guest |
| Power Users | Built-in | 547 |  |
| Print Operators | Built-in | 550 |  |
| Backup Operators | Built-in | 551 |  |
| Replicator | Built-in | 552 |  |
| Remote Desktop Users | Built-in | 555 |  |
| Network Configuration Operators | Built-in | 556 |  |
| Performance Monitor Users | Built-in | 558 |  |
| Performance Log Users | Built-in | 559 |  |
| Distributed COM Users | Built-in | 562 |  |
| IIS\_IUSRS | Built-in | 568 | IUSR |
| Cryptographic Operators | Built-in | 569 |  |
| Event Log Readers | Built-in | 573 |  |

DC configuration, user accounts.

| Name | Domain | Rid | userAccountControl |
| --- | --- | --- | --- |
| Administrator | Account | 500 | UF\_NORMAL\_ACCOUNT |  UF\_DONT\_EXPIRE\_PASSWORD |
| Guest | Account | 501 | UF\_NORMAL\_ACCOUNT |  UF\_ACCOUNTDISABLE |  UF\_DONT\_EXPIRE\_PASSWORD |
| krbtgt | Account | 502 | UF\_NORMAL\_ACCOUNT |  UF\_ACCOUNTDISABLE |

DC configuration, [**universal group**](#gt_f46053d6-0708-4094-ac63-57c1bcb73d32) accounts (only on root [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca)).

| Name | Domain | Rid | Member |
| --- | --- | --- | --- |
| Schema Admins | Account | 518 | Administrator |
| Enterprise Admins | Account | 519 | Administrator |
| Enterprise Read-only Domain Controllers | Account | 498 |  |

DC configuration, group accounts.

| Name | Domain | Rid | Member |
| --- | --- | --- | --- |
| Domain Admins | Account | 512 | Administrator |
| Domain Users | Account | 513 |  |
| Domain Guests | Account | 514 | Guest |
| Domain Computers | Account | 515 |  |
| Domain Controllers | Account | 516 |  |
| Group Policy Creator Owners | Account | 520 | Administrator |
| Read-only Domain Controllers | Account | 521 |  |

DC configuration, alias accounts.

| Name | Domain | Rid | Member |
| --- | --- | --- | --- |
| Administrators | Built-in | 544 | [**Domain Admins**](#gt_d6220a40-b608-4244-8f43-c5a4a0be89ed),  Administrator,  Enterprise Admins |
| Users | Built-in | 545 | Domain Users |
| Guests | Built-in | 546 | Domain Guests,  Guest |
| Account Operators | Built-in | 548 |  |
| System Operators | Built-in | 549 |  |
| Print Operators | Built-in | 550 |  |
| Backup Operators | Built-in | 551 |  |
| Replicator | Built-in | 552 |  |
| Cert Publishers | Account | 517 |  |
| RAS and IAS Servers | Account | 553 |  |
| \* Pre-Windows 2000 operating system Compatible Access | Built-in | 554 | Everyone,  Anonymous Logon,  Authenticated Users |
| Remote Desktop Users | Built-in | 555 |  |
| Network Configuration Operators | Built-in | 556 |  |
| Incoming Forest Trust Builders | Built-in | 557 |  |
| Performance Monitor Users | Built-in | 558 |  |
| Performance Log Users | Built-in | 559 |  |
| Windows Authorization Access Group | Built-in | 560 | Enterprise Domain Controllers |
| Terminal Server License Servers | Built-in | 561 |  |
| Distributed COM Users | Built-in | 562 |  |
| IIS\_IUSRS | Built-in | 568 | IUSR |
| Cryptographic Operators | Built-in | 569 |  |
| Allowed RODC Password Replication Group | Account | 571 |  |
| Denied RODC Password Replication Group | Account | 572 | Group Policy Creator Owners,  Domain Admins,  Cert Publishers,  Domain Controllers,  Krbtgt,  Enterprise Admins,  Schema Admins,  Read-only Domain Controllers |
| Event Log Readers | Built-in | 573 |  |
| Certificate Service DCOM Access | Built-in | 574 |  |

\* The information about Pre-Windows 2000 Compatible Access is qualified by the following product behavior note.[<38>](#Appendix_A_38" \o "Product behavior note 38)

### Message Processing Events and Sequencing Rules

This section specifies the methods of the protocol along with their processing.

The return value space of all methods is the **NTSTATUS** type, specified in [[MS-ERREF]](%5bMS-ERREF%5d.pdf#Section_1bc92ddfb79e413cbbaa99a5281a6c90) section 2.3. Unless specifically called out, error codes are returned to the client of the protocol and are not handled by any special processing at the client; therefore, the exact error code is implementation-specific. Cases in which the client might handle a specific error code are called out. The set of such error codes are found in section [2.2.1.15](#Section_7440cfac6052492584e4c32e417de300).

Methods in RPC Opnum Order

| Method | Description |
| --- | --- |
| [SamrConnect](#Section_defe20910a614dfabe9a2c1206d53a1f) | Returns a handle to a [**server object**](#gt_62a8c543-5998-480b-8fa7-41a8f04a18e5).  Opnum: 0 |
| [SamrCloseHandle](#Section_55d134dfe25748ad8afacb2ca45cd3cc) | Closes any context handle obtained from this RPC interface.  Opnum: 1 |
| [SamrSetSecurityObject](#Section_6666a06658cf4118bf4bdd54ed55ecf0) | Sets the access control on a server, [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca), user, group, or [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0).  Opnum: 2 |
| [SamrQuerySecurityObject](#Section_0ecf8fecd17e4a88b7f1e0f0f66790db) | Queries the access control on a server, domain, user, group, or alias object.  Opnum: 3 |
| Opnum4NotUsedOnWire | Reserved for local use.  Opnum: 4 |
| [SamrLookupDomainInSamServer](#Section_47492d59e0954398b03e8a062b989123) | Obtains the [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) of a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).  Opnum: 5 |
| [SamrEnumerateDomainsInSamServer](#Section_2142fd2d085442c1a9fb2fe964e381ce) | Obtains a listing of all domains hosted by the server side.  Opnum: 6 |
| [SamrOpenDomain](#Section_ba710c905b1242f89e5ad4aacc1329fa) | Obtains a handle to a domain object.  Opnum: 7 |
| [SamrQueryInformationDomain](#Section_5d6a2817caa941caa269fd13ecbb4fa8) | Obtains attributes from a domain object.  Opnum: 8 |
| [SamrSetInformationDomain](#Section_9b7ae0b0bd1141339c62fba7095aee12) | Updates attributes on a domain object.  Opnum: 9 |
| [SamrCreateGroupInDomain](#Section_175c1cf94fa248379e5bbb1f0f950bee) | Creates a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) within a domain.  Opnum: 10 |
| [SamrEnumerateGroupsInDomain](#Section_e0b7a4b7ecfc405f9d7d32b3cd2cd6c8) | Enumerates all groups.  Opnum: 11 |
| [SamrCreateUserInDomain](#Section_cd0efce013fb410687db4bef73bb4b1b) | Creates a user.  Opnum: 12 |
| [SamrEnumerateUsersInDomain](#Section_6bdc92c0c6924ffb9de765858b68da75) | Enumerates all users.  Opnum: 13 |
| [SamrCreateAliasInDomain](#Section_7e505875e44f4f9d922b7ff3871c752d) | Creates an alias.  Opnum: 14 |
| [SamrEnumerateAliasesInDomain](#Section_ce340cffedef4356ace033d2874d306b) | Enumerates all aliases.  Opnum: 15 |
| [SamrGetAliasMembership](#Section_0318404522084c02b38bef955d6dc3ef) | Obtains the union of all aliases of which a given set of SIDs is a member.  Opnum: 16 |
| [SamrLookupNamesInDomain](#Section_d91271c67b2e419499278fabfa429f90) | Translates a set of account names into a set of [**RIDs**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c).  Opnum: 17 |
| [SamrLookupIdsInDomain](#Section_c870951c74b347149857224595ffc61a) | Translates a set of RIDs into account names.  Opnum: 18 |
| [SamrOpenGroup](#Section_d396e6c9d04a4729b0d8f50f2748f3c8) | Obtains a handle to a group.  Opnum: 19 |
| [SamrQueryInformationGroup](#Section_8fd0003f807640ae872f9cdb6576b3c7) | Obtains attributes from a group object.  Opnum: 20 |
| [SamrSetInformationGroup](#Section_e66db19f600a481bbc4e23953433255d) | Updates attributes on a group object.  Opnum: 21 |
| [SamrAddMemberToGroup](#Section_3c70fec36a2e48efbd4eaa2b3a1cd96a) | Adds a member to a group.  Opnum: 22 |
| [SamrDeleteGroup](#Section_664ed55f198a4775b9d4398131dba577) | Removes a group object.  Opnum: 23 |
| [SamrRemoveMemberFromGroup](#Section_613ff6635863443089238f1edc05310b) | Removes a member from a group.  Opnum: 24 |
| [SamrGetMembersInGroup](#Section_3ed5030d88a342caa6e08c12aa2fdfbd) | Reads the members of a group.  Opnum: 25 |
| [SamrSetMemberAttributesOfGroup](#Section_a4db0c2755404d2fb11108db1ee0b5a4) | Sets the attributes of a member relationship.  Opnum: 26 |
| [SamrOpenAlias](#Section_c3696b7620054eba820d91fdded3422d) | Obtains a handle to an alias.  Opnum: 27 |
| [SamrQueryInformationAlias](#Section_599d6624d93b46de827a58eb0e7527bd) | Obtains attributes from an alias object.  Opnum: 28 |
| [SamrSetInformationAlias](#Section_ba787e4e3a4b47a2aca3c3ac2d2c511e) | Updates attributes on an alias object.  Opnum: 29 |
| [SamrDeleteAlias](#Section_ecb88df184d148c8b49cd3f444943100) | Removes an alias object.  Opnum: 30 |
| [SamrAddMemberToAlias](#Section_9a5d2c35e84b4e59b7b096c6fa0fc8d7) | Adds a member to an alias.  Opnum: 31 |
| [SamrRemoveMemberFromAlias](#Section_7e99402967a244fd84397640fe5a376b) | Removes a member from an alias.  Opnum: 32 |
| [SamrGetMembersInAlias](#Section_5909af7f858b4cc7a2e4f852070d479d) | Obtains the membership list of an alias.  Opnum: 33 |
| [SamrOpenUser](#Section_0aee1c31ec404633bb560cf8429093c0) | Obtains a handle to a user.  Opnum: 34 |
| [SamrDeleteUser](#Section_4643a57956ec4c66a1ef4ab78dd21d73) | Removes a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).  Opnum: 35 |
| [SamrQueryInformationUser](#Section_4ad8d54c0d5a4d5a9e9a1bc9ee008d47) | Obtains attributes from a user object.  Opnum: 36 |
| [SamrSetInformationUser](#Section_538222f71b894811949a0eac62e38dce) | Updates attributes on a user object.  Opnum: 37 |
| [SamrChangePasswordUser](#Section_9699d8cae1a4433ca8c3d7bebeb01476) | Changes the password of a user object.  Opnum: 38 |
| [SamrGetGroupsForUser](#Section_a4adbf20040f4416a960e5b7917fdae7) | Obtains a list of groups of which a user is a member.  Opnum: 39 |
| [SamrQueryDisplayInformation](#Section_c1458942f2d54317a888abd27abad504) | Obtains a list of [**accounts**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) in name-sorted order.  Opnum: 40 |
| [SamrGetDisplayEnumerationIndex](#Section_af95abf82e1142c394aaba72f8f96c09) | Obtains an index into an account-name–sorted list of accounts.  Opnum: 41 |
| Opnum42NotUsedOnWire | Reserved for local use.  Opnum: 42 |
| Opnum43NotUsedOnWire | Reserved for local use.  Opnum: 43 |
| [SamrGetUserDomainPasswordInformation](#Section_db4bedfe465147439381d74c3ad8c41c) | Obtains select password policy information.  Opnum: 44 |
| [SamrRemoveMemberFromForeignDomain](#Section_03afc843584d473b834a3f5a1ac86cce) | Removes a member from all aliases.  Opnum: 45 |
| [SamrQueryInformationDomain2](#Section_fbdbbd7401064ca2941e36ded0be0c62) | Obtains attributes from a domain object.  Opnum: 46 |
| [SamrQueryInformationUser2](#Section_29ab27f661da4c7d863ce228ee798f4d) | Obtains attributes from a user object.  Opnum: 47 |
| [SamrQueryDisplayInformation2](#Section_9aa37aa4f77e49b199935c92ba97cd62) | Obtains a list of accounts in name-sorted order.  Opnum: 48 |
| [SamrGetDisplayEnumerationIndex2](#Section_bd429624f2d547178aa2659952c3e209) | Obtains an index into an account-name–sorted list of accounts.  Opnum: 49 |
| [SamrCreateUser2InDomain](#Section_a98d7fbb17354fbfb41aef363c899002) | Creates a user.  Opnum: 50 |
| [SamrQueryDisplayInformation3](#Section_7cd1c6d088194a5897215be588c1d4ad) | Obtains a list of accounts in name-sorted order.  Opnum: 51 |
| [SamrAddMultipleMembersToAlias](#Section_6d981290a203408094b04a26de22d8c5) | Adds multiple members to an alias.  Opnum: 52 |
| [SamrRemoveMultipleMembersFromAlias](#Section_7b2455d23f1344a7996950c240562c42) | Removes multiple members from an alias.  Opnum: 53 |
| [SamrOemChangePasswordUser2](#Section_8d0bf63efa5f4c75be22558c52075842) | Changes a user's password.  Opnum: 54 |
| [SamrUnicodeChangePasswordUser2](#Section_acb3204ada8b478e91391ea589edb880) | Changes a user account's password.  Opnum: 55 |
| [SamrGetDomainPasswordInformation](#Section_1a8841487e904a93b4c47a0006f59026) | Obtains select password policy information.  Opnum: 56 |
| [SamrConnect2](#Section_1076eb2a4f514c5aa7c7a78323b06198) | Obtains a handle to a server object.  Opnum: 57 |
| [SamrSetInformationUser2](#Section_99ee9f3943e84bbaac3a82e0c0e0699e) | Updates attributes on a user object.  Opnum: 58 |
| Opnum59NotUsedOnWire | Reserved for local use.  Opnum: 59 |
| Opnum60NotUsedOnWire | Reserved for local use.  Opnum: 60 |
| Opnum61NotUsedOnWire | Reserved for local use.  Opnum: 61 |
| [SamrConnect4](#Section_c1b14e670f4647a69fa70689d284afc9) | Obtains a handle to a server object.  Opnum: 62 |
| Opnum63NotUsedOnWire | Reserved for local use.  Opnum: 63 |
| [SamrConnect5](#Section_c842a8970a424ca5a6072afd05271dae) | Obtains a handle to a server object.  Opnum: 64 |
| [SamrRidToSid](#Section_00ff8192a4f645ba9f65917e46b6a693) | Obtains the SID of an account.  Opnum: 65 |
| [SamrSetDSRMPassword](#Section_9bcad7d2b8e14d28a03386040b7b3ce9) | Sets a local recovery password.  Opnum: 66 |
| [SamrValidatePassword](#Section_c78a7239f8fc4a42bb71321e897dc046) | Validates an application password against the locally stored policy.  Opnum: 67 |
| Opnum68NotUsedOnWire | Reserved for local use.  Opnum: 68 |
| Opnum69NotUsedOnWire | Reserved for local use.  Opnum: 69 |
| Opnum70NotUsedOnWire | Reserved for local use.  Opnum 70 |
| Opnum71NotUsedOnWire | Reserved for local use.  Opnum 71 |
| Opnum72NotUsedOnWire | Reserved for local use.  Opnum 72 |
| SamrUnicodeChangePasswordUser4 | Changes a user account password.  Opnum 73 |

In the preceding table, the phrase "Reserved for local use" means that the client MUST NOT send the opnum, and the server behavior is undefined[<39>](#Appendix_A_39" \o "Product behavior note 39) because it does not affect interoperability.

All methods MUST NOT throw exceptions.

The SAM Remote Protocol (Client-to-Server) recognizes five types of handles: Server, Domain, Group, Alias, and User. A handle of each type can be obtained only by calling one of a well-defined set of methods. These handles are listed in the following table.

| Handle type | Methods that return this type of handle |
| --- | --- |
| Server | SamrConnect  SamrConnect2  SamrConnect4  SamrConnect5 |
| Domain | SamrOpenDomain |
| Group | SamrOpenGroup |
| Alias | SamrOpenAlias |
| User | SamrOpenUser |

For example, to obtain any context handle to the server, one of the following methods MUST be called: SamrConnect, SamrConnect2, SamrConnect4, or SamrConnect5. With the ServerHandle parameter returned from these methods, it is possible to obtain other context handles and call any associated methods on the handle. See section [4.1](#Section_3d8e23d8d9df481f83b39175f980294c) for an example.

The server MUST keep track of all handles of each type that every caller opens, from the moment of creation until the handle has been closed (by calling SamrCloseHandle, SamrDeleteGroup, SamrDeleteAlias, or SamrDeleteUser) or until the client disconnects. The object referenced by a handle can be edited, queried, deleted, or closed for as long as the handle is open, but not before or after this state.

The RPC protocol provides a mechanism to clean up any resources related to a context handle if a client that is holding the context handle exits, dies, disconnects, or reboots. An implementation of this protocol SHOULD use this functionality, as specified in [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824) section 5.1.6, Context Handle Rundown.

**Note**  Except for the methods listed in the preceding table, all other methods listed in this section can be called in any sequence to perform operations on the referenced object as long as its handle is open.

**Note**  The following methods do not require a context handle and can be called directly; they also do not return any context handle:

* SamrGetDomainPasswordInformation
* SamrSetDSRMPassword
* SamrValidatePassword
* SamrOemChangePasswordUser2
* SamrUnicodeChangePasswordUser2
* [SamrUnicodeChangePasswordUser4](#Section_bbc1c5e59b814038b2b9c87d3569ed38)

**Note**  A user account MUST be enabled by clearing the UF\_ACCOUNTDISABLE bit from the **userAccountControl** attribute before that account will be able to authenticate, as specified in [[MS-KILE]](%5bMS-KILE%5d.pdf#Section_2a32282edd484ad9a542609804b02cc9) section 3.3.5.7.1.

#### Open Pattern

These methods enable a client to obtain an RPC context handle to an existing object.

See section [1.7.2](#Section_b1e11eccf41a40aca38b4b894e119cb1) for details on how to choose between SamrConnect variations.

On success, each of these methods returns a handle that references a [**database object**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f) in the server's implementation.

For a description of the "open" pattern of methods, see section [1.3](#Section_d036091033dd4e3cb5a226daf95f5ffa).

##### SamrConnect5 (Opnum 64)

The SamrConnect5 method obtains a handle to a [**server object**](#gt_62a8c543-5998-480b-8fa7-41a8f04a18e5).

1. long SamrConnect5(
2. [in, unique, string] PSAMPR\_SERVER\_NAME ServerName,
3. [in] unsigned long DesiredAccess,
4. [in] unsigned long InVersion,
5. [in] [switch\_is(InVersion)] SAMPR\_REVISION\_INFO\* InRevisionInfo,
6. [out] unsigned long\* OutVersion,
7. [out, switch\_is(\*OutVersion)] SAMPR\_REVISION\_INFO\* OutRevisionInfo,
8. [out] SAMPR\_HANDLE\* ServerHandle
9. );

**ServerName:** The null-terminated NETBIOS name of the server; this parameter MAY[<40>](#Appendix_A_40" \o "Product behavior note 40) be ignored on receipt.

**DesiredAccess:** An ACCESS\_MASK that indicates the access requested for *ServerHandle* on output. For a listing of possible values, see section [2.2.1.3](#Section_e8afb15ec0534984b84b66877236e141).

**InVersion:** Indicates which field of the *InRevisionInfo* union is used.

**InRevisionInfo:** Revision information. For details, see the definition of the [SAMPR\_REVISION\_INFO\_V1](#Section_963e60b592334669b8a885bf4f0806dc) structure, which is contained in the [SAMPR\_REVISION\_INFO](#Section_9ee412b6d5e94c6c921f6f684d4c3079) union.

**OutVersion:** Indicates which field of the *OutRevisionInfo* union is used.

**OutRevisionInfo:** Revision information. For details, see the definition of the SAMPR\_REVISION\_INFO\_V1 structure, which is contained in the SAMPR\_REVISION\_INFO union.

**ServerHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a).

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST translate the following bits in *DesiredAccess* according to the following table. Translate means to remove the "Incoming Bit" and replace with the "Translated Bits".

| Incoming bit | Translated bits |
| --- | --- |
| GENERIC\_READ | SAM\_SERVER\_READ |
| GENERIC\_WRITE | SAM\_SERVER\_WRITE |
| GENERIC\_EXECUTE | SAM\_SERVER\_EXECUTE |
| GENERIC\_ALL | SAM\_SERVER\_ALL\_ACCESS |

1. Let S be the server object in the account domain.
2. Let GrantedAccess be the union of all bits in the *DesiredAccess* column in the following table, where the client has the specified access (shown in the Access Mask column) on the **ntSecurityDescriptor** on S. [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 5.1.3.3.3 specifies how to determine the client's access.

| DesiredAccess | Access mask |
| --- | --- |
| SAM\_SERVER\_CONNECT | ACTRL\_DS\_READ\_PROP |
| SAM\_SERVER\_SHUTDOWN | ACTRL\_DS\_WRITE\_PROP |
| SAM\_SERVER\_INITIALIZE | ACTRL\_DS\_WRITE\_PROP |
| SAM\_SERVER\_CREATE\_DOMAIN | ACTRL\_DS\_WRITE\_PROP |
| SAM\_SERVER\_ENUMERATE\_DOMAINS | ACTRL\_DS\_READ\_PROP |
| SAM\_SERVER\_LOOKUP\_DOMAIN | ACTRL\_DS\_READ\_PROP |
| ACCESS\_SYSTEM\_SECURITY | ACCESS\_SYSTEM\_SECURITY |
| WRITE\_OWNER | WRITE\_OWNER |
| WRITE\_DAC | WRITE\_DAC |
| DELETE | DELETE |

1. If GrantedAccess is 0, the server MUST return STATUS\_ACCESS\_DENIED.
2. If *DesiredAccess* contains the MAXIMUM\_ALLOWED bit, the server MUST create and return a SamContextHandle (section [3.1.1.10](#Section_cc2dc6cee45f481aa8ed5d0e273336b3)) via ServerHandle, with its fields initialized as follows:
   * SamContextHandle.HandleType = "Server"
   * SamContextHandle.Object = S
   * SamContextHandle.GrantedAccess = GrantedAccess
3. If *DesiredAccess* does not contain the MAXIMUM\_ALLOWED bit, the following constraint MUST be satisfied:
   * If *DesiredAccess* contains bits not in GrantedAccess, the server MUST return STATUS\_ACCESS\_DENIED. Otherwise, the server MUST create and return a SamContextHandle (section 3.1.1.10) via ServerHandle, with its fields initialized as follows:
     + SamContextHandle.HandleType = "Server"
     + SamContextHandle.Object = S
     + SamContextHandle.GrantedAccess = *DesiredAccess*
4. If *InVersion* is not equal to 1, the server MUST return STATUS\_NOT\_SUPPORTED.
5. The server MUST set *OutVersion* to 1 and OutRevisionInfo.Revision to 3. The remaining fields of *OutRevisionInfo* MUST be set to zero.
6. If any processing error occurred, the server MUST return that error. Otherwise, the server MUST return STATUS\_SUCCESS.

##### SamrConnect4 (Opnum 62)

The SamrConnect4 method obtains a handle to a [**server object**](#gt_62a8c543-5998-480b-8fa7-41a8f04a18e5).

1. long SamrConnect4(
2. [in, unique, string] PSAMPR\_SERVER\_NAME ServerName,
3. [out] SAMPR\_HANDLE\* ServerHandle,
4. [in] unsigned long ClientRevision,
5. [in] unsigned long DesiredAccess
6. );

**ServerName:** The null-terminated NETBIOS name of the server; this parameter MAY[<41>](#Appendix_A_41" \o "Product behavior note 41) be ignored on receipt.

**ServerHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a).

**ClientRevision:** Indicates the revision (for this protocol) of the client. The value MUST be set to 2 and MUST be ignored.

**DesiredAccess:** An ACCESS\_MASK that indicates the access requested for *ServerHandle* on output. See section [2.2.1.3](#Section_e8afb15ec0534984b84b66877236e141) for a listing of possible values.

The server MUST behave as with a call to [SamrConnect5](#Section_c842a8970a424ca5a6072afd05271dae), with the following parameter values.

| Parameter name | Parameter value |
| --- | --- |
| ServerName | SamrConnect4.ServerName |
| DesiredAccess | SamrConnect4.DesiredAccess |
| InVersion | 1 |
| InRevisionInfo | SAMPR\_REVISION\_INFO\_V1.Revision = {2}  SAMPR\_REVISION\_INFO\_V1.SupportedFeatures = {0} |
| OutVersion | Output ignored |
| OutRevisionInfo | Output ignored |
| ServerHandle | SamrConnect4.ServerHandle |

##### SamrConnect2 (Opnum 57)

The SamrConnect2 method returns a handle to a [**server object**](#gt_62a8c543-5998-480b-8fa7-41a8f04a18e5).

1. long SamrConnect2(
2. [in, unique, string] PSAMPR\_SERVER\_NAME ServerName,
3. [out] SAMPR\_HANDLE\* ServerHandle,
4. [in] unsigned long DesiredAccess
5. );

**ServerName:** The null-terminated NETBIOS name of the server; this parameter MAY[<42>](#Appendix_A_42" \o "Product behavior note 42) be ignored on receipt.

**ServerHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a).

**DesiredAccess:** An ACCESS\_MASK that indicates the access requested for *ServerHandle* on output. See section [2.2.1.3](#Section_e8afb15ec0534984b84b66877236e141) for a listing of possible values.

The server MUST behave as with a call to [SamrConnect5](#Section_c842a8970a424ca5a6072afd05271dae), with the following parameter values.

| Parameter name | Parameter value |
| --- | --- |
| ServerName | SamrConnect2.ServerName |
| DesiredAccess | SamrConnect2.DesiredAccess |
| InVersion | 1 |
| InRevisionInfo | SAMPR\_REVISION\_INFO\_V1.Revision = {1}  SAMPR\_REVISION\_INFO\_V1.SupportedFeatures = {0} |
| OutVersion | Output ignored |
| OutRevisionInfo | Output ignored |
| ServerHandle | SamrConnect2.ServerHandle |

##### SamrConnect (Opnum 0)

The SamrConnect method returns a handle to a [**server object**](#gt_62a8c543-5998-480b-8fa7-41a8f04a18e5).

1. long SamrConnect(
2. [in, unique] PSAMPR\_SERVER\_NAME ServerName,
3. [out] SAMPR\_HANDLE\* ServerHandle,
4. [in] unsigned long DesiredAccess
5. );

**ServerName:** The first character of the NETBIOS name of the server; this parameter MAY[<43>](#Appendix_A_43" \o "Product behavior note 43) be ignored on receipt.

**ServerHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a).

**DesiredAccess:** An ACCESS\_MASK that indicates the access requested for *ServerHandle* upon output. See section [2.2.1.3](#Section_e8afb15ec0534984b84b66877236e141) for a listing of possible values.

The server MUST behave as with a call to [SamrConnect5](#Section_c842a8970a424ca5a6072afd05271dae), with the following parameter values.

| Parameter name | Parameter value |
| --- | --- |
| ServerName | SamrConnect.ServerName |
| DesiredAccess | SamrConnect.DesiredAccess |
| InVersion | 1 |
| InRevisionInfo | SAMPR\_REVISION\_INFO\_V1.Revision = {0}  SAMPR\_REVISION\_INFO\_V1.SupportedFeatures = {10} |
| OutVersion | Output ignored |
| OutRevisionInfo | Output ignored |
| ServerHandle | SamrConnect.ServerHandle |

##### SamrOpenDomain (Opnum 7)

The SamrOpenDomain method obtains a handle to a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1), given a [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d).

1. long SamrOpenDomain(
2. [in] SAMPR\_HANDLE ServerHandle,
3. [in] unsigned long DesiredAccess,
4. [in] PRPC\_SID DomainId,
5. [out] SAMPR\_HANDLE\* DomainHandle
6. );

**ServerHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**server object**](#gt_62a8c543-5998-480b-8fa7-41a8f04a18e5).

**DesiredAccess:** An ACCESS\_MASK. See section [2.2.1.4](#Section_aef23495f6aa48e9aebc22e022a2b4eb) for a list of [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) access values.

**DomainId:** A SID value of a domain hosted by the server side of this protocol.

**DomainHandle:** An RPC context handle, as specified in section 2.2.7.2.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints in no particular order:

1. The server MUST return an error if *ServerHandle.HandleType* is not equal to "Server".
2. *ServerHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. The server MUST translate the following bits in *DesiredAccess* according to the following table. Translate means to remove the "Incoming bit" and replace with the "Translated bits", as follows.

| Incoming bit | Translated bits |
| --- | --- |
| GENERIC\_READ | DOMAIN\_READ |
| GENERIC\_WRITE | DOMAIN\_WRITE |
| GENERIC\_EXECUTE | DOMAIN\_EXECUTE |
| GENERIC\_ALL | DOMAIN\_ALL\_ACCESS |

1. Let D be the domain object whose objectSid is *DomainId*. If no such object exists, the server MUST return an error code.
2. Let GrantedAccess be the union of all bits in the "DesiredAccess" column in the following table where the client has the specified access (shown in the "Access mask" column) on the **ntSecurityDescriptor** on D. A missing value in the "Object ACE type" column means that the [**access mask**](#gt_462f2aa7-2cfc-404a-b479-30f127c512b1) applies to the entire object. [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 5.1.3.3.3 specifies how to determine the client's access.

| DesiredAccess | Access mask | Object ACE type |
| --- | --- | --- |
| DOMAIN\_READ\_PASSWORD\_PARAMETERS | ACTRL\_DS\_READ\_PROP | c7407360-20bf-11d0-a768-00aa006e0529 |
| DOMAIN\_WRITE\_PASSWORD\_PARAMS | ACTRL\_DS\_WRITE\_PROP | c7407360-20bf-11d0-a768-00aa006e0529 |
| DOMAIN\_READ\_OTHER\_PARAMETERS | ACTRL\_DS\_READ\_PROP | b8119fd0-04f6-4762-ab7a-4986c76b3f9a |
| DOMAIN\_WRITE\_OTHER\_PARAMETERS | ACTRL\_DS\_WRITE\_PROP | b8119fd0-04f6-4762-ab7a-4986c76b3f9a |
| DOMAIN\_CREATE\_USER | Always grant, if DOMAIN\_CREATE\_USER is requested or if MAXIMUM\_ALLOWED is present. |  |
| DOMAIN\_CREATE\_GROUP | Always grant, if DOMAIN\_CREATE\_GROUP is requested or if MAXIMUM\_ALLOWED is present. The default [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) for a non-[**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration's domain object does not grant DOMAIN\_CREATE\_GROUP to any security context. |  |
| DOMAIN\_CREATE\_ALIAS | Always grant, if DOMAIN\_CREATE\_ALIAS is requested or if MAXIMUM\_ALLOWED is present. |  |
| DOMAIN\_LIST\_ACCOUNTS | ACTRL\_DS\_LIST |  |
| DOMAIN\_LOOKUP | ACTRL\_DS\_LIST |  |
| DOMAIN\_ADMINISTER\_SERVER | ACTRL\_DS\_CONTROL\_ACCESS | ab721a52-1e2f-11d0-9819-00aa0040529b |
| ACCESS\_SYSTEM\_SECURITY | ACCESS\_SYSTEM\_SECURITY |  |
| WRITE\_OWNER | WRITE\_OWNER |  |
| WRITE\_DAC | WRITE\_DAC |  |
| DELETE | DELETE |  |

1. If *GrantedAccess* is 0, the server MUST return STATUS\_ACCESS\_DENIED.
2. If *DesiredAccess* contains the MAXIMUM\_ALLOWED bit, the server MUST create and return a *SamContextHandle* (section [3.1.1.10](#Section_cc2dc6cee45f481aa8ed5d0e273336b3)) via *DomainHandle* with its fields initialized as follows:
   * *SamContextHandle.HandleType* = "Domain"
   * *SamContextHandle.Object* = D
   * *SamContextHandle.GrantedAccess* = GrantedAccess
3. If *DesiredAccess* does not contain the MAXIMUM\_ALLOWED bit, the following constraint MUST be satisfied:
   1. If *DesiredAccess* contains bits not in *GrantedAccess*, the server MUST return STATUS\_ACCESS\_DENIED. Otherwise, the server MUST create and return a *SamContextHandle* (section 3.1.1.10) via *DomainHandle* with its fields initialized as follows:
      * *SamContextHandle.HandleType* = "Domain"
      * *SamContextHandle.Object* = D
      * *SamContextHandle.GrantedAccess* = *DesiredAccess*
4. If any processing error occurred, the server MUST return that error. Otherwise, the server MUST return STATUS\_SUCCESS to the client.

##### Common Processing for Group, Alias, and User

This section specifies the message processing for [SamrOpenGroup (section 3.1.5.1.7)](#Section_d396e6c9d04a4729b0d8f50f2748f3c8), [SamrOpenAlias (section 3.1.5.1.8)](#Section_c3696b7620054eba820d91fdded3422d), and [SamrOpenUser (section 3.1.5.1.9)](#Section_0aee1c31ec404633bb560cf8429093c0). Each one of these methods specifies the following "input" parameters for this common processing:

* Target-Rid: A [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) input parameter from the message.
* Target-Object-Type: The intended object type to be opened.
* Generic-Access-Mask-Mapping-Table: A mapping from a generic access (for example, GENERIC\_READ) to a specific mapping (for example, DOMAIN\_READ for domain objects).
* Desired-Access-Mapping-Table: A table that maps access masks specific to this protocol to object [**ACE**](#gt_b581857f-39aa-4979-876b-daba67a40f15) values. An example [**access mask**](#gt_462f2aa7-2cfc-404a-b479-30f127c512b1) specific to this protocol is USER\_READ (section [2.2.1.7](#Section_c0be3f43bcf943eeb0273d02ab372c53)).
* Output-Handle: An RPC context handle returned to the client that represents the object that is requested to be opened.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *DomainHandle.HandleType* (*DomainHandle* is an input parameter from the method) is not equal to "Domain".
2. *DomainHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. The server MUST translate the bits in *DesiredAccess* according to the Generic-Access-Mask-Mapping-Table.
4. Let A be the [**database object**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f), in the [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) referenced by *DomainHandle.Object*, whose **objectSid's** RID is Target-Rid, and whose database object type is Target-Object-Type. If no such object exists, the server MUST return an error code.
5. Let *GrantedAccess* be the union of all bits in the "DesiredAccess" column in the Desired-Access-Mapping-Table, where the client has the specified access (shown in the "Access mask" column) on the **ntSecurityDescriptor** on Target-Object. A missing value in the "Object ACE type" column means that the access mask applies to the entire object. [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 5.1.3.3.3 specifies how to determine the client's access.
6. If *DesiredAccess* contains the MAXIMUM\_ALLOWED bit, the server MUST create and return a *SamContextHandle* (section [3.1.1.10](#Section_cc2dc6cee45f481aa8ed5d0e273336b3)) via Output-Handle with its fields initialized as follows:
   * *SamContextHandle.HandleType* = "User" or "Group" or "Alias", depending on the type of A
   * *SamContextHandle.Object* = A
   * *SamContextHandle.GrantedAccess* = GrantedAccess
7. If *DesiredAccess* does not contain the MAXIMUM\_ALLOWED bit, the following constraint MUST be satisfied:
   1. If *DesiredAccess* contains bits not in *GrantedAccess*, the server MUST return STATUS\_ACCESS\_DENIED. Otherwise, the server MUST create and return a *SamContextHandle* (section 3.1.1.10) via Output-Handle with its fields initialized as follows:
      * *SamContextHandle.HandleType* = "User" or "Group" or "Alias", depending on the type of A
      * *SamContextHandle.Object* = A
      * *SamContextHandle.GrantedAccess* = DesiredAccess
8. If any processing error occurred, the server MUST return that error. Otherwise, the server MUST return STATUS\_SUCCESS to the client.

##### SamrOpenGroup (Opnum 19)

The SamrOpenGroup method obtains a handle to a group, given a [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c).

1. long SamrOpenGroup(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] unsigned long DesiredAccess,
4. [in] unsigned long GroupId,
5. [out] SAMPR\_HANDLE\* GroupHandle
6. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**DesiredAccess:** An ACCESS\_MASK that indicates the requested access for the returned handle. See section [2.2.1.5](#Section_f24f9fa8798d4e7da110a5eda6900f41) for a list of group access values.

**GroupId:** A RID of a group.

**GroupHandle:** An RPC context handle, as specified in section 2.2.7.2.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message according to the constraints in section [3.1.5.1.6](#Section_fb25440b269d422099f28b29841cb1ce), with the following values:

* Target-Rid: *GroupId*
* Target-Object-Type: a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) (that is, a database with the objectClass group or derived from group) and groupType containing GROUP\_TYPE\_ACCOUNT\_GROUP or GROUP\_TYPE\_UNIVERSAL\_GROUP.
* Generic-Access-Mask-Mapping-Table:

| Incoming bit | Translated bits |
| --- | --- |
| GENERIC\_READ | GROUP\_READ |
| GENERIC\_WRITE | GROUP\_WRITE |
| GENERIC\_EXECUTE | GROUP\_EXECUTE |
| GENERIC\_ALL | GROUP\_ALL\_ACCESS |

* Desired-Access-Mapping-Table:

| DesiredAccess | Access mask | Object ACE type |
| --- | --- | --- |
| GROUP\_READ\_INFORMATION | ACTRL\_DS\_READ\_PROP | 59ba2f42-79a2-11d0-9020-00c04fc2d3cf |
| GROUP\_WRITE\_ACCOUNT | ACTRL\_DS\_WRITE\_PROP | 59ba2f42-79a2-11d0-9020-00c04fc2d3cf |
| GROUP\_ADD\_MEMBER | ACTRL\_DS\_WRITE\_PROP | bf9679c0-0de6-11d0-a285-00aa003049e2 |
| GROUP\_REMOVE\_MEMBER | ACTRL\_DS\_WRITE\_PROP | bf9679c0-0de6-11d0-a285-00aa003049e2 |
| GROUP\_LIST\_MEMBERS | ACTRL\_DS\_READ\_PROP | bf9679c0-0de6-11d0-a285-00aa003049e2 |
| ACCESS\_SYSTEM\_SECURITY | ACCESS\_SYSTEM\_SECURITY |  |
| WRITE\_OWNER | WRITE\_OWNER |  |
| WRITE\_DAC | WRITE\_DAC |  |
| DELETE | DELETE |  |

* Output-Handle: *GroupHandle*

##### SamrOpenAlias (Opnum 27)

The SamrOpenAlias method obtains a handle to an alias, given a [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c).

1. long SamrOpenAlias(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] unsigned long DesiredAccess,
4. [in] unsigned long AliasId,
5. [out] SAMPR\_HANDLE\* AliasHandle
6. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**DesiredAccess:** An ACCESS\_MASK that indicates the requested access for the returned handle. See section [2.2.1.6](#Section_2da21c6c5b1546c8bd4e6a8443216e1a) for a list of alias access values.

**AliasId:** A RID of an alias.

**AliasHandle:** An RPC context handle, as specified in section 2.2.7.2.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message according to the constraints in section [3.1.5.1.6](#Section_fb25440b269d422099f28b29841cb1ce), with the following values:

* Target-Rid: *AliasId*
* Target-Object-Type: A [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) (that is, a database with the objectClass group or derived from group) and groupType containing GROUP\_TYPE\_RESOURCE\_GROUP.
* Generic-Access-Mask-Mapping-Table:

| Incoming bit | Translated bits |
| --- | --- |
| GENERIC\_READ | ALIAS\_READ |
| GENERIC\_WRITE | ALIAS\_WRITE |
| GENERIC\_EXECUTE | ALIAS\_EXECUTE |
| GENERIC\_ALL | ALIAS\_ALL\_ACCESS |

* Desired-Access-Mapping-Table:

| DesiredAccess | Access mask | Object ACE type |
| --- | --- | --- |
| ALIAS\_READ\_INFORMATION | ACTRL\_DS\_READ\_PROP | 59ba2f42-79a2-11d0-9020-00c04fc2d3cf |
| ALIAS\_WRITE\_ACCOUNT | ACTRL\_DS\_WRITE\_PROP | 59ba2f42-79a2-11d0-9020-00c04fc2d3cf |
| ALIAS\_ADD\_MEMBER | ACTRL\_DS\_WRITE\_PROP | bf9679c0-0de6-11d0-a285-00aa003049e2 |
| ALIAS\_REMOVE\_MEMBER | ACTRL\_DS\_WRITE\_PROP | bf9679c0-0de6-11d0-a285-00aa003049e2 |
| ALIAS\_LIST\_MEMBERS | ACTRL\_DS\_READ\_PROP | bf9679c0-0de6-11d0-a285-00aa003049e2 |
| ACCESS\_SYSTEM\_SECURITY | ACCESS\_SYSTEM\_SECURITY |  |
| WRITE\_OWNER | WRITE\_OWNER |  |
| WRITE\_DAC | WRITE\_DAC |  |
| DELETE | DELETE |  |

* Output-Handle: *AliasHandle*

##### SamrOpenUser (Opnum 34)

The SamrOpenUser method obtains a handle to a user, given a [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c).

1. long SamrOpenUser(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] unsigned long DesiredAccess,
4. [in] unsigned long UserId,
5. [out] SAMPR\_HANDLE\* UserHandle
6. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**DesiredAccess:** An ACCESS\_MASK that indicates the requested access for the returned handle. See section [2.2.1.7](#Section_c0be3f43bcf943eeb0273d02ab372c53) for a list of user access values.

**UserId:** A RID of a user account.

**UserHandle:** An RPC context handle, as specified in section 2.2.7.2.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message according to the constraints in section [3.1.5.1.6](#Section_fb25440b269d422099f28b29841cb1ce), with the following values:

* Target-Rid: *UserId*
* Target-Object-Type: A [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17) (that is, a database with the objectClass user or derived from user).
* Generic-Access-Mask-Mapping-Table:

| Incoming bit | Translated bits |
| --- | --- |
| GENERIC\_READ | USER\_READ |
| GENERIC\_WRITE | USER\_WRITE |
| GENERIC\_EXECUTE | USER\_EXECUTE |
| GENERIC\_ALL | USER\_ALL\_ACCESS |

* Desired-Access-Mapping-Table:

| DesiredAccess | Access mask | Object ACE type |
| --- | --- | --- |
| USER\_READ\_GENERAL | ACTRL\_DS\_READ\_PROP | 59ba2f42-79a2-11d0-9020-00c04fc2d3cf |
| USER\_READ\_PREFERENCES | ACTRL\_DS\_READ\_PROP | 59ba2f42-79a2-11d0-9020-00c04fc2d3cf |
| USER\_READ\_LOGON | ACTRL\_DS\_READ\_PROP | 5f202010-79a5-11d0-9020-00c04fc2d4cf |
| USER\_READ\_ACCOUNT | ACTRL\_DS\_READ\_PROP | 4c164200-20c0-11d0-a768-00aa006e0529 |
| USER\_WRITE\_PREFERENCES | ACTRL\_DS\_WRITE\_PROP | 59ba2f42-79a2-11d0-9020-00c04fc2d3cf |
| USER\_WRITE\_ACCOUNT | ACTRL\_DS\_WRITE\_PROP | 59ba2f42-79a2-11d0-9020-00c04fc2d3cf |
| USER\_WRITE\_ACCOUNT | ACTRL\_DS\_WRITE\_PROP | 5f202010-79a5-11d0-9020-00c04fc2d4cf |
| USER\_WRITE\_ACCOUNT | ACTRL\_DS\_WRITE\_PROP | 4c164200-20c0-11d0-a768-00aa006e0529 |
| USER\_CHANGE\_PASSWORD | ACTRL\_DS\_CONTROL\_ACCESS | ab721a53-1e2f-11d0-9819-00aa0040529b |
| USER\_FORCE\_PASSWORD\_CHANGE | ACTRL\_DS\_CONTROL\_ACCESS | 00299570-246d-11d0-a768-00aa006e0529 |
| USER\_LIST\_GROUPS | ACTRL\_DS\_READ\_PROP | bf967991-0de6-11d0-a285-00aa003049e2 |
| USER\_READ\_GROUP\_INFORMATION | ACTRL\_DS\_READ\_PROP |  |
| USER\_WRITE\_GROUP\_INFORMATION | ACTRL\_DS\_WRITE\_PROP |  |
| ACCESS\_SYSTEM\_SECURITY | ACCESS\_SYSTEM\_SECURITY |  |
| WRITE\_OWNER | WRITE\_OWNER |  |
| WRITE\_DAC | WRITE\_DAC |  |
| DELETE | DELETE |  |

* Output-Handle: *UserHandle*

#### Enumerate Pattern

These methods enable a client to obtain a listing of all objects of a certain type. With the exception of [SamrEnumerateDomainsInSamServer](#Section_2142fd2d085442c1a9fb2fe964e381ce), which requires a server handle, these methods require a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) handle from the "open" pattern of methods (section [3.1.5.1](#Section_6d92e4991d164c3596e6c564dfd2972b)).

For a description of the "enumerate" pattern of methods, see section [1.3](#Section_d036091033dd4e3cb5a226daf95f5ffa).

##### SamrEnumerateDomainsInSamServer (Opnum 6)

The SamrEnumerateDomainsInSamServer method obtains a listing of all [**domains**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) hosted by the server side of this protocol.

1. long SamrEnumerateDomainsInSamServer(
2. [in] SAMPR\_HANDLE ServerHandle,
3. [in, out] unsigned long\* EnumerationContext,
4. [out] PSAMPR\_ENUMERATION\_BUFFER\* Buffer,
5. [in] unsigned long PreferedMaximumLength,
6. [out] unsigned long\* CountReturned
7. );

**ServerHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**server object**](#gt_62a8c543-5998-480b-8fa7-41a8f04a18e5).

**EnumerationContext:** This value is a cookie that the server can use to continue an enumeration on a subsequent call. It is an opaque value to the client. To initiate a new enumeration, the client sets *EnumerationContext* to zero. Otherwise the client sets *EnumerationContext* to a value returned by a previous call to the method.

**Buffer:** A listing of domain information, as described in section [2.2.7.10](#Section_c53161a438e84a28a33e0d378fce03dd).

**PreferedMaximumLength:** The requested maximum number of bytes to return in *Buffer*.

**CountReturned:** The count of domain elements returned in *Buffer*.

This method asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

On receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *ServerHandle.HandleType* is not equal to "Server".
2. *ServerHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. The server MUST enable a client to obtain a listing, without duplicates, of the following two values: the name attribute of the [**account domain object**](#gt_3ff707e8-0783-4b16-9370-2899f1078c87) and the name attribute of the [**built-in domain**](#gt_22904540-f2fb-470e-af48-ce1b0e9a02b8) object.
4. *EnumerationContext* MUST be used to allow the client implementation to pass back to the server, on a subsequent call, information on the last [**database object**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f) that was returned using *EnumerationContext*.
5. Servers SHOULD[<44>](#Appendix_A_44" \o "Product behavior note 44) validate that *EnumerationContext* is an expected value for the server's implementation.
6. The server SHOULD[<45>](#Appendix_A_45" \o "Product behavior note 45) fill Buffer.Buffer with as many entries as possible, such that not more than *PreferedMaximumLength* bytes are returned in Buffer.Buffer. If the server returns more than *PreferedMaximumLength* bytes, the difference between *PreferedMaximumLength* and the actual number of bytes returned MUST be less than the maximum size, in bytes, of one entry in the array Buffer.Buffer.
7. Each element of Buffer.Buffer MUST represent one database object that matches the criteria from item 2 above, and MUST be filled as follows:
   1. Buffer.Buffer.Name is the name attribute value of the database object.
   2. Buffer.Buffer.RelativeId is 0.
8. On output, *CountReturned* MUST equal Buffer.EntriesRead.
9. STATUS\_MORE\_ENTRIES MUST be returned if the server returns less than all of the database objects in Buffer.Buffer because of the *PreferedMaximumLength* restriction described above. Note that this return value is not an error status.
10. If there are no entries or Buffer.Buffer contains all matching database objects that remain, the server MUST return STATUS\_SUCCESS.

##### Common Processing for Enumeration of Users, Groups, and Aliases

This section specifies message processing that is common for [SamrEnumerateGroupsInDomain](#Section_e0b7a4b7ecfc405f9d7d32b3cd2cd6c8), [SamrEnumerateAliasesInDomain](#Section_ce340cffedef4356ace033d2874d306b), and [SamrEnumerateUsersInDomain](#Section_6bdc92c0c6924ffb9de765858b68da75). The explanation of each method specifies a filter that is used to identify which objects to return to the client; this filter is referred to in this section by the term Enumerate-Filter.

Let the term "session" refer to a set of sequential enumerate method calls made by a client, starting with an *EnumerationContext* parameter of value 0 and ending with an enumerate method that returns STATUS\_SUCCESS. The methods MUST be the same type; for example, a session is a sequence of SamrEnumerateGroupsInDomain method calls, not a SamrEnumerateGroupsInDomain method call followed by a SamrEnumerateUsersInDomain method call.

Finally, a non-normative description of *EnumerationContext* is helpful to understand the processing of this parameter. This parameter is used as a "cookie" by the server in order to communicate to itself, between method calls within a session, which [**accounts**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) have already been returned to the client.

As an example, recall that *EnumerationContext* is a 32-bit value. Because of this fact, a possible choice of cookie could be the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of the last account that was returned. Upon receiving a nonzero cookie, the server can determine the next account that needs to be returned. Note that this example depends on the server returning the accounts in RID sort order; however, this method has no constraint about sort order.

[[MS-DRSR]](%5bMS-DRSR%5d.pdf#Section_f977faaa673e4f66b9bf48c640241d47) section 4.1.11.3 has information on another IDL method, IDL\_DRSGetNT4ChangeLog, that uses a "cookie" mechanism.

Upon receipt of one of the messages, the server MUST process the data from the message, subject to the following constraints:

1. *DomainHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
2. The server MUST return an error if *DomainHandle.HandleType* is not equal to "Domain".
3. The server MUST enable a client to obtain a listing, without duplicates, of all [**database objects**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f) that satisfy the criteria of Enumerate-Filter.
4. The server MUST use *EnumerationContext* to allow the client implementation to pass back to the server, on a subsequent call, information on the last database object that was returned using *EnumerationContext*.

If an object that satisfies Enumerate-Filter is added between successive Enumerate method calls in a session, and said object has a RID that is greater than the RIDs of all objects returned in previous calls, the server MUST return said object before the enumeration is complete.

If an object that satisfies Enumerate-Filter is deleted between successive Enumerate method calls in a session, and said object has not already been returned by a previous method call in the session, the server MUST NOT return said object before the enumeration is complete.

1. The server SHOULD[<46>](#Appendix_A_46" \o "Product behavior note 46) validate that *EnumerationContext* is an expected value for the server's implementation.
2. The server SHOULD[<47>](#Appendix_A_47" \o "Product behavior note 47) fill Buffer.Buffer with as many entries as possible, such that not more than *PreferedMaximumLength* bytes are returned in Buffer.Buffer. If the server returns more than *PreferedMaximumLength* bytes, the difference between *PreferedMaximumLength* and the actual number of bytes returned MUST be less than the maximum size, in bytes, of one entry in the array Buffer.Buffer.
3. Each element of Buffer.Buffer MUST represent one database object that matches the Enumerate-Filter and MUST be set as follows:
   1. Buffer.Buffer.Name is the **sAMAccountName** attribute value of the database object.
   2. Buffer.Buffer.RelativeId is the RID of the **objectSid** attribute of the database object.
4. On output, *CountReturned* MUST equal Buffer.EntriesRead.
5. STATUS\_MORE\_ENTRIES MUST be returned if the server returns less than all of the database objects in Buffer.Buffer because of the *PreferedMaximumLength* restriction described above. Note that this return value is not an error status.
6. If there are no entries or if Buffer.Buffer contains all matching database objects that remain, the server MUST return STATUS\_SUCCESS.

##### SamrEnumerateGroupsInDomain (Opnum 11)

The SamrEnumerateGroupsInDomain method enumerates all groups.

1. long SamrEnumerateGroupsInDomain(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in, out] unsigned long\* EnumerationContext,
4. [out] PSAMPR\_ENUMERATION\_BUFFER\* Buffer,
5. [in] unsigned long PreferedMaximumLength,
6. [out] unsigned long\* CountReturned
7. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**EnumerationContext:** This value is a cookie that the server can use to continue an enumeration on a subsequent call. It is an opaque value to the client. To initiate a new enumeration, the client sets *EnumerationContext* to zero. Otherwise, the client sets *EnumerationContext* to a value returned by a previous call to the method.

**Buffer:** A list of group information, as specified in section [2.2.7.10](#Section_c53161a438e84a28a33e0d378fce03dd).

**PreferedMaximumLength:** The requested maximum number of bytes to return in *Buffer*.

**CountReturned:** The count of [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) elements returned in *Buffer*.

This method asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

This method MUST be processed per the specifications in section [3.1.5.2.2](#Section_672e23b616eb45f7a0ebf7969d56c209) using the following object selection filter:

1. The **objectClass** attribute value MUST be group or derived from group.
2. The **groupType** attribute value MUST be one of GROUP\_TYPE\_SECURITY\_UNIVERSAL or GROUP\_TYPE\_SECURITY\_ACCOUNT.
3. The **objectSid** attribute value MUST have the [**domain prefix**](#gt_9066e9dc-8275-4452-9073-daab5fd427c5) of the domain referenced by *DomainHandle*.

##### SamrEnumerateAliasesInDomain (Opnum 15)

The SamrEnumerateAliasesInDomain method enumerates all aliases.

1. long SamrEnumerateAliasesInDomain(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in, out] unsigned long\* EnumerationContext,
4. [out] PSAMPR\_ENUMERATION\_BUFFER\* Buffer,
5. [in] unsigned long PreferedMaximumLength,
6. [out] unsigned long\* CountReturned
7. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**EnumerationContext:** This value is a cookie that the server can use to continue an enumeration on a subsequent call. It is an opaque value to the client. To initiate a new enumeration the client sets *EnumerationContext* to zero. Otherwise the client sets *EnumerationContext* to a value returned by a previous call to the method.

**Buffer:** A list of alias information, as specified in section [2.2.7.10](#Section_c53161a438e84a28a33e0d378fce03dd).

**PreferedMaximumLength:** The requested maximum number of bytes to return in *Buffer*.

**CountReturned:** The count of [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) elements returned in *Buffer*.

This method asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

This method MUST be processed per the specifications in section [3.1.5.2.2](#Section_672e23b616eb45f7a0ebf7969d56c209) using the following object selection filter:

1. The **objectClass** attribute value MUST be group or derived from group.
2. The **groupType** attribute value MUST be GROUP\_TYPE\_SECURITY\_RESOURCE.
3. The **objectSid** attribute value MUST have the [**domain prefix**](#gt_9066e9dc-8275-4452-9073-daab5fd427c5) of the domain referenced by *DomainHandle*.

##### SamrEnumerateUsersInDomain (Opnum 13)

The SamrEnumerateUsersInDomain method enumerates all users.

1. long SamrEnumerateUsersInDomain(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in, out] unsigned long\* EnumerationContext,
4. [in] unsigned long UserAccountControl,
5. [out] PSAMPR\_ENUMERATION\_BUFFER\* Buffer,
6. [in] unsigned long PreferedMaximumLength,
7. [out] unsigned long\* CountReturned
8. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**EnumerationContext:** This value is a cookie that the server can use to continue an enumeration on a subsequent call. It is an opaque value to the client. To initiate a new enumeration the client sets *EnumerationContext* to zero. Otherwise the client sets *EnumerationContext* to a value returned by a previous call to the method.

**UserAccountControl:** A filter value to be used on the **userAccountControl** attribute.

**Buffer:** A list of user information, as specified in section [2.2.7.10](#Section_c53161a438e84a28a33e0d378fce03dd).

**PreferedMaximumLength:** The requested maximum number of bytes to return in *Buffer*.

**CountReturned:** The count of [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) elements returned in *Buffer*.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

This method MUST be processed per the specifications in section [3.1.5.2.2](#Section_672e23b616eb45f7a0ebf7969d56c209), using the following object selection filter:

1. The **objectClass** attribute value MUST be user or derived from user.
2. The **userAccountControl** attribute value MUST contain all the bits in the method parameter *UserAccountControl*.
3. The **objectSid** attribute value MUST have the [**domain prefix**](#gt_9066e9dc-8275-4452-9073-daab5fd427c5) of the domain referenced by *DomainHandle*.

In addition, all of the following constraints MUST be satisfied before the constraints of section 3.1.5.2.2 are satisfied:

1. If *DomainHandle.Object* is a reference to the account domain and the configuration is [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd), the client MUST have the SAM-Enumerate-Entire-Domain [**control access right**](#gt_42f6c9e0-a2b3-4bc3-9b87-fdb902e5505e) ([[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 5.1.3.2.1) on the domain's **ntSecurityDescriptor** attribute value.
2. The server MUST ignore the UF\_LOCKOUT and UF\_PASSWORD\_EXPIRED bits in the *UserAccountControl* parameter.

#### Selective Enumerate Pattern

The client use pattern for these methods is a call to [SamrGetDisplayEnumerationIndex2](#Section_bd429624f2d547178aa2659952c3e209), followed by a call to [SamrQueryDisplayInformation3](#Section_7cd1c6d088194a5897215be588c1d4ad), passing in the state returned by SamrGetDisplayEnumerationIndex2. This state is used as an index to indicate the [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) at which SamrQueryDisplayInformation3 will start its enumeration. The client can also choose to skip the call to SamrGetDisplayEnumerationIndex2 and begin the enumeration by calling SamrQueryDisplayInformation3, specifying an index of zero. With either use pattern, the client can continue the enumeration process by calling SamrQueryDisplayInformation3 repeatedly, specifying on each call the Index value of the last account returned in the previous call.

These methods require a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) handle from the "open" pattern of methods (section [3.1.5.1](#Section_6d92e4991d164c3596e6c564dfd2972b)).

The server MAY[<48>](#Appendix_A_48" \o "Product behavior note 48) cache implementation-specific details about the ongoing state of the enumeration on the domain handle; clients therefore MUST follow one of the use patterns described previously in order to produce deterministic results.

See section [1.7.2](#Section_b1e11eccf41a40aca38b4b894e119cb1) for details on how to choose between [SamrQueryDisplayInformation](#Section_c1458942f2d54317a888abd27abad504) and [SamrGetDisplayEnumerationIndex](#Section_af95abf82e1142c394aaba72f8f96c09) variations.

See section [1.3](#Section_d036091033dd4e3cb5a226daf95f5ffa) for a description of the "selective enumerate" pattern of methods.

##### SamrQueryDisplayInformation3 (Opnum 51)

The SamrQueryDisplayInformation3 method obtains a listing of accounts in ascending name-sorted order, starting at a specified index.

1. long SamrQueryDisplayInformation3(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] DOMAIN\_DISPLAY\_INFORMATION DisplayInformationClass,
4. [in] unsigned long Index,
5. [in] unsigned long EntryCount,
6. [in] unsigned long PreferredMaximumLength,
7. [out] unsigned long\* TotalAvailable,
8. [out] unsigned long\* TotalReturned,
9. [out, switch\_is(DisplayInformationClass)]
10. PSAMPR\_DISPLAY\_INFO\_BUFFER Buffer
11. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**DisplayInformationClass:** An enumeration (see section [2.2.8.12](#Section_929775e003394f0b81616d3699c18185)) that indicates the type of accounts, as well as the type of attributes on the accounts, to return via the *Buffer* parameter.

**Index:** A cursor into an account-name–sorted list of accounts.

**EntryCount:** The number of accounts that the client is requesting on output.

**PreferredMaximumLength:** The requested maximum number of bytes to return in *Buffer*; this value overrides *EntryCount* if this value is reached before *EntryCount* is reached.

**TotalAvailable:** The number of bytes required to see a complete listing of accounts specified by the *DisplayInformationClass* parameter.

**TotalReturned:** The number of bytes returned.[<49>](#Appendix_A_49" \o "Product behavior note 49)

**Buffer:** The accounts that are returned.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *DomainHandle.HandleType* is not equal to "Domain".
2. *DomainHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. This method MUST return a set of [**database objects**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f), sorted by their **sAMAccountName** attribute value, that match the following criteria for the given *DisplayInformationClass*.

| DisplayInformationClass | Database object criteria |
| --- | --- |
| DomainDisplayUser | All [**user objects**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17) (or those derived from user) in the [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) referenced by *DomainHandle.Object* with **userAccountControl** containing the UF\_NORMAL\_ACCOUNT bit. |
| DomainDisplayMachine | All user objects (or those derived from user) in the domain referenced by *DomainHandle.Object* with **userAccountControl** containing the UF\_WORKSTATION\_TRUST\_ACCOUNT or UF\_SERVER\_TRUST\_ACCOUNT bit. |
| DomainDisplayGroup | All [**group objects**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) (or those derived from group) in the domain referenced by *DomainHandle.Object* with **groupType** equal to GROUP\_TYPE\_SECURITY\_UNIVERSAL or GROUP\_TYPE\_SECURITY\_ACCOUNT. |
| DomainDisplayOemUser | All user objects (or those derived from user) in both the account domain and the [**built-in domain**](#gt_22904540-f2fb-470e-af48-ce1b0e9a02b8) with **userAccountControl** containing the UF\_NORMAL\_ACCOUNT bit. |
| DomainDisplayOemGroup | All group objects (or those derived from group) in both the account domain and the built-in domain with **groupType** equal to GROUP\_TYPE\_SECURITY\_UNIVERSAL or GROUP\_TYPE\_SECURITY\_ACCOUNT. |

1. Let L be a list of accounts, sorted by **sAMAccountName**, that match the above criteria. If the *Index* parameter is nonzero, the server MUST return objects starting from the position in L implied by the implementation-specific cookie (carried in the *Index* parameter). If the *Index* parameter is zero, the server MUST start at the beginning of L. If the implementation-specific cookie refers to an object that has been deleted since the time at which the cookie was created, the server MUST return objects, if any, starting from the next position in L.
2. For each candidate object to return, the server MUST fill an element in the *Buffer* output parameter according to the following table.

| Element field | Value |
| --- | --- |
| Index | Any unsigned integer such that there are no duplicates in the set of values returned in *Buffer*; that is, each element has a unique *Index*. There is no requirement on the ordering of Index values.[<50>](#Appendix_A_50" \o "Product behavior note 50) |
| Rid | [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of the **objectSid** attribute. |
| AccountControl | **userAccountControl** attribute value. |
| AccountName | **sAMAccountName** attribute value. |
| AdminComment | **description** attribute value. |
| FullName | **displayName** attribute value. |
| Attributes | See section [3.1.5.14.7](#Section_87988bddb1004b19874e9fce919d9114) for a message processing specification. |

A call with *DisplayInformationClass* set to DomainDisplayOemUser or DomainDisplayOemGroup MUST behave identically to a call with *DisplayInformationClass* set to DomainDisplayUser or DomainDisplayGroup, respectively, with the following exceptions:

* + - The RPC\_UNICODE\_STRING structures in the Oem cases of *DisplayInformationClass* MUST be translated to [RPC\_STRING](#Section_28330caf78c449e091c146bbfeed633b) structures using the [**OEM code page**](#gt_442ab13f-d2c1-4128-b1db-f3bea4b8224e).
    - The value returned in TotalAvailable MUST be set to zero.

1. If a processing error occurs, the server MUST return that error. Otherwise, the server MUST return STATUS\_SUCCESS.

##### SamrQueryDisplayInformation2 (Opnum 48)

The SamrQueryDisplayInformation2 method obtains a list of accounts in ascending name-sorted order, starting at a specified index.

1. long SamrQueryDisplayInformation2(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] DOMAIN\_DISPLAY\_INFORMATION DisplayInformationClass,
4. [in] unsigned long Index,
5. [in] unsigned long EntryCount,
6. [in] unsigned long PreferredMaximumLength,
7. [out] unsigned long\* TotalAvailable,
8. [out] unsigned long\* TotalReturned,
9. [out, switch\_is(DisplayInformationClass)]
10. PSAMPR\_DISPLAY\_INFO\_BUFFER Buffer
11. );

See the description of [SamrQueryDisplayInformation3 (section 3.1.5.3.1)](#Section_7cd1c6d088194a5897215be588c1d4ad) for details, because the method-interface arguments and message processing are identical.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

The server MUST behave as with a call to SamrQueryDisplayInformation3, with the following parameter values.

| Parameter name | Parameter value |
| --- | --- |
| *DomainHandle* | SamrQueryDisplayInformation2.DomainHandle |
| *DisplayInformationClass* | SamrQueryDisplayInformation2.DisplayInformationClass |
| *Index* | SamrQueryDisplayInformation2.Index |
| *EntryCount* | SamrQueryDisplayInformation2.EntryCount |
| *PreferredMaximumLength* | SamrQueryDisplayInformation2.PreferredMaximumLength |
| *TotalAvailable* | SamrQueryDisplayInformation2.TotalAvailable |
| *TotalReturned* | SamrQueryDisplayInformation2.TotalReturned |
| *Buffer* | SamrQueryDisplayInformation2.Buffer |

##### SamrQueryDisplayInformation (Opnum 40)

The SamrQueryDisplayInformation method obtains a list of accounts in ascending name-sorted order, starting at a specified index.

1. long SamrQueryDisplayInformation(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] DOMAIN\_DISPLAY\_INFORMATION DisplayInformationClass,
4. [in] unsigned long Index,
5. [in] unsigned long EntryCount,
6. [in] unsigned long PreferredMaximumLength,
7. [out] unsigned long\* TotalAvailable,
8. [out] unsigned long\* TotalReturned,
9. [out, switch\_is(DisplayInformationClass)]
10. PSAMPR\_DISPLAY\_INFO\_BUFFER Buffer
11. );

See the description of [SamrQueryDisplayInformation3 (section 3.1.5.3.1)](#Section_7cd1c6d088194a5897215be588c1d4ad) for details, because the method interface arguments and message processing are identical.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

The server MUST behave as with a call to SamrQueryDisplayInformation3, with the following parameter values.

| Parameter name | Parameter value |
| --- | --- |
| *DomainHandle* | SamrQueryDisplayInformation.DomainHandle |
| *DisplayInformationClass* | SamrQueryDisplayInformation.DisplayInformationClass |
| *Index* | SamrQueryDisplayInformation.Index |
| *EntryCount* | SamrQueryDisplayInformation.EntryCount |
| *PreferredMaximumLength* | SamrQueryDisplayInformation.PreferredMaximumLength |
| *TotalAvailable* | SamrQueryDisplayInformation.TotalAvailable |
| *TotalReturned* | SamrQueryDisplayInformation.TotalReturned |
| *Buffer* | SamrQueryDisplayInformation.Buffer |

##### SamrGetDisplayEnumerationIndex2 (Opnum 49)

The SamrGetDisplayEnumerationIndex2 method obtains an index into an ascending account-name–sorted list of [**accounts**](#gt_8d10bd9d-392e-431b-8218-742bdac19815), such that the index is the position in the list of the accounts whose account name best matches a client-provided string.

1. long SamrGetDisplayEnumerationIndex2(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] DOMAIN\_DISPLAY\_INFORMATION DisplayInformationClass,
4. [in] PRPC\_UNICODE\_STRING Prefix,
5. [out] unsigned long\* Index
6. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**DisplayInformationClass:** An enumeration indicating which set of objects to return an index into (for a subsequent [SamrQueryDisplayInformation3](#Section_7cd1c6d088194a5897215be588c1d4ad) method call).

**Prefix:** A string matched against the account name to find a starting point for an enumeration. The *Prefix* parameter enables the client to obtain a listing of an account from SamrQueryDisplayInformation3 such that the accounts are returned in alphabetical order with respect to their account name, starting with the account name that most closely matches *Prefix*. See details later in this section.

**Index:** A value to use as input to SamrQueryDisplayInformation3 in order to control the accounts that are returned from that method.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *DomainHandle.HandleType* is not equal to "Domain".
2. *DomainHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. If *DisplayInformationClass* is not one of the following values, the server MUST return an error code: DomainDisplayUser, DomainDisplayMachine, DomainDisplayGroup.
4. If no accounts exist of the type specified in DisplayInformationClass, the server MUST return STATUS\_NO\_MORE\_ENTRIES.
5. The output parameter called *Index* MUST be returned as an index into a one-based-indexed list of [**database objects**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f) sorted by their **sAMAccountName** attribute value. The index is the position of the element that just precedes the element whose **sAMAccountName** generates the longest substring match starting at the beginning of the string with the *Prefix* input parameter. If no such element exists, the server MUST return STATUS\_NO\_MORE\_ENTRIES.
6. The list of directory objects MUST correspond to *DisplayInformationClass* as follows.

| DisplayInformationClass | Database object criteria |
| --- | --- |
| DomainDisplayUser | All [**user objects**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17) (or those derived from user) with **userAccountControl** containing the UF\_NORMAL\_ACCOUNT bit. |
| DomainDisplayMachine | All user objects (or those derived from user) with **userAccountControl** containing the UF\_WORKSTATION\_TRUST\_ACCOUNT or UF\_SERVER\_TRUST\_ACCOUNT bit. |
| DomainDisplayGroup | All [**group objects**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a). |

##### SamrGetDisplayEnumerationIndex (Opnum 41)

The SamrGetDisplayEnumerationIndex method obtains an index into an ascending account-name–sorted list of accounts.

1. long SamrGetDisplayEnumerationIndex(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] DOMAIN\_DISPLAY\_INFORMATION DisplayInformationClass,
4. [in] PRPC\_UNICODE\_STRING Prefix,
5. [out] unsigned long\* Index
6. );

See the description of [SamrGetDisplayEnumerationIndex2 (section 3.1.5.3.4)](#Section_bd429624f2d547178aa2659952c3e209) for details, because the method-interface arguments and processing are identical.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

The server MUST behave as with a call to SamrGetDisplayEnumerationIndex2, with the following parameter values.

| Parameter name | Parameter value |
| --- | --- |
| *DomainHandle* | SamrGetDisplayEnumerationIndex.DomainHandle |
| *DisplayInformationClass* | SamrGetDisplayEnumerationIndex.DisplayInformationClass |
| *Prefix* | SamrGetDisplayEnumerationIndex.Prefix |
| *Index* | SamrGetDisplayEnumerationIndex.Index |

#### Create Pattern

These methods enable a client to create a group, alias, or [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17). These methods require a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) handle from the "open" pattern of methods (section [3.1.5.1](#Section_6d92e4991d164c3596e6c564dfd2972b)).

See section [1.7.2](#Section_b1e11eccf41a40aca38b4b894e119cb1) for details on how to choose between the [SamrCreateUserInDomain](#Section_cd0efce013fb410687db4bef73bb4b1b) and [SamrCreateUser2InDomain](#Section_a98d7fbb17354fbfb41aef363c899002) variations.

See section [1.3](#Section_d036091033dd4e3cb5a226daf95f5ffa) for a description of the "create" pattern of methods.

##### Common Processing for Group and Alias Creation

This section specifies message processing that is common for [SamrCreateAliasInDomain](#Section_7e505875e44f4f9d922b7ff3871c752d) and [SamrCreateGroupInDomain](#Section_175c1cf94fa248379e5bbb1f0f950bee). The explanation of each method specifies a **groupType** attribute to use during group and alias creation, and a section containing valid [**access mask**](#gt_462f2aa7-2cfc-404a-b479-30f127c512b1) values; these values are referred to in this section by the terms Provided-Group-Type and Provided-Access-Mask-Section.

Upon receiving this message, the server MUST process the data from the message, subject to the following constraints:

1. The server MUST return an error if *DomainHandle.HandleType* is not equal to "Domain".
2. *DomainHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. If *DomainHandle.Object* refers to the [**built-in domain**](#gt_22904540-f2fb-470e-af48-ce1b0e9a02b8), the server MUST abort the request and return a failure code.
4. All updates caused by this request MUST be performed in the same transaction.
5. On successful completion of this method, a new [**database object**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f) MUST be created (subsequent constraints specify attributes for this new object).
6. The following database attribute MUST be updated from the values provided in the message per the following table.

| Database attribute | Message input |
| --- | --- |
| sAMAccountName | Name |

1. The **distinguishedName** database attribute MUST be updated with a value that conforms to the constraints as specified in section [3.1.5.14.1](#Section_21b0a86b89b94f668862a25c49620e7f).
2. The **objectClass** database attribute MUST be updated with the value group.
3. The **groupType** database attribute MUST be updated with the value Provided-Group-Type.
4. The security model for object creation specified in [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 5.1.3 MUST be adhered to.
5. Granted access MUST be set to *DesiredAccess* if *DesiredAccess* contains only valid access masks, according to Provided-Access-Mask-Section and section [2.2.1.1](#Section_15b9ebf7161d4c83a672dceb2ac8c448) (common Access Masks); otherwise, the request MUST be aborted and STATUS\_ACCESS\_DENIED MUST be returned.
6. If *DesiredAccess* contains the ACCESS\_SYSTEM\_SECURITY bit, the client's [**token**](#gt_95f17071-c8f1-403a-8a92-cf87aa7d40f5) MUST be retrieved using the method described in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.3.3.4.3. The **RpcImpersonationAccessToken.Privileges[]** field MUST have the SE\_SECURITY\_NAME privilege (defined in [[MS-LSAD]](%5bMS-LSAD%5d.pdf#Section_1b5471ef4c334a91b079dfcbb82f05cc) section 3.1.1.2.1). Otherwise, the server MUST abort processing and return STATUS\_ACCESS\_DENIED.

##### SamrCreateGroupInDomain (Opnum 10)

The SamrCreateGroupInDomain method creates a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) within a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca).

1. long SamrCreateGroupInDomain(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] PRPC\_UNICODE\_STRING Name,
4. [in] unsigned long DesiredAccess,
5. [out] SAMPR\_HANDLE\* GroupHandle,
6. [out] unsigned long\* RelativeId
7. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**Name:** The value to use as the name of the group. Details on how this value maps to the data model are provided later in this section.

**DesiredAccess:** The access requested on the *GroupHandle* on output. See section [2.2.1.5](#Section_f24f9fa8798d4e7da110a5eda6900f41) for a listing of possible values.

**GroupHandle:** An RPC context handle, as specified in section 2.2.7.2.

**RelativeId:** The [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of the newly created group.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

This method MUST be processed per the specifications in section [3.1.5.4.1](#Section_8df397babdda4f638f004502375f9cf6), using a group type of GROUP\_TYPE\_SECURITY\_ACCOUNT and using [**access mask**](#gt_462f2aa7-2cfc-404a-b479-30f127c512b1) values from section 2.2.1.5.

##### SamrCreateAliasInDomain (Opnum 14)

The SamrCreateAliasInDomain method creates an alias.

1. long SamrCreateAliasInDomain(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] PRPC\_UNICODE\_STRING AccountName,
4. [in] unsigned long DesiredAccess,
5. [out] SAMPR\_HANDLE\* AliasHandle,
6. [out] unsigned long\* RelativeId
7. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**AccountName:** The value to use as the name of the alias. Details on how this value maps to the data model are provided later in this section.

**DesiredAccess:** The access requested on the *AliasHandle* on output. See section [2.2.1.6](#Section_2da21c6c5b1546c8bd4e6a8443216e1a) for a listing of possible values.

**AliasHandle:** An RPC context handle, as specified in section 2.2.7.2.

**RelativeId:** The [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of the newly created alias.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

This method MUST be processed per the specifications in section [3.1.5.4.1](#Section_8df397babdda4f638f004502375f9cf6), using a group type of GROUP\_TYPE\_SECURITY\_RESOURCE and using [**access mask**](#gt_462f2aa7-2cfc-404a-b479-30f127c512b1) values from section 2.2.1.6.

##### SamrCreateUser2InDomain (Opnum 50)

The SamrCreateUser2InDomain method creates a user.

1. long SamrCreateUser2InDomain(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] PRPC\_UNICODE\_STRING Name,
4. [in] unsigned long AccountType,
5. [in] unsigned long DesiredAccess,
6. [out] SAMPR\_HANDLE\* UserHandle,
7. [out] unsigned long\* GrantedAccess,
8. [out] unsigned long\* RelativeId
9. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**Name:** The value to use as the name of the user. See the message processing shown later in this section for details on how this value maps to the data model.

**AccountType:** A 32-bit value indicating the type of [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) to create. See the message processing shown later in this section for possible values.

**DesiredAccess:** The access requested on the *UserHandle* on output. See section [2.2.1.7](#Section_c0be3f43bcf943eeb0273d02ab372c53) for a listing of possible values.

**UserHandle:** An RPC context handle, as specified in section 2.2.7.2.

**GrantedAccess:** The access granted on *UserHandle*.

**RelativeId:** The [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of the newly created user.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *DomainHandle.HandleType* is not equal to "Domain".
2. *DomainHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. If *DomainHandle.Object* refers to the [**built-in domain**](#gt_22904540-f2fb-470e-af48-ce1b0e9a02b8), the server MUST abort the request and return a failure code.
4. The *AccountType* parameter from the message MUST be equal to exactly one value from the following list. If there is no match, an error status MUST be returned.
   * USER\_NORMAL\_ACCOUNT
   * USER\_WORKSTATION\_TRUST\_ACCOUNT
   * USER\_SERVER\_TRUST\_ACCOUNT
5. All updates caused by this request MUST be performed in the same transaction.
6. On successful completion of this method, a new [**database object**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f) MUST be created (subsequent constraints specify attributes for this new object).
7. The following database attribute MUST be updated from the values provided in the message according to the following table.

| Database attribute | Message input |
| --- | --- |
| sAMAccountName | Name |

1. The **distinguishedName** attribute MUST be updated with a value that conforms to the constraints as specified in section [3.1.5.14.1](#Section_21b0a86b89b94f668862a25c49620e7f). Let the term Container-Object be the object with the **distinguishedName** of the suffix chosen in section 3.1.5.14.1 for the new object. For a computer object, for example, Container-Object is, by default, the object with the **distinguishedName** CN=Computers,<DN of account domain object>.
2. The **objectClass** database attribute MUST be updated with a value determined as follows:
   1. If the *AccountType* parameter is USER\_WORKSTATION\_TRUST\_ACCOUNT or USER\_SERVER\_TRUST\_ACCOUNT, use computer.
   2. Otherwise, use user.
3. The client's [**token**](#gt_95f17071-c8f1-403a-8a92-cf87aa7d40f5) MUST be retrieved using the method described in [MS-RPCE] section 3.3.3.4.3.
4. The **userAccountControl** attribute MUST be updated with a value from the following table. *AccountType* is the *AccountType* parameter from the message.

| AccountType | userAccountControl |
| --- | --- |
| USER\_NORMAL\_ACCOUNT | UF\_NORMAL\_ACCOUNT | UF\_ACCOUNTDISABLE |
| USER\_WORKSTATION\_TRUST\_ACCOUNT | UF\_WORKSTATION\_TRUST\_ACCOUNT | UF\_ACCOUNTDISABLE\* |
| USER\_SERVER\_TRUST\_ACCOUNT | UF\_SERVER\_TRUST\_ACCOUNT | UF\_ACCOUNTDISABLE |

\*If all the following conditions hold true, then the **userAccountControl** attribute MUST be updated only with the UF\_WORKSTATION\_TRUST\_ACCOUNT value.

* + The *AccountType* parameter is USER\_WORKSTATION\_TRUST\_ACCOUNT.
  + The client does not have the ACTRL\_DS\_CREATE\_CHILD access on the Container-Object object.
  + The **RpcImpersonationAccessToken.Privileges[]** field has the SE\_ MACHINE\_ACCOUNT NAME privilege (defined in [[MS-LSAD]](%5bMS-LSAD%5d.pdf#Section_1b5471ef4c334a91b079dfcbb82f05cc) section 3.1.1.2.1).

1. The security model for object creation specified in [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 5.1.3 MUST NOT be adhered to.
2. If the client does not have the ACTRL\_DS\_CREATE\_CHILD access right on the Container-Object object, the client is not otherwise denied access due to an explicit DENY [**ACE**](#gt_b581857f-39aa-4979-876b-daba67a40f15)[<51>](#Appendix_A_51" \o "Product behavior note 51), and the *AccountType* parameter is USER\_WORKSTATION\_TRUST\_ACCOUNT, then:
   1. On a [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration:
      1. If the **RpcImpersonationAccessToken.Privileges[]** field does not have the SE\_MACHINE\_ACCOUNT\_NAME privilege (defined in [MS-LSAD] section 3.1.1.2.1), return a processing error.
      2. Else:
         1. Let CallerSid be RpcImpersonationAccessToken.Sids[RpcImpersonationAccessToken.UserIndex].
         2. Let CallerPrimaryGroup be RpcImpersonationAccessToken.PrimaryGroup.
         3. If CallerPrimaryGroup is not equal to DOMAIN\_GROUP\_RID\_COMPUTERS, then:
            1. The number of computer objects in the [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) with msDS-creatorSID equal to CallerSid MUST be less than the value of ms-DS-MachineAccountQuota on the [**account domain object**](#gt_3ff707e8-0783-4b16-9370-2899f1078c87). On error, abort and return a failure code.
         4. If CallerPrimaryGroup is equal to DOMAIN\_GROUP\_RID\_COMPUTERS, then[<52>](#Appendix_A_52" \o "Product behavior note 52):
            1. If the domain [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) portion of CallerSid is different from the current domain SID, return a failure code.
            2. The server MUST compute the sum of all computer objects in the domain created by CallerSid and transitively created by other computer objects created by CallerSid. This sum MUST be less than the value of ms-DS-MachineAccountQuota on the account domain object. On error, abort and return a failure code.
         5. If the previous constraints are met, then:
            1. msDS-creatorSID MUST be set to CallerSid.
            2. The owner and group of the default [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) MUST be the [**Domain Admins**](#gt_d6220a40-b608-4244-8f43-c5a4a0be89ed) SID for the domain in which the account is created.
   2. On a non-DC configuration:
      * The server MUST abort processing and return STATUS\_ACCESS\_DENIED.
3. The return parameter of *GrantedAccess* MUST be set to *DesiredAccess* if *DesiredAccess* contains only valid access masks for the [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17) (see section 2.2.1.7); otherwise, the request MUST be aborted and STATUS\_ACCESS\_DENIED MUST be returned. Additionally, on a DC configuration, if the creation occurred because of a privilege (see step 13.1), the returned *GrantedAccess* MUST be restricted by the intersection of *DesiredAccess* and the following bits:
   * DELETE
   * USER\_WRITE
   * USER\_FORCE\_PASSWORD\_CHANGE
4. If *DesiredAccess* contains the ACCESS\_SYSTEM\_SECURITY bit, the **RpcImpersonationAccessToken.Privileges[]** field MUST have the SE\_SECURITY\_NAME privilege (defined in [MS-LSAD] section 3.1.1.2.1). Otherwise, the server MUST abort processing and return STATUS\_ACCESS\_DENIED.

##### SamrCreateUserInDomain (Opnum 12)

The SamrCreateUserInDomain method creates a user.

1. long SamrCreateUserInDomain(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] PRPC\_UNICODE\_STRING Name,
4. [in] unsigned long DesiredAccess,
5. [out] SAMPR\_HANDLE\* UserHandle,
6. [out] unsigned long\* RelativeId
7. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**Name:** The value to use as the name of the user. See the message processing shown later in this section for details on how this value maps to the data model.

**DesiredAccess:** The access requested on the *UserHandle* on output. See section [2.2.1.7](#Section_c0be3f43bcf943eeb0273d02ab372c53) for a listing of possible values.

**UserHandle:** An RPC context handle, as specified in section 2.2.7.2.

**RelativeId:** The [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of the newly created user.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

The server MUST behave as with a call to [SamrCreateUser2InDomain](#Section_a98d7fbb17354fbfb41aef363c899002) with the following parameter values.

| Parameter name | Parameter value |
| --- | --- |
| *DomainHandle* | SamrCreateUserInDomain.DomainHandle |
| *Name* | SamrCreateUserInDomain.Name |
| *AccountType* | USER\_NORMAL\_ACCOUNT |
| *DesiredAccess* | SamrCreateUserInDomain.DesiredAccess |
| *UserHandle* | SamrCreateUserInDomain.UserHandle |
| *RelativeId* | SamrCreateUserInDomain.RelativeId |

#### Query Pattern

These methods enable a client to read attributes about a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca), group, alias, or [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).

A client MUST first obtain a handle to the object through an "open" or a "create" method. See sections [3.1.5.1](#Section_6d92e4991d164c3596e6c564dfd2972b) and [3.1.5.4](#Section_2214fd297c2d450fb68d6a0e97ebe48f).

See section [1.7.2](#Section_b1e11eccf41a40aca38b4b894e119cb1) for details on how to choose between [SamrQueryInformationDomain](#Section_5d6a2817caa941caa269fd13ecbb4fa8) and [SamrQueryInformationDomain2](#Section_fbdbbd7401064ca2941e36ded0be0c62) variations.

See section [1.3](#Section_d036091033dd4e3cb5a226daf95f5ffa) for a description of the "query" pattern of methods.

##### SamrQueryInformationDomain2 (Opnum 46)

The SamrQueryInformationDomain2 method obtains attributes from a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

1. long SamrQueryInformationDomain2(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] DOMAIN\_INFORMATION\_CLASS DomainInformationClass,
4. [out, switch\_is(DomainInformationClass)]
5. PSAMPR\_DOMAIN\_INFO\_BUFFER\* Buffer
6. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a domain object.

**DomainInformationClass:** An enumeration indicating which attributes to return. See section [2.2.3.16](#Section_3e8738b25df6499f907dac2471bf0281) for a listing of possible values.

**Buffer:** The requested attributes on output. See section [2.2.3.17](#Section_1adc2142dbb84554aa24010c713698bf) for structure details.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *DomainHandle.HandleType* is not equal to "Domain".
2. *DomainHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. The following information levels MUST be processed by setting the appropriate output field name to the associated database attribute, as specified in section [3.1.5.14.8](#Section_bcfc423abc0b4f21ba793df04b60db46). Processing is completed by returning 0 on success.

| DomainInformationClass |
| --- |
| DomainPasswordInformation |
| DomainLockoutInformation |
| DomainLogoffInformation |
| DomainOemInformation |
| DomainNameInformation |
| DomainModifiedInformation |
| DomainModifiedInformation2 |
| DomainReplicationInformation |

1. If *DomainInformationClass* does not meet the criteria of constraint 3, the constraints associated with the *DomainInformationClass* input value in the following subsections MUST be satisfied; if there is no subsection for the *DomainInformationClass* value, an error MUST be returned to the client.

###### DomainGeneralInformation

1. The **Buffer.General.DomainServerState** field MUST be set to DomainStateEnabled.
2. If the server is not a [**domain controller (DC)**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd), the **Buffer.General.DomainServerRole** field MUST be set to DomainServerRolePrimary.

If the server is a DC and the **fsmoRoleOwner** attribute value of the [**account domain object**](#gt_3ff707e8-0783-4b16-9370-2899f1078c87) is equal to the **distinguishedName** attribute value of the server's computer object, the **Buffer.General.DomainServerRole** field MUST be set to DomainServerRolePrimary.

Otherwise, the **Buffer.General.DomainServerRole** field MUST be set to DomainServerRoleBackup.

1. **Buffer.General.UasCompatibilityRequired** MUST be set to 1 if the **uASCompat** database attribute value on the [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1) is nonzero.
2. The **Buffer.General.UserCount** field SHOULD[<53>](#Appendix_A_53" \o "Product behavior note 53) be the count of objects with the objectClass user (or derived from user).
3. The **Buffer.General.GroupCount** field SHOULD[<54>](#Appendix_A_54" \o "Product behavior note 54) be the count of objects with the objectClass group (or derived from group) and a **groupType** attribute value of GROUP\_TYPE\_SECURITY\_ACCOUNT.
4. The **Buffer.General.AliasCount** field SHOULD[<55>](#Appendix_A_55" \o "Product behavior note 55) be the count of objects with the objectClass group (or derived from group) and a groupType attribute value of GROUP\_TYPE\_SECURITY\_RESOURCE.
5. The server MUST use the database attribute value on the directory object referred to by *DomainHandle.Object* to set the *Buffer* fields not already set in the steps above, according to the table in section [3.1.5.14.8](#Section_bcfc423abc0b4f21ba793df04b60db46).

###### DomainServerRoleInformation

If the server is not a [**domain controller (DC)**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd), the **Buffer.Role.DomainServerRole** field MUST be set to DomainServerRolePrimary.

If the server is a DC and the **fsmoRoleOwner** attribute value of the [**account domain object**](#gt_3ff707e8-0783-4b16-9370-2899f1078c87) is equal to the **distinguishedName** attribute value of the server's computer object, the **Buffer.Role.DomainServerRole** field MUST be set to DomainServerRolePrimary.

Otherwise, the **Buffer.Role.DomainServerRole** field MUST be set to DomainServerRoleBackup.

###### DomainStateInformation

The server MUST set **Buffer.State.DomainServerState** to DomainServerEnabled.

###### DomainGeneralInformation2

The server MUST process this call as two calls to [SamrQueryInformationDomain](#Section_5d6a2817caa941caa269fd13ecbb4fa8) with the information levels of DomainGeneralInformation and DomainLockoutTime, but all in the same transaction. The output fields MUST be set as follows.

| Message output field | Value |
| --- | --- |
| Buffer.General2.I1 | [SAMPR\_DOMAIN\_GENERAL\_INFORMATION](#Section_85973e1c96f24c808135b24d74ad7794) |
| Buffer.General2.LockoutDuration | [SAMPR\_DOMAIN\_LOCKOUT\_INFORMATION](#Section_c9d789edc54a4450be56251e627e1f52).LockoutDuration |
| Buffer.General2.LockoutObservationWindow | SAMPR\_DOMAIN\_LOCKOUT\_INFORMATION.LockoutObservationWindow |
| Buffer.General2.LockoutThreshold | SAMPR\_DOMAIN\_LOCKOUT\_INFORMATION.LockoutThreshold |

##### SamrQueryInformationDomain (Opnum 8)

The SamrQueryInformationDomain method obtains attributes from a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

1. long SamrQueryInformationDomain(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] DOMAIN\_INFORMATION\_CLASS DomainInformationClass,
4. [out, switch\_is(DomainInformationClass)]
5. PSAMPR\_DOMAIN\_INFO\_BUFFER\* Buffer
6. );

See the description of [SamrQueryInformationDomain2 (section 3.1.5.5.1)](#Section_fbdbbd7401064ca2941e36ded0be0c62) for details, because the method interface arguments and message processing are identical.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

The server MUST behave as with a call to SamrQueryInformationDomain2, with the following parameter values.

| Parameter name | Parameter value |
| --- | --- |
| *DomainHandle* | SamrQueryInformationDomain.DomainHandle |
| *DomainInformationClass* | SamrQueryInformationDomain.DomainInformationClass |
| *Buffer* | SamrQueryInformationDomain.Buffer |

##### SamrQueryInformationGroup (Opnum 20)

The SamrQueryInformationGroup method obtains attributes from a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a).

1. long SamrQueryInformationGroup(
2. [in] SAMPR\_HANDLE GroupHandle,
3. [in] GROUP\_INFORMATION\_CLASS GroupInformationClass,
4. [out, switch\_is(GroupInformationClass)]
5. PSAMPR\_GROUP\_INFO\_BUFFER\* Buffer
6. );

**GroupHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a group object.

**GroupInformationClass:** An enumeration indicating which attributes to return. See section [2.2.4.6](#Section_46e6300dcb8a402f918701b41b13919a) for a listing of possible values.

**Buffer:** The requested attributes on output. See section [2.2.4.7](#Section_a37ef7399b2147a789b4d237dd6491f3) for structure details.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *GroupHandle.HandleType* is not equal to "Group".
2. *GroupHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. The following information levels MUST be processed by setting the appropriate output field name to either the associated database attribute or the value resulting from the associated processing rules, as specified in section [3.1.5.14.9](#Section_95c523dd932e40869483e58d1d848ceb). Processing is completed by returning 0 on success.

| GroupInformationClass |
| --- |
| GroupGeneralInformation |
| GroupNameInformation |
| GroupAttributeInformation |
| GroupAdminCommentInformation |

1. If *GroupInformationClass* does not meet the criteria of constraint 3, the constraints associated with the *GroupInformationClass* input value in the following subsections MUST be satisfied; if there is no subsection for the *GroupInformationClass* value, an error MUST be returned to the client.

###### GroupReplicationInformation

This information level is an anomaly in that it sets the *Buffer* fields for General, whereas in the union structure of [SAMPR\_GROUP\_INFO\_BUFFER (section 2.2.4.7)](#Section_a37ef7399b2147a789b4d237dd6491f3) the information level is associated with a different field (named **DoNotUse**).

The server MUST use the database attribute value on the directory object referred to by *GroupHandle.Object* to set the outgoing method parameters as shown in the following table.

| Message output | Database attribute |
| --- | --- |
| Buffer.General.Name | sAMAccountName |
| Buffer.General.Attributes | See section [3.1.5.14.7](#Section_87988bddb1004b19874e9fce919d9114) for a message processing specification. |
| Buffer.General.AdminComment | description |
| Buffer.General.MemberCount | 0 |

##### SamrQueryInformationAlias (Opnum 28)

The SamrQueryInformationAlias method obtains attributes from an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0).

1. long SamrQueryInformationAlias(
2. [in] SAMPR\_HANDLE AliasHandle,
3. [in] ALIAS\_INFORMATION\_CLASS AliasInformationClass,
4. [out, switch\_is(AliasInformationClass)]
5. PSAMPR\_ALIAS\_INFO\_BUFFER\* Buffer
6. );

**AliasHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing an alias object.

**AliasInformationClass:** An enumeration indicating which attributes to return. See section [2.2.5.5](#Section_c57a00edd49243e79be56170e5d89150) for a listing of possible values.

**Buffer:** The requested attributes on output. See section [2.2.5.6](#Section_aaf25a5522c3463996c3ef5ae3362cf4) for structure details.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *AliasHandle.HandleType* is not equal to "Alias".
2. *AliasHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. The following information levels MUST be processed by setting the appropriate output field name to the associated database attribute, as specified in section [3.1.5.14.10](#Section_81a0becd1e834043b66507afdd6a4064). Processing is completed by returning 0 on success. If the presented information level is not in the following table, the server MUST return an error.

| AliasInformationClass |
| --- |
| AliasGeneralInformation |
| AliasNameInformation |
| AliasAdminCommentInformation |

##### SamrQueryInformationUser2 (Opnum 47)

The SamrQueryInformationUser2 method obtains attributes from a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).

1. long SamrQueryInformationUser2(
2. [in] SAMPR\_HANDLE UserHandle,
3. [in] USER\_INFORMATION\_CLASS UserInformationClass,
4. [out, switch\_is(UserInformationClass)]
5. PSAMPR\_USER\_INFO\_BUFFER\* Buffer
6. );

**UserHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a user object.

**UserInformationClass:** An enumeration indicating which attributes to return. See section [2.2.6.28](#Section_6b0dff905ac0429a93aa150334adabf6) for a list of possible values.

**Buffer:** The requested attributes on output. See section [2.2.6.29](#Section_9496c26e490b4e76827f2695fc216f35) for structure details.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *UserHandle.HandleType* is not equal to "User".
2. *UserHandle.GrantedAccess* MUST have the required access specified in [Common Processing (section 3.1.5.5.5.1)](#Section_1c78087471104e3687c67dba52543eb1).
3. If *UserInformationClass* is set to UserAllInformation, the constraints in section [3.1.5.5.5.2](#Section_c348de4f12604312acfda000019feb87) ("UserAllInformation") MUST be satisfied. Otherwise, the constraints in section 3.1.5.5.5.1 ("Common Processing") MUST be satisfied.
4. The following bits in **Buffer.All.WhichFields**, and their corresponding field values, MUST never be returned by the server.

| WhichFields bits |
| --- |
| USER\_ALL\_NTPASSWORDPRESENT  0x01000000 |
| USER\_ALL\_LMPASSWORDPRESENT  0x02000000 |
| USER\_ALL\_PRIVATEDATA  0x04000000 |
| USER\_ALL\_PASSWORDEXPIRED  0x08000000 |
| USER\_ALL\_SECURITYDESCRIPTOR  0x10000000 |

###### Common Processing

1. *UserHandle.GrantedAccess* MUST have the required access shown in the following table; on error, the server MUST return STATUS\_ACCESS\_DENIED. If there is no match on Information Level, the server MUST return an error.

| Information level | Required access |
| --- | --- |
| UserAccountInformation | USER\_READ\_GENERAL | USER\_READ\_PREFERENCES | USER\_READ\_LOGON | USER\_READ\_ACCOUNT |
| UserGeneralInformation | USER\_READ\_GENERAL |
| UserPrimaryGroupInformation | USER\_READ\_GENERAL |
| UserNameInformation | USER\_READ\_GENERAL |
| UserAccountNameInformation | USER\_READ\_GENERAL |
| UserFullNameInformation | USER\_READ\_GENERAL |
| UserAdminCommentInformation | USER\_READ\_GENERAL |
| UserPreferencesInformation | USER\_READ\_PREFERENCES | USER\_READ\_GENERAL |
| UserLogonInformation | USER\_READ\_GENERAL | USER\_READ\_PREFERENCES | USER\_READ\_LOGON | USER\_READ\_ACCOUNT |
| UserLogonHoursInformation | USER\_READ\_LOGON |
| UserHomeInformation | USER\_READ\_LOGON |
| UserScriptInformation | USER\_READ\_LOGON |
| UserProfileInformation | USER\_READ\_LOGON |
| UserWorkStationsInformation | USER\_READ\_LOGON |
| UserControlInformation | USER\_READ\_ACCOUNT |
| UserExpiresInformation | USER\_READ\_ACCOUNT |
| UserParametersInformation | USER\_READ\_ACCOUNT (\*) |

(\*) In the [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration, this handle-based check MUST be relaxed if the client has ACTRL\_DS\_READ\_PROP access on the **userParameters** attribute ([**globally unique identifier (GUID)**](#gt_f49694cc-c350-462d-ab8e-816f0103c6c1) bf967a6d-0de6-11d0-a285-00aa003049e2).

1. The message processing MUST be similar to a SamrQueryInformationUser2 call with the *UserInformationClass* parameter set to UserAllInformation (section [3.1.5.5.5.2](#Section_C348DE4F12604312ACFDA000019FEB87)); that is, similar in the manner in which the fields are set from database attributes, but different in that the only processing errors that are propagated to the client are those errors related to the fields specifically requested. On return, the requested fields MUST be set to the value of the field with the same name in the [SAMPR\_USER\_ALL\_INFORMATION](#Section_dc966b81da274daea28cec16534f1cb9) structure.

The following table shows an example for an information level of UserGeneralInformation.

| Information level: UserGeneralInformation | |
| --- | --- |
| **Field of the Buffer parameter** | **Field value (from SAMPR\_USER\_ALL)** |
| General.UserName | UserName |
| General.FullName | FullName |
| General.PrimaryGroupId | PrimaryGroupId |
| General.AdminComment | AdminComment |
| General.UserComment | UserComment |

###### UserAllInformation

1. The server MUST set the fields of **Buffer.All** based on the access granted in *UserHandle.GrantedAccess*. The following table normatively specifies the value that the server MUST set in the **Buffer.All.WhichFields** field. If *UserHandle.GrantedAccess* does not have any of the Access Granted bits from this table, the server MUST return STATUS\_ACCESS\_DENIED.

| Access granted | WhichFields |
| --- | --- |
| USER\_READ\_GENERAL | USER\_ALL\_USERNAME  USER\_ALL\_FULLNAME  USER\_ALL\_USERID  USER\_ALL\_PRIMARYGROUPID  USER\_ALL\_ADMINCOMMENT  USER\_ALL\_USERCOMMENT |
| USER\_READ\_LOGON | USER\_ALL\_HOMEDIRECTORY  USER\_ALL\_HOMEDIRECTORYDRIVE  USER\_ALL\_SCRIPTPATH  USER\_ALL\_PROFILEPATH  USER\_ALL\_WORKSTATIONS  USER\_ALL\_LASTLOGON  USER\_ALL\_LASTLOGOFF  USER\_ALL\_LOGONHOURS  USER\_ALL\_BADPASSWORDCOUNT  USER\_ALL\_LOGONCOUNT  USER\_ALL\_PASSWORDCANCHANGE  USER\_ALL\_PASSWORDMUSTCHANGE |
| USER\_READ\_ACCOUNT | USER\_ALL\_PASSWORDLASTSET  USER\_ALL\_ACCOUNTEXPIRES  USER\_ALL\_USERACCOUNTCONTROL  USER\_ALL\_PARAMETERS |
| USER\_READ\_PREFERENCES | USER\_ALL\_COUNTRYCODE  USER\_ALL\_CODEPAGE |

1. Using the tables in sections [2.2.1.8](#Section_2675c17672e04ac9ae6dcdd87b8ba520) and [3.1.5.14.11](#Section_66cf69cd731847da914f24f4960b2c36), the server MUST set the appropriate fields in the *Buffer* parameter. The first table (section 2.2.1.8) lists the WhichFields-to-field-name mapping, and the second table (section 3.1.5.14.11) specifies the field-name-to-database-attribute mapping.

##### SamrQueryInformationUser (Opnum 36)

The SamrQueryInformationUser method obtains attributes from a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).

1. long SamrQueryInformationUser(
2. [in] SAMPR\_HANDLE UserHandle,
3. [in] USER\_INFORMATION\_CLASS UserInformationClass,
4. [out, switch\_is(UserInformationClass)]
5. PSAMPR\_USER\_INFO\_BUFFER\* Buffer
6. );

See the description of [SamrQueryInformationUser2 (section 3.1.5.5.5)](#Section_29ab27f661da4c7d863ce228ee798f4d) for details, because the method interface arguments and message processing are identical.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

The server MUST behave as with a call to SamrQueryInformationUser2, with the following parameter values.

| Parameter name | Parameter value |
| --- | --- |
| *UserHandle* | SamrQueryInformationUser.UserHandle |
| *UserInformationClass* | SamrQueryInformationUser.UserInformationClass |
| *Buffer* | SamrQueryInformationUser.Buffer |

#### Set Pattern

These methods enable a client to set attributes on a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca), group, alias, or [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).

A client MUST first obtain a handle to the object through an "open" or a "create" method. See sections [3.1.5.1](#Section_6d92e4991d164c3596e6c564dfd2972b) and [3.1.5.4](#Section_2214fd297c2d450fb68d6a0e97ebe48f).

See section [1.7.2](#Section_b1e11eccf41a40aca38b4b894e119cb1) for details on how to choose between [SamrSetInformationUser](#Section_538222f71b894811949a0eac62e38dce) and [SamrSetInformationUser2](#Section_99ee9f3943e84bbaac3a82e0c0e0699e).

See section [1.3](#Section_d036091033dd4e3cb5a226daf95f5ffa) for a description of the "set" pattern of methods.

##### SamrSetInformationDomain (Opnum 9)

The SamrSetInformationDomain method updates attributes on a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

1. long SamrSetInformationDomain(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] DOMAIN\_INFORMATION\_CLASS DomainInformationClass,
4. [in, switch\_is(DomainInformationClass)]
5. PSAMPR\_DOMAIN\_INFO\_BUFFER DomainInformation
6. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a domain object.

**DomainInformationClass:** An enumeration indicating which attributes to update. See section [2.2.3.16](#Section_3e8738b25df6499f907dac2471bf0281) for a list of possible values.

**DomainInformation:** The requested attributes and values to update. See section [2.2.3.17](#Section_1adc2142dbb84554aa24010c713698bf) for structure details.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints.

1. The server MUST return an error if *DomainHandle.HandleType* is not equal to "Domain".
2. *DomainHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. The following information levels MUST be processed by setting the database attribute on the domain object associated with *DomainHandle.Object* to the associated input field-name value using the mapping in section [3.1.5.14.8](#Section_bcfc423abc0b4f21ba793df04b60db46). All updates MUST be performed in the same transaction.

| DomainInformationClass |
| --- |
| DomainLockoutInformation |
| DomainLogoffInformation |
| DomainOemInformation |
| DomainReplicationInformation |

1. If *DomainInformationClass* does not meet the criteria of constraint 3, the constraints associated with the *DomainInformationClass* input value in the following subsections MUST be satisfied. If there is no subsection for the *DomainInformationClass* value, an error MUST be returned to the client.

###### DomainServerRoleInformation

1. If the server is not a [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd), an error status MUST be returned.
2. If *DomainHandle.Object* refers to the [**built-in domain**](#gt_22904540-f2fb-470e-af48-ce1b0e9a02b8), the server MUST abort and return STATUS\_SUCCESS.
3. If **DomainInformation.Role.DomainServerRole** is not equal to DomainServerRolePrimary, STATUS\_SUCCESS MUST be returned.
4. The **fsmoRoleOwner** attribute of the [**account domain object**](#gt_3ff707e8-0783-4b16-9370-2899f1078c87) is set to the value of the **distinguishedName** attribute of the server's computer object, and any resulting processing errors MUST be returned. Otherwise, return STATUS\_SUCCESS.

###### DomainStateInformation

The server MUST return STATUS\_SUCCESS.

###### DomainPasswordInformation

1. If **DomainInformation.Password.MaxPasswordAge** or **DomainInformation.Password.MinPasswordAge** is not a valid [**delta time**](#gt_8a861b7f-5224-44ab-8de7-18ab8f3e800d), then an error MUST be returned.
2. If **DomainInformation.Password.MaxPasswordAge** is less than or equal to **DomainInformation.Password.MinPasswordAge**, then an error MUST be returned.
3. If **DomainInformation.Password.MinPasswordLength** is greater than 1024, then an error MUST be returned.
4. The operation to update the password attributes on the [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1) MUST be processed by setting the database attribute on the domain object associated with *DomainHandle.Object* to the associated input field-name value using the mapping in section [3.1.5.14.8](#Section_BCFC423ABC0B4F21BA793DF04B60DB46). All updates MUST be performed in the same transaction. Any resulting processing errors MUST be returned. Otherwise, return STATUS\_SUCCESS.

##### SamrSetInformationGroup (Opnum 21)

The SamrSetInformationGroup method updates attributes on a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a).

1. long SamrSetInformationGroup(
2. [in] SAMPR\_HANDLE GroupHandle,
3. [in] GROUP\_INFORMATION\_CLASS GroupInformationClass,
4. [in, switch\_is(GroupInformationClass)]
5. PSAMPR\_GROUP\_INFO\_BUFFER Buffer
6. );

**GroupHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a group object.

**GroupInformationClass:** An enumeration indicating which attributes to update. See section [2.2.4.6](#Section_46e6300dcb8a402f918701b41b13919a) for a listing of possible values.

**Buffer:** The requested attributes and values to update. See section [2.2.4.7](#Section_a37ef7399b2147a789b4d237dd6491f3) for structure details.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *GroupHandle.HandleType* is not equal to "Group".
2. *GroupHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. The following information levels MUST be processed by setting the database attribute on the group object associated with *GroupHandle.Object* to the associated input field-name value using the mapping in section [3.1.5.14.9](#Section_95c523dd932e40869483e58d1d848ceb). All updates MUST be performed in the same transaction.

| GroupInformationClass |
| --- |
| GroupNameInformation |
| GroupAttributeInformation |
| GroupAdminCommentInformation |

1. If *GroupInformationClass* does not meet the criteria of constraint 2, the server MUST return an error code.

##### SamrSetInformationAlias (Opnum 29)

The SamrSetInformationAlias method updates attributes on an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0).

1. long SamrSetInformationAlias(
2. [in] SAMPR\_HANDLE AliasHandle,
3. [in] ALIAS\_INFORMATION\_CLASS AliasInformationClass,
4. [in, switch\_is(AliasInformationClass)]
5. PSAMPR\_ALIAS\_INFO\_BUFFER Buffer
6. );

**AliasHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing an alias object.

**AliasInformationClass:** An enumeration indicating which attributes to update. See section [2.2.5.5](#Section_c57a00edd49243e79be56170e5d89150) for a listing of possible values.

**Buffer:** The requested attributes and values to update. See section [2.2.5.6](#Section_aaf25a5522c3463996c3ef5ae3362cf4) for structure details.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *AliasHandle.HandleType* is not equal to "Alias".
2. *AliasHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. The following information levels MUST be processed by setting the database attribute on the alias object associated with *AliasHandle.Object* to the associated input field-name value using the mapping in section [3.1.5.14.10](#Section_81a0becd1e834043b66507afdd6a4064). All updates MUST be performed in the same transaction.

| AliasInformationClass |
| --- |
| AliasNameInformation |
| AliasAdminInformation |

1. If *AliasInformationClass* does not meet the criteria of constraint 2, the server MUST return an error code.

##### SamrSetInformationUser2 (Opnum 58)

The SamrSetInformationUser2 method updates attributes on a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).

1. long SamrSetInformationUser2(
2. [in] SAMPR\_HANDLE UserHandle,
3. [in] USER\_INFORMATION\_CLASS UserInformationClass,
4. [in, switch\_is(UserInformationClass)]
5. PSAMPR\_USER\_INFO\_BUFFER Buffer
6. );

**UserHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a user object.

**UserInformationClass:** An enumeration indicating which attributes to update. See section [2.2.6.28](#Section_6b0dff905ac0429a93aa150334adabf6) for a listing of possible values.

**Buffer:** The requested attributes and values to update. See section [2.2.6.29](#Section_9496c26e490b4e76827f2695fc216f35) for structure details.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *UserHandle.HandleType* is not equal to "User".
2. *UserHandle.GrantedAccess* MUST have the required access specified in [UserAllInformation (Common) (section 3.1.5.6.4.2)](#Section_f37122b8094c4e5c96333217d8988bd8).
3. The constraints in the following sections MUST be satisfied based on the *UserInformationClass* parameter. If there is no match in the table, the constraints of section [3.1.5.6.4.1](#Section_c00ca1dd5a294655bd6f6e680e154bc9) MUST be used.

| UserInformationClass | Constraint section |
| --- | --- |
| UserAllInformation | [3.1.5.6.4.3](#Section_29f69f33d0984c12be6d6b844214ee3b) |
| UserInternal4Information | [3.1.5.6.4.4](#Section_c73bd2ffb6a64ac89ddff08798f3f356) |
| UserInternal4InformationNew | [3.1.5.6.4.5](#Section_2d20cab4a0d4401baeebb72f17403ece) |
| UserInternal8Information | [3.1.5.6.4.6](#Section_ce4e6060629d4bedad7ec464f4b989f8) |

###### Common Processing

1. If the value of *UserInformationClass* is present in the following table, the message MUST be processed exactly as a call to [SamrSetInformationUser2](#Section_99ee9f3943e84bbaac3a82e0c0e0699e) with *UserInformationClass* set to UserAllInformation and Buffer of type [SAMPR\_USER\_ALL\_INFORMATION](#Section_dc966b81da274daea28cec16534f1cb9).

| UserInformationClass value |
| --- |
| UserPreferencesInformation |
| UserLogonHoursInformation |
| UserParametersInformation |
| UserNameInformation |
| UserAccountNameInformation |
| UserFullNameInformation |
| UserPrimaryGroupInformation |
| UserHomeInformation |
| UserScriptInformation |
| UserProfileInformation |
| UserAdminCommentInformation |
| UserWorkStationsInformation |
| UserControlInformation |
| UserExpiresInformation |
| UserInternal1Information |

All SAMPR\_USER\_ALL\_INFORMATION fields with the same name as the fields in the incoming structure MUST be set with the same value. Furthermore, the **WhichFields** field MUST be updated according to the table in section [2.2.1.8](#Section_2675c17672e04ac9ae6dcdd87b8ba520). All SAMPR\_USER\_ALL\_INFORMATION fields not covered MUST be zero.

As an example, the following table shows how a request for UserPreferencesInformation MUST be handled.

| Source UserInformationClass value: UserPreferencesInformation | |
| --- | --- |
| **Target: SAMPR\_USER\_ALL\_INFORMATION** | |
| **Field name** | **Value** |
| WhichFields | USER\_ALL\_USERCOMMENT |  USER\_ALL\_COUNTRYCODE |  USER\_ALL\_CODEPAGE |
| UserComment | Preferences.UserComment |
| CountryCode | Preferences.CountryCode |
| CodePage | Preferences.CodePage |

A request for Internal1Information is a slight exception and thus is shown explicitly in the following table.

| Source UserInformationClass value: UserInternal1Information | |
| --- | --- |
| **Target: SAMPR\_USER\_ALL\_INFORMATION** | |
| **Field name** | **Value** |
| WhichFields | USER\_ALL\_NTPASSWORDPRESENT |  USER\_ALL\_LMPASSWORDPRESENT |  USER\_ALL\_PASSWORDEXPIRED |
| NtPasswordPresent | Internal1.NtPasswordPresent |
| LmPasswordPresent | Internal1.LmPasswordPresent |
| PasswordExpired | Internal1.PasswordExpired |
| LmOwfPassword.Length | 0x10 |
| LmOwfPassword.MaximumLength | 0x10 |
| LmOwfPassword.Buffer | Internal1.LmOwfPassword |
| NtOwfPassword.Length | 0x10 |
| NtOwfPassword.MaximumLength | 0x10 |
| NtOwfPassword.Buffer | Internal1.NtOwfPassword |

1. If the value of *UserInformationClass* is UserInternal5InformationNew, the message MUST be processed exactly as a call to SamrSetInformationUser2, with *UserInformationClass* set to UserInternal4InformationNew and Buffer of type [SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW](#Section_b2f614b90312421aabed10ee002ef780) with the fields set as shown in the following table. All SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW fields not covered by the table MUST be zero.

| Source UserInformationClass value: UserInternal5InformationNew | |
| --- | --- |
| **Target: SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW** | |
| **Field name** | **Value** |
| I1.WhichFields | USER\_ALL\_NTPASSWORDPRESENT |  USER\_ALL\_LMPASSWORDPRESENT |  USER\_ALL\_PASSWORDEXPIRED |
| I1.PasswordExpired | Internal5.PasswordExpired |
| UserPassword | Internal5.UserPassword |

1. If the value of *UserInformationClass* is UserInternal5Information, the message MUST be processed exactly as a call to SamrSetInformationUser2, with *UserInformationClass* set to UserInternal4Information and Buffer of type [SAMPR\_USER\_INTERNAL4\_INFORMATION](#Section_e28bf420898944fb8b08f5a7c2f2e33c) with the fields set as shown in the following table. All SAMPR\_USER\_INTERNAL4\_INFORMATION fields not covered by the table MUST be zero.

| Source UserInformationClass value: UserInternal5Information | |
| --- | --- |
| **Target: SAMPR\_USER\_INTERNAL4\_INFORMATION** | |
| **Field name** | **Value** |
| I1.WhichFields | USER\_ALL\_NTPASSWORDPRESENT |  USER\_ALL\_LMPASSWORDPRESENT |  USER\_ALL\_PASSWORDEXPIRED |
| I1.PasswordExpired | Internal5.PasswordExpired |
| UserPassword | Internal5.UserPassword |

1. If the value of **UserInformationClass** (section [2.2.6.28](#Section_6b0dff905ac0429a93aa150334adabf6)) is **UserInternal7Information**, the message MUST be processed exactly as a call to **SamrSetInformationUser2** (section 3.1.5.6.4), with **UserInformationClass** set to **UserInternal8Information** and Buffer of type SAMPR\_USER\_INTERNAL8\_INFORMATION (section [2.2.6.31](#Section_1d946f0cc09942f287ec0206d5680c2f)), and with field settings set as indicated in the following table. All SAMPR\_USER\_INTERNAL8\_INFORMATION fields not specified in the table MUST be set to zero.

| **Source UserInformationClass value: UserInternal7Information** | |
| --- | --- |
| **Target: SAMPR\_USER\_INTERNAL8\_INFORMATION** | |
| **Field name** | **Value** |
| I1.WhichFields | USER\_ALL\_NTPASSWORDPRESENT | USER\_ALL\_PASSWORDEXPIRED |
| I1.PasswordExpired | Internal7.PasswordExpired |
| UserPassword | Internal7.UserPassword |

1. If the value of *UserInformationClass* was not found in the previous four constraints, the server MUST return an error.

###### UserAllInformation (Common)

The server MUST process the message subject to the following constraints on the [SAMPR\_USER\_ALL\_INFORMATION](#Section_dc966b81da274daea28cec16534f1cb9) message parameter:

1. If the **WhichFields** field is 0 or contains any of the following bits, the server MUST abort and return an error.

| Bit |
| --- |
| USER\_ALL\_USERID |
| USER\_ALL\_PASSWORDCANCHANGE |
| USER\_ALL\_PASSWORDMUSTCHANGE |
| USER\_ALL\_UNDEFINED\_MASK |
| USER\_ALL\_LASTLOGON |
| USER\_ALL\_LASTLOGOFF |
| USER\_ALL\_BADPASSWORDCOUNT |
| USER\_ALL\_LOGONCOUNT |
| USER\_ALL\_PASSWORDLASTSET |
| USER\_ALL\_SECURITYDESCRIPTOR |
| USER\_ALL\_PRIVATEDATA |

1. The *UserHandle* MUST be granted the following access based on the value of the **WhichFields** field.

| WhichFields | Required access |
| --- | --- |
| USER\_ALL\_USERNAME | USER\_WRITE\_ACCOUNT |
| USER\_ALL\_FULLNAME | USER\_WRITE\_ACCOUNT |
| USER\_ALL\_PRIMARYGROUPID | USER\_WRITE\_ACCOUNT |
| USER\_ALL\_HOMEDIRECTORY | USER\_WRITE\_ACCOUNT |
| USER\_ALL\_HOMEDIRECTORYDRIVE | USER\_WRITE\_ACCOUNT |
| USER\_ALL\_SCRIPTPATH | USER\_WRITE\_ACCOUNT |
| USER\_ALL\_PROFILEPATH | USER\_WRITE\_ACCOUNT |
| USER\_ALL\_ADMINCOMMENT | USER\_WRITE\_ACCOUNT |
| USER\_ALL\_WORKSTATIONS | USER\_WRITE\_ACCOUNT |
| USER\_ALL\_LOGONHOURS | USER\_WRITE\_ACCOUNT |
| USER\_ALL\_ACCOUNTEXPIRES | USER\_WRITE\_ACCOUNT |
| USER\_ALL\_USERACCOUNTCONTROL | USER\_WRITE\_ACCOUNT |
| USER\_ALL\_PARAMETERS | USER\_WRITE\_ACCOUNT |
| USER\_ALL\_USERCOMMENT | USER\_WRITE\_PREFERENCES |
| USER\_ALL\_COUNTRYCODE | USER\_WRITE\_PREFERENCES |
| USER\_ALL\_CODEPAGE | USER\_WRITE\_PREFERENCES |
| USER\_ALL\_NTPASSWORDPRESENT | USER\_FORCE\_PASSWORD\_CHANGE |
| USER\_ALL\_LMPASSWORDPRESENT | USER\_FORCE\_PASSWORD\_CHANGE |
| USER\_ALL\_PASSWORDEXPIRED | USER\_FORCE\_PASSWORD\_CHANGE |

1. The server MUST update the corresponding database attributes for each bit that is present in the **WhichFields** field. In addition, the server MUST enforce that the client has ACTRL\_DS\_READ\_PROP access to the database attribute being updated, according to the *UserHandle* passed into the method. Section [2.2.1.8](#Section_2675c17672e04ac9ae6dcdd87b8ba520) specifies a WhichFields-to-field mapping, and section [3.1.5.14.11](#Section_66cf69cd731847da914f24f4960b2c36) specifies a field-to-database-attribute mapping.
2. If the USER\_ALL\_USERACCOUNTCONTROL bit is present in the **WhichFields** field, the server MUST:
   1. Enforce that the client has ACTRL\_DS\_READ\_PROP access to the database attribute of **userAccountControl**, according to the *UserHandle.GrantedAccess* passed into the method.
   2. Translate the bits according to the table in section [3.1.5.14.2](#Section_8a193181a7a249dfa8b1f689aaa6987c). If a bit does not translate, abort with a processing error.
   3. Update the **userAccountControl** attribute in the database.
3. If the USER\_ALL\_PASSWORDEXPIRED flag is present in the **WhichFields** field, the server MUST:
   1. If **Buffer.All.PasswordExpired** is nonzero, then:
      * Update the **pwdLastSet** with a value of 0.
   2. If **Buffer.All.PasswordExpired** is 0 and the value of the current time minus the **pwdLastSet** attribute is greater than the Effective-MaximumPasswordAge (see section [3.1.1.5](#Section_6da47028e678475eb015a5a15ee98212)), then:
      * Update the **pwdLastSet** attribute with a value of the current time.
   3. Enforce that this update to **pwdLastSet** MUST take precedence over any other writes to this attribute during the message processing and associated triggers.

###### UserAllInformation

The server MUST process the message subject to the following constraints:

1. All updates MUST be done in the same transaction.
2. The server MUST satisfy the constraints listed in [UserAllInformation (Common) (section 3.1.5.6.4.2)](#Section_f37122b8094c4e5c96333217d8988bd8).
3. If the USER\_ALL\_NTPASSWORDPRESENT flag is present in the **WhichFields** field, the server MUST:
   1. If **Buffer.All.NtPasswordPresent** is true:
      * Update the **unicodePwd** attribute with the (decrypted) value of **Buffer.All.NtOwfPassword.Buffer**.
   2. If **Buffer.All.NtPasswordPresent** is false:
      * Update the **unicodePwd** attribute with the [**NT hash**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f) of a zero-length string.
4. If the USER\_ALL\_LMPASSWORDPRESENT flag is present in the **WhichFields** field, the server MUST:
   1. If **Buffer.All.LmPasswordPresent** is true, update the **dBCSPwd** attribute with the (decrypted) value of **Buffer.All.LmOwfPassword.Buffer**.
   2. If **Buffer.All.LmPasswordPresent** is false, update **dBCSPwd** attribute with the [**LM hash**](#gt_58a72159-b1e9-4770-9b06-d7a6b16279a8) of a zero-length string.

###### UserInternal4Information

The server MUST process the message subject to the following constraints:

1. All updates MUST be done in the same transaction.
2. The server MUST satisfy the constraints listed in section [3.1.5.6.4.2](#Section_f37122b8094c4e5c96333217d8988bd8).
3. If the USER\_ALL\_NTPASSWORDPRESENT or USER\_ALL\_LMPASSWORDPRESENT flag is present in the **WhichFields** field, the server MUST update the **clearTextPassword** attribute with the (decrypted) value of **SAMPR\_USER\_INTERNAL4\_INFORMATION.UserPassword**, using, as the decryption key, the 16-byte SMB session key obtained as specified in section [3.1.2.4](#Section_c718b655919f43a88897be080553e746).

###### UserInternal4InformationNew

The server MUST process the message subject to the following constraints:

1. All updates MUST be done in the same transaction.
2. The server MUST satisfy the constraints listed in section [3.1.5.6.4.2](#Section_f37122b8094c4e5c96333217d8988bd8).
3. If the USER\_ALL\_NTPASSWORDPRESENT or USER\_ALL\_LMPASSWORDPRESENT flag is present in the **WhichFields** field, the server MUST update the **clearTextPassword** attribute with the (decrypted) value of **SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW.UserPassword**.

###### UserInternal8Information

The server MUST process the message subject to the following constraints:

1. All updates MUST be done in the same transaction.
2. The server MUST satisfy the constraints listed in section [3.1.5.6.4.2](#Section_f37122b8094c4e5c96333217d8988bd8).
3. If the section [2.2.1.8](#Section_2675c17672e04ac9ae6dcdd87b8ba520) **USER\_ALL\_NTPASSWORDPRESENT** or **USER\_ALL\_LMPASSWORDPRESENT** flag is present in the **WhichFields** field, the server MUST update the **clearTextPassword** attribute with the (decrypted) value of **SAMPR\_USER\_INTERNAL8\_INFORMATION.UserPassword** presented by the client in the format of SAMPR\_ENCRYPTED\_PASSWORD\_AES (section [2.2.6.32](#Section_2cbb49a2be3844908be7b0e4951bc221)), while using as the decryption key the 16-byte SMB session key obtained as specified in section [3.1.2.4](#Section_c718b655919f43a88897be080553e746) and the AES Cipher as specified in section [3.2.2.4](#Section_0fc25352662b40e79deef10c6244c4c9).
4. The value of **UserPassword.PBKDFIterations**, as specified in SAMPR\_ENCRYPTED\_PASSWORD\_AES (section 2.2.6.32), is ignored by the server.

##### SamrSetInformationUser (Opnum 37)

The SamrSetInformationUser method updates attributes on a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).

1. long SamrSetInformationUser(
2. [in] SAMPR\_HANDLE UserHandle,
3. [in] USER\_INFORMATION\_CLASS UserInformationClass,
4. [in, switch\_is(UserInformationClass)]
5. PSAMPR\_USER\_INFO\_BUFFER Buffer
6. );

See the description of [SamrSetInformationUser2 (section 3.1.5.6.4)](#Section_99ee9f3943e84bbaac3a82e0c0e0699e) for details, because the method interface arguments and message processing are identical.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

The server MUST behave as with a call to SamrSetInformationUser2, with the following parameter values.

| Parameter name | Parameter value |
| --- | --- |
| UserHandle | SamrSetInformationUser.UserHandle |
| UserInformationClass | SamrSetInformationUser.UserInformationClass |
| Buffer | SamrSetInformationUser.Buffer |

#### Delete Pattern

These methods enable a client to delete a group, alias, or [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).

A client MUST first obtain a handle to the object through an "open" or a "create" method. See sections [3.1.5.1](#Section_6d92e4991d164c3596e6c564dfd2972b) and [3.1.5.4](#Section_2214fd297c2d450fb68d6a0e97ebe48f).

See section [1.3](#Section_d036091033dd4e3cb5a226daf95f5ffa) for a description of the "delete" pattern of methods.

##### SamrDeleteGroup (Opnum 23)

The SamrDeleteGroup method removes a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a).

1. long SamrDeleteGroup(
2. [in, out] SAMPR\_HANDLE\* GroupHandle
3. );

**GroupHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a group object.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *GroupHandle.HandleType* is not equal to "Group".
2. *GroupHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. All database operations MUST occur in a single transaction.
4. Let G be the group referenced by the *GroupHandle.Object*.
5. If the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of G's **objectSid** attribute is less than 1000, an error MUST be returned.
6. In the non-[**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration, if G has any values in the member attribute, an error MUST be returned.
7. If any user in the same [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) as G has, as its **primaryGroupId** attribute, the RID of G's **objectSid** attribute, an error MUST be returned.
8. In the DC configuration, if G is a parent to another object, an error MUST be returned.[<56>](#Appendix_A_56" \o "Product behavior note 56)
9. G MUST be removed from the database.
10. The server MUST delete the **SamContextHandle** ADM element (section [3.1.1.10](#Section_cc2dc6cee45f481aa8ed5d0e273336b3)) represented by *GroupHandle*, and then MUST return 0 for the value of *GroupHandle* and a return code of STATUS\_SUCCESS.

##### SamrDeleteAlias (Opnum 30)

The SamrDeleteAlias method removes an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0).

1. long SamrDeleteAlias(
2. [in, out] SAMPR\_HANDLE\* AliasHandle
3. );

**AliasHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing an alias object.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *AliasHandle.HandleType* is not equal to "Alias".
2. *AliasHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. All database operations MUST occur in a single transaction.
4. Let A be the alias object referenced by *AliasHandle.Object*.
5. If the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of A's **objectSid** attribute value is less than 1000, an error MUST be returned.
6. In the [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration, if A is a parent to another object, an error MUST be returned.[<57>](#Appendix_A_57" \o "Product behavior note 57)
7. A MUST be removed from the database.
8. The server MUST delete the **SamContextHandle** ADM element (section [3.1.1.10](#Section_cc2dc6cee45f481aa8ed5d0e273336b3)) represented by *AliasHandle*, and then MUST return 0 for the value of *AliasHandle* and a return code of STATUS\_SUCCESS.

##### SamrDeleteUser (Opnum 35)

The SamrDeleteUser method removes a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).

1. long SamrDeleteUser(
2. [in, out] SAMPR\_HANDLE\* UserHandle
3. );

**UserHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a user object.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *UserHandle.HandleType* is not equal to "User".
2. *UserHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. Let U be the object referenced by *UserHandle.Object*.
4. All database operations MUST occur in a single transaction.
5. If the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of U's **objectSid** attribute value is less than 1000, an error MUST be returned.
6. In the [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration, if U is a parent to another object, an error MUST be returned.[<58>](#Appendix_A_58" \o "Product behavior note 58)
7. U MUST be removed from the database.
8. The server MUST delete the **SamContextHandle** ADM element (section [3.1.1.10](#Section_cc2dc6cee45f481aa8ed5d0e273336b3)) represented by *UserHandle*, and then MUST return 0 for the value of *UserHandle* and a return code of STATUS\_SUCCESS.

#### Membership Pattern

These methods enable a client to set and query the membership of a group or alias.

A client MUST first obtain a handle to the group or [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0) through an "open" or a "create" method. See sections [3.1.5.1](#Section_6d92e4991d164c3596e6c564dfd2972b) and [3.1.5.4](#Section_2214fd297c2d450fb68d6a0e97ebe48f).

See section [1.3](#Section_d036091033dd4e3cb5a226daf95f5ffa) for a description of the "membership" pattern of methods.

##### SamrAddMemberToGroup (Opnum 22)

The SamrAddMemberToGroup method adds a member to a group.

1. long SamrAddMemberToGroup(
2. [in] SAMPR\_HANDLE GroupHandle,
3. [in] unsigned long MemberId,
4. [in] unsigned long Attributes
5. );

**GroupHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a).

**MemberId:** A [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) representing an [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) to add to the group's membership list.

**Attributes:** The characteristics of the membership relationship. See section [2.2.1.10](#Section_9e093bd2e4514dd5970097b977d7ebb2) for legal values and semantics.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *GroupHandle.HandleType* is not equal to "Group".
2. *GroupHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. All database operations MUST occur in a single transaction.
4. Let G be the group referenced by *GroupHandle.Object*.
5. Let TargetSid be the [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) composed by making the MemberId a suffix to the [**domain prefix**](#gt_9066e9dc-8275-4452-9073-daab5fd427c5) of G's objectSid.
6. If there is no object whose **objectSid** attribute is TargetSid, the server MUST return STATUS\_NO\_SUCH\_USER.
7. If G's member attribute already has as a [**dsname**](#gt_4d5e1f08-aa00-4dde-9411-7dd6e09ed85a) value that references the object whose **objectSid** is TargetSid, the server MUST return an error.
8. G's member attribute MUST be updated to add a dsname value that references the object with the **objectSid** value TargetSid.
9. The message processing specified in section [3.1.5.14.7](#Section_87988bddb1004b19874e9fce919d9114) for the *Attributes* parameter MUST be adhered to.

##### SamrRemoveMemberFromGroup (Opnum 24)

The SamrRemoveMemberFromGroup method removes a member from a group.

1. long SamrRemoveMemberFromGroup(
2. [in] SAMPR\_HANDLE GroupHandle,
3. [in] unsigned long MemberId
4. );

**GroupHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a).

**MemberId:** A [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) representing an [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) to remove from the group's membership list.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *GroupHandle.HandleType* is not equal to "Group".
2. *GroupHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. All database operations MUST occur in a single transaction.
4. Let G be the group referenced by the *GroupHandle.Object*.
5. Let TargetSid be the [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) composed by making the MemberId a suffix to the [**domain prefix**](#gt_9066e9dc-8275-4452-9073-daab5fd427c5) of G's objectSid.
6. If G's member attribute does not have a [**dsname**](#gt_4d5e1f08-aa00-4dde-9411-7dd6e09ed85a) value that references the object whose **objectSid** is TargetSid, the server MUST return an error.
7. G's member attribute MUST be updated to remove a dsname value that references the object with the **objectSid** value TargetSid.

##### SamrGetMembersInGroup (Opnum 25)

The SamrGetMembersInGroup method reads the members of a group.

1. long SamrGetMembersInGroup(
2. [in] SAMPR\_HANDLE GroupHandle,
3. [out] PSAMPR\_GET\_MEMBERS\_BUFFER\* Members
4. );

**GroupHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a).

**Members:** A structure containing an array of [**RIDs**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c), as well as an array of attribute values.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *GroupHandle.HandleType* is not equal to "Group".
2. *GroupHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. Let G be the group object referenced by *GroupHandle.Object*.
4. Let M be the set of values of G's member attribute such that the groupType of the object referenced by each value is GROUP\_TYPE\_SECURITY\_ACCOUNT or GROUP\_TYPE\_SECURITY\_UNIVERSAL. Objects with groupType GROUP\_TYPE\_SECURITY\_RESOURCE are ignored.
5. If the [**domain prefix**](#gt_9066e9dc-8275-4452-9073-daab5fd427c5) of the **objectSid** attribute of any object in set M is different from the domain prefix of G's **objectSid**, the server SHOULD[<59>](#Appendix_A_59" \o "Product behavior note 59) return STATUS\_DS\_GLOBAL\_CANT\_HAVE\_CROSSDOMAIN\_MEMBER.
6. On output:
   1. **Members.MemberCount** MUST be equal to the number of values in M.
   2. The **Members.Members** array MUST contain the RelativeIds of the **objectSid** attribute values for all objects in set M.
   3. For each element in the **Members.Members** array, see section [3.1.5.14.7](#Section_87988bddb1004b19874e9fce919d9114) for a message processing specification of each element in the **Members.Attributes** array.

##### SamrAddMemberToAlias (Opnum 31)

The SamrAddMemberToAlias method adds a member to an alias.

1. long SamrAddMemberToAlias(
2. [in] SAMPR\_HANDLE AliasHandle,
3. [in] PRPC\_SID MemberId
4. );

**AliasHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0).

**MemberId:** The [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) of an [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) to add to the alias.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *AliasHandle.HandleType* is not equal to "Alias".
2. *AliasHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90) Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. All database operations MUST occur in a single transaction.
4. Let A be the alias referenced by *AliasHandle.Object*.
5. If the [**domain prefix**](#gt_9066e9dc-8275-4452-9073-daab5fd427c5) of MemberId is the same domain prefix as the account domain and there is no object whose **objectSid** attribute is MemberId, the server MUST return an error.
6. If A's member attribute already has a [**dsname**](#gt_4d5e1f08-aa00-4dde-9411-7dd6e09ed85a) value that references the object whose **objectSid** is MemberId, the server MUST return an error.
7. A's member attribute MUST be updated to add a dsname value that references the object with the **objectSid** value MemberId.

##### SamrRemoveMemberFromAlias (Opnum 32)

The SamrRemoveMemberFromAlias method removes a member from an alias.

1. long SamrRemoveMemberFromAlias(
2. [in] SAMPR\_HANDLE AliasHandle,
3. [in] PRPC\_SID MemberId
4. );

**AliasHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0).

**MemberId:** The [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) of an [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) to remove from the alias.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *AliasHandle.HandleType* is not equal to "Alias".
2. *AliasHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. All database operations MUST occur in a single transaction.
4. Let A be the alias object referenced by the *AliasHandle.Object*.
5. If A's member attribute does not have a [**dsname**](#gt_4d5e1f08-aa00-4dde-9411-7dd6e09ed85a) value that references the object whose **objectSid** is MemberId, the server MUST return an error.
6. A's member attribute MUST be updated to remove a dsname value that references the object with the **objectSid** value MemberId.

##### SamrGetMembersInAlias (Opnum 33)

The SamrGetMembersInAlias method obtains the membership list of an alias.

1. long SamrGetMembersInAlias(
2. [in] SAMPR\_HANDLE AliasHandle,
3. [out] PSAMPR\_PSID\_ARRAY\_OUT Members
4. );

**AliasHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0).

**Members:** A structure containing an array of [**SIDs**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) that represent the membership list of the alias referenced by *AliasHandle*.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *AliasHandle.HandleType* is not equal to "Alias".
2. *AliasHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. On output, **Members.Count** MUST be equal to the number of values in the member attribute, and **Members.Sids** MUST have **Member.Count** number of elements. Each element MUST contain the **objectSid** value of the object referenced in the member attribute.

##### SamrRemoveMemberFromForeignDomain (Opnum 45)

The SamrRemoveMemberFromForeignDomain method removes a member from all aliases.

1. long SamrRemoveMemberFromForeignDomain(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] PRPC\_SID MemberSid
4. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**MemberSid:** The [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) to remove from the membership.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *DomainHandle.HandleType* is not equal to "Domain".
2. *DomainHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. All database operations MUST occur in a single transaction.
4. If the server is not a [**domain controller**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd), for all [**alias objects**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0) in the [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) referenced by *DomainHandle.Object*, the server MUST remove any member value that references the object with the **objectSid** attribute value of MemberSid.
5. If the server is a domain controller, the server MUST return STATUS\_SUCCESS without making any modifications to any alias objects.

##### SamrAddMultipleMembersToAlias (Opnum 52)

The SamrAddMultipleMembersToAlias method adds multiple members to an alias.

1. long SamrAddMultipleMembersToAlias(
2. [in] SAMPR\_HANDLE AliasHandle,
3. [in] PSAMPR\_PSID\_ARRAY MembersBuffer
4. );

**AliasHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0).

**MembersBuffer:** A structure containing a list of [**SIDs**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) to add as members to the alias.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

The server MUST behave as with N successive message calls to [SamrAddMemberToAlias](#Section_9a5d2c35e84b4e59b7b096c6fa0fc8d7), once for each SID value in *MembersBuffer*, where *MembersBuffer* contains N elements. The server MUST ignore the processing error of a member value already being present in the member attribute and abort the request on any other processing error.

##### SamrRemoveMultipleMembersFromAlias (Opnum 53)

The SamrRemoveMultipleMembersFromAlias method removes multiple members from an alias.

1. long SamrRemoveMultipleMembersFromAlias(
2. [in] SAMPR\_HANDLE AliasHandle,
3. [in] PSAMPR\_PSID\_ARRAY MembersBuffer
4. );

**AliasHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0).

**MembersBuffer:** A structure containing a list of [**SIDs**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) to remove from the alias's membership list.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

The server MUST behave as with N successive message calls to [SamrRemoveMemberFromAlias](#Section_7e99402967a244fd84397640fe5a376b), once for each SID value in *MembersBuffer*, where *MembersBuffer* contains N elements. The server MUST ignore the processing error triggered by a value not existing in the member attribute's values and abort the request on any other processing error.

#### Membership-Of Pattern

These methods enable a client to obtain the group membership of a user or the alias membership of a set of [**SIDs**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d). In [**mixed mode**](#gt_06c1c70e-f2c6-4efd-bff8-474409e69660) [**domains**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca), these methods are useful in approximating the authorization data associated with an authentication request for a given user. However, in [**native mode**](#gt_8adcb510-b25d-459e-9832-90079803d3c0) domains, these methods are not accurate because the authorization building process is more complex than what these methods enable. This means that in native mode domains, these methods MUST NOT be used to approximate the authorization data for a given user accessing a resource.

A client MUST first obtain a handle to the user or domain, depending on the method.

See section [1.3](#Section_d036091033dd4e3cb5a226daf95f5ffa) for a description of the "membership-of" pattern of methods.

##### SamrGetGroupsForUser (Opnum 39)

The SamrGetGroupsForUser method obtains a listing of groups that a user is a member of.

1. long SamrGetGroupsForUser(
2. [in] SAMPR\_HANDLE UserHandle,
3. [out] PSAMPR\_GET\_GROUPS\_BUFFER\* Groups
4. );

**UserHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).

**Groups:** An array of [**RIDs**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of the groups that the user referenced by *UserHandle* is a member of.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *UserHandle.HandleType* is not equal to "User".
2. *UserHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. The server MUST determine the union of all [**database objects**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f) that meet the following criteria:

* They are of class group.
* Their groupType is GROUP\_TYPE\_SECURITY\_ACCOUNT or GROUP\_TYPE\_SECURITY\_UNIVERSAL.
* Their member value contains the [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) of the user referenced by *UserHandle.Object*.
* They are in the same [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) as the user referenced by *UserHandle.Object*.

The union MUST also contain the group identified by the **primaryGroupId** attribute of the user that is referenced by *UserHandle.Object*.

1. The returned **Groups.MembershipCount** MUST be set to the cardinality that the union determined from step 3.
2. For each group in the union determined from step 3, the server MUST set a corresponding element in **Groups.Groups** as follows:
   1. **RelativeId** MUST contain the RID of the SID of the [**dsname**](#gt_4d5e1f08-aa00-4dde-9411-7dd6e09ed85a) member value.
   2. Set the **Attributes** field according to the message processing rules in section [3.1.5.14.7](#Section_87988bddb1004b19874e9fce919d9114).

##### SamrGetAliasMembership (Opnum 16)

The SamrGetAliasMembership method obtains the union of all aliases that a given set of [**SIDs**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) is a member of.

1. long SamrGetAliasMembership(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in] PSAMPR\_PSID\_ARRAY SidArray,
4. [out] PSAMPR\_ULONG\_ARRAY Membership
5. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**SidArray:** A list of SIDs.

**Membership:** The union of all aliases (represented by [**RIDs**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c)) that all SIDs in SidArray are a member of.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *DomainHandle.HandleType* is not equal to "Domain".
2. *DomainHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. For each SID value in SidArray, the server MUST determine the union of all [**database objects**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f) in the [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) referenced by *DomainHandle.Object* with class group and groupType GROUP\_TYPE\_SECURITY\_RESOURCE whose member value contains the SID.
4. The returned Membership parameter MUST contain the RIDs of the **objectSid** attribute of the union of all groups found by constraint 2.

#### Change Password Pattern

The "change password" methods enable a client to change the password of a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17). All these methods require that the client has knowledge of the current password in order for the message to be processed successfully.

It is important to note that [SamrChangePasswordUser](#Section_9699d8cae1a4433ca8c3d7bebeb01476) requires a handle to a user object (obtained through an "open" or a "create" method, sections [3.1.5.1](#Section_6d92e4991d164c3596e6c564dfd2972b) and [3.1.5.4](#Section_2214fd297c2d450fb68d6a0e97ebe48f)) and therefore requires an authentication connect. [SamrUnicodeChangePasswordUser2](#Section_acb3204ada8b478e91391ea589edb880) and [SamrOemChangePasswordUser2](#Section_8d0bf63efa5f4c75be22558c52075842) do not require any handle and can be sent directly to the targeted server using no security or by authenticating as anonymous. This characteristic allows end users, whose passwords have expired and therefore cannot logon, to change their passwords without an authenticated connection. See section [1.3](#Section_d036091033dd4e3cb5a226daf95f5ffa) for a description of the "change password" pattern of methods.

In the following descriptions, when a value is said to be "presented by the client", that value is provided by the client side of the protocol. In a canonical password-change scenario, an end user enters his or her old and new passwords into a password-change application. That application acts as a client for this method.

To encrypt password data, these methods use the fact that the client (an end user in the canonical scenario) and the server (a [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) in the canonical scenario) share a common secret: the user's existing password. The [**LM**](#gt_58a72159-b1e9-4770-9b06-d7a6b16279a8) and/or [**NT hash**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f) (specified in the following sections) of the existing password's cleartext value is used as an encryption key. Because the DC stores the existing password as well, the DC is able to decrypt the data sent by the client. Of course, if the end user did not enter the correct existing password, the decryption does not result in meaningful data, and an error is returned.

SamrUnicodeChangePasswordUser2 is preferred if the Unicode-encoded cleartext password is available to the client.

##### SamrChangePasswordUser (Opnum 38)

The SamrChangePasswordUser method changes the password of a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).

1. long SamrChangePasswordUser(
2. [in] SAMPR\_HANDLE UserHandle,
3. [in] unsigned char LmPresent,
4. [in, unique] PENCRYPTED\_LM\_OWF\_PASSWORD OldLmEncryptedWithNewLm,
5. [in, unique] PENCRYPTED\_LM\_OWF\_PASSWORD NewLmEncryptedWithOldLm,
6. [in] unsigned char NtPresent,
7. [in, unique] PENCRYPTED\_NT\_OWF\_PASSWORD OldNtEncryptedWithNewNt,
8. [in, unique] PENCRYPTED\_NT\_OWF\_PASSWORD NewNtEncryptedWithOldNt,
9. [in] unsigned char NtCrossEncryptionPresent,
10. [in, unique] PENCRYPTED\_NT\_OWF\_PASSWORD NewNtEncryptedWithNewLm,
11. [in] unsigned char LmCrossEncryptionPresent,
12. [in, unique] PENCRYPTED\_LM\_OWF\_PASSWORD NewLmEncryptedWithNewNt
13. );

**UserHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a user object.

**LmPresent:** If this parameter is zero, the *OldLmEncryptedWithNewLm* and *NewLmEncryptedWithOldLm* fields MUST be ignored by the server; otherwise these fields MUST be processed.

**OldLmEncryptedWithNewLm:** The [**LM hash**](#gt_58a72159-b1e9-4770-9b06-d7a6b16279a8) of the target user's existing password (as presented by the client) encrypted according to the specification of [ENCRYPTED\_LM\_OWF\_PASSWORD (section 2.2.7.3)](#Section_ce061fef6d4f4802bd5d26b11f14f4a6), where the key is the LM hash of the new password for the target user (as presented by the client in the *NewLmEncryptedWithOldLm* parameter).

**NewLmEncryptedWithOldLm:** The LM hash of the target user's new password (as presented by the client) encrypted according to the specification of ENCRYPTED\_LM\_OWF\_PASSWORD, where the key is the LM hash of the existing password for the target user (as presented by the client in the *OldLmEncryptedWithNewLm* parameter).

**NtPresent:** If this parameter is zero, *OldNtEncryptedWithNewNt* and *NewNtEncryptedWithOldNt* MUST be ignored by the server; otherwise these fields MUST be processed.

**OldNtEncryptedWithNewNt:** The [**NT hash**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f) of the target user's existing password (as presented by the client) encrypted according to the specification of ENCRYPTED\_NT\_OWF\_PASSWORD (section 2.2.7.3), where the key is the NT hash of the new password for the target user (as presented by the client).

**NewNtEncryptedWithOldNt:** The NT hash of the target user's new password (as presented by the client) encrypted according to the specification of ENCRYPTED\_NT\_OWF\_PASSWORD, where the key is the NT hash of the existing password for the target user (as presented by the client).

**NtCrossEncryptionPresent:** If this parameter is zero, *NewNtEncryptedWithNewLm* MUST be ignored; otherwise, this field MUST be processed.

**NewNtEncryptedWithNewLm:** The NT hash of the target user's new password (as presented by the client) encrypted according to the specification of ENCRYPTED\_NT\_OWF\_PASSWORD, where the key is the LM hash of the new password for the target user (as presented by the client).

**LmCrossEncryptionPresent:** If this parameter is zero, *NewLmEncryptedWithNewNt* MUST be ignored; otherwise, this field MUST be processed.

**NewLmEncryptedWithNewNt:** The LM hash of the target user's new password (as presented by the client) encrypted according to the specification of ENCRYPTED\_LM\_OWF\_PASSWORD, where the key is the NT hash of the new password for the target user (as presented by the client).

The processing for this method is quite complex. To aid comprehension, a brief, non-normative description of the method's intent follows.

The method requires that the client presents both the NT and the LM hash of the new password (and will fail otherwise). However, because the old password might not be stored in either the NT or LM hash format on the receiver, and thus the new hash values cannot be decrypted using the old hash values, the method allows for the new NT and LM hashes to be "cross-encrypted" using the new LM or NT hash value (instead of the old values). As such, there are three combinations that can lead to successful processing, which are listed below.

1. *NtPresent* is nonzero, *LmPresent* is nonzero, and both the LM and NT hashes are present in the database. No "cross-encryption" is required. The cross-encryption–related parameters are ignored.
2. *LmPresent* is nonzero, *NtCrossEncryptionPresent* is nonzero, and the LM hash is present in the database. This combination is used when the NT hash is not stored at the server; the client can send the NT hash encrypted with the new LM hash instead. The NT-hash–related parameters are ignored.
3. *NtPresent* is nonzero, *LmCrossEncryptionPresent* is nonzero, and the NT hash is present in the database. This combination is used when the LM hash is not stored at the server; the client can send the LM hash encrypted with the new NT hash instead. The LM-hash–related parameters are ignored.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints applied in the presented order:

1. All database operations MUST occur in a single transaction.
2. The constraints in section [3.1.5.14.5](#Section_9a859446f921410dbcc18a8c3dac751e) MUST be satisfied.
3. If *LmPresent* is nonzero and *NewLmEncryptedWithOldLm* or *OldLmEncryptedWithNewLm* is "NULL", the server MUST return an error.
4. If *NtPresent* is nonzero and *NewNtEncryptedWithOldNt* or *OldNtEncryptedWithNewNt* is "NULL", the server MUST return an error.
5. If *NtCrossEncryptionPresent* is nonzero and *NewNtEncryptedWithNewLm* is "NULL", the server MUST return an error.
6. If *LmCrossEncryptionPresent* is nonzero and *NewLmEncryptedWithNewNt* is "NULL", the server MUST return an error.
7. If *LmPresent* and *NtPresent* are zero, the server MUST return an error.
8. Let U be the user account referenced by *UserHandle.Object*.
9. Let Stored-LM-Hash be the value of the **dBCSPwd** attribute from the database decrypted using the algorithm specified in section [2.2.11.1](#Section_3f5ec79db4494ab29423c4dccbe0b184), using U's RelativeId (an unsigned integer) as the key. If the **dBCSPwd** attribute does not exist, let Stored-LM-Hash be "NULL".
10. Let Stored-NT-Hash be the value of the **unicodePwd** attribute from the database decrypted using the algorithm specified in section 2.2.11.1, using U's RelativeId (an unsigned integer) as the key. If the **unicodePwd** attribute does not exist, let Stored-NT-Hash be "NULL".
11. If *LmPresent* is nonzero and Stored-LM-Hash is not NULL, let Presented-New-LM-Hash be *NewLmEncryptedWithOldLm*, decrypted as specified in section 2.2.11.1, using Stored-LM-Hash as the key; and let Presented-Old-LM-Hash be *OldLmEncryptedWithNewLm*, decrypted as specified in section 2.2.11.1, using Presented-New-LM-Hash as the key. The values are not referenced below if *LmPresent* is zero.
12. If *NtPresent* is nonzero and Stored-NT-Hash is not NULL, let Presented-New-NT-Hash be *NewNtEncryptedWithOldNt*, decrypted as specified in section 2.2.11.1, using Stored-NT-Hash as the key; and let Presented-Old-NT-Hash be *OldNtEncryptedWithNewNt*, decrypted as specified in section 2.2.11.1, using Presented-New-NT-Hash as the key. The values are not referenced below if *NtPresent* is zero.
13. If all of the following conditions are true, the server MUST abort processing and return the error status STATUS\_LM\_CROSS\_ENCRYPTION\_REQUIRED:
    1. *NtPresent* is nonzero.
    2. *LmPresent* is zero.
    3. *LmCrossEncryptionPresent* is zero.
    4. Stored-NT-Hash is non-NULL and equals Presented-Old-NT-Hash.
14. If all of the following conditions are true, the server MUST abort processing and return the error status STATUS\_NT\_CROSS\_ENCRYPTION\_REQUIRED.
    1. *NtPresent* is nonzero.
    2. *LmPresent* is nonzero.
    3. *NtCrossEncryptionPresent* is zero.
    4. Stored-NT-Hash is NULL.
    5. Stored-LM-Hash is non-NULL and equals Presented-Old-LM-Hash.
15. Exactly one of the three following conditions MUST be true; otherwise, the server MUST satisfy the constraints in section [3.1.5.14.6](#Section_11b31f55f4474a63bbf7be4c4853c23e) and then return STATUS\_WRONG\_PASSWORD.
    1. *LmPresent* is nonzero, Stored-LM-Hash is non-NULL and equals Presented-Old-LM-Hash, *NtPresent* is nonzero, Stored-NT-Hash is non-NULL, and Stored-NT-Hash equals Presented-Old-NT-Hash.
    2. *LmPresent* is nonzero, Stored-LM-Hash is non-NULL and equals Presented-Old-LM-Hash, *NtPresent* is zero, and Stored-NT-Hash is NULL.
    3. *NtPresent* is nonzero, Stored-NT-Hash is non-NULL and equals Presented-Old-NT-Hash, *LmPresent* is zero, and Stored-LM-Hash is NULL.
16. If *LmPresent* is nonzero, the **dBCSPwd** attribute MUST be updated with Presented-New-LM-Hash.
17. If *LmPresent* is zero and *LmCrossEncryptionPresent* is nonzero, the **dBCSPwd** attribute MUST be updated with the value of *NewLmEncryptedWithNewNt*, decrypted using the algorithm specified in section 2.2.11.1, using Presented-New-NT-Hash as the decryption key.
18. If *NtPresent* is nonzero, the **unicodePwd** attribute MUST be updated with Presented-New-NT-Hash.
19. If *NtPresent* is zero and *NtCrossEncryptionPresent* is nonzero, the **unicodePwd** attribute MUST be updated with the value of *NewNtEncryptedWithNewLm*, decrypted using the algorithm specified in section 2.2.11.1, using Presented-New-LM-Hash as the decryption key.
20. On database error, the server MUST return the data error; on general processing error, the server MUST return STATUS\_WRONG\_PASSWORD; otherwise, return STATUS\_SUCCESS.

##### SamrOemChangePasswordUser2 (Opnum 54)

The SamrOemChangePasswordUser2 method changes a user's password.

1. long SamrOemChangePasswordUser2(
2. [in] handle\_t BindingHandle,
3. [in, unique] PRPC\_STRING ServerName,
4. [in] PRPC\_STRING UserName,
5. [in, unique] PSAMPR\_ENCRYPTED\_USER\_PASSWORD NewPasswordEncryptedWithOldLm,
6. [in, unique] PENCRYPTED\_LM\_OWF\_PASSWORD OldLmOwfPasswordEncryptedWithNewLm

);

**BindingHandle:** An RPC binding handle parameter as specified in [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824) section 1.

**ServerName:** A counted string, encoded in the OEM character set, containing the NETBIOS name of the server; this parameter MAY[<60>](#Appendix_A_60" \o "Product behavior note 60) be ignored by the server.

**UserName:** A counted string, encoded in the OEM character set, containing the name of the user whose password is to be changed; see message processing later in this section for details on how this value is used as a database key to locate the [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) that is the target of this password change operation.

**NewPasswordEncryptedWithOldLm:** A cleartext password encrypted according to the specification of [SAMPR\_ENCRYPTED\_USER\_PASSWORD (section 2.2.6.21)](#Section_23f9ef4ccf3e43309287ea4799b03201), where the key is the [**LM hash**](#gt_58a72159-b1e9-4770-9b06-d7a6b16279a8) of the existing password for the target user (as presented by the client). The cleartext password MUST be encoded in an [**OEM code page**](#gt_442ab13f-d2c1-4128-b1db-f3bea4b8224e) character set (as opposed to UTF-16).

**OldLmOwfPasswordEncryptedWithNewLm:** The LM hash of the target user's existing password (as presented by the client) encrypted according to the specification of [ENCRYPTED\_LM\_OWF\_PASSWORD (section 2.2.7.3)](#Section_ce061fef6d4f4802bd5d26b11f14f4a6), where the key is the LM hash of the cleartext password obtained from decrypting *NewPasswordEncryptedWithOldLm* (see the preceding description for decryption details).

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. On a [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration if [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) is not running, the server MUST abort the request and return an error status.
2. All database operations MUST occur in a single transaction.
3. The server MUST encode the *UserName* parameter into UTF-16 using the OEM code page.
4. Let U be the user account with the **sAMAccountName** attribute value of *UserName*. The server MUST return STATUS\_WRONG\_PASSWORD if no such account exists.
5. Let Stored-LM-Hash be the value of the **dBCSPwd** attribute from the database decrypted using the algorithm specified in section [2.2.11.1](#Section_3f5ec79db4494ab29423c4dccbe0b184), using U's RelativeId as the key. If this attribute does not exist, STATUS\_WRONG\_PASSWORD MUST be returned.
6. Let Presented-Clear-Text be the cleartext value sent by the client. This value is obtained by decrypting *NewPasswordEncryptedWithOldLm* according to the specification of SAMPR\_ENCRYPTED\_USER\_PASSWORD using Stored-LM-Hash as the key, and then translating the result into a UTF-16 encoded string (using the OEM code page).
7. Let Presented-Old-LM-Hash be the value of *OldLmOwfPasswordEncryptedWithNewLm* that has been decrypted per the specification of ENCRYPTED\_LM\_OWF\_PASSWORD, using the LM hash of Presented-Clear-Text as the key.
8. If Presented-Old-LM-Hash is not equal to Stored-LM-Hash, the server MUST satisfy the constraints in section [3.1.5.14.6](#Section_11b31f55f4474a63bbf7be4c4853c23e), abort processing, and return STATUS\_WRONG\_PASSWORD.
9. The server MUST update the **clearTextPassword** attribute with Presented-Clear-Text.

##### SamrUnicodeChangePasswordUser2 (Opnum 55)

The SamrUnicodeChangePasswordUser2 method changes a user account's password.

1. long SamrUnicodeChangePasswordUser2(
2. [in] handle\_t BindingHandle,
3. [in, unique] PRPC\_UNICODE\_STRING ServerName,
4. [in] PRPC\_UNICODE\_STRING UserName,
5. [in, unique] PSAMPR\_ENCRYPTED\_USER\_PASSWORD NewPasswordEncryptedWithOldNt,
6. [in, unique] PENCRYPTED\_NT\_OWF\_PASSWORD OldNtOwfPasswordEncryptedWithNewNt,
7. [in] unsigned char LmPresent,
8. [in, unique] PSAMPR\_ENCRYPTED\_USER\_PASSWORD NewPasswordEncryptedWithOldLm,
9. [in, unique] PENCRYPTED\_LM\_OWF\_PASSWORD OldLmOwfPasswordEncryptedWithNewNt
10. );

**BindingHandle:** An RPC binding handle parameter as specified in [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824) section 1.

**ServerName:** A null-terminated string containing the NETBIOS name of the server; this parameter MAY[<61>](#Appendix_A_61" \o "Product behavior note 61) be ignored by the server.

**UserName:** The name of the user. See the message processing later in this section for details on how this value is used as a database key to locate the [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) that is the target of this password change operation.

**NewPasswordEncryptedWithOldNt:** A cleartext password encrypted according to the specification of [SAMPR\_ENCRYPTED\_USER\_PASSWORD (section 2.2.6.21)](#Section_23f9ef4ccf3e43309287ea4799b03201), where the key is the [**NT hash**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f) of the existing password for the target user (as presented by the client in the *OldNtOwfPasswordEncryptedWithNewNt* parameter).

**OldNtOwfPasswordEncryptedWithNewNt:** The NT hash of the target user's existing password (as presented by the client) encrypted according to the specification of [ENCRYPTED\_LM\_OWF\_PASSWORD (section 2.2.7.3)](#Section_ce061fef6d4f4802bd5d26b11f14f4a6), where the key is the NT hash of the cleartext password obtained from decrypting *NewPasswordEncryptedWithOldNt*.

**LmPresent:** If this parameter is zero, *NewPasswordEncryptedWithOldLm* and *OldLmOwfPasswordEncryptedWithNewNt* MUST be ignored; otherwise these fields MUST be processed.

**NewPasswordEncryptedWithOldLm:** A cleartext password encrypted according to the specification of SAMPR\_ENCRYPTED\_USER\_PASSWORD, where the key is the [**LM hash**](#gt_58a72159-b1e9-4770-9b06-d7a6b16279a8) of the existing password for the target user (as presented by the client).

**OldLmOwfPasswordEncryptedWithNewNt:** The LM hash the target user's existing password (as presented by the client) encrypted according to the specification of ENCRYPTED\_LM\_OWF\_PASSWORD, where the key is the NT hash of the cleartext password obtained from decrypting *NewPasswordEncryptedWithOldNt*.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. On a [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration if [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) is not running, the server MUST abort the request and return an error status.
2. All database operations MUST occur in a single transaction.
3. Let U be the user account with the **sAMAccountName** attribute value of UserName. The server MUST return STATUS\_WRONG\_PASSWORD if no such account exists.
4. Let Stored-NT-Hash be the value of the **unicodePwd** attribute from the database decrypted using the algorithm specified in section [2.2.11.1](#Section_3f5ec79db4494ab29423c4dccbe0b184), using U's *RelativeId* as the key. If the attribute does not exist, let Stored-NT-Hash be "NULL".
5. Let Stored-LM-Hash be the value of the **dBCSPwd** attribute from the database decrypted using the algorithm specified in section 2.2.11.1, using U's *RelativeId* as the key. If the attribute does not exist, let Stored-LM-Hash be "NULL".
6. If Stored-NT-Hash is NULL and LmPresent is zero or Stored-LM-Hash is NULL, the server MUST abort processing and return STATUS\_WRONG\_PASSWORD.
7. If Stored-NT-Hash is not NULL, then:
   1. Let Presented-Clear-Text be the cleartext value sent by the client, obtained by decrypting *NewPasswordEncryptedWithOldNt* according to the specification of SAMPR\_ENCRYPTED\_USER\_PASSWORD, using Stored-NT-Hash as the key, AND
   2. Let Presented-Old-NT-Hash be the value of *OldNtOwfPasswordEncryptedWithNewNt* decrypted according to the specification of ENCRYPTED\_LM\_OWF\_PASSWORD, using the NT hash of Presented-Clear-Text as the key.
8. If Stored-NT-Hash is NULL, then:
   1. Let Presented-Clear-Text be the cleartext value sent by the client, obtained by decrypting *NewPasswordEncryptedWithOldLm* according to the specification of SAMPR\_ENCRYPTED\_USER\_PASSWORD, using Stored-LM-Hash as the key, AND
   2. Let Presented-Old-LM-Hash be the value of *OldLmOwfPasswordEncryptedWithNewNt* decrypted according to the specification of ENCRYPTED\_LM\_OWF\_PASSWORD, using the NT hash of Presented-Clear-Text as the key.
9. Exactly one of the two following conditions MUST be true; otherwise, the server MUST satisfy the constraints in section [3.1.5.14.6](#Section_11b31f55f4474a63bbf7be4c4853c23e) and return STATUS\_WRONG\_PASSWORD.
   1. Stored-NT-Hash is non-NULL and equals Presented-Old-NT-Hash.
   2. Stored-NT-Hash is NULL, and Stored-LM-Hash is non-NULL and equals Presented-Old-LM-Hash.
10. The server MUST update the **clearTextPassword** attribute with Presented-Clear-Text.

##### SamrUnicodeChangePasswordUser4 (Opnum 73)

Use the SamrUnicodeChangePasswordUser4 method to change a user account password.

1. NTSTATUS
2. SamrUnicodeChangePasswordUser4(
3. [in] handle\_t BindingHandle,
4. [in,unique] PRPC\_UNICODE\_STRING ServerName,
5. [in] PRPC\_UNICODE\_STRING UserName,
6. [in] PSAMPR\_ENCRYPTED\_PASSWORD\_AES EncryptedPassword
7. );

**BindingHandle**: An RPC binding handle parameter as specified in [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824) section 1.

**ServerName**: A null-terminated string containing the NETBIOS name of the server[<62>](#Appendix_A_62" \o "Product behavior note 62).

**UserName**: The name of the user. The processing instructions that follow describe how this value is used as a database key to locate the [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) that is the target of this password change operation.

**EncryptedPassword**: A cleartext password encrypted to the specification of SAMPR\_ENCRYPTED\_PASSWORD\_AES (section [3.2.2.4](#Section_0fc25352662b40e79deef10c6244c4c9)), where the key is derived using the PBKDF2 algorithm and the NT-hash of the users existing password, the EncryptedPassword.Salt, and EncryptedPassword.PBKDF2 Iteration count.

EncryptedPassword.PBKDFIterations MUST be present and MUST be between 5000 and 1,000,000 inclusive.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. On a [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration, if [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) is not running, the server MUST abort the request and return an error status.
2. All database operations MUST occur in a single transaction.
3. Let ‘U’ be the user account with the **sAMAccountName** attribute value of UserName. The server MUST return STATUS\_WRONG\_PASSWORD (section [2.2.1.15](#Section_7440cfac6052492584e4c32e417de300)) if no such account exists.
4. Let Stored-NT-Hash be the value of the unicodePwd attribute from the database decrypted using the algorithm specified in section [2.2.11.1](#Section_3f5ec79db4494ab29423c4dccbe0b184), using U's RelativeId (section [3.1.5.11.2](#Section_d91271c67b2e419499278fabfa429f90)) as the key. If the attribute does not exist, let Stored-NT-Hash be "NULL".
5. If Stored-NT-Hash is NULL, the server MUST abort processing and return STATUS\_WRONG\_PASSWORD.
6. If EncryptedPassword.PBKDF2Iterations is not present or not valid, the server will return STATUS\_WRONG\_PASSWORD and abort processing.
7. If Stored-NT-Hash is not NULL and EncryptedPassword.PBKDF2Iterations is valid, the server will decrypt EncryptedPassword as follows:

* Let CEK be a 16-byte encryption key derived with PBKDF2 from the Stored-NT-Hash, EncryptedPassword.PBKDF2Iterations, and EncryptedPassword.Salt.
* Compute a MAC as specified in AES Cipher Usage (section 3.2.2.4) and verify whether EncryptedPassword.AuthData matches this MAC.
* Decrypt the EncryptedPassword per the AES Cipher Usage specifications using the CEK.
* Let Presented-Clear-Text be the cleartext value sent by the client, obtained by decrypting EncryptedPassword according to the specifications of SAMPR\_ENCRYPTED\_PASSWORD\_AES and AES Cipher Usage, using a 16-byte CEK derived from the Stored-NT-Hash using PBKDF2, EncryptedPassword.PBKDF2Iterations, and EncryptedPassword.Salt.

1. The following conditions MUST be true; otherwise, the server MUST satisfy the constraints in section [3.1.5.14.6](#Section_11b31f55f4474a63bbf7be4c4853c23e) and return STATUS\_WRONG\_PASSWORD:

* Stored-NT-Hash is non-NULL.
* EncryptedPassword.PBKDFIterations MUST be between 5000 and 1,000,000 inclusive.
* MAC computed by the Server according to the AES Cipher Usage specifications (section 3.2.2.4), with CEK matching the AuthData presented by the client.
* Decrypting the EncryptedPasswork.Ciphertext succeeds.

1. The server MUST update the clearTextPassword attribute with Presented-Clear-Text.

#### Lookup Pattern

These methods enable a client to translate from a security ID (either a [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) or a [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c)) to a user-friendly name, and vice versa. This action is useful when an end user is setting access control via a [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350). However, the translation methods specified in [[MS-LSAT]](%5bMS-LSAT%5d.pdf#Section_1ba21e6fd8a9462c91534375f2020894) sections 3.1.4.5 and 3.1.4.9 are superior because they translate a wider range of SIDs.

A client MUST first obtain a handle to the object of interest through an "open" method. See section [3.1.5.1](#Section_6d92e4991d164c3596e6c564dfd2972b).

See section [1.3](#Section_d036091033dd4e3cb5a226daf95f5ffa) for a description of the "lookup" pattern of methods.

##### SamrLookupDomainInSamServer (Opnum 5)

The SamrLookupDomainInSamServer method obtains the [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) of a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1), given the object's name.

1. long SamrLookupDomainInSamServer(
2. [in] SAMPR\_HANDLE ServerHandle,
3. [in] PRPC\_UNICODE\_STRING Name,
4. [out] PRPC\_SID\* DomainId
5. );

**ServerHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**server object**](#gt_62a8c543-5998-480b-8fa7-41a8f04a18e5).

**Name:** A UTF-16 encoded string.

**DomainId:** A SID value of a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) that corresponds to the *Name* passed in. The match MUST be exact (no wildcard characters are permitted). See message processing later in this section for more details.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *ServerHandle.HandleType* is not equal to "Server".
2. *ServerHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. If the *Name* input parameter matches an attribute value as shown in the following table, the associated value in the "Return attribute" column MUST be returned via the *DomainId* parameter.

| Matching object | Matching attribute | Return object | Return attribute |
| --- | --- | --- | --- |
| domain object | name | domain object | objectSid |
| built-in object | name | built-in object | objectSid |

1. If there is no match, an error MUST be returned.

##### SamrLookupNamesInDomain (Opnum 17)

The SamrLookupNamesInDomain method translates a set of account names into a set of [**RIDs**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c).

1. long SamrLookupNamesInDomain(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in, range(0,1000)] unsigned long Count,
4. [in, size\_is(1000), length\_is(Count)]
5. RPC\_UNICODE\_STRING Names[\*],
6. [out] PSAMPR\_ULONG\_ARRAY RelativeIds,
7. [out] PSAMPR\_ULONG\_ARRAY Use
8. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**Count:** The number of elements in *Names*. The maximum value of 1,000 is chosen to limit the amount of memory that the client can force the server to allocate.

**Names:** An array of strings that are to be mapped to RIDs.

**RelativeIds:** An array of RIDs of [**accounts**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) that correspond to the elements in *Names*.

**Use:** An array of SID\_NAME\_USE enumeration values that describe the type of account for each entry in *RelativeIds*.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

On receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *DomainHandle.HandleType* is not equal to "Domain".
2. *DomainHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. Let U be the set of all [**database objects**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f) whose **objectSid**'s [**domain prefix**](#gt_9066e9dc-8275-4452-9073-daab5fd427c5) matches the domain prefix of the [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) referenced by *DomainHandle.Object*.
4. For each element in *Names* that matches a database object's **sAMAccountName** attribute value in the set U, the server MUST fill in *RelativeIds* and *Use* as follows:
   1. Let 'i' be the current element of *Names*.
   2. **RelativeIds.Element[i]** is the RID of the matched object's **objectSid** attribute value.
   3. **Use.Element[i]** is set as follows.

| objectClass | GroupType | Use |
| --- | --- | --- |
| User | n/a | SidTypeUser |
| Group | GROUP\_TYPE\_ACCOUNT\_GROUP | SidTypeGroup |
| Group | GROUP\_TYPE\_UNIVERSAL\_GROUP | SidTypeGroup |
| Group | Any value not matching the above criteria for Group | SidTypeAlias |

1. For each element in *Names* that does not match a database object's **sAMAccountName** attribute value in the set U, the server MUST fill in *RelativeIds* and *Use* as follows:
   1. Let 'i' be the current element of *Names*.
   2. RelativeIds.Element[i] is 0.
   3. **Use.Element[i]** is SidTypeUnknown.
2. Otherwise:
   1. **RelativeIds.Count** MUST be set to the input parameter *Count* on successful completion of the method.
   2. **Use.Count** MUST be set to the input parameter *Count* on successful completion of the method.
   3. If the number of matched accounts is equal to the input parameter *Count*, STATUS\_SUCCESS MUST be returned.
   4. If the number of matched accounts is less than the input parameter *Count* but greater than 0, STATUS\_SOME\_NOT\_MAPPED MUST be returned. Note that this is not an error condition.
   5. If the number of matched accounts is 0, STATUS\_NONE\_MAPPED MUST be returned.

##### SamrLookupIdsInDomain (Opnum 18)

The SamrLookupIdsInDomain method translates a set of [**RIDs**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) into account names.

1. long SamrLookupIdsInDomain(
2. [in] SAMPR\_HANDLE DomainHandle,
3. [in, range(0,1000)] unsigned long Count,
4. [in, size\_is(1000), length\_is(Count)]
5. unsigned long\* RelativeIds,
6. [out] PSAMPR\_RETURNED\_USTRING\_ARRAY Names,
7. [out] PSAMPR\_ULONG\_ARRAY Use
8. );

**DomainHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1).

**Count:** The number of elements in *RelativeIds*. The maximum value of 1,000 is chosen to limit the amount of memory that the client can force the server to allocate.

**RelativeIds:** An array of RIDs that are to be mapped to account names.

**Names:** A structure containing an array of account names that correspond to the elements in *RelativeIds*.

**Use:** A structure containing an array of SID\_NAME\_USE enumeration values that describe the type of [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) for each entry in *RelativeIds*.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

On receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *DomainHandle.HandleType* is not equal to "Domain".
2. *DomainHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. Let U be the set of all [**database objects**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f) whose objectSid's [**domain prefix**](#gt_9066e9dc-8275-4452-9073-daab5fd427c5) matches the domain prefix of the [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) referenced by *DomainHandle.Object*.
4. For each element in *RelativeIds* that matches the RID of a database object's **objectSid** attribute value in the set U, the server MUST fill in *Names* and *Use* as follows:
   1. Let 'i' be the current element of *RelativeIds*.
   2. **Names.Element[i]** is the **sAMAccoutName** attribute value of the matched object.
   3. **Use.Element[i]** is set as follows.

| objectClass | GroupType | Use |
| --- | --- | --- |
| User | n/a | SidTypeUser |
| Group | GROUP\_TYPE\_ACCOUNT\_GROUP | SidTypeGroup |
| Group | GROUP\_TYPE\_UNIVERSAL\_GROUP | SidTypeGroup |
| Group | Any value not matching the above criteria for Group | SidTypeAlias |

1. For each element in *RelativeIds* that does not match the RID of a database object's **objectSid** attribute value, the server MUST fill in *Names* and *Use* as follows:
   1. Let 'i' be the current element of *RelativeIds*.
   2. All fields of **Names.Element[i]** MUST be set to 0.
   3. **Use.Element[i]** is SidTypeUnknown.
2. Otherwise:
   1. **Names.Count** MUST be set to the input parameter *Count* on successful completion of the method.
   2. **Use.Count** MUST be set to the input parameter *Count* on successful completion of the method.
   3. If the number of matched accounts is equal to the input parameter *Count*, 0 MUST be returned.
   4. If the number of matched accounts is less than the input parameter *Count* but greater than 0, STATUS\_SOME\_NOT\_MAPPED MUST be returned. Note that this is not an error condition.
   5. If the number of matched accounts is 0, STATUS\_NONE\_MAPPED MUST be returned.

#### Security Pattern

These methods enable a client to set the access control on a server, [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca), group, alias, or [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).

These methods require a handle obtained from an "open" or a "create" method. See sections [3.1.5.1](#Section_6d92e4991d164c3596e6c564dfd2972b) and [3.1.5.4](#Section_2214fd297c2d450fb68d6a0e97ebe48f).

A non-normative description of these methods is helpful to understand the intent of the message processing. The remainder of this section contains such a description.

Two points are significant:

1. The message processing requirements between [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) and non-DC configurations are very different.
2. All known clients use a very small subset of the functionality exposed in these methods.

The DC message processing requirements differ from the non-DC case because the [**database objects**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f) on which the server operates are also exposed through the LDAP model for read and update, and have a different [**ACE**](#gt_b581857f-39aa-4979-876b-daba67a40f15) format than what this protocol exposes. Specifically, in the DC case, the database objects have [**security descriptors**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) with an object ACE format (specified in [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 5.1.3), whereas these methods expect and return security descriptors with a simple ACE format (specified in [MS-ADTS] section 5.1.3). Therefore, the message processing for these methods converts between these two models. In general, this would be an intractable problem because new access masks and object ACE types can be added that are not expressible through this protocol.

Fortunately, all known clients use a small subset of the functionality exposed through these methods. Specifically, all known clients use [SamrQuerySecurityObject](#Section_0ecf8fecd17e4a88b7f1e0f0f66790db) and [SamrSetSecurityObject](#Section_6666a06658cf4118bf4bdd54ed55ecf0) only to control whether a password can be changed for a user account (the relevant [**access mask**](#gt_462f2aa7-2cfc-404a-b479-30f127c512b1) is USER\_CHANGE\_PASSWORD, specified in section [2.2.1.7](#Section_c0be3f43bcf943eeb0273d02ab372c53)). Accordingly, the server of these methods is required to support only this narrow request; other requests can be safely ignored.

In the DC case, general security-descriptor manipulation is best achieved through LDAP. [MS-ADTS] section 5 specifies the [**Active Directory**](#gt_e467d927-17bf-49c9-98d1-96ddf61ddd90) security model in detail.

For the non-DC case, because the security descriptor on the database objects is not exposed through any other protocol, a server implementation has much greater breadth in implementing the access control specified in the security descriptor presented in a method call to SamrSetSecurityObject. Furthermore, because no other protocol can modify the security descriptor on the database objects in a non-DC configuration, it is possible to translate an object ACE format security descriptor to a simple ACE format. Non-DC servers have the requirement to return, via SamrQuerySecurityObject, the same access control specification that was specified to a previous call to SamrSetSecurityObject, and to enforce all access control permissions specified through SamrSetSecurityObject.

See section [1.3](#Section_d036091033dd4e3cb5a226daf95f5ffa) for a description of the "security" pattern of methods.

##### SamrSetSecurityObject (Opnum 2)

The SamrSetSecurityObject method sets the access control on a server, [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca), user, group, or [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0).

1. long SamrSetSecurityObject(
2. [in] SAMPR\_HANDLE ObjectHandle,
3. [in] SECURITY\_INFORMATION SecurityInformation,
4. [in] PSAMPR\_SR\_SECURITY\_DESCRIPTOR SecurityDescriptor
5. );

**ObjectHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a server, domain, user, group, or alias object.

**SecurityInformation:** A bit field that indicates the fields of *SecurityDescriptor* that are requested to be set.

The SECURITY\_INFORMATION type is defined in [[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2) section 2.4.7. The following bits are valid; all other bits MUST be zero when sent and ignored on receipt. If none of the bits below are present, the server MUST return STATUS\_INVALID\_PARAMETER.

| Value | Meaning |
| --- | --- |
| OWNER\_SECURITY\_INFORMATION  0x00000001 | Refers to the Owner member of the [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350). |
| GROUP\_SECURITY\_INFORMATION  0x00000002 | Refers to the Group member of the security descriptor. |
| DACL\_SECURITY\_INFORMATION  0x00000004 | Refers to the [**DACL**](#gt_d727f612-7a45-48e4-9d87-71735d62b321) of the security descriptor. |
| SACL\_SECURITY\_INFORMATION  0x00000008 | Refers to the [**system access control list (SACL)**](#gt_c189801e-3752-4715-88f4-17804dad5782) of the security descriptor. |

**SecurityDescriptor:** A security descriptor expressing access that is specific to the *ObjectHandle*.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Message processing for this method is specified in the following two sections.

###### SamrSetSecurityObject (DC Configuration)

Upon receiving this message, the server MUST process the data from the message subject to all of the following constraints:

1. The access control specified in *SecurityDescriptor* MUST be a valid [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) containing simple [**ACEs**](#gt_b581857f-39aa-4979-876b-daba67a40f15); otherwise the server MUST return an error status. [[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2) section 2.4.6 contains the specification for a valid security descriptor. On error, the server MUST abort processing and return an error.
2. *ObjectHandle.GrantedAccess* MUST have the required access specified in the following table, based on the set bits in the *SecurityInformation* parameter. The server MUST ignore set bits in *SecurityInformation* that are not specified in the table. On error, the server MUST abort processing and return STATUS\_ACCESS\_DENIED.

| Security information bits | Required access |
| --- | --- |
| SACL\_SECURITY\_INFORMATION | ACCESS\_SYSTEM\_SECURITY |
| OWNER\_SECURITY\_INFORMATION | WRITE\_OWNER |
| GROUP\_SECURITY\_INFORMATION | WRITE\_OWNER |
| DACL\_SECURITY\_INFORMATION | WRITE\_DAC |

1. If the DACL\_SECURITY\_INFORMATION bit is set in *SecurityInformation*, the server MUST determine whether the [**DACL**](#gt_d727f612-7a45-48e4-9d87-71735d62b321) of *SecurityDescriptor* of the input message matches one of the following DACLs. The ordering of the ACEs is not relevant. Let Self denote the [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) of the [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17) referenced by *ObjectHandle.Object*.
   * DACL a.

| SID | Access mask |
| --- | --- |
| [**WorldSid**](#gt_3aed7180-e8fa-46c9-a0a1-428f9f14f532) | USER\_EXECUTE | USER\_READ |
| [**AdministratorSid**](#gt_64473c0e-1ce4-499c-bcb2-5264406871c2) | USER\_ALL\_ACCESS |
| [**AccountOperatorsSid**](#gt_1771f746-746d-4079-ae9c-c5339ff9c61b) | USER\_ALL\_ACCESS |
| Self | USER\_WRITE |

* + DACL b.

| SID | Access mask |
| --- | --- |
| WorldSid | (USER\_EXECUTE | USER\_READ) & ~ USER\_CHANGE\_PASSWORD |
| AdministratorSid | USER\_ALL\_ACCESS |
| AccountOperatorsSid | USER\_ALL\_ACCESS |
| Self | USER\_WRITE & ~ USER\_CHANGE\_PASSWORD |

* + DACL c.

| SID | Access mask |
| --- | --- |
| WorldSid | (USER\_EXECUTE | USER\_READ) & ~ USER\_CHANGE\_PASSWORD |
| AdministratorSid | USER\_ALL\_ACCESS |
| AccountOperatorsSid | USER\_ALL\_ACCESS |

* + DACL d.

| SID | Access mask |
| --- | --- |
| WorldSid | USER\_EXECUTE | USER\_READ |
| AdministratorSid | USER\_ALL\_ACCESS |
| Self | USER\_WRITE |

1. If there is no match from the preceding constraint, the server MUST silently ignore the request by aborting processing and returning 0.
2. If the matching DACL grants USER\_CHANGE\_PASSWORD to World, the server MUST update the **ntSecurityDescriptor** attribute for the target user such that the target user has the ability to change his or her password; otherwise, the server MUST update the **ntSecurityDescriptor** attribute for the target user such that the target does not have the ability to change his or her password. For an example of how to do this, see the following citation in Appendix B: Product Behavior.[<63>](#Appendix_A_63" \o "Product behavior note 63)

###### SamrSetSecurityObject (Non-DC Configuration)

Upon receiving this message, the server MUST process the data from the message subject to all the following constraints:

1. The access control specified in *SecurityDescriptor* MUST be a valid [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) containing simple [**ACEs**](#gt_b581857f-39aa-4979-876b-daba67a40f15); otherwise the server MUST return an error status. [[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2) section 2.4.6 contains the specification for a valid security descriptor.
2. *ObjectHandle.GrantedAccess* MUST have the required access specified in the following table, based on the set bits in the *SecurityInformation* parameter. The server MUST ignore set bits in *SecurityInformation* that are not specified in the table. On error, the server MUST abort processing and return STATUS\_ACCESS\_DENIED.

| Security information bits | Required access |
| --- | --- |
| SACL\_SECURITY\_INFORMATION | ACCESS\_SYSTEM\_SECURITY |
| OWNER\_SECURITY\_INFORMATION | WRITE\_OWNER |
| GROUP\_SECURITY\_INFORMATION | WRITE\_OWNER |
| DACL\_SECURITY\_INFORMATION | WRITE\_DAC |

1. The server MUST update the **ntSecurityDescriptor** attribute value on the object referenced by *ObjectHandle.Object* such that all of the following constraints are satisfied:
   1. All accesses granted and denied in the input security descriptor (*SecurityDescriptor*) are granted and denied during subsequent method calls across this interface (for all time).
   2. If the target object is a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1), all ACEs containing DOMAIN\_CREATE\_USER, DOMAIN\_CREATE\_ALIAS, or DOMAIN\_CREATE\_GROUP MUST grant or deny (depending on the type of ACE) the trustee of the ACE the ability to create a user, alias, or group as specified in [SamrCreateUser2InDomain (section 3.1.5.4.4)](#Section_a98d7fbb17354fbfb41aef363c899002), [SamrCreateAliasInDomain (section 3.1.5.4.3)](#Section_7e505875e44f4f9d922b7ff3871c752d), or [SamrCreateGroupInDomain (section 3.1.5.4.2)](#Section_175c1cf94fa248379e5bbb1f0f950bee).
   3. If the target object is a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17), all ACEs containing the specified [**access mask**](#gt_462f2aa7-2cfc-404a-b479-30f127c512b1) in the following table MUST grant or deny (depending on the type of ACE) the trustee to update associated attributes.

| Access mask | Attribute |
| --- | --- |
| USER\_WRITE\_ACCOUNT | sAMAccountName  displayName  primaryGroupId  homeDirectory  homeDrive  scriptPath  profilePath  Description  userWorkstations  logonHours  accountExpires  userAccountControl  userParameters |
| USER\_WRITE\_PREFERENCE | comment  countryCode  codePage |
| USER\_FORCE\_PASSWORD\_CHANGE | clearTextPassword  pwdLastSet  dBCSPwd  unicodePwd |

##### SamrQuerySecurityObject (Opnum 3)

The SamrQuerySecurityObject method queries the access control on a server, [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca), user, group, or [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0).

1. long SamrQuerySecurityObject(
2. [in] SAMPR\_HANDLE ObjectHandle,
3. [in] SECURITY\_INFORMATION SecurityInformation,
4. [out] PSAMPR\_SR\_SECURITY\_DESCRIPTOR\* SecurityDescriptor
5. );

**ObjectHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a server, domain, user, group, or alias object.

**SecurityInformation:** A bit field that specifies which fields of *SecurityDescriptor* the client is requesting to be returned.

The SECURITY\_INFORMATION type is defined in [[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2) section 2.4.7. The following bits are valid; all other bits MUST be zero when sent and ignored on receipt.

| Value | Meaning |
| --- | --- |
| OWNER\_SECURITY\_INFORMATION  0x00000001 | If this bit is set, the client requests that the **Owner** member be returned.  If this bit is not set, the client requests that the **Owner** member not be returned. |
| GROUP\_SECURITY\_INFORMATION  0x00000002 | If this bit is set, the client requests that the **Group** member be returned.  If this bit is not set, the client requests that the **Group** member not be returned. |
| DACL\_SECURITY\_INFORMATION  0x00000004 | If this bit is set, the client requests that the [**DACL**](#gt_d727f612-7a45-48e4-9d87-71735d62b321) be returned.  If this bit is not set, the client requests that the DACL not be returned. |
| SACL\_SECURITY\_INFORMATION  0x00000008 | If this bit is set, the client requests that the [**SACL**](#gt_c189801e-3752-4715-88f4-17804dad5782) be returned.  If this bit is not set, the client requests that the SACL not be returned. |

**SecurityDescriptor:** A [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) expressing accesses that are specific to the *ObjectHandle* and the owner and group of the object. [MS-DTYP] section 2.4.6 contains the specification for a valid security descriptor.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Message processing for this method is specified in the following two sections.

###### SamrQuerySecurityObject (DC Configuration)

Let Self denote the **objectSid** attribute value, if any, of the object referenced by *ObjectHandle.Object*.

Upon receiving this message, the server MUST process the data from the message subject to all of the following constraints:

1. *ObjectHandle.GrantedAccess* MUST have the required access specified in the following table, based on the bits contained in the *SecurityInformation* parameter. On error, the server MUST abort processing and return STATUS\_ACCESS\_DENIED.

| Security information bits | Required access |
| --- | --- |
| SACL\_SECURITY\_INFORMATION | ACCESS\_SYSTEM\_SECURITY |
| OWNER\_SECURITY\_INFORMATION | READ\_CONTROL |
| GROUP\_SECURITY\_INFORMATION | READ\_CONTROL |
| DACL\_SECURITY\_INFORMATION | READ\_CONTROL |

1. The server MUST return, via the *SecurityDescriptor* parameter, a [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) that only contains fields based on the bits contained in the *SecurityInformation* parameter (the fields of the security descriptor that are not returned are set to zero) and that satisfies all of the following constraints:
   1. The **Owner** and **Group** fields of the security descriptor MUST be the administrator's [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) (S-1-5-32-544).
   2. The [**DACL**](#gt_d727f612-7a45-48e4-9d87-71735d62b321) MUST contain the following specified [**ACEs**](#gt_b581857f-39aa-4979-876b-daba67a40f15):

If *ObjectHandle.Object* refers to the [**server object**](#gt_62a8c543-5998-480b-8fa7-41a8f04a18e5), the DACL MUST contain the following ACEs.

| SID | Access mask |
| --- | --- |
| [**WorldSid**](#gt_3aed7180-e8fa-46c9-a0a1-428f9f14f532) | SAM\_SERVER\_EXECUTE |  SAM\_SERVER\_READ |
| [**AdministratorSid**](#gt_64473c0e-1ce4-499c-bcb2-5264406871c2) | SAM\_SERVER\_ALL\_ACCESS |

Else, if *ObjectHandle.Object* refers to a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1), the DACL MUST contain the following ACEs.

| SID | Access mask |
| --- | --- |
| WorldSid | DOMAIN\_EXECUTE |  DOMAIN\_READ |
| AdministratorSid | DOMAIN\_ALL\_ACCESS |
| [**AccountOperatorsSid**](#gt_1771f746-746d-4079-ae9c-c5339ff9c61b) | DOMAIN\_EXECUTE |  DOMAIN\_READ |  DOMAIN\_CREATE\_USER |  DOMAIN\_CREATE\_GROUP |  DOMAIN\_CREATE\_ALIAS |

Else, if *ObjectHandle.Object* refers to a group or [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0) that is the Domain Admins group (Domain Admins) or Administrators alias, or a member of [**Domain Admins**](#gt_d6220a40-b608-4244-8f43-c5a4a0be89ed) or [**Administrators**](#gt_eb335f5b-619b-46ed-9138-06e9910108c3), the DACL MUST contain the following ACEs.

| SID | Access mask |
| --- | --- |
| WorldSid | GROUP\_EXECUTE |  GROUP\_READ |
| AdministratorSid | GROUP\_ALL\_ACCESS |

Else, if *ObjectHandle.Object* refers to any [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) that does not satisfy the previous condition, the DACL MUST contain the following ACEs.

| SID | Access mask |
| --- | --- |
| WorldSid | GROUP\_EXECUTE |  GROUP\_READ |
| AdministratorSid | GROUP\_ALL\_ACCESS |
| AccountOperatorsSid | GROUP\_ALL\_ACCESS |

Else, if *ObjectHandle.Object* refers to any alias object that does not satisfy the previous condition, the DACL MUST contain the following ACEs.

| SID | Access mask |
| --- | --- |
| WorldSid | ALIAS\_EXECUTE |  ALIAS\_READ |
| AdministratorSid | ALIAS\_ALL\_ACCESS |
| AccountOperatorsSid | ALIAS\_ALL\_ACCESS |

Else, if *ObjectHandle.Object* refers to a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17) that is a member of Domain Admins or Administrators, the DACL MUST contain the following ACEs.

| SID | Access mask |
| --- | --- |
| WorldSid | USER\_EXECUTE |  USER\_READ |
| AdministratorSid | USER\_ALL\_ACCESS |
| The SID of the user referenced by *ObjectHandle.Object* | USER\_WRITE |

Else, if *ObjectHandle.Object* refers to a user object whose **ntSecurityDescriptor** does not grant Self or World the User-Change-Password [**control access right**](#gt_42f6c9e0-a2b3-4bc3-9b87-fdb902e5505e) ([[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 5.1.3.2.1), the DACL MUST contain the following ACEs.

| SID | Access mask |
| --- | --- |
| WorldSid | USER\_EXECUTE |  USER\_READ |  ~USER\_CHANGE\_PASSWORD |
| AdministratorSid | USER\_ALL\_ACCESS |
| AccountOperatorsSid | USER\_ALL\_ACCESS |
| The SID of the user referenced by *ObjectHandle.Object* | USER\_WRITE |  ~USER\_CHANGE\_PASSWORD |

Otherwise, the DACL MUST contain the following ACEs.

| SID | Access mask |
| --- | --- |
| WorldSid | USER\_EXECUTE |  USER\_READ |
| AdministratorSid | USER\_ALL\_ACCESS |
| AccountOperatorsSid | USER\_ALL\_ACCESS |
| The SID of the user referenced by *ObjectHandle.Object* | USER\_WRITE |

###### SamrQuerySecurityObject (Non-DC Configuration)

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. *ObjectHandle.GrantedAccess* MUST have the required access specified in the following table, based on the bits contained in the *SecurityInformation* parameter. On error, the server MUST abort processing and return STATUS\_ACCESS\_DENIED.

| Security information bits | Required access |
| --- | --- |
| SACL\_SECURITY\_INFORMATION | ACCESS\_SYSTEM\_SECURITY |
| OWNER\_SECURITY\_INFORMATION | READ\_CONTROL |
| GROUP\_SECURITY\_INFORMATION | READ\_CONTROL |
| DACL\_SECURITY\_INFORMATION | READ\_CONTROL |

1. The server MUST return, via the *SecurityDescriptor* parameter, a [**security descriptor**](#gt_e5213722-75a9-44e7-b026-8e4833f0d350) that only contains fields based on the bits contained in the *SecurityInformation* parameter; the fields of the security descriptor that are not returned are set to zero. The security descriptor expresses the owner and group of the referenced object and an access control ([**SACL**](#gt_c189801e-3752-4715-88f4-17804dad5782) and [**DACL**](#gt_d727f612-7a45-48e4-9d87-71735d62b321)) that has been specified either by default settings or by previous calls to [SamrSetSecurityObject](#Section_6666a06658cf4118bf4bdd54ed55ecf0). The security descriptor MUST be in terms of simple [**ACEs**](#gt_b581857f-39aa-4979-876b-daba67a40f15) and ACCESS\_MASK values as specified in the following table, based on the object type that *ObjectHandle.HandleType* references.

| Object type | ACCESS\_MASK section |
| --- | --- |
| Server | [2.2.1.1](#Section_15b9ebf7161d4c83a672dceb2ac8c448) |
| Domain | [2.2.1.4](#Section_aef23495f6aa48e9aebc22e022a2b4eb) |
| Group | [2.2.1.5](#Section_f24f9fa8798d4e7da110a5eda6900f41) |
| Alias | [2.2.1.6](#Section_2da21c6c5b1546c8bd4e6a8443216e1a) |
| User | [2.2.1.7](#Section_c0be3f43bcf943eeb0273d02ab372c53) |

#### Miscellaneous

See section [1.3](#Section_d036091033dd4e3cb5a226daf95f5ffa) for a description of these methods.

##### SamrCloseHandle (Opnum 1)

The SamrCloseHandle method closes (that is, releases server-side resources used by) any context handle obtained from this RPC interface.

1. long SamrCloseHandle(
2. [in, out] SAMPR\_HANDLE\* SamHandle
3. );

**SamHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing any context handle returned from this interface.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. If *SamHandle* is 0, the server MUST return an error.
2. Otherwise, the server MUST delete the SamContextHandle (section [3.1.1.10](#Section_cc2dc6cee45f481aa8ed5d0e273336b3)) represented by SamHandle, and then MUST return 0 for the value of SamHandle and a return code of STATUS\_SUCCESS.

##### SamrSetMemberAttributesOfGroup (Opnum 26)

The SamrSetMemberAttributesOfGroup method sets the attributes of a member relationship.

1. long SamrSetMemberAttributesOfGroup(
2. [in] SAMPR\_HANDLE GroupHandle,
3. [in] unsigned long MemberId,
4. [in] unsigned long Attributes
5. );

**GroupHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a).

**MemberId:** A [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) that represents a member of a group (which is a user or [**machine account**](#gt_ccf481e1-fb56-43d1-b6c6-0733fb52e001)).

**Attributes:** The characteristics of the membership relationship. For legal values, see section [2.2.1.10](#Section_9e093bd2e4514dd5970097b977d7ebb2).

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

On receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *GroupHandle.HandleType* is not equal to "Group".
2. *GroupHandle.GrantedAccess* MUST have the required access specified in section [3.1.2.2](#Section_87bacbd07b8b429fabc64b3d895d4e90). Otherwise, the server MUST return STATUS\_ACCESS\_DENIED.
3. In a non-[**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration, the *MemberId* parameter MUST be a member of the group referenced by *GroupHandle.Object*; otherwise, processing MUST be aborted and an error returned.
4. For a message processing specification of the *Attributes* parameter, see section [3.1.5.14.7](#Section_87988bddb1004b19874e9fce919d9114).

##### SamrGetUserDomainPasswordInformation (Opnum 44)

The SamrGetUserDomainPasswordInformation method obtains select password policy information (without requiring a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) handle).

1. long SamrGetUserDomainPasswordInformation(
2. [in] SAMPR\_HANDLE UserHandle,
3. [out] PUSER\_DOMAIN\_PASSWORD\_INFORMATION PasswordInformation
4. );

**UserHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a), representing a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17).

**PasswordInformation:** Password policy information from the user's domain.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

On receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The server MUST return an error if *UserHandle.HandleType* is not equal to "User".
2. The security identity of the client MUST have DOMAIN\_READ\_PASSWORD\_PARAMETERS access to the [**account domain object**](#gt_3ff707e8-0783-4b16-9370-2899f1078c87); if not, the server MUST abort processing and return STATUS\_ACCESS\_DENIED.
3. If the RelativeId of the **objectSid** attribute of the user object referenced by *UserHandle.Object* is DOMAIN\_USER\_RID\_KRBTGT, or if the **userAccountControl** attribute contains UF\_INTERDOMAIN\_TRUST\_ACCOUNT, UF\_WORKSTATION\_TRUST\_ACCOUNT, or UF\_SERVER\_TRUST\_ACCOUNT, then *PasswordInformation* MUST be set to all zeros, and the server MUST end processing and return STATUS\_SUCCESS.
4. The output parameter **PasswordInformation.MinPasswordLength** MUST be set to the Effective-MinimumPasswordLength attribute value (see section [3.1.1.5](#Section_6da47028e678475eb015a5a15ee98212)).
5. The output parameter **PasswordInformation.PasswordProperties** MUST be set to the **pwdProperties** attribute value on the account domain object. In addition:
   1. If the Effective-PasswordComplexityEnabled value (see section 3.1.1.5) is set, **PasswordInformation.PasswordProperties** MUST contain DOMAIN\_PASSWORD\_COMPLEX.
   2. If the Effective-PasswordReversibleEncryptionEnabled value (see section 3.1.1.5) is set, **PasswordInformation.PasswordProperties** MUST contain DOMAIN\_PASSWORD\_STORE\_CLEARTEXT.

##### SamrGetDomainPasswordInformation (Opnum 56)

The SamrGetDomainPasswordInformation method obtains select password policy information (without authenticating to the server).

1. long SamrGetDomainPasswordInformation(
2. [in] handle\_t BindingHandle,
3. [in, unique] PRPC\_UNICODE\_STRING Unused,
4. [out] PUSER\_DOMAIN\_PASSWORD\_INFORMATION PasswordInformation
5. );

**BindingHandle:** An RPC binding handle parameter, as specified in [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824) section 1.

**Unused:** A string value that is unused by the protocol. It is ignored by the server. The client MAY[<64>](#Appendix_A_64" \o "Product behavior note 64) set any value.

**PasswordInformation:** Password policy information from the account domain.

There is no security enforced for this method beyond the server-wide access check specified in section [3.1.2.1](#Section_aaa1fc5d851d4f0096890b9950e88e7b).

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The output parameter **PasswordInformation.MinPasswordLength** MUST be set to the **minPwdLength** attribute value on the [**account domain object**](#gt_3ff707e8-0783-4b16-9370-2899f1078c87).
2. The output parameter **PasswordInformation.PasswordProperties** MUST be set to the **pwdProperties** attribute value on the account domain object.
3. The method MUST return STATUS\_SUCCESS.

##### SamrRidToSid (Opnum 65)

The SamrRidToSid method obtains the [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) of an [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815), given a [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c).

1. long SamrRidToSid(
2. [in] SAMPR\_HANDLE ObjectHandle,
3. [in] unsigned long Rid,
4. [out] PRPC\_SID\* Sid
5. );

**ObjectHandle:** An RPC context handle, as specified in section [2.2.7.2](#Section_8138c078d8814d1fb37894b4c34a287a). The message processing shown later in this section contains details on which types of *ObjectHandle* are accepted by the server.

**Rid:** A RID of an account.

**Sid:** The SID of the account referenced by *Rid*.

This protocol asks the RPC runtime, via the **strict\_context\_handle** attribute, to reject the use of context handles created by a method of a different RPC interface than this one, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 3.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The *ObjectHandle.HandleType* MUST be "Domain", "User", "Group", or "Alias".
2. The output parameter *Sid* MUST be set to a SID whose [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) SID prefix is equal to the domain SID prefix of the **objectSid** attribute of the object identified by *ObjectHandle*, and whose RID suffix is equal to the *Rid* parameter.

##### SamrSetDSRMPassword (Opnum 66)

The SamrSetDSRMPassword method sets a local recovery password.

1. long SamrSetDSRMPassword(
2. [in] handle\_t BindingHandle,
3. [in, unique] PRPC\_UNICODE\_STRING Unused,
4. [in] unsigned long UserId,
5. [in, unique] PENCRYPTED\_NT\_OWF\_PASSWORD EncryptedNtOwfPassword
6. );

**BindingHandle:** An RPC binding handle parameter, as specified in [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824) section 1.

**Unused:** A string value. This value is not used in the protocol and is ignored by the server.

**UserId:** A [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of a user account. See the message processing later in this section for details on restrictions on this value.

**EncryptedNtOwfPassword:** The [**NT hash**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f) of the new password (as presented by the client) encrypted according to the specification of [ENCRYPTED\_NT\_OWF\_PASSWORD](#Section_ce061fef6d4f4802bd5d26b11f14f4a6), where the key is the *UserId*.

Upon receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The client MUST be a member of the Administrators alias, which is an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0) with the [**security identifier (SID)**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) S-1-5-32-544.
2. On a non-[**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration, the server MUST return an error code.
3. The server MAY[<65>](#Appendix_A_65" \o "Product behavior note 65) enforce parameter checks on the *UserId* parameter.
4. The server MAY[<66>](#Appendix_A_66" \o "Product behavior note 66) decrypt *EncryptedNtOwfPassword* using *UserId* as a key and use the result to store the password of a local recovery account.

##### SamrValidatePassword (Opnum 67)

The SamrValidatePassword method validates an application password against the locally stored policy.

1. long SamrValidatePassword(
2. [in] handle\_t Handle,
3. [in] PASSWORD\_POLICY\_VALIDATION\_TYPE ValidationType,
4. [in, switch\_is(ValidationType)]
5. PSAM\_VALIDATE\_INPUT\_ARG InputArg,
6. [out, switch\_is(ValidationType)]
7. PSAM\_VALIDATE\_OUTPUT\_ARG\* OutputArg
8. );

**Handle:** An RPC binding handle parameter, as specified in [[C706]](https://go.microsoft.com/fwlink/?LinkId=89824) section 1.

**ValidationType:** The password policy validation requested.

**InputArg:** The password-related material to validate.

**OutputArg:** The result of the validation.

On receiving this message, the server MUST process the data from the message subject to the following constraints:

1. The client MUST have SAM\_SERVER\_LOOKUP\_DOMAIN access on the [**server object**](#gt_62a8c543-5998-480b-8fa7-41a8f04a18e5) and DOMAIN\_READ\_PASSWORD\_PARAMETERS on the [**account domain object**](#gt_3ff707e8-0783-4b16-9370-2899f1078c87). To implement the SAM\_SERVER\_LOOKUP\_DOMAIN [**access check**](#gt_d7906f17-bb2c-4193-a3f0-848bcc351dec), the server MUST internally invoke [SamrConnect5 (section 3.1.5.1.1)](#Section_c842a8970a424ca5a6072afd05271dae) with *DesiredAccess* set to SAM\_SERVER\_LOOKUP\_DOMAIN. To implement the DOMAIN\_READ\_PASSWORD\_PARAMETERS access check, the server MUST internally invoke [SamrOpenDomain (section 3.1.5.1.5)](#Section_BA710C905B1242F89E5AD4AACC1329FA) with *ServerHandle* set to the handle returned by SamrConnect5, and with *DesiredAccess* set to DOMAIN\_READ\_PASSWORD\_PARAMETERS. If both calls succeed, the client is granted access.
2. Let the following symbolic names correspond to the values specified in the table.

| Symbolic name | Attribute value on the account domain object |
| --- | --- |
| DomainPasswordHistoryLength | pwdHistoryLength |
| DomainLockoutDuration | lockoutDuration |
| DomainLockoutObservationWindow | lockOutObservationWindow |
| DomainLockoutThreshold | lockoutThreshold |
| DomainMinimumPasswordLength | minPwdLength |
| DomainMaximumPasswordAge | maxPwdAge |
| DomainMinimumPasswordAge | minPwdAge |

1. Any field of OutputArg that is modified MUST cause the associated bit in PresentFields (in the [SAM\_VALIDATE\_PERSISTED\_FIELDS](#Section_e0b2d21d0b1c4fc08f4a895bef6ffc4e) structure) to be set according to the following table.

| Bit | Corresponding field |
| --- | --- |
| SAM\_VALIDATE\_PASSWORD\_LAST\_SET | PasswordLastSet |
| SAM\_VALIDATE\_BAD\_PASSWORD\_TIME | BadPasswordTime |
| SAM\_VALIDATE\_LOCKOUT\_TIME | LockoutTime |
| SAM\_VALIDATE\_BAD\_PASSWORD\_COUNT | BadPasswordCount |
| SAM\_VALIDATE\_PASSWORD\_HISTORY | PasswordHistoryLength |

1. Additional constraints in the following sections MUST be satisfied based on the *ValidationType* input parameter according to the following table. If the *ValidationType* input parameter does not match a row in the table, an error MUST be returned.

| ValidationType | Section |
| --- | --- |
| SamValidateAuthentication | [3.1.5.13.7.1](#Section_6f716311e64a4d83b00a8d74fb9b9fd4) |
| SamValidatePasswordChange | [3.1.5.13.7.2](#Section_f9f32118aa9f45fa85739777118deb0e) |
| SamValidatePasswordReset | [3.1.5.13.7.3](#Section_6b1f62f2c34040359c1af42b0af33398) |

###### SamValidateAuthentication

The following table lists the constraints that MUST be satisfied (in the order presented) in order to return the associated output parameters to the client. All fields of ValidateAuthenticationOutput MUST be set to 0 before any constraints are met.

| Constraint | Condition (fields based on ValidateAuthenticationInput) | ValidateAuthenticationOutput changes |
| --- | --- | --- |
| 1 | If the current time is less than or equal to LockoutTime plus DomainLockoutDuration. | ValidationStatus MUST be set to SamValidateAccountLockedOut. |
| 2 | If the current time is greater than LockoutTime plus DomainLockoutDuration. | LockoutTime MUST be set to 0 (and continue processing). |
| 3 | PasswordMatch is zero, and BadPasswordTime plus DomainLockoutObservationWindow is greater than or equal to the current time. | 1. ValidationStatus MUST be set to SamValidatePasswordIncorrect. 2. BadPasswordCount MUST be set to ValidateAuthenticationInput.BadPasswordCount plus 1. 3. BadPasswordTime MUST be set to the current time. 4. If DomainLockoutThreshold is greater than 0 and BadPasswordCount is greater than or equal to DomainLockoutThreshold, LockoutTime MUST be set to the current time. |
| 4 | PasswordMatch is zero, and BadPasswordTime plus DomainLockoutObservationWindow is less than the current time. | 1. ValidationStatus MUST be set to SamValidatePasswordIncorrect. 2. BadPasswordCount MUST be set to 1. 3. BadPasswordTime MUST be set to the current time. |
| 5 | PasswordLastSet is zero.1 | ValidationStatus MUST be set to SamValidatePasswordMustChange. |
| 6 | PasswordLastSet plus DomainMaximumPasswordAge is less than the current time. 1 | ValidationStatus MUST be set to SamValidatePasswordExpired. |
| 7 | PasswordMatched is nonzero. | 1. ValidationStatus MUST be set to SamValidateSuccess. 2. If BadPasswordCount is nonzero, BadPasswordCount MUST be set to 0. |

1 The order in which these conditions are tested SHOULD[<67>](#Appendix_A_67" \o "Product behavior note 67) follow the order shown in the preceding table.

###### SamValidatePasswordChange

The following table lists the constraints that MUST be satisfied (in the order presented) in order to return the associated output parameters to the client. All fields of ValidatePasswordChangeOutput MUST be set to 0 before any constraints are met.

| Constraint | Condition (fields based on ValidatePasswordChangeInput) | ValidatePasswordChangeOutput changes |
| --- | --- | --- |
| 1 | LockoutTime plus DomainLockoutDuration is greater than the current time. | ValidationStatus MUST be set to SamValidateAccountLockedOut. |
| 2 | LockoutTime plus DomainLockoutDuration is less than or equal to the current time. | LockoutTime MUST be set to 0. |
| 3 | PasswordLastSet plus DomainMinimumPasswordAge is greater than the current time. | ValidationStatus MUST be set to SamValidatePasswordTooRecent. |
| 4 | PasswordMatch is zero, and BadPasswordTime plus DomainLockoutObservationWindow is greater than or equal to the current time. | 1. ValidationStatus MUST be set to SamValidatePasswordIncorrect. 2. BadPasswordCount MUST be set to ValidatePasswordChangeInput.BadPasswordCount plus 1. 3. BadPasswordTime MUST be set to the current time. |
| 5 | PasswordMatch is zero, and BadPasswordTime plus DomainLockoutObservationWindow is less than the current time. | 1. ValidationStatus MUST be set to SamValidatePasswordIncorrect. 2. BadPasswordCount MUST be set to 1. 3. BadPasswordTime MUST be set to the current time. 4. If DomainLockoutThreshold is greater than 0 and BadPasswordCount is greater than or equal to DomainLockoutThreshold, LockoutTime MUST be set to the current time. |
| 6 | PasswordMatch is nonzero, and HashedPassword is equal to at least one of the first DomainPasswordHistoryLength elements of PasswordHistory (without exceeding the number of elements in PasswordHistory) where the Length field of HashedPassword is equal to the Length field of the PasswordHistory element. | ValidateStatus MUST be set to SamValidatePasswordIsInHistory. |
| 7 | PasswordMatch is nonzero. | 1. The constraints in section [3.1.1.8.5](#Section_56a9c9d72ec64c49926631b9b6ed82fb) MUST be satisfied, where sAMAccountName is ValidatePasswordChangeInput.UserAccountName and userAccountControl is UF\_NORMAL\_ACCOUNT; on error, ValidationStatus MUST be set as follows:    1. If the minimum password length constraint fails, ValidationStatus MUST be SamValidatePasswordTooShort.    2. If the maximum password length constraint fails, ValidationStatus MUST be SamValidatePasswordTooLong.    3. If any other constraint in section [3.1.1.7.2](#Section_cb1f380a0a51470584cdeea1b85953e0) or section 3.1.1.8.5 fails, ValidationStatus MUST be SamValidatePasswordNotComplexEnough.[<68>](#Appendix_A_68" \o "Product behavior note 68) 2. If any constraint from item 1 failed, the server MUST return STATUS\_SUCCESS. 3. Otherwise (if no constraint from item 1 failed), PasswordHistory MUST be updated such that ValidatePasswordChangeInput.HashedPassword is the first element in PasswordHistory, and ValidatePasswordChangeInput.InputPersistedFields.PasswordHistory elements are used, starting from the left, to fill the remaining elements of PasswordHistory such that PasswordHistory contains as many elements as possible up to DomainPasswordHistoryLength elements. 4. PasswordHistoryLength MUST be updated to be DomainPasswordHistoryLength. 5. PasswordLastSet MUST be set to the current time. 6. BadPasswordCount is set to 0. 7. ValidationStatus MUST be set to SamValidateSuccess. 8. The server MUST return any processing errors; otherwise, it MUST return STATUS\_SUCCESS. |

###### SamValidatePasswordReset

The following table lists the constraints that MUST be satisfied (in the order presented) in order to return the associated output parameters to the client. All fields of ValidatePasswordResetOutput MUST be set to 0 before any constraints are met.

| Constraint | Condition (fields based on ValidatePasswordResetInput) | ValidatePasswordResetOutput changes |
| --- | --- | --- |
| 1 | Always | 1. The constraints in section [3.1.1.8.5](#Section_56a9c9d72ec64c49926631b9b6ed82fb) MUST be satisfied, where sAMAccountName is ValidatePasswordChangeInput.UserAccountName and userAccountControl is UF\_NORMAL\_ACCOUNT; on error, ValidationStatus MUST be set as follows:    1. If the minimum password length constraint fails, ValidationStatus MUST be SamValidatePasswordTooShort.    2. If the maximum password length constraint fails, ValidationStatus MUST be SamValidatePasswordTooLong.    3. If any other constraint in section [3.1.1.7.2](#Section_cb1f380a0a51470584cdeea1b85953e0) or section 3.1.1.8.5 fails, ValidationStatus MUST be SamValidatePasswordNotComplexEnough.[<69>](#Appendix_A_69" \o "Product behavior note 69) 2. If any constraint from item 1 failed, the server MUST return STATUS\_SUCCESS. |
| 2 | PasswordMustChangeAtNextLogon is nonzero. | PasswordLastSet MUST be set to zero. |
| 3 | PasswordMustChangeAtNextLogon is zero. | PasswordLastSet MUST be set to the current time. |
| 4 | ClearLockout is nonzero. | 1. LockoutTime MUST be set to 0. 2. If ValidatePasswordResetInput.InputPersistedFields.BadPasswordCount is nonzero, BadPasswordCount MUST be set to 0. |
| 5 | Always | 1. PasswordHistory MUST be updated such that ValidatePasswordResetInput.HashedPassword is the first element in PasswordHistory and ValidatePasswordResetInput.InputPersistedFields.PasswordHistory elements are used, starting from the left, to fill the remaining elements of PasswordHistory such that PasswordHistory contains as many elements as possible up to DomainPasswordHistoryLength elements. 2. PasswordHistoryLength MUST be updated to be DomainPasswordHistoryLength. 3. BadPasswordCount MUST be set to 0. 4. ValidationStatus MUST be set to SamValidateSuccess. 5. The server MUST return any processing errors; otherwise, it MUST return STATUS\_SUCCESS. |

#### Supplemental Message Processing

##### distinguishedName Generation

This section contains constraints pertaining to the generation of a distinguishedName attribute value for objects created through this protocol. This section is referenced by the "create" pattern of methods, section [3.1.5.4](#Section_2214fd297c2d450fb68d6a0e97ebe48f). The constraints refer to an AccountType parameter from the referring section; if the object being created has the objectClass of a group, there is no AccountType parameter in the message. In this case, use an Account Type value of USER\_NORMAL\_ACCOUNT.

1. If the wellKnownObjects attribute on the [**account domain object**](#gt_3ff707e8-0783-4b16-9370-2899f1078c87) exists and contains a value that matches the [**GUID**](#gt_f49694cc-c350-462d-ab8e-816f0103c6c1) associated with Account Type, where Account Type is the AccountType parameter from the message referencing this section, the distinguishedName MUST be suffixed with the associated value from the wellKnownObject attribute. Information about the syntax of the wellKnownObject attribute is specified in [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 6.1.1.4. Unless otherwise specified, GUIDs in this document are represented using the string form of a [**universally unique identifier (UUID)**](#gt_c4813fc3-b2e5-4aa3-bde7-421d950d68d3), as specified in [[RFC4122]](https://go.microsoft.com/fwlink/?LinkId=90460) section 3.

| AccountType | wellKnownObject GUID |
| --- | --- |
| USER\_NORMAL\_ACCOUNT | a9d1ca15-7688-11d1-aded-00c04fd8d5cd |
| USER\_WORKSTATION\_TRUST\_ACCOUNT | aa312825-7688-11d1-aded-00c04fd8d5cd |
| USER\_SERVER\_TRUST\_ACCOUNT | a361b2ff-ffd2-11d1-aa4b-00c04fd7d83a |

1. If the wellKnownObjects attribute does not exist or if there is no match according to constraint 1, the distinguishedName MUST be suffixed with the associated value according to the following table.

| AccountType | distinguishedName suffix |
| --- | --- |
| USER\_NORMAL\_ACCOUNT | CN=Users,<DN of account domain object> |
| USER\_WORKSTATION\_TRUST\_ACCOUNT | CN=Computers,<DN of account domain object> |
| USER\_SERVER\_TRUST\_ACCOUNT | CN=Domain Controllers,<DN of account domain object> |

1. The server MUST prefix the [**RDN**](#gt_22198321-b40b-4c24-b8a2-29e44d9d92b9) directly in front of the suffix determined from steps 1 and 2. Implementations SHOULD[<70>](#Appendix_A_70" \o "Product behavior note 70) use the sAMAccountName as the value for the RDN, with the component type of "CN", if this choice matches the constraints of the distinguishedName attribute.

##### userAccountControl Mapping Table

| Protocol UserAccountControl | Database userAccountControl |
| --- | --- |
| USER\_ACCOUNT\_DISABLED | UF\_ACCOUNTDISABLE |
| USER\_HOME\_DIRECTORY\_REQUIRED | UF\_HOMEDIR\_REQUIRED |
| USER\_PASSWORD\_NOT\_REQUIRED | UF\_PASSWD\_NOTREQD |
| USER\_TEMP\_DUPLICATE\_ACCOUNT | UF\_TEMP\_DUPLICATE\_ACCOUNT |
| USER\_ENCRYPTED\_TEXT\_PASSWORD\_ALLOWED | UF\_ENCRYPTED\_TEXT\_PASSWORD\_ALLOWED |
| USER\_NORMAL\_ACCOUNT | UF\_NORMAL\_ACCOUNT |
| USER\_INTERDOMAIN\_TRUST\_ACCOUNT | UF\_INTERDOMAIN\_TRUST\_ACCOUNT |
| USER\_WORKSTATION\_TRUST\_ACCOUNT | UF\_WORKSTATION\_TRUST\_ACCOUNT |
| USER\_SERVER\_TRUST\_ACCOUNT | UF\_SERVER\_TRUST\_ACCOUNT |
| USER\_DONT\_EXPIRE\_PASSWORD | UF\_DONT\_EXPIRE\_PASSWD |
| USER\_MNS\_LOGON\_ACCOUNT | UF\_MNS\_LOGON\_ACCOUNT |
| USER\_SMARTCARD\_REQUIRED | UF\_SMARTCARD\_REQUIRED |
| USER\_TRUSTED\_FOR\_DELEGATION | UF\_TRUSTED\_FOR\_DELEGATION |
| USER\_NOT\_DELEGATED | UF\_NOT\_DELEGATED |
| USER\_USE\_DES\_KEY\_ONLY | UF\_USE\_DES\_KEY\_ONLY |
| USER\_DONT\_REQUIRE\_PREAUTH | UF\_DONT\_REQUIRE\_PREAUTH |
| USER\_TRUSTED\_TO\_AUTHENTICATE\_FOR\_DELEGATION | UF\_TRUSTED\_TO\_AUTHENTICATE\_FOR\_DELEGATION |
| USER\_NO\_AUTH\_DATA\_REQUIRED | UF\_NO\_AUTH\_DATA\_REQUIRED |
| USER\_ACCOUNT\_AUTO\_LOCKED | UF\_LOCKOUT |
| USER\_PASSWORD\_EXPIRED | UF\_PASSWORD\_EXPIRED |
| USER\_PARTIAL\_SECRETS\_ACCOUNT | UF\_PARTIAL\_SECRETS\_ACCOUNT |
| USER\_USE\_AES\_KEYS | UF\_USE\_AES\_KEYS |

##### PasswordCanChange Generation

The PasswordCanChange value is computed as follows:

1. If either the dBCSPwd attribute or the unicodePwd attribute does not have a value, or if either of these is equal to the respective hash of a zero-length string, PasswordCanChange MUST be 0.
2. Otherwise, the PasswordCanChange value MUST be the pwdLastSet attribute value on the [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17) plus the Effective-MinimumPasswordAge attribute value (see section [3.1.1.5](#Section_6da47028e678475eb015a5a15ee98212)).

##### PasswordMustChange Generation

The PasswordMustChange value is computed as follows:

1. If the userAccountControl attribute value on the target [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17) contains any of the following bits: UF\_DONT\_EXPIRE\_PASSWD, UF\_SMARTCARD\_REQUIRED, UF\_INTERDOMAIN\_TRUST\_ACCOUNT, UF\_WORKSTATION\_TRUST\_ACCOUNT, or UF\_SERVER\_TRUST\_ACCOUNT, the PasswordMustChange value MUST be 0x7FFFFFFF FFFFFFFF.
2. Else, if the pwdLastSet attribute value on the user object is 0, the PasswordMustChange value MUST be 0.
3. Else, if the Effective-MaximumPasswordAge attribute value (see section [3.1.1.5](#Section_6da47028e678475eb015a5a15ee98212)) is 0, the PasswordMustChange value MUST be 0x7FFFFFFF FFFFFFFF.
4. Otherwise, the PasswordMustChange value MUST be the pwdLastSet attribute value on the user object plus the Effective-MaximumPasswordAge attribute value (see section 3.1.1.5).

##### Account Lockout Enforcement and Reset

1. Let U be the user account that is the subject of a change password request.
2. If U's lockoutTime attribute value plus the attribute value of Effective-LockoutDuration (see section [3.1.1.5](#Section_6da47028e678475eb015a5a15ee98212)) is less than the current time, the server MUST abort the request and return STATUS\_ACCOUNT\_LOCKED\_OUT.
3. Otherwise, U's lockoutTime MUST be updated to the value 0.

##### Account Lockout State Maintenance

1. Let U be the user account that is the subject of a change password request.
2. If the Effective-LockoutThreshold attribute value (see section [3.1.1.5](#Section_6da47028e678475eb015a5a15ee98212)) is greater than zero and U's lockoutTime attribute value is zero or nonexistent, all of the following constraints apply:
   1. If the time period between U's badPwdTime attribute value and the current time is greater than the attribute value of the Effective-LockoutObservationWindow (see section 3.1.1.5), the server MUST set U's badPwdCount attribute value to one. Otherwise, the server MUST increment U's badPwdCount attribute value by one.
   2. The server MUST update U's badPwdTime attribute value to the current time (with FILETIME syntax).
   3. If the Effective-LockoutThreshold attribute value (see section 3.1.1.5) is greater than zero, and BadPasswordCount is greater than or equal to lockoutThreshold, the server MUST update U's lockoutTime attribute to the current time (with FILETIME syntax).

##### Attributes Field Handling

This protocol associates a field called "Attributes" with a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a) and a user membership for a group. This field is a bit field that uses values from the space specified in section [2.2.1.10](#Section_9e093bd2e4514dd5970097b977d7ebb2).

For a group object, this field can be set via [SamrSetInformationGroup](#Section_e66db19f600a481bbc4e23953433255d) and queried via [SamrQueryInformationGroup](#Section_8fd0003f807640ae872f9cdb6576b3c7) and the [SamrQueryDisplayInformation](#Section_c1458942f2d54317a888abd27abad504) family of methods.

For a user membership, this field can be set via [SamrAddMemberToGroup](#Section_3c70fec36a2e48efbd4eaa2b3a1cd96a) and [SamrSetMemberAttributesOfGroup](#Section_a4db0c2755404d2fb11108db1ee0b5a4) and queried via [SamrGetGroupsForUser](#Section_a4adbf20040f4416a960e5b7917fdae7) and [SamrGetMembersInGroup](#Section_3ed5030d88a342caa6e08c12aa2fdfbd).

This section specifies the message processing for this field for the aforementioned methods.

On a [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) configuration:

* On query, the returned value MUST be a logical union of the following bits: SE\_GROUP\_MANDATORY, SE\_GROUP\_ENABLED\_BY\_DEFAULT, and SE\_GROUP\_ENABLED.
* On set, this field is ignored. The client SHOULD[<71>](#Appendix_A_71" \o "Product behavior note 71) set the value to the logical union of the following bits: SE\_GROUP\_MANDATORY, SE\_GROUP\_ENABLED\_BY\_DEFAULT, and SE\_GROUP\_ENABLED.

On a non-DC configuration:

* Any value set via SamrSetInformationGroup MUST be returned via a subsequent call to SamrQueryInformationGroup or the SamrQueryDisplayInformation family of methods at any time in the future (not just within the current session). If no such SamrSetInformationGroup call has been made, a default value of zero MUST be returned.
* Any value set via SamrAddMemberToGroup or SamrSetMemberAttributesOfGroup MUST be returned via a subsequent call to SamrGetGroupsForUser or SamrGetMembersInGroup at any time in the future (not just within the current session). If no such call to SamrAddMemberToGroup or SamrSetMemberAttributesOfGroup has been made, a default value of zero MUST be returned.

##### Domain Field to Attribute Name Mapping

This table specifies the field-to-database-attribute mapping, where the field is a field in a domain-related structure such as [SAMPR\_DOMAIN\_GENERAL\_INFORMATION (section 2.2.3.10)](#Section_85973e1c96f24c808135b24d74ad7794) and the database attribute is an attribute defined on a [**domain object**](#gt_2346e8d0-b0a0-481f-b644-85c45c60d4d1). These attributes are from the data model specified in section [3.1.1](#Section_814e12f610374b648d288f1b899dc57f).

| Field name | Database attribute |
| --- | --- |
| CreationTime | creationTime |
| DomainModifiedCount | modifiedCount |
| DomainName | Name |
| ForceLogoff | forceLogoff |
| LockoutDuration | lockoutDuration |
| LockoutObservationWindow | lockOutObservationWindow |
| LockoutThreshold | lockoutThreshold |
| ModifiedCountAtLastPromotion | modifiedCountAtLastProm |
| MaxPasswordAge | maxPwdAge |
| MinPasswordAge | minPwdAge |
| MinPasswordLength | minPwdLength |
| PasswordHistoryLength | pwdHistoryLength |
| PasswordProperties | pwdProperties |
| OemInformation | oEMInformation |
| ReplicaSourceNodeName | domainReplica |
| UasCompatibilityRequired | uASCompat |

##### Group Field to Attribute Name Mapping

This table specifies the field-to-database-attribute mapping, where the field is a field in a group-related structure such as [SAMPR\_GROUP\_GENERAL\_INFORMATION (section 2.2.4.3)](#Section_fe821071d3334eef862af45408343e4e) and the database attribute is an attribute defined on a [**group object**](#gt_7ce4771c-2043-49b8-85d3-0c60c7789f9a). These attributes are from the data model specified in section [3.1.1](#Section_814e12f610374b648d288f1b899dc57f).

| Field name | Database attribute or value |
| --- | --- |
| AdminComment | Description |
| Attributes | See section [3.1.5.14.7](#Section_87988bddb1004b19874e9fce919d9114) for a message processing specification. |
| MemberCount | The number of values in the member attribute. |
| Name | sAMAccountName |

##### Alias Field to Attribute Name Mapping

This table specifies the field-to-database-attribute mapping, where the field is a field in a group-related structure such as [SAMPR\_ALIAS\_GENERAL\_INFORMATION (section 2.2.5.2)](#Section_025890e9c48347e08761632caf30be3f) and the database attribute is an attribute defined on an [**alias object**](#gt_0387e636-5654-4910-9519-1f8326cf5ec0). These attributes are from the data model specified in section [3.1.1](#Section_814e12f610374b648d288f1b899dc57f).

| Field name | Database attribute or value |
| --- | --- |
| AdminComment | Description |
| MemberCount | The number of values in the member attribute. |
| Name | sAMAccountName |

##### User Field to Attribute Name Mapping

This table specifies the field-to-database-attribute mapping, where the field is a field in a user-related structure such as [SAMPR\_USER\_ALL\_INFORMATION (section 2.2.6.6)](#Section_dc966b81da274daea28cec16534f1cb9) and the database attribute is an attribute defined on a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17). These attributes are from the data model specified in section [3.1.1](#Section_814e12f610374b648d288f1b899dc57f).

| Field name | Database attribute |
| --- | --- |
| LastLogon | lastLogon |
| LastLogoff | lastLogoff |
| PasswordLastSet | pwdLastSet |
| AccountExpires | accountExpires |
| PasswordCanChange | See section [3.1.5.14.3](#Section_d756a2fb3704488aab1ecb57f91f7ab4) for message processing. |
| PasswordMustChange | See section [3.1.5.14.4](#Section_e61ba8096ddd41968534d5585c9b2e3f) for message processing. |
| UserName | sAMAccountName |
| FullName | displayName |
| HomeDirectory | homeDirectory |
| HomeDirectoryDrive | homeDrive |
| ScriptPath | scriptPath |
| ProfilePath | profilePath |
| AdminComment | description |
| WorkStations | userWorkstations |
| UserComment | comment |
| Parameters | userParameters |
| UserId | [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) of objectSid |
| PrimaryGroupId | primaryGroupId |
| UserAccountControl\* | userAccountControl |
| LogonHours | logonHours |
| BadPasswordCount | badPwdCount |
| LogonCount | logonCount |
| CountryCode | countryCode |
| CodePage | codePage |
| NtOwfPassword\*\* | unicodePwd |
| LmOwfPassword\*\* | dBCSPwd |
| NtPasswordPresent\*\* | Not persisted as a database attribute |
| LmPasswordPresent\*\* | Not persisted as a database attribute |
| PrivateData\*\* | Not persisted as a database attribute |
| PasswordExpired\*\* | Not persisted as a database attribute |
| SecurityDescriptor\*\* | ntSecurityDescriptor |

\*On read of UserAccountControl, the database attribute value MUST be:

1. Augmented with the UF\_LOCKOUT bit if the lockoutTime attribute value on the target object is nonzero and if its value plus the Effective-LockoutDuration attribute value (section [3.1.1.5](#Section_6da47028e678475eb015a5a15ee98212)) is less than the current time.
2. Augmented with UF\_PASSWORD\_EXPIRED if PasswordMustChange is less than the current time.
3. Translated according to the table in section [3.1.5.14.2](#Section_8a193181a7a249dfa8b1f689aaa6987c).

\*\*NtOwfPassword, NtPasswordPresent, LmOwfPassword, LmPasswordPresent, PrivateData, PasswordExpired, and SecurityDescriptor cannot be returned by the SAM Remote Protocol, as indicated by the processing instructions specified in sections [3.1.5.5.6](#Section_4AD8D54C0D5A4D5A9E9A1BC9EE008D47) and [3.1.5.5.5](#Section_29AB27F661DA4C7D863CE228EE798F4D)

### Timer Events

None.

### Other Local Events

#### Domain Join Processing

This event accepts the following parameter:

DomainSID: A [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) ([[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2) section 2.4.2) that identifies the [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) being joined.

Upon invocation of this event, the server MUST perform the following processing:

1. Let A be the [**database object**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f) whose **objectSid** is S-1-5-32-544, whose database object type is group (that is, an object with **objectClass** group or derived from group), and with **groupType** containing GROUP\_TYPE\_RESOURCE\_GROUP. A's member attribute MUST be updated to add a [**dsname**](#gt_4d5e1f08-aa00-4dde-9411-7dd6e09ed85a) value that references the object whose **objectSid** specifies the SID for the Domain Administrators group. The SID for the Domain Administrators group is constructed by joining the DomainSID parameter with the well-known [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) for Domain Administrators ([[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 6.1.1.6.5).
2. Let B be the database object whose **objectSid** is S-1-5-32-545, whose database object type is group (that is, an object with **objectClass** group or derived from group), and with **groupType** containing GROUP\_TYPE\_RESOURCE\_GROUP. B's **member** attribute MUST be updated to add a dsname value that references the object whose **objectSid** specifies the SID for the Domain Users group. The SID for the Domain Users group is constructed by joining the DomainSID parameter with the well-known RID for Domain Users ([MS-ADTS] section 6.1.1.6.9).

#### Domain Unjoin Processing

This event accepts the following parameter:

DomainSID: A [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) ([[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2) section 2.4.2) identifying the [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) being joined.

Upon invocation of this event, the server MUST perform the following processing:

1. Let A be the [**database object**](#gt_60122e64-9d34-40ee-a387-782eaf7adf2f) whose **objectSid** is S-1-5-32-544, whose database object type is group (that is, an object with **objectClass** group or derived from group), and with **groupType** containing GROUP\_TYPE\_RESOURCE\_GROUP. If A's member attribute contains a [**dsname**](#gt_4d5e1f08-aa00-4dde-9411-7dd6e09ed85a) value that references the object whose **objectSid** specifies the SID for the Domain Administrators group, the server MUST remove that value. The SID for the Domain Administrators group is constructed by joining the DomainSID parameter with the well-known [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) for Domain Administrators ([[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 6.1.1.6.5).
2. Let B be the database object whose **objectSid** is S-1-5-32-545, whose database object type is group (that is, an object with **objectClass** group or derived from group), and with **groupType** containing GROUP\_TYPE\_RESOURCE\_GROUP. If B's **member** attribute contains a dsname value that references the object whose **objectSid** specifies the SID for the Domain Users group, the server MUST remove that value. The SID for the Domain Users group is constructed by joining the DomainSID parameter with the well-known RID for Domain Users ([MS-ADTS] section 6.1.1.6.9).

## Client Details

### Abstract Data Model

As discussed in section [1.5](#Section_1080b153835e44f0966daa9ad474e203), an [**original equipment manufacturer (OEM) code page**](#gt_442ab13f-d2c1-4128-b1db-f3bea4b8224e) MUST be configured in the server implementation for the server to accept data that is encoded in an OEM code page and to return select results that are encoded in an OEM code page. In particular, the client MUST use an OEM code page to encode or decode an [RPC\_STRING](#Section_28330caf78c449e091c146bbfeed633b) structure when participating in the SAM Remote Protocol (Client-to-Server).

### Security Model

The client MUST create a secure RPC session such that the server can identify and determine the authorization for the client. (For more information on secure RPC, see [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15).) This requirement exists so that the server can implement its security model (section [3.1.2](#Section_5e3f7f39df374f9cbb99a27d79cc957f)).

#### RC4 Cipher Usage

The data MUST be encrypted and decrypted using the [**RC4**](#gt_d57eac33-f561-4a08-b148-dfcf29cfb4d8) algorithm (for more information about RC4, see [[SCHNEIER]](https://go.microsoft.com/fwlink/?LinkId=817338) section 17.1). The key, required during runtime by the RC4 algorithm, MUST be the 16-byte key specified by the method using this structure (for examples, see sections [3.1.5.10.2](#Section_8d0bf63efa5f4c75be22558c52075842) and [3.1.5.10.3](#Section_acb3204ada8b478e91391ea589edb880)). The encrypted portion of the SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW.Buffer structure MUST be protected in the same way, but the 16-byte key is specified in section [3.2.2.2](#Section_d1310f9523af4c2d92c0aa5166da8450).

#### MD5 Usage

The key required during runtime by the [**RC4**](#gt_d57eac33-f561-4a08-b148-dfcf29cfb4d8) encryption algorithm that encrypts and decrypts the protected portion of the SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW.Buffer is specified by the following pseudocode.

1. CALL MD5Init(md5context)
2. CALL MD5Update(md5context, SAMPR\_USER\_PASSWORD\_NEW.ClearSalt, 16)
3. CALL MD5Update(md5context, user-session-key, 16)
4. CALL MD5Final(md5context)

Where:

* MD5Init, MD5Update, and MD5Final are predicates/functions defined in [[RFC1321]](https://go.microsoft.com/fwlink/?LinkId=90275).
* md5Context is a variable of type MD5\_CTX, as specified in [RFC1321].
* user-session-key is the 16-byte SMB session key obtained as specified in section [3.2.2.3](#Section_a26ab2cb3e0949809f2d8d75f73005ae).

#### Acquiring an SMB Session Key

The client MUST retrieve the SMB session key as specified in [[MS-CIFS]](%5bMS-CIFS%5d.pdf#Section_d416ff7cc536406ea9514f04b2fd1d2b) section 3.4.4.6.

#### AES Cipher Usage

Advanced Encryption Standard (AES) encryption is achieved in this protocol by using the AEAD-AES-256-CBC-HMAC-SHA512 cipher, as specified in [[AES-CBC]](https://go.microsoft.com/fwlink/?linkid=2162731). In order to use an encryption key, AES encryption requires a shared secret between the server and the client. Create this encryption key as a content encryption key (CEK), as specified in following methods, as appropriate:

* For the SamrSetInformationUser2 method (section [3.1.5.6.4](#Section_99ee9f3943e84bbaac3a82e0c0e0699e)) with the UserInternal7Information (section [3.1.5.6.4.1](#Section_c00ca1dd5a294655bd6f6e680e154bc9)) or UserInternal8Information (section [3.1.5.6.4.6](#Section_ce4e6060629d4bedad7ec464f4b989f8)) value, obtain the shared secret and CEK as the 16-byte user SMB session key, as specified in section [3.2.2.3](#Section_a26ab2cb3e0949809f2d8d75f73005ae).
* For the SamrUnicodeChangePasswordUser4 method (section [3.1.5.10.4](#Section_bbc1c5e59b814038b2b9c87d3569ed38)), the shared secret is the plaintext old password and the CEK is generated as specified in section [3.2.2.5](#Section_c37085ecc4564a1cad98a4c2b39a35fd).

The data MUST be encrypted and decrypted using AEAD-AES-256-CBC-HMAC-SHA512 as follows:

* Let IV be a random 16-byte number.
* Then the encryption is done as follows:

Let enc\_key ::= HMAC-SHA-512(CEK, SAM\_AES256\_ENC\_KEY\_STRING)

Let mac\_key ::= HMAC-SHA-512(CEK, SAM\_AES256\_MAC\_KEY\_STRING)

Let Cipher ::= AES-CBC(enc\_key, IV, secret\_plaintext)

Let AuthData ::= HMAC-SHA-512(mac\_key, versionbyte + IV + Cipher + versionbyte\_length)

#### Deriving an Encryption Key from a Plaintext Password

The client MUST derive the CEK in the following manner:

CEK :: = (PBKDF2(NT HASH of “OldPassword”, Salt, IterationCount, 512))

### Timers

None.

### Initialization

None.

### Message Processing Events and Sequencing Rules

To obtain any context handle to the server, one of the following methods MUST be called initially: [SamrConnect2](#Section_1076eb2a4f514c5aa7c7a78323b06198), [SamrConnect4](#Section_c1b14e670f4647a69fa70689d284afc9), or [SamrConnect5](#Section_c842a8970a424ca5a6072afd05271dae). With the *ServerHandle* parameter returned from these methods, it is possible to obtain other context handles and call any associated methods on the handle. See section [4.1](#Section_3d8e23d8d9df481f83b39175f980294c) for an example.

**Note**  The following methods do not require a context handle and can be called directly; they also do not return any context handle:

* [SamrGetDomainPasswordInformation](#Section_1a8841487e904a93b4c47a0006f59026)
* [SamrSetDSRMPassword](#Section_9bcad7d2b8e14d28a03386040b7b3ce9)
* [SamrValidatePassword](#Section_c78a7239f8fc4a42bb71321e897dc046)
* [SamrOemChangePasswordUser2](#Section_8d0bf63efa5f4c75be22558c52075842)
* [SamrUnicodeChangePasswordUser2](#Section_acb3204ada8b478e91391ea589edb880)

**Note**  A user account MUST be enabled by clearing the UF\_ACCOUNTDISABLE bit from the **userAccountControl** attribute before that account will be able to authenticate, as specified in [[MS-KILE]](%5bMS-KILE%5d.pdf#Section_2a32282edd484ad9a542609804b02cc9) section 3.3.5.7.1.

### Timer Events

The protocol does not include its own timer events. Information about any transport-level timers is specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15).

### Other Local Events

None.

# Protocol Examples

## Creating a User Account

The following sequence of methods and parameters creates a user account given a network address of "msdc-1", a [**domain name**](#gt_45a1c9f1-0263-49a8-97c7-7aca1a99308c) of "ms", and a user name of "testuser".

1. Send [SamrConnect](#Section_defe20910a614dfabe9a2c1206d53a1f).

| Parameter field | Parameter value |
| --- | --- |
| ServerName | msdc-1 |
| DesiredAccess | 0x31 |

1. Receive SamrConnect.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |
| ServerHandle | [implementation-specific value] serverHandle |

1. Send [SamrLookupDomainInSamServer](#Section_47492d59e0954398b03e8a062b989123).

| Parameter field | Parameter value |
| --- | --- |
| ServerHandle | serverHandle |
| Name.Length | 4 |
| Name.MaximumLength | 4 |
| Name.Buffer | ms |

1. Receive SamrLookupDomainInSamServer.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |
| DomainId | [implementation-specific [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d)]. For example: S-1-5-21-3448151421-356457007-600757626 |

1. Send [SamrOpenDomain](#Section_ba710c905b1242f89e5ad4aacc1329fa).

| Parameter field | Parameter value |
| --- | --- |
| ServerHandle | serverHandle |
| DesiredAccess | 0x00000010 |
| DomainId | S-1-5-21-3448151421-356457007-600757626 |

1. Receive SamrOpenDomain.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |
| DomainHandle | [implementation-specific value] domainHandle |

1. Send [SamrCreateUser2InDomain](#Section_a98d7fbb17354fbfb41aef363c899002).

| Parameter field | Parameter value |
| --- | --- |
| DomainHandle | domainHandle |
| Name.Length | 16 |
| Name.MaximumLength | 16 |
| Name.Buffer | testuser |
| AccountType | 0x00000080 |
| DesiredAccess | 0x02000000 |

1. Receive SamrCreateUser2InDomain.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |
| UserHandle | [implementation-specific value] userHandle |
| GrantedAccess | 0xf07ff |
| RelativeId | 2810 |

1. Send [SamrCloseHandle](#Section_55d134dfe25748ad8afacb2ca45cd3cc).

| Parameter field | Parameter value |
| --- | --- |
| Handle | userHandle |

1. Receive SamrCloseHandle.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |
| Handle | 0 |

1. Send SamrCloseHandle.

| Parameter field | Parameter value |
| --- | --- |
| Handle | domainHandle |

1. Receive SamrCloseHandle.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |
| Handle | 0 |

1. Send SamrCloseHandle.

| Parameter field | Parameter value |
| --- | --- |
| Handle | serverHandle |

1. Receive SamrCloseHandle.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |
| Handle | 0 |

## Enabling a User Account

The following sequence of methods and parameters enables the user account created in the previous example. This is performed on the machine with the network address of "msdc-1", a [**domain name**](#gt_45a1c9f1-0263-49a8-97c7-7aca1a99308c) of "ms", and a user name of "testuser" with Relative ID = 2810.

1. Send [SamrConnect](#Section_defe20910a614dfabe9a2c1206d53a1f).

| Parameter field | Parameter value |
| --- | --- |
| ServerName | msdc-1 |
| DesiredAccess | 0x31 |

1. Receive SamrConnect.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |
| ServerHandle | [implementation-specific value] serverHandle |

1. Send [SamrLookupDomainInSamServer](#Section_47492d59e0954398b03e8a062b989123).

| Parameter field | Parameter value |
| --- | --- |
| ServerHandle | serverHandle |
| Name.Length | 4 |
| Name.MaximumLength | 4 |
| Name.Buffer | ms |

1. Receive SamrLookupDomainInSamServer.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |
| DomainId | [implementation-specific [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d)]. For example: S-1-5-21-3448151421-356457007-600757626 |

1. Send [SamrOpenDomain](#Section_ba710c905b1242f89e5ad4aacc1329fa).

| Parameter field | Parameter value |
| --- | --- |
| ServerHandle | serverHandle |
| DesiredAccess | 0x00000200 |
| DomainId | S-1-5-21-3448151421-356457007-600757626 |

1. Receive SamrOpenDomain.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |
| DomainHandle | [implementation-specific value] domainHandle |

1. Send [SamrOpenUser](#Section_0aee1c31ec404633bb560cf8429093c0).

| Parameter field | Parameter value |
| --- | --- |
| DomainHandle | domainHandle |
| DesiredAccess | 0x02000000 |
| UserId | 2810 |

1. Receive SamrOpenUser.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |
| UserHandle | [implementation-specific value] userHandle |

1. Send [SamrSetInformationUser2](#Section_99ee9f3943e84bbaac3a82e0c0e0699e).

| Parameter field | Parameter value |
| --- | --- |
| UserHandle | userHandle |
| UserInformationClass | 16 |
| Buffer | Control = { 0x00000010 } |

1. Receive SamrSetInformationUser2.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |

1. Send [SamrCloseHandle](#Section_55d134dfe25748ad8afacb2ca45cd3cc).

| Parameter field | Parameter value |
| --- | --- |
| Handle | userHandle |

1. Receive SamrCloseHandle.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |
| Handle | 0 |

1. Send SamrCloseHandle.

| Parameter field | Parameter value |
| --- | --- |
| Handle | domainHandle |

1. Receive SamrCloseHandle.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |
| Handle | 0 |

1. Send SamrCloseHandle.

| Parameter field | Parameter value |
| --- | --- |
| Handle | serverHandle |

1. Receive SamrCloseHandle.

| Parameter field | Parameter value |
| --- | --- |
| Status | 0 |
| Handle | 0 |

## Encrypting an NT or LM Hash

The following example shows actual values for the cleartext passwords and password hashes as well as the key derivations necessary to apply [[FIPS81]](https://go.microsoft.com/fwlink/?LinkId=89874).

* Old password is "OLDPASSWORD".

[**LM hash**](#gt_58a72159-b1e9-4770-9b06-d7a6b16279a8) of "OLDPASSWORD":

1. c9 b8 1d 93 9d 6f d8 0c d4 08 e6 b1 05 74 18 64

[**NT hash**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f) of "OLDPASSWORD":

1. 66 77 b2 c3 94 31 13 55 b5 4f 25 ee c5 bf ac f5

* New password is "NEWPASSWORD".

LM hash of "NEWPASSWORD":

1. 09 ee ab 5a a4 15 d6 e4 d4 08 e6 b1 05 74 18 64

NT hash of "NEWPASSWORD":

1. 25 67 81 a6 20 31 28 9d 3c 2c 98 c1 4f 1e fc 8c

To demonstrate sample data values for the 7-byte InputKey and 8-byte OutputKey used in section [2.2.11.1.2](#Section_ebdb15df8d0d43479d62082e6eccac40), the following values are used for the encryption of the old NT hash with the new NT hash shown above.

1. Split the NT hash of the old password into two blocks ([2.2.11.1.1](#Section_a5252e8c25e74616a37555ced086b19b)).
2. Block 1: 66 77 b2 c3 94 31 13 55
3. Block 2: b5 4f 25 ee c5 bf ac f5
4. Split the NT hash of the new password into two blocks (2.2.11.1.1).
5. Block 1: 25 67 81 a6 20 31 28 9d
6. Block 2: 3c 2c 98 c1 4f 1e fc 8c
7. The 7-byte keys are derived as stated in section [2.2.11.1.4](#Section_ba4377867de847b4b2695a595c255327) using the 16-byte hash value. Apply the algorithm in section 2.2.11.1.2 to transform the 7-byte key into an 8-byte key.
8. 7 byte InputKey for block 1: 25 67 81 a6 20 31 28
9. 8 byte OutputKey for block 1: 25 b3 e0 34 62 01 c4 51
10. 7 byte InputKey for block 2: 9d 3c 2c 98 c1 4f 1e
11. 8 byte OutputKey for block 2: 9d 9e 0b 92 8c 0b 3d 3d
12. Apply [FIPS81] to encrypt both blocks using these keys.

OldNtOwfEncryptedWithNewNt:

1. da 39 84 64 27 f5 e6 c9 48 2c 8f e9 b3 3a 16 07

Likewise, the following values are used for encryption of the old LM hash with the new NT hash.

1. Split the LM hash of the old password into two blocks (2.2.11.1.1).
2. Block 1: c9 b8 1d 93 9d 6f d8 0c
3. Block 2: d4 08 e6 b1 05 74 18 64
4. As before, split the NT hash of the new password into two blocks (2.2.11.1.1).
5. Block 1: 25 67 81 a6 20 31 28 9d
6. Block 2: 3c 2c 98 c1 4f 1e fc 8c
7. 7 byte InputKey for block 1: 25 67 81 a6 20 31 28
8. 8 byte OutputKey for block 1: 25 b3 e0 34 62 01 c4 51
9. 7 byte InputKey for block 2: 9d 3c 2c 98 c1 4f 1e
10. 8 byte OutputKey for block 2: 9d 9e 0b 92 8c 0b 3d 3d
11. Apply [FIPS81] to encrypt both blocks using these keys.

OldLmOwfEncryptedWithNewNt:

1. 80 45 7a 72 72 5a 37 9c ed 8b 07 d2 fd 6f 46 ff

# Security

## Security Considerations for Implementers

Sensitive information, such as the cleartext password for accounts, is communicated through this protocol; therefore, implementers have to pay special attention to the secrecy of this data. Although this protocol does not use transport-level encryption (with the exception of [SamrValidatePassword](#Section_c78a7239f8fc4a42bb71321e897dc046)), it does rely on the key strength of the SMB transport for encrypting cleartext data.

Using [SamrSetInformationUser2](#Section_99ee9f3943e84bbaac3a82e0c0e0699e) with UserInternal8Information and UserInternal7Information is the best choice that a client can make for setting a cleartext password through this protocol, because the cryptography used is the strongest in this protocol.

Creating a [**user object**](#gt_e767a471-c3fa-4e4b-a40c-daeb08f82a17) is a multi-step process in this protocol. These steps are outlined in the example in section [4.1](#Section_3d8e23d8d9df481f83b39175f980294c). After completing these steps correctly, the server creates a user object in its abstract database. However, the user object is not usable for authentication in this state. The user object needs to be enabled for authentication. The steps for enabling a user object are outlined in the example in section [4.2](#Section_bf8cfb7624f742dea95fe5b9ec7435d0). Optionally, a password can be set on the user object. As specified in the previous paragraph, SamrSetInformationUser2 with UserInternal8Information and UserInternal7Information is the best choice for setting a cleartext password in this protocol.

## Index of Security Parameters

| Security Parameter | Section |
| --- | --- |
| Service principal name for server | [2.1](#Section_084da2e70ba044fc8f17e8a200c69eb5) |
| Encryption algorithm for hashes | [2.2.11.1](#Section_3f5ec79db4494ab29423c4dccbe0b184) |
| End-user password (to set) | [3.1.5.6.4](#Section_99ee9f3943e84bbaac3a82e0c0e0699e) |
| End-user password (to change) | [3.1.5.10.1](#Section_9699d8cae1a4433ca8c3d7bebeb01476) |
| End-user password (to change) | [3.1.5.10.2](#Section_8d0bf63efa5f4c75be22558c52075842) |
| End-user password (to change) | [3.1.5.10.3](#Section_acb3204ada8b478e91391ea589edb880) |
| Recovery password (to set) | [3.1.5.13.6](#Section_9bcad7d2b8e14d28a03386040b7b3ce9) |
| End-user application password (set, change, and authenticate) | [3.1.5.13.7](#Section_c78a7239f8fc4a42bb71321e897dc046) |
| Encryption key for storing an encrypted [**LM hash**](#gt_58a72159-b1e9-4770-9b06-d7a6b16279a8) | [3.1.1.8.6](#Section_9a014fcd83f64432811031fae3d78f6b) |
| Encryption key for storing an encrypted [**NT hash**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f) | [3.1.1.8.7](#Section_22b79fa813bd4952a3999d5a93b2256f) |

# Appendix A: Full IDL

For ease of implementation, the full IDL is provided below, where "ms-dtyp.idl" is the IDL specified in [[MS-DTYP]](%5bMS-DTYP%5d.pdf#Section_cca2742956894a16b2b49325d93e4ba2) Appendix A.

1. import "ms-dtyp.idl";
2. [
3. uuid(12345778-1234-ABCD-EF00-0123456789AC),
4. version(1.0),
5. ms\_union,
6. pointer\_default(unique)
7. ]
8. interface samr{
9. typedef struct \_RPC\_STRING {
10. unsigned short Length;
11. unsigned short MaximumLength;
12. [size\_is(MaximumLength), length\_is(Length)] char \* Buffer;
13. } RPC\_STRING, \*PRPC\_STRING;
14. typedef struct \_OLD\_LARGE\_INTEGER {
15. unsigned long LowPart;
16. long HighPart;
17. } OLD\_LARGE\_INTEGER, \*POLD\_LARGE\_INTEGER;
18. typedef [handle] wchar\_t \* PSAMPR\_SERVER\_NAME;
19. typedef [context\_handle] void \* SAMPR\_HANDLE;
20. typedef struct \_ENCRYPTED\_LM\_OWF\_PASSWORD {
21. char data[16];
22. } ENCRYPTED\_LM\_OWF\_PASSWORD,
23. \*PENCRYPTED\_LM\_OWF\_PASSWORD,
24. ENCRYPTED\_NT\_OWF\_PASSWORD,
25. \*PENCRYPTED\_NT\_OWF\_PASSWORD;
26. typedef struct \_SAMPR\_ULONG\_ARRAY {
27. unsigned long Count;
28. [size\_is(Count)] unsigned long \* Element;
29. } SAMPR\_ULONG\_ARRAY, \*PSAMPR\_ULONG\_ARRAY;
30. typedef struct \_SAMPR\_SID\_INFORMATION {
31. PRPC\_SID SidPointer;
32. } SAMPR\_SID\_INFORMATION, \*PSAMPR\_SID\_INFORMATION;
33. typedef struct \_SAMPR\_PSID\_ARRAY {
34. [range(0, 1024)] unsigned long Count;
35. [size\_is(Count)] PSAMPR\_SID\_INFORMATION Sids;
36. } SAMPR\_PSID\_ARRAY, \*PSAMPR\_PSID\_ARRAY;
37. typedef struct \_SAMPR\_PSID\_ARRAY\_OUT {
38. unsigned long Count;
39. [size\_is(Count)] PSAMPR\_SID\_INFORMATION Sids;
40. } SAMPR\_PSID\_ARRAY\_OUT, \*PSAMPR\_PSID\_ARRAY\_OUT;
41. typedef struct \_SAMPR\_RETURNED\_USTRING\_ARRAY {
42. unsigned long Count;
43. [size\_is(Count)] PRPC\_UNICODE\_STRING Element;
44. } SAMPR\_RETURNED\_USTRING\_ARRAY, \*PSAMPR\_RETURNED\_USTRING\_ARRAY;
45. typedef enum \_SID\_NAME\_USE {
46. SidTypeUser = 1,
47. SidTypeGroup,
48. SidTypeDomain,
49. SidTypeAlias,
50. SidTypeWellKnownGroup,
51. SidTypeDeletedAccount,
52. SidTypeInvalid,
53. SidTypeUnknown,
54. SidTypeComputer, // Not used.
55. SidTypeLabel // Not used.
56. } SID\_NAME\_USE, \*PSID\_NAME\_USE;
57. typedef struct \_RPC\_SHORT\_BLOB {
58. unsigned short Length;
59. unsigned short MaximumLength;
60. [size\_is(MaximumLength/2), length\_is(Length/2)]
61. unsigned short\* Buffer;
62. } RPC\_SHORT\_BLOB, \*PRPC\_SHORT\_BLOB;
63. typedef struct \_SAMPR\_RID\_ENUMERATION {
64. unsigned long RelativeId;
65. RPC\_UNICODE\_STRING Name;
66. } SAMPR\_RID\_ENUMERATION, \*PSAMPR\_RID\_ENUMERATION;
67. typedef struct \_SAMPR\_ENUMERATION\_BUFFER {
68. unsigned long EntriesRead;
69. [size\_is(EntriesRead)] PSAMPR\_RID\_ENUMERATION Buffer;
70. } SAMPR\_ENUMERATION\_BUFFER, \*PSAMPR\_ENUMERATION\_BUFFER;
71. typedef struct \_SAMPR\_SR\_SECURITY\_DESCRIPTOR {
72. [range(0, 256 \* 1024)] unsigned long Length;
73. [size\_is(Length)] unsigned char\* SecurityDescriptor;
74. } SAMPR\_SR\_SECURITY\_DESCRIPTOR, \*PSAMPR\_SR\_SECURITY\_DESCRIPTOR;
75. typedef struct \_GROUP\_MEMBERSHIP {
76. unsigned long RelativeId;
77. unsigned long Attributes;
78. } GROUP\_MEMBERSHIP, \*PGROUP\_MEMBERSHIP;
79. typedef struct \_SAMPR\_GET\_GROUPS\_BUFFER {
80. unsigned long MembershipCount;
81. [size\_is(MembershipCount)] PGROUP\_MEMBERSHIP Groups;
82. } SAMPR\_GET\_GROUPS\_BUFFER, \*PSAMPR\_GET\_GROUPS\_BUFFER;
83. typedef struct \_SAMPR\_GET\_MEMBERS\_BUFFER {
84. unsigned long MemberCount;
85. [size\_is(MemberCount)] unsigned long\* Members;
86. [size\_is(MemberCount)] unsigned long\* Attributes;
87. } SAMPR\_GET\_MEMBERS\_BUFFER, \*PSAMPR\_GET\_MEMBERS\_BUFFER;
88. typedef struct \_SAMPR\_REVISION\_INFO\_V1 {
89. unsigned long Revision;
90. unsigned long SupportedFeatures;
91. } SAMPR\_REVISION\_INFO\_V1, \*PSAMPR\_REVISION\_INFO\_V1;
92. typedef [switch\_type(unsigned long)] union {
93. [case(1)] SAMPR\_REVISION\_INFO\_V1 V1;
94. }SAMPR\_REVISION\_INFO, \*PSAMPR\_REVISION\_INFO;
95. typedef struct \_USER\_DOMAIN\_PASSWORD\_INFORMATION {
96. unsigned short MinPasswordLength;
97. unsigned long PasswordProperties;
98. } USER\_DOMAIN\_PASSWORD\_INFORMATION,
99. \*PUSER\_DOMAIN\_PASSWORD\_INFORMATION;
100. typedef enum \_DOMAIN\_SERVER\_ENABLE\_STATE {
101. DomainServerEnabled = 1,
102. DomainServerDisabled
103. } DOMAIN\_SERVER\_ENABLE\_STATE, \*PDOMAIN\_SERVER\_ENABLE\_STATE;
104. typedef struct \_DOMAIN\_STATE\_INFORMATION {
105. DOMAIN\_SERVER\_ENABLE\_STATE DomainServerState;
106. } DOMAIN\_STATE\_INFORMATION, \*PDOMAIN\_STATE\_INFORMATION;
107. typedef enum \_DOMAIN\_SERVER\_ROLE {
108. DomainServerRoleBackup = 2,
109. DomainServerRolePrimary = 3
110. } DOMAIN\_SERVER\_ROLE, \*PDOMAIN\_SERVER\_ROLE;
111. typedef struct \_DOMAIN\_PASSWORD\_INFORMATION {
112. unsigned short MinPasswordLength;
113. unsigned short PasswordHistoryLength;
114. unsigned long PasswordProperties;
115. OLD\_LARGE\_INTEGER MaxPasswordAge;
116. OLD\_LARGE\_INTEGER MinPasswordAge;
117. } DOMAIN\_PASSWORD\_INFORMATION, \*PDOMAIN\_PASSWORD\_INFORMATION;
118. typedef struct \_DOMAIN\_LOGOFF\_INFORMATION {
119. OLD\_LARGE\_INTEGER ForceLogoff;
120. } DOMAIN\_LOGOFF\_INFORMATION, \*PDOMAIN\_LOGOFF\_INFORMATION;
121. typedef struct \_DOMAIN\_SERVER\_ROLE\_INFORMATION {
122. DOMAIN\_SERVER\_ROLE DomainServerRole;
123. } DOMAIN\_SERVER\_ROLE\_INFORMATION, \*PDOMAIN\_SERVER\_ROLE\_INFORMATION;
124. typedef struct \_DOMAIN\_MODIFIED\_INFORMATION {
125. OLD\_LARGE\_INTEGER DomainModifiedCount;
126. OLD\_LARGE\_INTEGER CreationTime;
127. } DOMAIN\_MODIFIED\_INFORMATION, \*PDOMAIN\_MODIFIED\_INFORMATION;
128. typedef struct \_DOMAIN\_MODIFIED\_INFORMATION2 {
129. OLD\_LARGE\_INTEGER DomainModifiedCount;
130. OLD\_LARGE\_INTEGER CreationTime;
131. OLD\_LARGE\_INTEGER ModifiedCountAtLastPromotion;
132. } DOMAIN\_MODIFIED\_INFORMATION2, \*PDOMAIN\_MODIFIED\_INFORMATION2;
133. #pragma pack(4)
134. typedef struct \_SAMPR\_DOMAIN\_GENERAL\_INFORMATION {
135. OLD\_LARGE\_INTEGER ForceLogoff;
136. RPC\_UNICODE\_STRING OemInformation;
137. RPC\_UNICODE\_STRING DomainName;
138. RPC\_UNICODE\_STRING ReplicaSourceNodeName;
139. OLD\_LARGE\_INTEGER DomainModifiedCount;
140. unsigned long DomainServerState;
141. unsigned long DomainServerRole;
142. unsigned char UasCompatibilityRequired;
143. unsigned long UserCount;
144. unsigned long GroupCount;
145. unsigned long AliasCount;
146. } SAMPR\_DOMAIN\_GENERAL\_INFORMATION,
147. \*PSAMPR\_DOMAIN\_GENERAL\_INFORMATION;
148. #pragma pack()
149. #pragma pack(4)
150. typedef struct \_SAMPR\_DOMAIN\_GENERAL\_INFORMATION2 {
151. SAMPR\_DOMAIN\_GENERAL\_INFORMATION I1;
152. LARGE\_INTEGER LockoutDuration;
153. LARGE\_INTEGER LockoutObservationWindow;
154. unsigned short LockoutThreshold;
155. } SAMPR\_DOMAIN\_GENERAL\_INFORMATION2,
156. \*PSAMPR\_DOMAIN\_GENERAL\_INFORMATION2;
157. #pragma pack()
158. typedef struct \_SAMPR\_DOMAIN\_OEM\_INFORMATION {
159. RPC\_UNICODE\_STRING OemInformation;
160. } SAMPR\_DOMAIN\_OEM\_INFORMATION, \*PSAMPR\_DOMAIN\_OEM\_INFORMATION;
161. typedef struct \_SAMPR\_DOMAIN\_NAME\_INFORMATION {
162. RPC\_UNICODE\_STRING DomainName;
163. } SAMPR\_DOMAIN\_NAME\_INFORMATION, \*PSAMPR\_DOMAIN\_NAME\_INFORMATION;
164. typedef struct SAMPR\_DOMAIN\_REPLICATION\_INFORMATION {
165. RPC\_UNICODE\_STRING ReplicaSourceNodeName;
166. } SAMPR\_DOMAIN\_REPLICATION\_INFORMATION,
167. \*PSAMPR\_DOMAIN\_REPLICATION\_INFORMATION;
168. typedef struct \_SAMPR\_DOMAIN\_LOCKOUT\_INFORMATION {
169. LARGE\_INTEGER LockoutDuration;
170. LARGE\_INTEGER LockoutObservationWindow;
171. unsigned short LockoutThreshold;
172. } SAMPR\_DOMAIN\_LOCKOUT\_INFORMATION,
173. \*PSAMPR\_DOMAIN\_LOCKOUT\_INFORMATION;
174. typedef enum \_DOMAIN\_INFORMATION\_CLASS {
175. DomainPasswordInformation = 1,
176. DomainGeneralInformation = 2,
177. DomainLogoffInformation = 3,
178. DomainOemInformation = 4,
179. DomainNameInformation = 5,
180. DomainReplicationInformation = 6,
181. DomainServerRoleInformation = 7,
182. DomainModifiedInformation = 8,
183. DomainStateInformation = 9,
184. DomainGeneralInformation2 = 11,
185. DomainLockoutInformation = 12,
186. DomainModifiedInformation2 = 13
187. } DOMAIN\_INFORMATION\_CLASS;
188. typedef [switch\_type(DOMAIN\_INFORMATION\_CLASS)] union
189. \_SAMPR\_DOMAIN\_INFO\_BUFFER {
190. [case(DomainPasswordInformation)]
191. DOMAIN\_PASSWORD\_INFORMATION Password;
192. [case(DomainGeneralInformation)]
193. SAMPR\_DOMAIN\_GENERAL\_INFORMATION General;
194. [case(DomainLogoffInformation)]
195. DOMAIN\_LOGOFF\_INFORMATION Logoff;
196. [case(DomainOemInformation)]
197. SAMPR\_DOMAIN\_OEM\_INFORMATION Oem;
198. [case(DomainNameInformation)]
199. SAMPR\_DOMAIN\_NAME\_INFORMATION Name;
200. [case(DomainServerRoleInformation)]
201. DOMAIN\_SERVER\_ROLE\_INFORMATION Role;
202. [case(DomainReplicationInformation)]
203. SAMPR\_DOMAIN\_REPLICATION\_INFORMATION Replication;
204. [case(DomainModifiedInformation)]
205. DOMAIN\_MODIFIED\_INFORMATION Modified;
206. [case(DomainStateInformation)]
207. DOMAIN\_STATE\_INFORMATION State;
208. [case(DomainGeneralInformation2)]
209. SAMPR\_DOMAIN\_GENERAL\_INFORMATION2 General2;
210. [case(DomainLockoutInformation)]
211. SAMPR\_DOMAIN\_LOCKOUT\_INFORMATION Lockout;
212. [case(DomainModifiedInformation2)]
213. DOMAIN\_MODIFIED\_INFORMATION2 Modified2;
214. } SAMPR\_DOMAIN\_INFO\_BUFFER, \*PSAMPR\_DOMAIN\_INFO\_BUFFER;
215. typedef enum \_DOMAIN\_DISPLAY\_INFORMATION {
216. DomainDisplayUser = 1,
217. DomainDisplayMachine,
218. DomainDisplayGroup,
219. DomainDisplayOemUser,
220. DomainDisplayOemGroup
221. } DOMAIN\_DISPLAY\_INFORMATION, \*PDOMAIN\_DISPLAY\_INFORMATION;
222. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_USER {
223. unsigned long Index;
224. unsigned long Rid;
225. unsigned long AccountControl;
226. RPC\_UNICODE\_STRING AccountName;
227. RPC\_UNICODE\_STRING AdminComment;
228. RPC\_UNICODE\_STRING FullName;
229. } SAMPR\_DOMAIN\_DISPLAY\_USER, \*PSAMPR\_DOMAIN\_DISPLAY\_USER;
230. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_MACHINE {
231. unsigned long Index;
232. unsigned long Rid;
233. unsigned long AccountControl;
234. RPC\_UNICODE\_STRING AccountName;
235. RPC\_UNICODE\_STRING AdminComment;
236. } SAMPR\_DOMAIN\_DISPLAY\_MACHINE, \*PSAMPR\_DOMAIN\_DISPLAY\_MACHINE;
237. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_GROUP {
238. unsigned long Index;
239. unsigned long Rid;
240. unsigned long Attributes;
241. RPC\_UNICODE\_STRING AccountName;
242. RPC\_UNICODE\_STRING AdminComment;
243. } SAMPR\_DOMAIN\_DISPLAY\_GROUP, \*PSAMPR\_DOMAIN\_DISPLAY\_GROUP;
244. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER {
245. unsigned long Index;
246. RPC\_STRING OemAccountName;
247. } SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER, \*PSAMPR\_DOMAIN\_DISPLAY\_OEM\_USER;
248. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP {
249. unsigned long Index;
250. RPC\_STRING OemAccountName;
251. } SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP, \*PSAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP;
252. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_USER\_BUFFER {
253. unsigned long EntriesRead;
254. [size\_is(EntriesRead)] PSAMPR\_DOMAIN\_DISPLAY\_USER Buffer;
255. } SAMPR\_DOMAIN\_DISPLAY\_USER\_BUFFER,
256. \*PSAMPR\_DOMAIN\_DISPLAY\_USER\_BUFFER;
257. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_MACHINE\_BUFFER {
258. unsigned long EntriesRead;
259. [size\_is(EntriesRead)] PSAMPR\_DOMAIN\_DISPLAY\_MACHINE Buffer;
260. } SAMPR\_DOMAIN\_DISPLAY\_MACHINE\_BUFFER,
261. \*PSAMPR\_DOMAIN\_DISPLAY\_MACHINE\_BUFFER;
262. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_GROUP\_BUFFER {
263. unsigned long EntriesRead;
264. [size\_is(EntriesRead)] PSAMPR\_DOMAIN\_DISPLAY\_GROUP Buffer;
265. } SAMPR\_DOMAIN\_DISPLAY\_GROUP\_BUFFER,
266. \*PSAMPR\_DOMAIN\_DISPLAY\_GROUP\_BUFFER;
267. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER\_BUFFER {
268. unsigned long EntriesRead;
269. [size\_is(EntriesRead)] PSAMPR\_DOMAIN\_DISPLAY\_OEM\_USER Buffer;
270. } SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER\_BUFFER,
271. \*PSAMPR\_DOMAIN\_DISPLAY\_OEM\_USER\_BUFFER;
272. typedef struct \_SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP\_BUFFER {
273. unsigned long EntriesRead;
274. [size\_is(EntriesRead)] PSAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP Buffer;
275. } SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP\_BUFFER,
276. \*PSAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP\_BUFFER;
277. typedef [switch\_type(DOMAIN\_DISPLAY\_INFORMATION)] union
278. \_SAMPR\_DISPLAY\_INFO\_BUFFER {
279. [case(DomainDisplayUser)]
280. SAMPR\_DOMAIN\_DISPLAY\_USER\_BUFFER UserInformation;
281. [case(DomainDisplayMachine)]
282. SAMPR\_DOMAIN\_DISPLAY\_MACHINE\_BUFFER MachineInformation;
283. [case(DomainDisplayGroup)]
284. SAMPR\_DOMAIN\_DISPLAY\_GROUP\_BUFFER GroupInformation;
285. [case(DomainDisplayOemUser)]
286. SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER\_BUFFER OemUserInformation;
287. [case(DomainDisplayOemGroup)]
288. SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP\_BUFFER OemGroupInformation;
289. } SAMPR\_DISPLAY\_INFO\_BUFFER, \*PSAMPR\_DISPLAY\_INFO\_BUFFER;
290. typedef struct \_GROUP\_ATTRIBUTE\_INFORMATION {
291. unsigned long Attributes;
292. } GROUP\_ATTRIBUTE\_INFORMATION, \*PGROUP\_ATTRIBUTE\_INFORMATION;
293. typedef struct \_SAMPR\_GROUP\_GENERAL\_INFORMATION {
294. RPC\_UNICODE\_STRING Name;
295. unsigned long Attributes;
296. unsigned long MemberCount;
297. RPC\_UNICODE\_STRING AdminComment;
298. } SAMPR\_GROUP\_GENERAL\_INFORMATION,
299. \*PSAMPR\_GROUP\_GENERAL\_INFORMATION;
300. typedef struct \_SAMPR\_GROUP\_NAME\_INFORMATION {
301. RPC\_UNICODE\_STRING Name;
302. } SAMPR\_GROUP\_NAME\_INFORMATION, \*PSAMPR\_GROUP\_NAME\_INFORMATION;
303. typedef struct \_SAMPR\_GROUP\_ADM\_COMMENT\_INFORMATION {
304. RPC\_UNICODE\_STRING AdminComment;
305. } SAMPR\_GROUP\_ADM\_COMMENT\_INFORMATION,
306. \*PSAMPR\_GROUP\_ADM\_COMMENT\_INFORMATION;
307. typedef enum \_GROUP\_INFORMATION\_CLASS {
308. GroupGeneralInformation = 1,
309. GroupNameInformation,
310. GroupAttributeInformation,
311. GroupAdminCommentInformation,
312. GroupReplicationInformation
313. } GROUP\_INFORMATION\_CLASS;
314. typedef [switch\_type(GROUP\_INFORMATION\_CLASS)] union
315. \_SAMPR\_GROUP\_INFO\_BUFFER {
316. [case(GroupGeneralInformation)]
317. SAMPR\_GROUP\_GENERAL\_INFORMATION General;
318. [case(GroupNameInformation)]
319. SAMPR\_GROUP\_NAME\_INFORMATION Name;
320. [case(GroupAttributeInformation)]
321. GROUP\_ATTRIBUTE\_INFORMATION Attribute;
322. [case(GroupAdminCommentInformation)]
323. SAMPR\_GROUP\_ADM\_COMMENT\_INFORMATION AdminComment;
324. [case(GroupReplicationInformation)]
325. SAMPR\_GROUP\_GENERAL\_INFORMATION DoNotUse;
326. } SAMPR\_GROUP\_INFO\_BUFFER, \*PSAMPR\_GROUP\_INFO\_BUFFER;
327. typedef struct \_SAMPR\_ALIAS\_GENERAL\_INFORMATION {
328. RPC\_UNICODE\_STRING Name;
329. unsigned long MemberCount;
330. RPC\_UNICODE\_STRING AdminComment;
331. } SAMPR\_ALIAS\_GENERAL\_INFORMATION,
332. \*PSAMPR\_ALIAS\_GENERAL\_INFORMATION;
333. typedef struct \_SAMPR\_ALIAS\_NAME\_INFORMATION {
334. RPC\_UNICODE\_STRING Name;
335. } SAMPR\_ALIAS\_NAME\_INFORMATION, \*PSAMPR\_ALIAS\_NAME\_INFORMATION;
336. typedef struct \_SAMPR\_ALIAS\_ADM\_COMMENT\_INFORMATION {
337. RPC\_UNICODE\_STRING AdminComment;
338. } SAMPR\_ALIAS\_ADM\_COMMENT\_INFORMATION,
339. \*PSAMPR\_ALIAS\_ADM\_COMMENT\_INFORMATION;
340. typedef enum \_ALIAS\_INFORMATION\_CLASS {
341. AliasGeneralInformation = 1,
342. AliasNameInformation,
343. AliasAdminCommentInformation
344. } ALIAS\_INFORMATION\_CLASS;
345. typedef [switch\_type(ALIAS\_INFORMATION\_CLASS)] union
346. \_SAMPR\_ALIAS\_INFO\_BUFFER {
347. [case(AliasGeneralInformation)]
348. SAMPR\_ALIAS\_GENERAL\_INFORMATION General;
349. [case(AliasNameInformation)]
350. SAMPR\_ALIAS\_NAME\_INFORMATION Name;
351. [case(AliasAdminCommentInformation)]
352. SAMPR\_ALIAS\_ADM\_COMMENT\_INFORMATION AdminComment;
353. } SAMPR\_ALIAS\_INFO\_BUFFER, \*PSAMPR\_ALIAS\_INFO\_BUFFER;
354. typedef struct \_SAMPR\_ENCRYPTED\_USER\_PASSWORD {
355. unsigned char Buffer[ (256 \* 2) + 4 ];
356. } SAMPR\_ENCRYPTED\_USER\_PASSWORD, \*PSAMPR\_ENCRYPTED\_USER\_PASSWORD;
357. typedef struct \_SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW {
358. unsigned char Buffer[ (256 \* 2) + 4 + 16];
359. } SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW,
360. \*PSAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW;
361. typedef struct \_USER\_PRIMARY\_GROUP\_INFORMATION {
362. unsigned long PrimaryGroupId;
363. } USER\_PRIMARY\_GROUP\_INFORMATION, \*PUSER\_PRIMARY\_GROUP\_INFORMATION;
364. typedef struct \_USER\_CONTROL\_INFORMATION {
365. unsigned long UserAccountControl;
366. } USER\_CONTROL\_INFORMATION, \*PUSER\_CONTROL\_INFORMATION;
367. typedef struct \_USER\_EXPIRES\_INFORMATION {
368. OLD\_LARGE\_INTEGER AccountExpires;
369. } USER\_EXPIRES\_INFORMATION, \*PUSER\_EXPIRES\_INFORMATION;
370. typedef struct \_SAMPR\_LOGON\_HOURS {
371. unsigned short UnitsPerWeek;
372. [size\_is(1260), length\_is((UnitsPerWeek+7)/8)]
373. unsigned char\* LogonHours;
374. } SAMPR\_LOGON\_HOURS, \*PSAMPR\_LOGON\_HOURS;
375. typedef struct \_SAMPR\_USER\_ALL\_INFORMATION {
376. OLD\_LARGE\_INTEGER LastLogon;
377. OLD\_LARGE\_INTEGER LastLogoff;
378. OLD\_LARGE\_INTEGER PasswordLastSet;
379. OLD\_LARGE\_INTEGER AccountExpires;
380. OLD\_LARGE\_INTEGER PasswordCanChange;
381. OLD\_LARGE\_INTEGER PasswordMustChange;
382. RPC\_UNICODE\_STRING UserName;
383. RPC\_UNICODE\_STRING FullName;
384. RPC\_UNICODE\_STRING HomeDirectory;
385. RPC\_UNICODE\_STRING HomeDirectoryDrive;
386. RPC\_UNICODE\_STRING ScriptPath;
387. RPC\_UNICODE\_STRING ProfilePath;
388. RPC\_UNICODE\_STRING AdminComment;
389. RPC\_UNICODE\_STRING WorkStations;
390. RPC\_UNICODE\_STRING UserComment;
391. RPC\_UNICODE\_STRING Parameters;
392. RPC\_SHORT\_BLOB LmOwfPassword;
393. RPC\_SHORT\_BLOB NtOwfPassword;
394. RPC\_UNICODE\_STRING PrivateData;
395. SAMPR\_SR\_SECURITY\_DESCRIPTOR SecurityDescriptor;
396. unsigned long UserId;
397. unsigned long PrimaryGroupId;
398. unsigned long UserAccountControl;
399. unsigned long WhichFields;
400. SAMPR\_LOGON\_HOURS LogonHours;
401. unsigned short BadPasswordCount;
402. unsigned short LogonCount;
403. unsigned short CountryCode;
404. unsigned short CodePage;
405. unsigned char LmPasswordPresent;
406. unsigned char NtPasswordPresent;
407. unsigned char PasswordExpired;
408. unsigned char PrivateDataSensitive;
409. } SAMPR\_USER\_ALL\_INFORMATION, \*PSAMPR\_USER\_ALL\_INFORMATION;
410. typedef struct \_SAMPR\_USER\_GENERAL\_INFORMATION {
411. RPC\_UNICODE\_STRING UserName;
412. RPC\_UNICODE\_STRING FullName;
413. unsigned long PrimaryGroupId;
414. RPC\_UNICODE\_STRING AdminComment;
415. RPC\_UNICODE\_STRING UserComment;
416. } SAMPR\_USER\_GENERAL\_INFORMATION, \*PSAMPR\_USER\_GENERAL\_INFORMATION;
417. typedef struct \_SAMPR\_USER\_PREFERENCES\_INFORMATION {
418. RPC\_UNICODE\_STRING UserComment;
419. RPC\_UNICODE\_STRING Reserved1;
420. unsigned short CountryCode;
421. unsigned short CodePage;
422. } SAMPR\_USER\_PREFERENCES\_INFORMATION,
423. \*PSAMPR\_USER\_PREFERENCES\_INFORMATION;
424. typedef struct \_SAMPR\_USER\_PARAMETERS\_INFORMATION {
425. RPC\_UNICODE\_STRING Parameters;
426. } SAMPR\_USER\_PARAMETERS\_INFORMATION,
427. \*PSAMPR\_USER\_PARAMETERS\_INFORMATION;
428. typedef struct \_SAMPR\_USER\_LOGON\_INFORMATION {
429. RPC\_UNICODE\_STRING UserName;
430. RPC\_UNICODE\_STRING FullName;
431. unsigned long UserId;
432. unsigned long PrimaryGroupId;
433. RPC\_UNICODE\_STRING HomeDirectory;
434. RPC\_UNICODE\_STRING HomeDirectoryDrive;
435. RPC\_UNICODE\_STRING ScriptPath;
436. RPC\_UNICODE\_STRING ProfilePath;
437. RPC\_UNICODE\_STRING WorkStations;
438. OLD\_LARGE\_INTEGER LastLogon;
439. OLD\_LARGE\_INTEGER LastLogoff;
440. OLD\_LARGE\_INTEGER PasswordLastSet;
441. OLD\_LARGE\_INTEGER PasswordCanChange;
442. OLD\_LARGE\_INTEGER PasswordMustChange;
443. SAMPR\_LOGON\_HOURS LogonHours;
444. unsigned short BadPasswordCount;
445. unsigned short LogonCount;
446. unsigned long UserAccountControl;
447. } SAMPR\_USER\_LOGON\_INFORMATION, \*PSAMPR\_USER\_LOGON\_INFORMATION;
448. typedef struct \_SAMPR\_USER\_ACCOUNT\_INFORMATION {
449. RPC\_UNICODE\_STRING UserName;
450. RPC\_UNICODE\_STRING FullName;
451. unsigned long UserId;
452. unsigned long PrimaryGroupId;
453. RPC\_UNICODE\_STRING HomeDirectory;
454. RPC\_UNICODE\_STRING HomeDirectoryDrive;
455. RPC\_UNICODE\_STRING ScriptPath;
456. RPC\_UNICODE\_STRING ProfilePath;
457. RPC\_UNICODE\_STRING AdminComment;
458. RPC\_UNICODE\_STRING WorkStations;
459. OLD\_LARGE\_INTEGER LastLogon;
460. OLD\_LARGE\_INTEGER LastLogoff;
461. SAMPR\_LOGON\_HOURS LogonHours;
462. unsigned short BadPasswordCount;
463. unsigned short LogonCount;
464. OLD\_LARGE\_INTEGER PasswordLastSet;
465. OLD\_LARGE\_INTEGER AccountExpires;
466. unsigned long UserAccountControl;
467. } SAMPR\_USER\_ACCOUNT\_INFORMATION, \*PSAMPR\_USER\_ACCOUNT\_INFORMATION;
468. typedef struct \_SAMPR\_USER\_A\_NAME\_INFORMATION {
469. RPC\_UNICODE\_STRING UserName;
470. } SAMPR\_USER\_A\_NAME\_INFORMATION, \*PSAMPR\_USER\_A\_NAME\_INFORMATION;
471. typedef struct \_SAMPR\_USER\_F\_NAME\_INFORMATION {
472. RPC\_UNICODE\_STRING FullName;
473. } SAMPR\_USER\_F\_NAME\_INFORMATION, \*PSAMPR\_USER\_F\_NAME\_INFORMATION;
474. typedef struct \_SAMPR\_USER\_NAME\_INFORMATION {
475. RPC\_UNICODE\_STRING UserName;
476. RPC\_UNICODE\_STRING FullName;
477. } SAMPR\_USER\_NAME\_INFORMATION, \*PSAMPR\_USER\_NAME\_INFORMATION;
478. typedef struct \_SAMPR\_USER\_HOME\_INFORMATION {
479. RPC\_UNICODE\_STRING HomeDirectory;
480. RPC\_UNICODE\_STRING HomeDirectoryDrive;
481. } SAMPR\_USER\_HOME\_INFORMATION, \*PSAMPR\_USER\_HOME\_INFORMATION;
482. typedef struct \_SAMPR\_USER\_SCRIPT\_INFORMATION {
483. RPC\_UNICODE\_STRING ScriptPath;
484. } SAMPR\_USER\_SCRIPT\_INFORMATION, \*PSAMPR\_USER\_SCRIPT\_INFORMATION;
485. typedef struct \_SAMPR\_USER\_PROFILE\_INFORMATION {
486. RPC\_UNICODE\_STRING ProfilePath;
487. } SAMPR\_USER\_PROFILE\_INFORMATION, \*PSAMPR\_USER\_PROFILE\_INFORMATION;
488. typedef struct \_SAMPR\_USER\_ADMIN\_COMMENT\_INFORMATION {
489. RPC\_UNICODE\_STRING AdminComment;
490. } SAMPR\_USER\_ADMIN\_COMMENT\_INFORMATION,
491. \*PSAMPR\_USER\_ADMIN\_COMMENT\_INFORMATION;
492. typedef struct \_SAMPR\_USER\_WORKSTATIONS\_INFORMATION {
493. RPC\_UNICODE\_STRING WorkStations;
494. } SAMPR\_USER\_WORKSTATIONS\_INFORMATION,
495. \*PSAMPR\_USER\_WORKSTATIONS\_INFORMATION;
496. typedef struct \_SAMPR\_USER\_LOGON\_HOURS\_INFORMATION {
497. SAMPR\_LOGON\_HOURS LogonHours;
498. } SAMPR\_USER\_LOGON\_HOURS\_INFORMATION,
499. \*PSAMPR\_USER\_LOGON\_HOURS\_INFORMATION;
500. typedef struct \_SAMPR\_USER\_INTERNAL1\_INFORMATION {
501. ENCRYPTED\_NT\_OWF\_PASSWORD EncryptedNtOwfPassword;
502. ENCRYPTED\_LM\_OWF\_PASSWORD EncryptedLmOwfPassword;
503. unsigned char NtPasswordPresent;
504. unsigned char LmPasswordPresent;
505. unsigned char PasswordExpired;
506. } SAMPR\_USER\_INTERNAL1\_INFORMATION,
507. \*PSAMPR\_USER\_INTERNAL1\_INFORMATION;
508. typedef struct \_SAMPR\_USER\_INTERNAL4\_INFORMATION {
509. SAMPR\_USER\_ALL\_INFORMATION I1;
510. SAMPR\_ENCRYPTED\_USER\_PASSWORD UserPassword;
511. } SAMPR\_USER\_INTERNAL4\_INFORMATION,
512. \*PSAMPR\_USER\_INTERNAL4\_INFORMATION;
513. typedef struct \_SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW {
514. SAMPR\_USER\_ALL\_INFORMATION I1;
515. SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW UserPassword;
516. } SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW,
517. \*PSAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW;
518. typedef struct \_SAMPR\_USER\_INTERNAL5\_INFORMATION {
519. SAMPR\_ENCRYPTED\_USER\_PASSWORD UserPassword;
520. unsigned char PasswordExpired;
521. } SAMPR\_USER\_INTERNAL5\_INFORMATION,
522. \*PSAMPR\_USER\_INTERNAL5\_INFORMATION;
523. typedef struct \_SAMPR\_USER\_INTERNAL5\_INFORMATION\_NEW {
524. SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW UserPassword;
525. unsigned char PasswordExpired;
526. } SAMPR\_USER\_INTERNAL5\_INFORMATION\_NEW,
527. \*PSAMPR\_USER\_INTERNAL5\_INFORMATION\_NEW;
528. typedef struct \_SAMPR\_ENCRYPTED\_PASSWORD\_AES {
529. UCHAR AuthData[64];
530. UCHAR Salt[16];
531. ULONG cbCipher;
532. [size\_is(cbCipher)] PUCHAR Cipher;
533. ULONGLONG PBKDF2Iterations;
534. } SAMPR\_ENCRYPTED\_PASSWORD\_AES, \*PSAMPR\_ENCRYPTED\_PASSWORD\_AES;
535. typedef struct \_SAMPR\_USER\_INTERNAL7\_INFORMATION {
536. SAMPR\_ENCRYPTED\_PASSWORD\_AES UserPassword;
537. BOOLEAN PasswordExpired;
538. } SAMPR\_USER\_INTERNAL7\_INFORMATION, \*PSAMPR\_USER\_INTERNAL7\_INFORMATION;
539. typedef struct \_SAMPR\_USER\_INTERNAL8\_INFORMATION {
540. SAMPR\_USER\_ALL\_INFORMATION I1;
541. SAMPR\_ENCRYPTED\_PASSWORD\_AES UserPassword;
542. } SAMPR\_USER\_INTERNAL8\_INFORMATION, \*PSAMPR\_USER\_INTERNAL8\_INFORMATION;
543. typedef enum \_USER\_INFORMATION\_CLASS {
544. UserGeneralInformation = 1,
545. UserPreferencesInformation = 2,
546. UserLogonInformation = 3,
547. UserLogonHoursInformation = 4,
548. UserAccountInformation = 5,
549. UserNameInformation = 6,
550. UserAccountNameInformation = 7,
551. UserFullNameInformation = 8,
552. UserPrimaryGroupInformation = 9,
553. UserHomeInformation = 10,
554. UserScriptInformation = 11,
555. UserProfileInformation = 12,
556. UserAdminCommentInformation = 13,
557. UserWorkStationsInformation = 14,
558. UserControlInformation = 16,
559. UserExpiresInformation = 17,
560. UserInternal1Information = 18,
561. UserParametersInformation = 20,
562. UserAllInformation = 21,
563. UserInternal4Information = 23,
564. UserInternal5Information = 24,
565. UserInternal4InformationNew = 25,
566. UserInternal5InformationNew = 26,
567. UserInternal7Information = 31,
568. UserInternal8Information = 32
569. } USER\_INFORMATION\_CLASS, \*PUSER\_INFORMATION\_CLASS;
570. typedef [switch\_type(USER\_INFORMATION\_CLASS)] union
571. \_SAMPR\_USER\_INFO\_BUFFER {
572. [case(UserGeneralInformation)]
573. SAMPR\_USER\_GENERAL\_INFORMATION General;
574. [case(UserPreferencesInformation)]
575. SAMPR\_USER\_PREFERENCES\_INFORMATION Preferences;
576. [case(UserLogonInformation)]
577. SAMPR\_USER\_LOGON\_INFORMATION Logon;
578. [case(UserLogonHoursInformation)]
579. SAMPR\_USER\_LOGON\_HOURS\_INFORMATION LogonHours;
580. [case(UserAccountInformation)]
581. SAMPR\_USER\_ACCOUNT\_INFORMATION Account;
582. [case(UserNameInformation)]
583. SAMPR\_USER\_NAME\_INFORMATION Name;
584. [case(UserAccountNameInformation)]
585. SAMPR\_USER\_A\_NAME\_INFORMATION AccountName;
586. [case(UserFullNameInformation)]
587. SAMPR\_USER\_F\_NAME\_INFORMATION FullName;
588. [case(UserPrimaryGroupInformation)]
589. USER\_PRIMARY\_GROUP\_INFORMATION PrimaryGroup;
590. [case(UserHomeInformation)]
591. SAMPR\_USER\_HOME\_INFORMATION Home;
592. [case(UserScriptInformation)]
593. SAMPR\_USER\_SCRIPT\_INFORMATION Script;
594. [case(UserProfileInformation)]
595. SAMPR\_USER\_PROFILE\_INFORMATION Profile;
596. [case(UserAdminCommentInformation)]
597. SAMPR\_USER\_ADMIN\_COMMENT\_INFORMATION AdminComment;
598. [case(UserWorkStationsInformation)]
599. SAMPR\_USER\_WORKSTATIONS\_INFORMATION WorkStations;
600. [case(UserControlInformation)]
601. USER\_CONTROL\_INFORMATION Control;
602. [case(UserExpiresInformation)]
603. USER\_EXPIRES\_INFORMATION Expires;
604. [case(UserInternal1Information)]
605. SAMPR\_USER\_INTERNAL1\_INFORMATION Internal1;
606. [case(UserParametersInformation)]
607. SAMPR\_USER\_PARAMETERS\_INFORMATION Parameters;
608. [case(UserAllInformation)]
609. SAMPR\_USER\_ALL\_INFORMATION All;
610. [case(UserInternal4Information)]
611. SAMPR\_USER\_INTERNAL4\_INFORMATION Internal4;
612. [case(UserInternal5Information)]
613. SAMPR\_USER\_INTERNAL5\_INFORMATION Internal5;
614. [case(UserInternal4InformationNew)]
615. SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW Internal4New;
616. [case(UserInternal5InformationNew)]
617. SAMPR\_USER\_INTERNAL5\_INFORMATION\_NEW Internal5New;
618. [case(UserInternal7Information)]
619. SAMPR\_USER\_INTERNAL7\_INFORMATION Internal7;
620. [case(UserInternal8Information)]
621. SAMPR\_USER\_INTERNAL8\_INFORMATION Internal8;
622. } SAMPR\_USER\_INFO\_BUFFER, \*PSAMPR\_USER\_INFO\_BUFFER;
623. typedef enum \_PASSWORD\_POLICY\_VALIDATION\_TYPE{
624. SamValidateAuthentication = 1,
625. SamValidatePasswordChange,
626. SamValidatePasswordReset
627. } PASSWORD\_POLICY\_VALIDATION\_TYPE;
628. typedef struct \_SAM\_VALIDATE\_PASSWORD\_HASH{
629. unsigned long Length;
630. [unique,size\_is(Length)]
631. unsigned char\* Hash;
632. } SAM\_VALIDATE\_PASSWORD\_HASH, \*PSAM\_VALIDATE\_PASSWORD\_HASH;
633. typedef struct \_SAM\_VALIDATE\_PERSISTED\_FIELDS{
634. unsigned long PresentFields;
635. LARGE\_INTEGER PasswordLastSet;
636. LARGE\_INTEGER BadPasswordTime;
637. LARGE\_INTEGER LockoutTime;
638. unsigned long BadPasswordCount;
639. unsigned long PasswordHistoryLength;
640. [unique,size\_is(PasswordHistoryLength)]
641. PSAM\_VALIDATE\_PASSWORD\_HASH PasswordHistory;
642. } SAM\_VALIDATE\_PERSISTED\_FIELDS, \*PSAM\_VALIDATE\_PERSISTED\_FIELDS;
643. typedef enum \_SAM\_VALIDATE\_VALIDATION\_STATUS{
644. SamValidateSuccess = 0,
645. SamValidatePasswordMustChange,
646. SamValidateAccountLockedOut,
647. SamValidatePasswordExpired,
648. SamValidatePasswordIncorrect,
649. SamValidatePasswordIsInHistory,
650. SamValidatePasswordTooShort,
651. SamValidatePasswordTooLong,
652. SamValidatePasswordNotComplexEnough,
653. SamValidatePasswordTooRecent,
654. SamValidatePasswordFilterError
655. } SAM\_VALIDATE\_VALIDATION\_STATUS, \*PSAM\_VALIDATE\_VALIDATION\_STATUS;
656. typedef struct \_SAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG{
657. SAM\_VALIDATE\_PERSISTED\_FIELDS ChangedPersistedFields;
658. SAM\_VALIDATE\_VALIDATION\_STATUS ValidationStatus;
659. } SAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG,
660. \*PSAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG;
661. typedef struct \_SAM\_VALIDATE\_AUTHENTICATION\_INPUT\_ARG{
662. SAM\_VALIDATE\_PERSISTED\_FIELDS InputPersistedFields;
663. unsigned char PasswordMatched;
664. } SAM\_VALIDATE\_AUTHENTICATION\_INPUT\_ARG,
665. \*PSAM\_VALIDATE\_AUTHENTICATION\_INPUT\_ARG;
666. typedef struct \_SAM\_VALIDATE\_PASSWORD\_CHANGE\_INPUT\_ARG{
667. SAM\_VALIDATE\_PERSISTED\_FIELDS InputPersistedFields;
668. RPC\_UNICODE\_STRING ClearPassword;
669. RPC\_UNICODE\_STRING UserAccountName;
670. SAM\_VALIDATE\_PASSWORD\_HASH HashedPassword;
671. unsigned char PasswordMatch;
672. } SAM\_VALIDATE\_PASSWORD\_CHANGE\_INPUT\_ARG,
673. \*PSAM\_VALIDATE\_PASSWORD\_CHANGE\_INPUT\_ARG;
674. typedef struct \_SAM\_VALIDATE\_PASSWORD\_RESET\_INPUT\_ARG{
675. SAM\_VALIDATE\_PERSISTED\_FIELDS InputPersistedFields;
676. RPC\_UNICODE\_STRING ClearPassword;
677. RPC\_UNICODE\_STRING UserAccountName;
678. SAM\_VALIDATE\_PASSWORD\_HASH HashedPassword;
679. unsigned char PasswordMustChangeAtNextLogon;
680. unsigned char ClearLockout;
681. } SAM\_VALIDATE\_PASSWORD\_RESET\_INPUT\_ARG,
682. \*PSAM\_VALIDATE\_PASSWORD\_RESET\_INPUT\_ARG;
683. typedef
684. [switch\_type(PASSWORD\_POLICY\_VALIDATION\_TYPE)]
685. union \_SAM\_VALIDATE\_INPUT\_ARG{
686. [case(SamValidateAuthentication)]
687. SAM\_VALIDATE\_AUTHENTICATION\_INPUT\_ARG
688. ValidateAuthenticationInput;
689. [case(SamValidatePasswordChange)]
690. SAM\_VALIDATE\_PASSWORD\_CHANGE\_INPUT\_ARG
691. ValidatePasswordChangeInput;
692. [case(SamValidatePasswordReset)]
693. SAM\_VALIDATE\_PASSWORD\_RESET\_INPUT\_ARG
694. ValidatePasswordResetInput;
695. } SAM\_VALIDATE\_INPUT\_ARG, \*PSAM\_VALIDATE\_INPUT\_ARG;
696. typedef
697. [switch\_type(PASSWORD\_POLICY\_VALIDATION\_TYPE)]
698. union \_SAM\_VALIDATE\_OUTPUT\_ARG{
699. [case(SamValidateAuthentication)]
700. SAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG
701. ValidateAuthenticationOutput;
702. [case(SamValidatePasswordChange)]
703. SAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG
704. ValidatePasswordChangeOutput;
705. [case(SamValidatePasswordReset)]
706. SAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG
707. ValidatePasswordResetOutput;
708. } SAM\_VALIDATE\_OUTPUT\_ARG, \*PSAM\_VALIDATE\_OUTPUT\_ARG;
709. // opnum 0
710. long SamrConnect(
711. [in, unique] PSAMPR\_SERVER\_NAME ServerName,
712. [out] SAMPR\_HANDLE \* ServerHandle,
713. [in] unsigned long DesiredAccess
714. );
715. // opnum 1
716. long
717. SamrCloseHandle(
718. [in,out] SAMPR\_HANDLE \* SamHandle
719. );
720. // opnum 2
721. long
722. SamrSetSecurityObject(
723. [in] SAMPR\_HANDLE ObjectHandle,
724. [in] SECURITY\_INFORMATION SecurityInformation,
725. [in] PSAMPR\_SR\_SECURITY\_DESCRIPTOR SecurityDescriptor
726. );
727. // opnum 3
728. long
729. SamrQuerySecurityObject(
730. [in] SAMPR\_HANDLE ObjectHandle,
731. [in] SECURITY\_INFORMATION SecurityInformation,
732. [out] PSAMPR\_SR\_SECURITY\_DESCRIPTOR \* SecurityDescriptor
733. );
734. // opnum 4
735. void Opnum4NotUsedOnWire(void);
736. // opnum 5
737. long
738. SamrLookupDomainInSamServer(
739. [in] SAMPR\_HANDLE ServerHandle,
740. [in] PRPC\_UNICODE\_STRING Name,
741. [out] PRPC\_SID \* DomainId
742. );
743. // opnum 6
744. long
745. SamrEnumerateDomainsInSamServer(
746. [in] SAMPR\_HANDLE ServerHandle,
747. [in,out] unsigned long \* EnumerationContext,
748. [out] PSAMPR\_ENUMERATION\_BUFFER \* Buffer,
749. [in] unsigned long PreferedMaximumLength,
750. [out] unsigned long \* CountReturned
751. );
752. // opnum 7
753. long
754. SamrOpenDomain(
755. [in] SAMPR\_HANDLE ServerHandle,
756. [in] unsigned long DesiredAccess,
757. [in] PRPC\_SID DomainId,
758. [out] SAMPR\_HANDLE \* DomainHandle
759. );
760. // opnum 8
761. long
762. SamrQueryInformationDomain(
763. [in] SAMPR\_HANDLE DomainHandle,
764. [in] DOMAIN\_INFORMATION\_CLASS DomainInformationClass,
765. [out, switch\_is(DomainInformationClass)]
766. PSAMPR\_DOMAIN\_INFO\_BUFFER \* Buffer
767. );
768. // opnum 9
769. long
770. SamrSetInformationDomain(
771. [in] SAMPR\_HANDLE DomainHandle,
772. [in] DOMAIN\_INFORMATION\_CLASS DomainInformationClass,
773. [in, switch\_is(DomainInformationClass)]
774. PSAMPR\_DOMAIN\_INFO\_BUFFER DomainInformation
775. );
776. // opnum 10
777. long
778. SamrCreateGroupInDomain(
779. [in] SAMPR\_HANDLE DomainHandle,
780. [in] PRPC\_UNICODE\_STRING Name,
781. [in] unsigned long DesiredAccess,
782. [out] SAMPR\_HANDLE \* GroupHandle,
783. [out] unsigned long \* RelativeId
784. );
785. // opnum 11
786. long
787. SamrEnumerateGroupsInDomain(
788. [in] SAMPR\_HANDLE DomainHandle,
789. [in,out] unsigned long \* EnumerationContext,
790. [out] PSAMPR\_ENUMERATION\_BUFFER \* Buffer,
791. [in] unsigned long PreferedMaximumLength,
792. [out] unsigned long \* CountReturned
793. );
794. // opnum 12
795. long
796. SamrCreateUserInDomain(
797. [in] SAMPR\_HANDLE DomainHandle,
798. [in] PRPC\_UNICODE\_STRING Name,
799. [in] unsigned long DesiredAccess,
800. [out] SAMPR\_HANDLE \* UserHandle,
801. [out] unsigned long \* RelativeId
802. );
803. // opnum 13
804. long
805. SamrEnumerateUsersInDomain(
806. [in] SAMPR\_HANDLE DomainHandle,
807. [in,out] unsigned long \* EnumerationContext,
808. [in] unsigned long UserAccountControl,
809. [out] PSAMPR\_ENUMERATION\_BUFFER \* Buffer,
810. [in] unsigned long PreferedMaximumLength,
811. [out] unsigned long \* CountReturned
812. );
813. // opnum 14
814. long
815. SamrCreateAliasInDomain(
816. [in] SAMPR\_HANDLE DomainHandle,
817. [in] PRPC\_UNICODE\_STRING AccountName,
818. [in] unsigned long DesiredAccess,
819. [out] SAMPR\_HANDLE \* AliasHandle,
820. [out] unsigned long \* RelativeId
821. );
822. // opnum 15
823. long
824. SamrEnumerateAliasesInDomain(
825. [in] SAMPR\_HANDLE DomainHandle,
826. [in,out] unsigned long \* EnumerationContext,
827. [out] PSAMPR\_ENUMERATION\_BUFFER \* Buffer,
828. [in] unsigned long PreferedMaximumLength,
829. [out] unsigned long \* CountReturned
830. );
831. // opnum 16
832. long
833. SamrGetAliasMembership(
834. [in] SAMPR\_HANDLE DomainHandle,
835. [in] PSAMPR\_PSID\_ARRAY SidArray,
836. [out] PSAMPR\_ULONG\_ARRAY Membership
837. );
838. // opnum 17
839. long
840. SamrLookupNamesInDomain(
841. [in] SAMPR\_HANDLE DomainHandle,
842. [in, range(0, 1000)] unsigned long Count,
843. [in, size\_is(1000), length\_is(Count)] RPC\_UNICODE\_STRING Names[\*],
844. [out] PSAMPR\_ULONG\_ARRAY RelativeIds,
845. [out] PSAMPR\_ULONG\_ARRAY Use
846. );
847. // opnum 18
848. long
849. SamrLookupIdsInDomain(
850. [in] SAMPR\_HANDLE DomainHandle,
851. [in, range(0, 1000)] unsigned long Count,
852. [in, size\_is(1000), length\_is(Count)] unsigned long \*RelativeIds,
853. [out] PSAMPR\_RETURNED\_USTRING\_ARRAY Names,
854. [out] PSAMPR\_ULONG\_ARRAY Use
855. );
856. // opnum 19
857. long
858. SamrOpenGroup(
859. [in] SAMPR\_HANDLE DomainHandle,
860. [in] unsigned long DesiredAccess,
861. [in] unsigned long GroupId,
862. [out] SAMPR\_HANDLE \* GroupHandle
863. );
864. // opnum 20
865. long
866. SamrQueryInformationGroup(
867. [in] SAMPR\_HANDLE GroupHandle,
868. [in] GROUP\_INFORMATION\_CLASS GroupInformationClass,
869. [out, switch\_is(GroupInformationClass)]
870. PSAMPR\_GROUP\_INFO\_BUFFER \* Buffer
871. );
872. // opnum 21
873. long
874. SamrSetInformationGroup(
875. [in] SAMPR\_HANDLE GroupHandle,
876. [in] GROUP\_INFORMATION\_CLASS GroupInformationClass,
877. [in, switch\_is(GroupInformationClass)]
878. PSAMPR\_GROUP\_INFO\_BUFFER Buffer
879. );
880. // opnum 22
881. long
882. SamrAddMemberToGroup(
883. [in] SAMPR\_HANDLE GroupHandle,
884. [in] unsigned long MemberId,
885. [in] unsigned long Attributes
886. );
887. // opnum 23
888. long
889. SamrDeleteGroup(
890. [in,out] SAMPR\_HANDLE \* GroupHandle
891. );
892. // opnum 24
893. long
894. SamrRemoveMemberFromGroup(
895. [in] SAMPR\_HANDLE GroupHandle,
896. [in] unsigned long MemberId
897. );
898. // opnum 25
899. long
900. SamrGetMembersInGroup(
901. [in] SAMPR\_HANDLE GroupHandle,
902. [out] PSAMPR\_GET\_MEMBERS\_BUFFER \* Members
903. );
904. // opnum 26
905. long
906. SamrSetMemberAttributesOfGroup(
907. [in] SAMPR\_HANDLE GroupHandle,
908. [in] unsigned long MemberId,
909. [in] unsigned long Attributes
910. );
911. // opnum 27
912. long
913. SamrOpenAlias(
914. [in] SAMPR\_HANDLE DomainHandle,
915. [in] unsigned long DesiredAccess,
916. [in] unsigned long AliasId,
917. [out] SAMPR\_HANDLE \* AliasHandle
918. );
919. // opnum 28
920. long
921. SamrQueryInformationAlias(
922. [in] SAMPR\_HANDLE AliasHandle,
923. [in] ALIAS\_INFORMATION\_CLASS AliasInformationClass,
924. [out, switch\_is(AliasInformationClass)]
925. PSAMPR\_ALIAS\_INFO\_BUFFER \* Buffer
926. );
927. // opnum 29
928. long
929. SamrSetInformationAlias(
930. [in] SAMPR\_HANDLE AliasHandle,
931. [in] ALIAS\_INFORMATION\_CLASS AliasInformationClass,
932. [in, switch\_is(AliasInformationClass)]
933. PSAMPR\_ALIAS\_INFO\_BUFFER Buffer
934. );
935. // opnum 30
936. long
937. SamrDeleteAlias(
938. [in, out] SAMPR\_HANDLE \* AliasHandle
939. );
940. // opnum 31
941. long
942. SamrAddMemberToAlias(
943. [in] SAMPR\_HANDLE AliasHandle,
944. [in] PRPC\_SID MemberId
945. );
946. // opnum 32
947. long
948. SamrRemoveMemberFromAlias(
949. [in] SAMPR\_HANDLE AliasHandle,
950. [in] PRPC\_SID MemberId
951. );
952. // opnum 33
953. long
954. SamrGetMembersInAlias(
955. [in] SAMPR\_HANDLE AliasHandle,
956. [out] PSAMPR\_PSID\_ARRAY\_OUT Members
957. );
958. // opnum 34
959. long
960. SamrOpenUser(
961. [in] SAMPR\_HANDLE DomainHandle,
962. [in] unsigned long DesiredAccess,
963. [in] unsigned long UserId,
964. [out] SAMPR\_HANDLE \* UserHandle
965. );
966. // opnum 35
967. long
968. SamrDeleteUser(
969. [in,out] SAMPR\_HANDLE \* UserHandle
970. );
971. // opnum 36
972. long
973. SamrQueryInformationUser(
974. [in] SAMPR\_HANDLE UserHandle,
975. [in] USER\_INFORMATION\_CLASS UserInformationClass,
976. [out, switch\_is(UserInformationClass)]
977. PSAMPR\_USER\_INFO\_BUFFER \* Buffer
978. );
979. // opnum 37
980. long
981. SamrSetInformationUser(
982. [in] SAMPR\_HANDLE UserHandle,
983. [in] USER\_INFORMATION\_CLASS UserInformationClass,
984. [in, switch\_is(UserInformationClass)]
985. PSAMPR\_USER\_INFO\_BUFFER Buffer
986. );
987. // opnum 38
988. long
989. SamrChangePasswordUser(
990. [in] SAMPR\_HANDLE UserHandle,
991. [in] unsigned char LmPresent,
992. [in, unique] PENCRYPTED\_LM\_OWF\_PASSWORD OldLmEncryptedWithNewLm,
993. [in, unique] PENCRYPTED\_LM\_OWF\_PASSWORD NewLmEncryptedWithOldLm,
994. [in] unsigned char NtPresent,
995. [in, unique] PENCRYPTED\_NT\_OWF\_PASSWORD OldNtEncryptedWithNewNt,
996. [in, unique] PENCRYPTED\_NT\_OWF\_PASSWORD NewNtEncryptedWithOldNt,
997. [in] unsigned char NtCrossEncryptionPresent,
998. [in, unique] PENCRYPTED\_NT\_OWF\_PASSWORD NewNtEncryptedWithNewLm,
999. [in] unsigned char LmCrossEncryptionPresent,
1000. [in, unique] PENCRYPTED\_LM\_OWF\_PASSWORD NewLmEncryptedWithNewNt
1001. );
1002. // opnum 39
1003. long
1004. SamrGetGroupsForUser(
1005. [in] SAMPR\_HANDLE UserHandle,
1006. [out] PSAMPR\_GET\_GROUPS\_BUFFER \* Groups
1007. );
1008. // opnum 40
1009. long
1010. SamrQueryDisplayInformation (
1011. [in] SAMPR\_HANDLE DomainHandle,
1012. [in] DOMAIN\_DISPLAY\_INFORMATION DisplayInformationClass,
1013. [in] unsigned long Index,
1014. [in] unsigned long EntryCount,
1015. [in] unsigned long PreferredMaximumLength,
1016. [out] unsigned long \* TotalAvailable,
1017. [out] unsigned long \* TotalReturned,
1018. [out, switch\_is(DisplayInformationClass)]
1019. PSAMPR\_DISPLAY\_INFO\_BUFFER Buffer
1020. );
1021. // opnum 41
1022. long
1023. SamrGetDisplayEnumerationIndex (
1024. [in] SAMPR\_HANDLE DomainHandle,
1025. [in] DOMAIN\_DISPLAY\_INFORMATION DisplayInformationClass,
1026. [in] PRPC\_UNICODE\_STRING Prefix,
1027. [out] unsigned long \* Index
1028. );
1029. // opnum 42
1030. void Opnum42NotUsedOnWire(void);
1031. // opnum 43
1032. void Opnum43NotUsedOnWire(void);
1033. // opnum 44
1034. long
1035. SamrGetUserDomainPasswordInformation (
1036. [in] SAMPR\_HANDLE UserHandle,
1037. [out] PUSER\_DOMAIN\_PASSWORD\_INFORMATION PasswordInformation
1038. );
1039. // opnum 45
1040. long
1041. SamrRemoveMemberFromForeignDomain (
1042. [in] SAMPR\_HANDLE DomainHandle,
1043. [in] PRPC\_SID MemberSid
1044. );
1045. // opnum 46
1046. long
1047. SamrQueryInformationDomain2(
1048. [in] SAMPR\_HANDLE DomainHandle,
1049. [in] DOMAIN\_INFORMATION\_CLASS DomainInformationClass,
1050. [out, switch\_is(DomainInformationClass)]
1051. PSAMPR\_DOMAIN\_INFO\_BUFFER \* Buffer
1052. );
1053. // opnum 47
1054. long
1055. SamrQueryInformationUser2(
1056. [in] SAMPR\_HANDLE UserHandle,
1057. [in] USER\_INFORMATION\_CLASS UserInformationClass,
1058. [out, switch\_is(UserInformationClass)]
1059. PSAMPR\_USER\_INFO\_BUFFER \* Buffer
1060. );
1061. // opnum 48
1062. long
1063. SamrQueryDisplayInformation2 (
1064. [in] SAMPR\_HANDLE DomainHandle,
1065. [in] DOMAIN\_DISPLAY\_INFORMATION DisplayInformationClass,
1066. [in] unsigned long Index,
1067. [in] unsigned long EntryCount,
1068. [in] unsigned long PreferredMaximumLength,
1069. [out] unsigned long \*TotalAvailable,
1070. [out] unsigned long \*TotalReturned,
1071. [out, switch\_is(DisplayInformationClass)]
1072. PSAMPR\_DISPLAY\_INFO\_BUFFER Buffer
1073. );
1074. // opnum 49
1075. long
1076. SamrGetDisplayEnumerationIndex2 (
1077. [in] SAMPR\_HANDLE DomainHandle,
1078. [in] DOMAIN\_DISPLAY\_INFORMATION DisplayInformationClass,
1079. [in] PRPC\_UNICODE\_STRING Prefix,
1080. [out] unsigned long \* Index
1081. );
1082. // opnum 50
1083. long
1084. SamrCreateUser2InDomain(
1085. [in] SAMPR\_HANDLE DomainHandle,
1086. [in] PRPC\_UNICODE\_STRING Name,
1087. [in] unsigned long AccountType,
1088. [in] unsigned long DesiredAccess,
1089. [out] SAMPR\_HANDLE \* UserHandle,
1090. [out] unsigned long \* GrantedAccess,
1091. [out] unsigned long \* RelativeId
1092. );
1093. // opnum 51
1094. long
1095. SamrQueryDisplayInformation3 (
1096. [in] SAMPR\_HANDLE DomainHandle,
1097. [in] DOMAIN\_DISPLAY\_INFORMATION DisplayInformationClass,
1098. [in] unsigned long Index,
1099. [in] unsigned long EntryCount,
1100. [in] unsigned long PreferredMaximumLength,
1101. [out] unsigned long \* TotalAvailable,
1102. [out] unsigned long \* TotalReturned,
1103. [out, switch\_is(DisplayInformationClass)]
1104. PSAMPR\_DISPLAY\_INFO\_BUFFER Buffer
1105. );
1106. // opnum 52
1107. long
1108. SamrAddMultipleMembersToAlias(
1109. [in] SAMPR\_HANDLE AliasHandle,
1110. [in] PSAMPR\_PSID\_ARRAY MembersBuffer
1111. );
1112. // opnum 53
1113. long
1114. SamrRemoveMultipleMembersFromAlias(
1115. [in] SAMPR\_HANDLE AliasHandle,
1116. [in] PSAMPR\_PSID\_ARRAY MembersBuffer
1117. );
1118. // opnum 54
1119. long
1120. SamrOemChangePasswordUser2(
1121. [in] handle\_t BindingHandle,
1122. [in,unique] PRPC\_STRING ServerName,
1123. [in] PRPC\_STRING UserName,
1124. [in,unique]
1125. PSAMPR\_ENCRYPTED\_USER\_PASSWORD NewPasswordEncryptedWithOldLm,
1126. [in,unique]
1127. PENCRYPTED\_LM\_OWF\_PASSWORD OldLmOwfPasswordEncryptedWithNewLm
1128. );
1129. // opnum 55
1130. long
1131. SamrUnicodeChangePasswordUser2(
1132. [in] handle\_t BindingHandle,
1133. [in,unique] PRPC\_UNICODE\_STRING ServerName,
1134. [in] PRPC\_UNICODE\_STRING UserName,
1135. [in,unique]
1136. PSAMPR\_ENCRYPTED\_USER\_PASSWORD NewPasswordEncryptedWithOldNt,
1137. [in,unique]
1138. PENCRYPTED\_NT\_OWF\_PASSWORD OldNtOwfPasswordEncryptedWithNewNt,
1139. [in] unsigned char LmPresent,
1140. [in,unique]
1141. PSAMPR\_ENCRYPTED\_USER\_PASSWORD NewPasswordEncryptedWithOldLm,
1142. [in,unique]
1143. PENCRYPTED\_LM\_OWF\_PASSWORD OldLmOwfPasswordEncryptedWithNewNt
1144. );
1145. // opnum 56
1146. long
1147. SamrGetDomainPasswordInformation (
1148. [in] handle\_t BindingHandle,
1149. [in,unique] PRPC\_UNICODE\_STRING Unused,
1150. [out] PUSER\_DOMAIN\_PASSWORD\_INFORMATION PasswordInformation
1151. );
1152. // opnum 57
1153. long
1154. SamrConnect2(
1155. [in,unique,string] PSAMPR\_SERVER\_NAME ServerName,
1156. [out] SAMPR\_HANDLE \*ServerHandle,
1157. [in] unsigned long DesiredAccess
1158. );
1159. // opnum 58
1160. long
1161. SamrSetInformationUser2(
1162. [in] SAMPR\_HANDLE UserHandle,
1163. [in] USER\_INFORMATION\_CLASS UserInformationClass,
1164. [in, switch\_is(UserInformationClass)]
1165. PSAMPR\_USER\_INFO\_BUFFER Buffer
1166. );
1167. // opnum 59
1168. void Opnum59NotUsedOnWire(void);
1169. // opnum 60
1170. void Opnum60NotUsedOnWire(void);
1171. // opnum 61
1172. void Opnum61NotUsedOnWire(void);
1173. // opnum 62
1174. long
1175. SamrConnect4(
1176. [in,unique,string] PSAMPR\_SERVER\_NAME ServerName,
1177. [out] SAMPR\_HANDLE \*ServerHandle,
1178. [in] unsigned long ClientRevision,
1179. [in] unsigned long DesiredAccess
1180. );
1181. // opnum 63
1182. void Opnum63NotUsedOnWire(void);
1183. // opnum 64
1184. long
1185. SamrConnect5(
1186. [in,unique,string] PSAMPR\_SERVER\_NAME ServerName,
1187. [in] unsigned long DesiredAccess,
1188. [in] unsigned long InVersion,
1189. [in] [switch\_is(InVersion)] SAMPR\_REVISION\_INFO \*InRevisionInfo,
1190. [out] unsigned long \*OutVersion,
1191. [out] [switch\_is(\*OutVersion)]
1192. SAMPR\_REVISION\_INFO \*OutRevisionInfo,
1193. [out] SAMPR\_HANDLE \*ServerHandle
1194. );
1195. // opnum 65
1196. long
1197. SamrRidToSid(
1198. [in] SAMPR\_HANDLE ObjectHandle,
1199. [in] unsigned long Rid,
1200. [out] PRPC\_SID \* Sid
1201. );
1202. // opnum 66
1203. long
1204. SamrSetDSRMPassword(
1205. [in] handle\_t BindingHandle,
1206. [in,unique] PRPC\_UNICODE\_STRING Unused,
1207. [in] unsigned long UserId,
1208. [in,unique] PENCRYPTED\_NT\_OWF\_PASSWORD EncryptedNtOwfPassword
1209. );
1210. // opnum 67
1211. long
1212. SamrValidatePassword(
1213. [in] handle\_t Handle,
1214. [in] PASSWORD\_POLICY\_VALIDATION\_TYPE ValidationType,
1215. [in, switch\_is(ValidationType)] PSAM\_VALIDATE\_INPUT\_ARG InputArg,
1216. [out, switch\_is(ValidationType)]
1217. PSAM\_VALIDATE\_OUTPUT\_ARG \* OutputArg
1218. );
1219. // Opnum 68
1220. void Opnum68NotUsedOnWire(void);
1221. // Opnum 69
1222. void Opnum69NotUsedOnWire(void);
1223. // Opnum 70
1224. void Opnum70NotUsedOnWire(void);
1225. // Opnum 71
1226. void Opnum71NotUsedOnWire(void);
1227. // Opnum 72
1228. void Opnum72NotUsedOnWire(void);
1229. // Opnum 73
1230. NTSTATUS
1231. SamrUnicodeChangePasswordUser4(
1232. [in] handle\_t BindingHandle,
1233. [in,unique] PRPC\_UNICODE\_STRING ServerName,
1234. [in] PRPC\_UNICODE\_STRING UserName,
1235. [in] PSAMPR\_ENCRYPTED\_PASSWORD\_AES EncryptedPassword

); }

# Appendix B: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include updates to those products.

The terms "earlier" and "later", when used with a product version, refer to either all preceding versions or all subsequent versions, respectively. The term "through" refers to the inclusive range of versions. Applicable Microsoft products are listed chronologically in this section.

The following tables show the relationships between Microsoft product versions or supplemental software and the roles they perform.

| Windows Client releases | Client role | Server role |
| --- | --- | --- |
| Windows NT 3.1 operating system | Yes | Yes |
| Windows NT 3.5 operating system | Yes | Yes |
| Windows NT 3.51 operating system | Yes | Yes |
| Windows NT 4.0 operating system | Yes | Yes |
| Windows 2000 Professional operating system | Yes | Yes |
| Windows XP operating system | Yes | Yes |
| Windows Vista operating system | Yes | Yes |
| Windows 7 operating system | Yes | Yes |
| Windows 8 operating system | Yes | Yes |
| Windows 8.1 operating system | Yes | Yes |
| Windows 10 operating system | Yes | Yes |

| Windows Server releases | Client role | Server role |
| --- | --- | --- |
| Windows NT 3.1 | Yes | Yes |
| Windows NT 3.5 | Yes | Yes |
| Windows NT 3.51 | Yes | Yes |
| Windows NT 4.0 | Yes | Yes |
| Windows 2000 Server operating system | Yes | Yes |
| Windows Server 2003 operating system | Yes | Yes |
| Windows Server 2008 operating system | Yes | Yes |
| Windows Server 2008 R2 operating system | Yes | Yes |
| Windows Server 2012 operating system | Yes | Yes |
| Windows Server 2012 R2 operating system | Yes | Yes |
| Windows Server 2016 operating system | Yes | Yes |
| Windows Server operating system | Yes | Yes |
| Windows Server 2019 operating system | Yes | Yes |
| Windows Server 2022 operating system | Yes | Yes |

Exceptions, if any, are noted in this section. If an update version, service pack or Knowledge Base (KB) number appears with a product name, the behavior changed in that update. The new behavior also applies to subsequent updates unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

[<1> Section 1.3.2](#Appendix_A_Target_1): There is no supported configuration in which this method is called from Windows clients. See section [2.2.7.15](#Section_963e60b592334669b8a885bf4f0806dc) for details on the conditions under which this method is called from a client.

[<2> Section 1.6](#Appendix_A_Target_2): The [**DC**](#gt_76a05049-3531-4abd-aec8-30e19954b4bd) implementation of this protocol is largely for backward compatibility with Windows NT 4.0–style applications. The LDAP protocol can be used to access a superset of the information exposed in this protocol (see [[MS-ADTS]](%5bMS-ADTS%5d.pdf#Section_d243592709994c628c6d13ba31a52e1a) section 3.1.1.3). The notable exceptions to this rule are that Windows clients use this protocol to join a [**domain**](#gt_b0276eb2-4e65-4cf1-a718-e0920a614aca) ([[MS-ADOD]](%5bMS-ADOD%5d.pdf#Section_5ff67bf4c14548cb89cd4f5482d94664) sections 2.7.7 and 3.1) and that they use the [SamrUnicodeChangePasswordUser2](#Section_acb3204ada8b478e91391ea589edb880) method to change passwords.

[<3> Section 1.6](#Appendix_A_Target_3): Windows clients depend on this protocol in order to perform an end-user password change and join computers to a domain (as specified in [MS-ADTS] section 6.4).

[<4> Section 1.7.1](#Appendix_A_Target_4): The following table depicts a timeline of when each method was introduced. The Product column indicates the Windows version in which each method was introduced. Unless otherwise noted, all methods listed in the table continue to be supported in subsequent versions of Windows according to the applicability lists at the beginning of this section.

| Opnum | Friendly name | Product |
| --- | --- | --- |
| 0 | [SamrConnect](#Section_defe20910a614dfabe9a2c1206d53a1f) | Windows NT 3.1 |
| 1 | [SamrCloseHandle](#Section_55d134dfe25748ad8afacb2ca45cd3cc) | Windows NT 3.1 |
| 2 | [SamrSetSecurityObject](#Section_6666a06658cf4118bf4bdd54ed55ecf0) | Windows NT 3.1 |
| 3 | [SamrQuerySecurityObject](#Section_0ecf8fecd17e4a88b7f1e0f0f66790db) | Windows NT 3.1 |
| 4 | Reserved (not intended for network traffic) | - |
| 5 | [SamrLookupDomainInSamServer](#Section_47492d59e0954398b03e8a062b989123) | Windows NT 3.1 |
| 6 | [SamrEnumerateDomainsInSamServer](#Section_2142fd2d085442c1a9fb2fe964e381ce) | Windows NT 3.1 |
| 7 | [SamrOpenDomain](#Section_ba710c905b1242f89e5ad4aacc1329fa) | Windows NT 3.1 |
| 8 | [SamrQueryInformationDomain](#Section_5d6a2817caa941caa269fd13ecbb4fa8) | Windows NT 3.1 |
| 9 | [SamrSetInformationDomain](#Section_9b7ae0b0bd1141339c62fba7095aee12) | Windows NT 3.1 |
| 10 | [SamrCreateGroupInDomain](#Section_175c1cf94fa248379e5bbb1f0f950bee) | Windows NT 3.1 |
| 11 | [SamrEnumerateGroupsInDomain](#Section_e0b7a4b7ecfc405f9d7d32b3cd2cd6c8) | Windows NT 3.1 |
| 12 | [SamrCreateUserInDomain](#Section_cd0efce013fb410687db4bef73bb4b1b) | Windows NT 3.1 |
| 13 | [SamrEnumerateUsersInDomain](#Section_6bdc92c0c6924ffb9de765858b68da75) | Windows NT 3.1 |
| 14 | [SamrCreateAliasInDomain](#Section_7e505875e44f4f9d922b7ff3871c752d) | Windows NT 3.1 |
| 15 | [SamrEnumerateAliasesInDomain](#Section_ce340cffedef4356ace033d2874d306b) | Windows NT 3.1 |
| 16 | [SamrGetAliasMembership](#Section_0318404522084c02b38bef955d6dc3ef) | Windows NT 3.1 |
| 17 | [SamrLookupNamesInDomain](#Section_d91271c67b2e419499278fabfa429f90) | Windows NT 3.1 |
| 18 | [SamrLookupIdsInDomain](#Section_c870951c74b347149857224595ffc61a) | Windows NT 3.1 |
| 19 | [SamrOpenGroup](#Section_d396e6c9d04a4729b0d8f50f2748f3c8) | Windows NT 3.1 |
| 20 | [SamrQueryInformationGroup](#Section_8fd0003f807640ae872f9cdb6576b3c7) | Windows NT 3.1 |
| 21 | [SamrSetInformationGroup](#Section_e66db19f600a481bbc4e23953433255d) | Windows NT 3.1 |
| 22 | [SamrAddMemberToGroup](#Section_3c70fec36a2e48efbd4eaa2b3a1cd96a) | Windows NT 3.1 |
| 23 | [SamrDeleteGroup](#Section_664ed55f198a4775b9d4398131dba577) | Windows NT 3.1 |
| 24 | [SamrRemoveMemberFromGroup](#Section_613ff6635863443089238f1edc05310b) | Windows NT 3.1 |
| 25 | [SamrGetMembersInGroup](#Section_3ed5030d88a342caa6e08c12aa2fdfbd) | Windows NT 3.1 |
| 26 | [SamrSetMemberAttributesOfGroup](#Section_a4db0c2755404d2fb11108db1ee0b5a4) | Windows NT 3.1 |
| 27 | [SamrOpenAlias](#Section_c3696b7620054eba820d91fdded3422d) | Windows NT 3.1 |
| 28 | [SamrQueryInformationAlias](#Section_599d6624d93b46de827a58eb0e7527bd) | Windows NT 3.1 |
| 29 | [SamrSetInformationAlias](#Section_ba787e4e3a4b47a2aca3c3ac2d2c511e) | Windows NT 3.1 |
| 30 | [SamrDeleteAlias](#Section_ecb88df184d148c8b49cd3f444943100) | Windows NT 3.1 |
| 31 | [SamrAddMemberToAlias](#Section_9a5d2c35e84b4e59b7b096c6fa0fc8d7) | Windows NT 3.1 |
| 32 | [SamrRemoveMemberFromAlias](#Section_7e99402967a244fd84397640fe5a376b) | Windows NT 3.1 |
| 33 | [SamrGetMembersInAlias](#Section_5909af7f858b4cc7a2e4f852070d479d) | Windows NT 3.1 |
| 34 | [SamrOpenUser](#Section_0aee1c31ec404633bb560cf8429093c0) | Windows NT 3.1 |
| 35 | [SamrDeleteUser](#Section_4643a57956ec4c66a1ef4ab78dd21d73) | Windows NT 3.1 |
| 36 | [SamrQueryInformationUser](#Section_4ad8d54c0d5a4d5a9e9a1bc9ee008d47) | Windows NT 3.1 |
| 37 | [SamrSetInformationUser](#Section_538222f71b894811949a0eac62e38dce) | Windows NT 3.1 |
| 38 | [SamrChangePasswordUser](#Section_9699d8cae1a4433ca8c3d7bebeb01476) | Windows NT 3.1 |
| 39 | [SamrGetGroupsForUser](#Section_a4adbf20040f4416a960e5b7917fdae7) | Windows NT 3.1 |
| 40 | [SamrQueryDisplayInformation](#Section_c1458942f2d54317a888abd27abad504) | Windows NT 3.1 |
| 41 | [SamrGetDisplayEnumerationIndex](#Section_af95abf82e1142c394aaba72f8f96c09) | Windows NT 3.1 |
| 42 | Reserved (not intended for network traffic) | - |
| 43 | Reserved (not intended for network traffic) | - |
| 44 | [SamrGetUserDomainPasswordInformation](#Section_db4bedfe465147439381d74c3ad8c41c) | Windows NT 3.1 |
| 45 | [SamrRemoveMemberFromForeignDomain](#Section_03afc843584d473b834a3f5a1ac86cce) | Windows NT 3.1 |
| 46 | [SamrQueryInformationDomain2](#Section_fbdbbd7401064ca2941e36ded0be0c62) | Windows NT 3.5 |
| 47 | [SamrQueryInformationUser2](#Section_29ab27f661da4c7d863ce228ee798f4d) | Windows NT 3.5 |
| 48 | [SamrQueryDisplayInformation2](#Section_9aa37aa4f77e49b199935c92ba97cd62) | Windows NT 3.5 |
| 49 | [SamrGetDisplayEnumerationIndex2](#Section_bd429624f2d547178aa2659952c3e209) | Windows NT 3.5 |
| 50 | [SamrCreateUser2InDomain](#Section_a98d7fbb17354fbfb41aef363c899002) | Windows NT 3.5 |
| 51 | [SamrQueryDisplayInformation3](#Section_7cd1c6d088194a5897215be588c1d4ad) | Windows NT 3.5 |
| 52 | [SamrAddMultipleMembersToAlias](#Section_6d981290a203408094b04a26de22d8c5) | Windows NT 3.51 |
| 53 | [SamrRemoveMultipleMembersFromAlias](#Section_7b2455d23f1344a7996950c240562c42) | Windows NT 3.51 |
| 54 | [SamrOemChangePasswordUser2](#Section_8d0bf63efa5f4c75be22558c52075842) | Windows NT 3.51 |
| 55 | SamrUnicodeChangePasswordUser2 | Windows NT 3.51 |
| 56 | [SamrGetDomainPasswordInformation](#Section_1a8841487e904a93b4c47a0006f59026) | Windows NT 3.51 |
| 57 | [SamrConnect2](#Section_1076eb2a4f514c5aa7c7a78323b06198) | Windows NT 3.51 |
| 58 | [SamrSetInformationUser2](#Section_99ee9f3943e84bbaac3a82e0c0e0699e) | Windows NT 3.51 |
| 59 | Reserved (not intended for network traffic) | - |
| 60 | Reserved (not intended for network traffic) | - |
| 61 | Reserved (not intended for network traffic) | - |
| 62 | [SamrConnect4](#Section_c1b14e670f4647a69fa70689d284afc9) | Windows 2000 operating system |
| 63 | Reserved (not intended for network traffic) | - |
| 64 | [SamrConnect5](#Section_c842a8970a424ca5a6072afd05271dae) | Windows XP and Windows Server 2003 |
| 65 | [SamrRidToSid](#Section_00ff8192a4f645ba9f65917e46b6a693) | Windows XP and Windows Server 2003 |
| 66 | [SamrSetDSRMPassword](#Section_9bcad7d2b8e14d28a03386040b7b3ce9) | Windows 2000 Server SP2 and Windows XP |
| 67 | [SamrValidatePassword](#Section_c78a7239f8fc4a42bb71321e897dc046) | Windows Server 2003 and Windows Vista |
| 68 | Reserved (not intended for network traffic) | - |
| 69 | Reserved (not intended for network traffic) | - |

[<5> Section 1.7.2](#Appendix_A_Target_5): Windows clients call deprecated methods under the following conditions. There is no benefit in doing so.

| Deprecated method | Condition |
| --- | --- |
| SamrQueryInformationDomain | Windows clients call this method for information levels less than or equal to DomainStateInformation (see section [2.2.3.16](#Section_3e8738b25df6499f907dac2471bf0281) for a description of the information levels). |
| SamrQueryDisplayInformation | Windows clients call this method for information levels less than or equal to DomainDisplayMachine (see section [2.2.8.12](#Section_929775e003394f0b81616d3699c18185) for a description of the information levels). |
| SamrQueryDisplayInformation2 | Windows clients call this method for information levels less than or equal to DomainDisplayGroup (see section 2.2.8.12 for a description of the information levels). |
| SamrGetDisplayEnumerationIndex | Windows clients call this method for information levels less than or equal to DomainDisplayMachine (see section 2.2.8.12 for a description of the information levels). |
| SamrQueryInformationUser | Windows clients call this method under all conditions; even though SamrQueryInformationUser2 is available to be called, it is not called from any Windows clients. |
| SamrSetInformationUser | Windows clients call this method for information levels other than [UserInternal4InformationNew](#Section_2d20cab4a0d4401baeebb72f17403ece) and UserInternal5InformationNew (see section [2.2.6.28](#Section_6b0dff905ac0429a93aa150334adabf6) for a description of the information levels). |

[<6> Section 1.7.3](#Appendix_A_Target_6): All information levels are supported in Windows NT 4.0, Windows 2000 Server, and later with the exception of GroupReplicationInformation for SamrQueryInformationGroup. This information level is supported in Windows Server 2003, and later.

[<7> Section 2.1](#Appendix_A_Target_7): Windows NT operating system, Windows 2000, Windows Server 2003, and Windows Server 2003 R2 operating system implementations of the server for this protocol can be configured to use the SPX (NCACN\_SPX) protocol, as specified in [[MS-RPCE]](%5bMS-RPCE%5d.pdf#Section_290c38b192fe422991e64fc376610c15) section 2.1.1.3; the AppleTalk (NCACN\_AT\_DSP) protocol, as specified in [MS-RPCE] section 2.1.1.7; and the Banyan VINES protocol. This configuration can be enabled by adding the following registry values of type REG\_DWORD and by modifying the value to be nonzero:

HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\LSA

* For SPX: NetWareClientSupport
* For Appletalk: AppletalkClientSupport
* For Banyan VINES: VinesClientSupport

In addition, none of the Windows implementations of the client for this protocol can be configured to use protocols that are not listed in section [2.1](#Section_084da2e70ba044fc8f17e8a200c69eb5).

[<8> Section 2.1](#Appendix_A_Target_8): Windows 2000, Windows XP, Windows Server 2003, and Windows Server 2003 R2 process calls for all opnums over the RPC-over-named-pipes (NCACN\_NP) protocol. Windows Vista operating system with Service Pack 2 (SP2), Windows 7, and later, and Windows Server 2008 operating system with Service Pack 2 (SP2), Windows Server 2008 R2, and later behave in the same way, except that calls made to SamrValidatePassword using NCACN\_NP are rejected with RPC\_S\_ACCESS\_DENIED.

[<9> Section 2.1](#Appendix_A_Target_9): By default, the endpoint "\PIPE\samr" allows anonymous access on Windows NT 3.1, Windows NT 3.5, Windows NT 3.51, Windows 2000, Windows XP, Windows Server 2003, Windows Server 2003 R2, and Windows Vista. Anonymous access to this pipe on non–domain controller machines is removed by default on Windows Vista operating system with Service Pack 1 (SP1), Windows 7, and later, and on Windows Server 2008 and later. The pipe [**access check**](#gt_d7906f17-bb2c-4193-a3f0-848bcc351dec) happens before any other access check, and therefore overrides any other access.

[<10> Section 2.1](#Appendix_A_Target_10): Windows 2000, Windows XP, Windows Server 2003, and Windows Server 2003 R2 process calls for all opnums over TCP (NCACN\_IP\_TCP). Windows Vista SP2, Windows 7, and later, and Windows Server 2008 with SP2, Windows Server 2008 R2, and later behave in the same way, except that calls made to SamrSetDSRMPassword using NCACN\_IP\_TCP are rejected with RPC\_S\_ACCESS\_DENIED.

[<11> Section 2.1](#Appendix_A_Target_11): A service-specific service principal name is not registered for this protocol. Windows-based clients use the host-based service principal name to identify the server for mutual authentication for the SMB and TCP RPC transports.

[<12> Section 2.1](#Appendix_A_Target_12): Servers running Windows 2000, Windows XP, and Windows Server 2003 accept calls at any authentication level. Without [[MSKB-3149090]](https://go.microsoft.com/fwlink/?LinkId=786963) installed, servers running Windows Vista, Windows Server 2008, Windows 7, Windows Server 2008 R2, Windows 8, Windows Server 2012, Windows 8.1, Windows Server 2012 R2, Windows 10 v1507 operating system, or Windows 10 v1511 operating system also accept calls at any authentication level.

[<13> Section 2.1](#Appendix_A_Target_13): The Windows-based client uses transport security to encrypt the message for SamrValidatePassword.

[<14> Section 2.2.6.1](#Appendix_A_Target_14): Windows interactive-logon applications expect this value to be a UNC path (for example, \\machine-name\share-name\directory-name), or a fully qualified local path, including the drive letter (for example, "c:\directory\folder").

[<15> Section 2.2.6.1](#Appendix_A_Target_15): Windows interactive-logon applications expect this value to be either a zero-length string or a string with two characters: an alphabetic character, 'a' through 'z', in lower- or uppercase, followed by a colon (':').

[<16> Section 2.2.6.1](#Appendix_A_Target_16): This value is not accurate in multiple-DC configurations, as this value is not replicated among DCs. Therefore, this field is not to be used by clients. Windows clients do not use this field.

[<17> Section 2.2.6.1](#Appendix_A_Target_17): This value is not accurate in multiple-DC configurations, because this value is not replicated among DCs. Windows clients do not use this field.

[<18> Section 2.2.6.1](#Appendix_A_Target_18): This value is not accurate in multiple-DC configurations, because this value is not replicated among DCs. Therefore, this field is not to be used by clients. Windows clients do not use this field.

[<19> Section 2.2.7.15](#Appendix_A_Target_19): There is no supported configuration in which Windows implementations of the server of this protocol (for example, a DC) return nonzero values for the **SupportedFeatures** field. However, Windows protocol clients running Windows XP and later are implemented to behave as specified in the description for the **SupportedFeatures** field. For example, after calling SamrCreateUser2InDomain (section 3.1.5.4.4), Windows NT 4.0–style client applications assume that the [**RID**](#gt_df3d0b61-56cd-4dac-9402-982f1fedc41c) returned by SamrCreateUser2InDomain can be concatenated with the domain [**SID**](#gt_83f2020d-0804-4840-a5ac-e06439d50f8d) in which the user was created to obtain the SID of the newly created user. This assumption limits the server's ability to create SIDs that differ in format from this assumption, and thus limits the number of accounts ever created to 2^32 (the maximum size of an unsigned integer, which is the datatype of a RID). For more information about the extensible structure of SIDs, see [[MS-AZOD]](%5bMS-AZOD%5d.pdf#Section_5a0a0a3ec7a742e1b5f2cc8d8bd9739e) section 1.1.1.2.

To allow servers (in future implementations) to generate SIDs such that the RID is not an unsigned integer (for example, a 64-bit value), the SupportedFeatures value of 1 specifies to the client that the SamrRidToSid method is to be called to obtain the SID of a RID value returned from this protocol. In this scenario, the RID returned from the protocol is modeled as a "handle" to the [**account**](#gt_8d10bd9d-392e-431b-8218-742bdac19815) that SamrRidToSid uses to return the SID value.

The two reserved values (0x00000002 and 0x00000004) have no effect on the protocol; however, when these values are set, the Windows NET API ([[MSDN-NMF]](https://go.microsoft.com/fwlink/?LinkId=90052)) on the client behaves as shown in the following table. These values are mutually exclusive with each other, though they can be combined using a logical OR with other bits.

| Value | Description |
| --- | --- |
| 0x00000002 | All fields that return a RID value return the value 0 instead of the RID value returned from the SAM Remote Protocol (Client-to-Server). |
| 0x00000004 | All method calls that accept information levels that return a RID fail with a Windows error code of ERROR\_NOT\_SUPPORTED (defined in [[MS-ERREF]](%5bMS-ERREF%5d.pdf#Section_1bc92ddfb79e413cbbaa99a5281a6c90) section 2.2). |

[<20> Section 2.2.10.1](#Appendix_A_Target_20): Windows sets this buffer to the repeating pattern 0x20 0x00 on update.

[<21> Section 2.2.10.1](#Appendix_A_Target_21): Windows implementations of the protocol server set the **Reserved5** field to arbitrary values.

[<22> Section 2.2.10.2](#Appendix_A_Target_22): Windows sets this value to 1 or 2, but does not use the value.

[<23> Section 2.2.10.3](#Appendix_A_Target_23): Windows sets this value to 0x31 and ignores it on read.

[<24> Section 2.2.10.8](#Appendix_A_Target_24): When the current domain functional level is DS\_BEHAVIOR\_WIN2003 or less, a Windows Server 2008 and later DC includes a **KeyType** of -140 in each of KERB\_STORED\_CREDENTIAL and KERB\_STORED\_CREDENTIAL\_NEW, which is not needed and can be ignored; it is a dummy type in the supplemental credentials that is not present when the [**domain functional level**](#gt_f9b5a663-12d5-46d4-b23a-147704f026fc) is raised to DS\_BEHAVIOR\_WIN2008 or greater. The key data is the [**NT hash**](#gt_7eb5bce8-bdc5-4d61-a3d9-263ada14ba1f) of the password.

[<25> Section 3.1.1.5](#Appendix_A_Target_25): Windows 2000 Server, Windows Server 2003, and Windows Server 2003 R2 do not support the msDS-ResultantPSO attribute.

[<26> Section 3.1.1.6](#Appendix_A_Target_26): This modification is always allowed in Windows 2000 and in the following products that do NOT have [[MSKB-3072595]](https://go.microsoft.com/fwlink/?LinkId=626613) installed: Windows Server 2003, Windows Server 2003 R2, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2.

[<27> Section 3.1.1.8.3](#Appendix_A_Target_27): On a DC configuration, Windows initiates urgent replication (described in [MS-ADTS] section 3.1.1.1.14, under event-driven replication) when this attribute value changes.

[<28> Section 3.1.1.8.8](#Appendix_A_Target_28): On a DC configuration, Windows initiates urgent replication (described in [MS-ADTS] section 3.1.1.1.14, under event-driven replication) when this attribute value is set to 0 or when this attribute value changes due to a password change request (as opposed to set) and **userAccountControl** contains the UF\_NORMAL\_ACCOUNT flag.

[<29> Section 3.1.1.8.10](#Appendix_A_Target_29): On a DC configuration, if the UF\_SERVER\_TRUST\_ACCOUNT bit or the UF\_WORKSTATION\_TRUST\_ACCOUNT bit changes on commit, an urgent replication is initiated. (Information about urgent replication is specified in [MS-ADTS] section 3.1.1.1.14.)

[<30> Section 3.1.1.8.11.4](#Appendix_A_Target_30): Windows uses the account's userPrincipalName as the **DefaultSalt** value. However, it does not use this value in any calculation.

[<31> Section 3.1.1.8.11.4](#Appendix_A_Target_31): Windows implementations of the protocol server include irrelevant bytes in the [KERB\_STORED\_CREDENTIAL](#Section_866f018086944e14be7edfd91d91694f) structure for a single [KERB\_KEY\_DATA](#Section_9921c3edf4704f369edf7afe24c468ec) structure (20 bytes). The bytes appear directly prior to the start of **DefaultSalt**. They are not referenced by any offset value or necessary for interoperability. All bits in these bytes are 0.

[<32> Section 3.1.1.8.11.6](#Appendix_A_Target_32): Windows implementations of the protocol server include irrelevant bytes in the [KERB\_STORED\_CREDENTIAL\_NEW](#Section_08cb3ca7954b45e3902e77512fe3ba8e) structure for a single [KERB\_KEY\_DATA\_NEW](#Section_447520a5e1cc48cc8fdcb90db57f7eac) structure (24 bytes). The bytes appear directly prior to the start of **DefaultSalt**. They are not referenced by any offset value or necessary for interoperability. All bits in these bytes are 0.

[<33> Section 3.1.1.8.11.7](#Appendix_A_Target_33): Windows 2000 Server, Windows Server 2003, Windows Server 2003 R2, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2 do not set the NTLM-Strong-NTOWF property.

[<34> Section 3.1.1.9.2.1](#Appendix_A_Target_34): If the constraints in step 1 cannot be satisfied, the server returns an error code to the client and initiates an asynchronous call to IDL\_DRSGetNCChanges to obtain a new **rIDAllocationPool**, if such an asynchronous call is not already active.

[<35> Section 3.1.2](#Appendix_A_Target_35): In Windows 2000 operating system Service Pack 4 (SP4), Windows Server 2003 operating system with Service Pack 1 (SP1), Windows Server 2003 R2, and Windows XP operating system Service Pack 2 (SP2), the Windows implementation of RPC does not satisfy this requirement. Consequently, a security check is enforced by the server of this protocol to ensure this constraint. Specifically, the server ensures that the SID of the client matches the SID of the client that opened the handle. If this condition is not met, a processing error is returned to the client.

[<36> Section 3.1.2.1](#Appendix_A_Target_36): Windows enforces this check on Windows 10 version 1607 and later, Windows Server 2016 and later, Windows 10 version 1511 with [[MSKB-4013198]](https://go.microsoft.com/fwlink/?linkid=2163543) installed, Windows 10 version 1507 with [[MSKB-4012606]](https://go.microsoft.com/fwlink/?linkid=2163542) installed, Windows 8.1 with [[MSKB-4102219]](https://go.microsoft.com/fwlink/?linkid=2163445) installed, Windows 7 with [[MSKB-4012218]](https://support.microsoft.com/en-us/topic/march-2017-preview-of-monthly-quality-rollup-for-windows-8-1-and-windows-server-2012-r2-3f3d4166-4b8c-6f88-17c9-64cb0378acd4) installed, Windows Server 2012 R2 with [MSKB-4102219] installed, Windows Server 2012 with [[MSKB-4012220]](https://go.microsoft.com/fwlink/?linkid=2163626) installed, and Windows Server 2008 R2 with [MSKB-4012218] installed. For more information, see [[MSDOCS-RESTRICTRMTSAM]](https://go.microsoft.com/fwlink/?linkid=2163448).

[<37> Section 3.1.4.2](#Appendix_A_Target_37): The following tables list the Windows versions in which various accounts were introduced. All accounts continue to exist in subsequent versions of Windows according to the applicability lists at the beginning of this section.

Non-DC configuration, user accounts.

| Name | Revision introduced |
| --- | --- |
| Administrator | Windows NT 3.1 |
| Guest | Windows NT 3.1 |

Non-DC configuration, alias accounts.

| Name | Revision introduced |
| --- | --- |
| Administrators | Windows NT 3.1 |
| Users | Windows NT 3.1 |
| Guests | Windows NT 3.1 |
| Power Users | Windows NT 3.1 |
| Print Operators | Windows NT 3.1 |
| Backup Operators | Windows NT 3.1 |
| Replicator | Windows NT 3.1 |
| Remote Desktop Users | Windows XP  Windows Server 2003 |
| Network Configuration Operators | Windows XP  Windows Server 2003 |
| Performance Monitor Users | Windows Server 2003  Windows Vista |
| Performance Log Users | Windows Server 2003  Windows Vista |
| Distributed COM Users | Windows Server 2003 with SP1  Windows Vista |
| IIS\_IUSRS | Windows Vista  Windows Server 2008 |
| Cryptographic Operators | Windows Vista  Windows Server 2008 |
| Event Log Readers | Windows Vista  Windows Server 2008 |

DC configuration, user accounts.

| Name | Revision introduced |
| --- | --- |
| Administrator | Windows NT 3.1 |
| Guest | Windows NT 3.1 |
| krbtgt | Windows 2000 |

DC configuration, [**universal group**](#gt_f46053d6-0708-4094-ac63-57c1bcb73d32) accounts (only on root domain).

| Name | Revision introduced |
| --- | --- |
| Schema Admins | Windows 2000 |
| Enterprise Admins | Windows 2000 |
| Enterprise Read-only Domain Controllers | Windows Server 2008 |

DC configuration, group accounts.

| Name | Revision introduced |
| --- | --- |
| Domain Admins | Windows NT 3.1 |
| Domain Users | Windows NT 3.1 |
| Domain Guests | Windows NT 3.1 |
| Domain Computers | Windows NT 3.1 |
| Domain Controllers | Windows NT 3.1 |
| Group Policy Creator Owners | Windows 2000 Server  Windows XP |
| Read-only Domain Controllers | Windows Server 2008 |

DC configuration, alias accounts.

| Name | Revision introduced |
| --- | --- |
| Administrators | Windows NT 3.1 |
| Users | Windows NT 3.1 |
| Guests | Windows NT 3.1 |
| Account Operators | Windows NT 3.1 |
| System Operators | Windows NT 3.1 |
| Print Operators | Windows NT 3.1 |
| Backup Operators | Windows NT 3.1 |
| Replicator | Windows NT 3.1 |
| Cert Publishers | Windows 2000 |
| RAS and IAS Servers | Windows 2000 |
| Pre-Windows 2000 Compatible Access | Windows 2000 |
| Remote Desktop Users | Windows Server 2003 |
| Network Configuration Operators | Windows Server 2003 |
| Incoming Forest Trust Builders | Windows Server 2003 |
| Performance Monitor Users | Windows Server 2003 |
| Performance Log Users | Windows Server 2003 |
| Windows Authorization Access Group | Windows Server 2003 |
| Terminal Server License Servers | Windows Server 2003 |
| Distributed COM Users | Windows Server 2003 with SP1 |
| IIS\_IUSRS | Windows Vista  Windows Server 2008 |
| Cryptographic Operators | Windows Vista  Windows Server 2008 |
| Allowed RODC Password Replication Group | Windows Vista  Windows Server 2008 |
| Denied RODC Password Replication Group | Windows Vista  Windows Server 2008 |
| Event Log Readers | Windows Vista  Windows Server 2008 |
| Certificate Service DCOM Access | Windows Vista SP1  Windows Server 2008 |

[<38> Section 3.1.4.2](#Appendix_A_Target_38): In Windows 2000 Server, Windows Server 2003, and Windows Server 2003 R2, the initial membership of this group depends on the version of Windows running on the first DC of the domain and on the administrator's choice between "Pre-Windows 2000–compatible permissions mode" and "Windows 2000–only permissions mode".

Membership of the "Pre-Windows 2000 Compatible Access" group in Windows 2000 Server, Windows Server 2003, and Windows Server 2003 R2 is shown in the following table.

| Operating system version | "Pre-Windows 2000-compatible permissions mode" | "Windows 2000-only permissions mode" |
| --- | --- | --- |
| Windows 2000 Server | "Everyone" (S-1-1-0) | No members |
| Windows Server 2003 | "Everyone" (S-1-1-0)  "Anonymous" (S-1-5-7) | "Authenticated Users" (S-1-5-11) |
| Windows Server 2003 R2 | "Everyone" (S-1-1-0)  "Anonymous" (S-1-5-7) | "Authenticated Users" (S-1-5-11) |

Membership of the "Pre-Windows 2000 Compatible Access" group in Windows Server 2008 and later is "Authenticated Users" (S-1-5-11).

[<39> Section 3.1.5](#Appendix_A_Target_39): Opnums reserved for local use apply to Windows as follows.

| Opnum | Description |
| --- | --- |
| 4 | Not used by Windows. |
| 42 | Just returns STATUS\_NOT\_IMPLEMENTED. It is never used. |
| 43 | Just returns STATUS\_NOT\_IMPLEMENTED. It is never used. |
| 59 | Used only locally by Windows, never remotely. |
| 60 | Used only locally by Windows, never remotely. |
| 61 | Not used by Windows. |
| 63 | Not used by Windows. |
| 68 | Used only locally by Windows, never remotely. |
| 69 | Used only locally by Windows, never remotely. |

[<40> Section 3.1.5.1.1](#Appendix_A_Target_40): *ServerName* is ignored on receipt.

[<41> Section 3.1.5.1.2](#Appendix_A_Target_41): *ServerName* is ignored on receipt.

[<42> Section 3.1.5.1.3](#Appendix_A_Target_42): *ServerName* is ignored on receipt.

[<43> Section 3.1.5.1.4](#Appendix_A_Target_43): *ServerName* is ignored on receipt.

[<44> Section 3.1.5.2.1](#Appendix_A_Target_44): Windows does NOT validate the input, though the result of malformed information merely results in inconsistent output to the client.

[<45> Section 3.1.5.2.1](#Appendix_A_Target_45): Windows estimates the number of entries to return by dividing *PreferedMaximumLength* by the number of bytes of a maximum-sized entry.

[<46> Section 3.1.5.2.2](#Appendix_A_Target_46): Windows does not validate the input, though the result of malformed information merely results in inconsistent output to the client.

[<47> Section 3.1.5.2.2](#Appendix_A_Target_47): Windows estimates the number of entries to return by dividing *PreferedMaximumLength* by the number of bytes of a maximum-sized entry.

[<48> Section 3.1.5.3](#Appendix_A_Target_48): Non-DC configurations do not cache implementation-specific enumeration state on the domain handle; DC configurations do.

[<49> Section 3.1.5.3.1](#Appendix_A_Target_49): This value is estimated and is not accurate. Windows clients do not rely on the accuracy of this value.

[<50> Section 3.1.5.3.1](#Appendix_A_Target_50): On a non-DC configuration, *Index* is a per-element monotonically increasing number. If *Index* (the message parameter) is 0, the start value is 0; otherwise, the start value is one greater than *Index* (the message parameter).

On a DC, this value is an implementation-specific value that satisfies the requirement shown earlier.

[<51> Section 3.1.5.4.4](#Appendix_A_Target_51): The test for an explicit DENY [**ACE**](#gt_b581857f-39aa-4979-876b-daba67a40f15) is NOT performed in Windows 2000. This test is also NOT performed in the following products that do not have [MSKB-3072595] installed: Windows Server 2003, Windows Server 2003 R2, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2.

[<52> Section 3.1.5.4.4](#Appendix_A_Target_52): This behavior is NOT performed in Windows 2000, and is also NOT performed in the following products that do not have [MSKB-3072595] installed: Windows Server 2003, Windows Server 2003 R2, Windows Server 2008, Windows Server 2008 R2, Windows Server 2012, and Windows Server 2012 R2. In these cases, the server behaves as if CallerPrimaryGroup is NOT equal to DOMAIN\_GROUP\_RID\_COMPUTERS.

[<53> Section 3.1.5.5.1.1](#Appendix_A_Target_53): On non-DC configurations, the exact value is returned. On DC configurations, Windows estimates this count with no guarantees as to accuracy.

[<54> Section 3.1.5.5.1.1](#Appendix_A_Target_54): On non-DC configurations, the exact value is returned. On DC configurations, Windows estimates this count with no guarantees as to accuracy.

[<55> Section 3.1.5.5.1.1](#Appendix_A_Target_55): On non-DC configurations, the exact value is returned. On DC configurations, Windows estimates this count with no guarantees as to accuracy.

[<56> Section 3.1.5.7.1](#Appendix_A_Target_56): Applicable Windows Server releases return error STATUS\_DS\_BUSY (0xc00002a5).

[<57> Section 3.1.5.7.2](#Appendix_A_Target_57): Applicable Windows Server releases return error STATUS\_DS\_BUSY (0xc00002a5).

[<58> Section 3.1.5.7.3](#Appendix_A_Target_58): Applicable Windows Server releases return error STATUS\_DS\_BUSY (0xc00002a5).

[<59> Section 3.1.5.8.3](#Appendix_A_Target_59): Servers running Windows 2000 Server, Windows Server 2003, Windows Server 2003 R2, and Windows Server 2008 do not check whether the [**domain prefixes**](#gt_9066e9dc-8275-4452-9073-daab5fd427c5) of **objectSid** attributes from objects in M and G match.

[<60> Section 3.1.5.10.2](#Appendix_A_Target_60): Windows implementations of the protocol server ignore the *ServerName* parameter.

[<61> Section 3.1.5.10.3](#Appendix_A_Target_61): Windows implementations of the protocol server ignore the *ServerName* parameter.

[<62> Section 3.1.5.10.4](#Appendix_A_Target_62): This parameter MAY be ignored by the server.

[<63> Section 3.1.5.12.1.1](#Appendix_A_Target_63): If USER\_CHANGE\_PASSWORD is not granted to World on receipt, Windows adds the following (deny) ACEs to the **ntSecurityDescriptor** value.

| Field name | Value |
| --- | --- |
| Ace Type | ACCESS\_DENIED\_OBJECT\_ACE\_TYPE |
| SID | PRINCIPAL\_SELF\_SID |
| Access Mask | ACTRL\_DS\_CONTROL\_ACCESS |
| ObjectGuid | ab721a53-1e2f-11d0-9819-00aa0040529b |

| Field name | Value |
| --- | --- |
| Ace Type | ACCESS\_DENIED\_OBJECT\_ACE\_TYPE |
| SID | World |
| Access Mask | ACTRL\_DS\_CONTROL\_ACCESS |
| ObjectGuid | ab721a53-1e2f-11d0-9819-00aa0040529b |

If USER\_CHANGE\_PASSWORD is granted to Self or World on receipt, Windows removes the above two ACEs (if present) and adds the following two ACEs, if not already present.

| Field name | Value |
| --- | --- |
| Ace Type | ACCESS\_ALLOWED\_OBJECT\_ACE\_TYPE |
| SID | Self |
| Access Mask | ACTRL\_DS\_CONTROL\_ACCESS |
| ObjectGuid | ab721a53-1e2f-11d0-9819-00aa0040529b |

| Field name | Value |
| --- | --- |
| Ace Type | ACCESS\_ALLOWED\_OBJECT\_ACE\_TYPE |
| SID | World |
| Access Mask | ACTRL\_DS\_CONTROL\_ACCESS |
| ObjectGuid | ab721a53-1e2f-11d0-9819-00aa0040529b |

[<64> Section 3.1.5.13.4](#Appendix_A_Target_64): Windows clients set this value to be the null-terminated NETBIOS name of the server.

[<65> Section 3.1.5.13.6](#Appendix_A_Target_65): Windows 2000 Server and later enforce that the *UserId* parameter is 0x1F4.

[<66> Section 3.1.5.13.6](#Appendix_A_Target_66): Windows does not decrypt the value but stores the encrypted value directly in an implementation-specific store.

[<67> Section 3.1.5.13.7.1](#Appendix_A_Target_67): Windows Server 2003, Windows Server 2003 R2, and Windows Server 2008 test the PasswordLastSet conditions (constraints 5 and 6) immediately after testing the LockoutTime conditions (constraints 1 and 2).

[<68> Section 3.1.5.13.7.2](#Appendix_A_Target_68): Starting with Windows 2000 Server, if there is a custom password filter installed, and that password filter fails to validate the password, Windows implementations of the protocol server set ValidationStatus to SamValidatePasswordFilterError.

[<69> Section 3.1.5.13.7.3](#Appendix_A_Target_69): Starting with Windows 2000 Server, if there is a custom password filter installed, and that password filter fails to validate the password, Windows implementations of the protocol server set ValidationStatus to SamValidatePasswordFilterError.

[<70> Section 3.1.5.14.1](#Appendix_A_Target_70): Windows uses the sAMAccountName attribute unless the sAMAccountName attribute contains characters that are not allowed for an [**RDN**](#gt_22198321-b40b-4c24-b8a2-29e44d9d92b9) (RDN syntax is specified in [MS-ADTS] section 3.1.1.1.4), in which case the objectSid is used (in string form). If the sAMAccountName is not a unique RDN for the given container, the server returns STATUS\_USER\_EXISTS to the client.

[<71> Section 3.1.5.14.7](#Appendix_A_Target_71): Windows clients do not set this field.

# Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

* A document revision that incorporates changes to interoperability requirements.
* A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact [dochelp@microsoft.com](mailto:dochelp@microsoft.com).

| Section | Description | Revision class |
| --- | --- | --- |
| [1.2.1](#Section_8626acfed4f3409b89763ae772b5120b) Normative References | Updated Normative References to add [AES-CBC] in support of the Advanced Encryption Standard in Cipher Block Chaining mode in this protocol. | Major |
| [2.1](#Section_084da2e70ba044fc8f17e8a200c69eb5) Transport | Added UserInternal7Information and UserInternal8Information to the list of UserInformationClass values required for the SamrSetInformationUser and SamrSetInformationUser2 methods that RPC clients must use with RPC over SMB. | Major |
| [2.2.1.18](#Section_6dd243f4ce4f471fa5166277ab511b19) AEAD-AES-256-CBC-HMAC-SHA512 Constants | Created new section to specify the constants used with the AEAD-AES-256-CBC-HMAC-SHA512 cipher for wire encryption of sensitive data. | Major |
| [2.2.6.29](#Section_9496c26e490b4e76827f2695fc216f35) SAMPR\_USER\_INFO\_BUFFER | Updated this section by adding UserInternal7Information and UserInternal8Information cases and related Internal7 and Internal8 structure types to the SAMPR\_USER\_INFO\_BUFFER union. | Major |
| [2.2.6.30](#Section_fec80c3f10044166a7292a06d0a856c5) SAMPR\_USER\_INTERNAL7\_INFORMATION | Created new section SAMPR\_USER\_INTERNAL7\_INFORMATION to define the SAMPR\_USER\_INTERNAL7\_INFORMATION structure. | Major |
| [2.2.6.31](#Section_1d946f0cc09942f287ec0206d5680c2f) SAMPR\_USER\_INTERNAL8\_INFORMATION | Created new section SAMPR\_USER\_INTERNAL8\_INFORMATION to define the SAMPR\_USER\_INTERNAL8\_INFORMATION structure. | Major |
| [2.2.6.32](#Section_2cbb49a2be3844908be7b0e4951bc221) SAMPR\_ENCRYPTED\_PASSWORD\_AES | Created new section SAMPR\_ENCRYPTED\_PASSWORD\_AES to define the SAMPR\_ENCRYPTED\_PASSWORD\_AES structure. | Major |
| [2.2.7.5](#Section_1d40622e52e44aaabc77aa626089f116) SAMPR\_PSID\_ARRAY | Changed section title to 'SAMPR\_PSID\_ARRAY' from 'SAMPR\_SID\_INFORMATION' and added new content to define the SAMPR\_PSID\_ARRAY structure. Moved former content to section 2.2.7.6, which is retitled 'SAMPR\_SID\_INFORMATION'. | Major |
| [2.2.7.6](#Section_0c138399f6634039b4e7b3c9f82bff65) SAMPR\_SID\_INFORMATION | Created new section 2.2.7.6 SAMPR\_SID\_INFORMATION to hold the former content of section 2.2.7.5. | Major |
| [2.2.7.15](#Section_963e60b592334669b8a885bf4f0806dc) SAMPR\_REVISION\_INFO\_V1 | Updated the SupportedFeatures field table to specify the received value that signifies when the client should use the SAMPR\_ENCRYPTED\_PASSWORD\_AES structure to encrypt password buffers when sent over the wire. | Major |
| [3.1.1.11](#Section_1ed7ed5bd7b5444f86b8ebe8bad1403a) Server Access Control List | 11131 : Created new section 'Server Access Control List' to specify the manner in which the server access control list authorizes clients to make calls to the server. | Major |
| [3.1.2](#Section_5e3f7f39df374f9cbb99a27d79cc957f) Security Model | 11131 : Updated by adding content clarifying the two layers that are the basis of the Security Model. | Major |
| [3.1.2.1](#Section_aaa1fc5d851d4f0096890b9950e88e7b) Server-wide Access Check | 11131 : Created new section to specify the manner in which the server performs an authorization check at the beginning of each call before method-specific processing begins. The scope of this applicability is described in a newly added product behavior note. | Major |
| [3.1.5](#Section_3f15814e46004647abfd3890f5f3570c) Message Processing Events and Sequencing Rules | Added OpNum methods to method table that are reserved for local use: Opnum70NotUsedOnWire, Opnum71NotUsedOnWire, and Opnum72NotUsedOnWire; in addition to SamrUnicodeChangePasswordUser4. The latter is also added to the list of methods that do not require a context handle and can be called directly. | Major |
| [3.1.5.6.4](#Section_99ee9f3943e84bbaac3a82e0c0e0699e) SamrSetInformationUser2 (Opnum 58) | Updated section 3.1.5.6.4 SamrSetInformationUser2 to add UserInternal8Information (3.1.5.6.4.6) to the constraints table of step 3. | Major |
| [3.1.5.6.4.1](#Section_c00ca1dd5a294655bd6f6e680e154bc9) Common Processing | Added constraints of new step 4, added a new table of field settings, and specified the processing instructions required when the UserInformationClass is set to the UserInternal7Information value. | Major |
| [3.1.5.6.4.6](#Section_ce4e6060629d4bedad7ec464f4b989f8) UserInternal8Information | Created new section to specify the constraints to which the server must process the UserInternal8Information message. | Major |
| [3.1.5.10.4](#Section_bbc1c5e59b814038b2b9c87d3569ed38) SamrUnicodeChangePasswordUser4 (Opnum 73) | Created new section to define the SamrUnicodeChangePasswordUser4 method and the processing instructions for changing a user account password. | Major |
| [3.1.5.13.4](#Section_1a8841487e904a93b4c47a0006f59026) SamrGetDomainPasswordInformation (Opnum 56) | 11131 : Updated the description of the out parameter 'PasswordInformation' for the SamrGetDomainPasswordInformation method to clarify the scope of security checks. | Major |
| [3.2.2.4](#Section_0fc25352662b40e79deef10c6244c4c9) AES Cipher Usage | Created new section to describe how to achieve AES encryption by using the AEAD-AES-256-CBC-HMAC-SHA512 cipher. | Major |
| [3.2.2.5](#Section_c37085ecc4564a1cad98a4c2b39a35fd) Deriving an Encryption Key from a Plaintext Password | Created new section 'Deriving an Encryption Key from a Plaintext Password' to explain how to derive an encryption key from a plaintext password. | Major |
| [5.1](#Section_801e68e3b26d4a65b95becce817becd9) Security Considerations for Implementers | Updated to specify that using UserInternal8Information and UserInternal7Information are the best choices for setting a cleartext password with the SamrSetInformationUser2 method in this protocol. | Major |
| [6](#Section_1cd138b9cc1b4706b11549e53189e32e) Appendix A: Full IDL | Updated the IDL to reflect the new and updated structures and messages for this revision. | Major |

# Index

A

Abstract data model

[client](#section_d31e5664a6af4a9698522a1c30eb444f) 229

[server](#section_814e12f610374b648d288f1b899dc57f) 99

[Access - default](#section_82fbaa9eefc64a66a51d489878be3f0a) 132

Access checks

[Active Directory in DC configuration](#section_95e09531b9af4e3b9e9cc9720f7c5dc8) 131

[standard handle-based](#section_87bacbd07b8b429fabc64b3d895d4e90) 126

[ACCESS\_SYSTEM\_SECURITY](#section_15b9ebf7161d4c83a672dceb2ac8c448) 29

[Accounts - default](#section_2c0c5129c259452f81cf792f1ba2c265) 132

[ACTRL\_DS\_CONTROL\_ACCESS](#section_508fad1c4e924806930dbda2e71ba7ec) 42

[ACTRL\_DS\_DELETE\_TREE](#section_508fad1c4e924806930dbda2e71ba7ec) 42

[ACTRL\_DS\_LIST](#section_508fad1c4e924806930dbda2e71ba7ec) 42

[ACTRL\_DS\_READ\_PROP](#section_508fad1c4e924806930dbda2e71ba7ec) 42

[ACTRL\_DS\_WRITE\_PROP](#section_508fad1c4e924806930dbda2e71ba7ec) 42

Algorithms

[DES-ECB-LM](#section_3f5ec79db4494ab29423c4dccbe0b184) 96

[Kerberos encryption](#section_1355fa6bd0974ecc8d5e75b3a6533e04) 96

Alias

[creating](#section_8df397babdda4f638f004502375f9cf6) 161

[fields](#section_e60b8ed7aeb64627a19a7ac37a74bc10) 54

[query/set data types](#section_67e7d6a2ec3642e5b6c30903d7611f22) 54

[ALIAS\_ADD\_MEMBER](#section_2da21c6c5b1546c8bd4e6a8443216e1a) 32

[ALIAS\_ALL\_ACCESS](#section_2da21c6c5b1546c8bd4e6a8443216e1a) 32

[ALIAS\_EXECUTE](#section_2da21c6c5b1546c8bd4e6a8443216e1a) 32

[ALIAS\_INFORMATION\_CLASS enumeration](#section_c57a00edd49243e79be56170e5d89150) 55

[ALIAS\_LIST\_MEMBERS](#section_2da21c6c5b1546c8bd4e6a8443216e1a) 32

[ALIAS\_READ](#section_2da21c6c5b1546c8bd4e6a8443216e1a) 32

[ALIAS\_READ\_INFORMATION](#section_2da21c6c5b1546c8bd4e6a8443216e1a) 32

[ALIAS\_REMOVE\_MEMBER](#section_2da21c6c5b1546c8bd4e6a8443216e1a) 32

[ALIAS\_WRITE](#section_2da21c6c5b1546c8bd4e6a8443216e1a) 32

[ALIAS\_WRITE\_ACCOUNT](#section_2da21c6c5b1546c8bd4e6a8443216e1a) 32

[Applicability](#section_2e160098baca419caf17fd515c3d110f) 26

Attributes

[constraints](#section_e270cd0a529541be9e892c3dc3a39536) 104

[listing](#section_2a79a0a80d564b5bbee5c8df393cde57) 101

[password settings](#section_6da47028e678475eb015a5a15ee98212) 103

[triggers for originating updates](#section_90f7a608d9f14831b2feb9a50cf26aec) 112

B

[Basic data types](#section_414e52d7fa31486d8f38234f661ad157) 42

C

[Capability negotiation](#section_4e7d249a98c04c44ab177c63b67d337e) 27

[Change password pattern](#section_41d7ca60909f4d0db85ac9a35b5f2aaa) 194

[Change Password Pattern method](#section_41d7ca60909f4d0db85ac9a35b5f2aaa) 194

[Change tracking](#section_fc6206af03894af584bf28713d70b17c) 275

[Classes - object class list](#section_a4ebabe2e00e49e7b8315a5eb345eb94) 103

Client

[abstract data model](#section_d31e5664a6af4a9698522a1c30eb444f) 229

[initialization](#section_91cddc62b6fb4a65a76e9fb1018e5a0b) 231

[local events](#section_77e451fa33c4470496f7191499c97f75) 231

[message processing](#section_4d35b12bbdbe42afa6e946c85ee8d1d1) 231

[security model](#section_a533f08da1664274a6e8e59125afa433) 230

[sequencing rules](#section_4d35b12bbdbe42afa6e946c85ee8d1d1) 231

[timer events](#section_c0a840d832ce4eb98bc4c33bb39be76e) 231

[timers](#section_2e602eb4fafb47f28bee64da82b060f2) 231

[Common data types](#section_1639f1a09cb74e4093548dd7dfa2b4a3) 29

[Constant value definitions](#section_3dd38771546240d285ae641d3cdf7383) 29

[Constraints - attributes](#section_e270cd0a529541be9e892c3dc3a39536) 104

[Create pattern](#section_2214fd297c2d450fb68d6a0e97ebe48f) 161

[Create Pattern method](#section_2214fd297c2d450fb68d6a0e97ebe48f) 161

[Creating a user account example](#section_3d8e23d8d9df481f83b39175f980294c) 232

[Creating user account example](#section_3d8e23d8d9df481f83b39175f980294c) 232

[Credentials structures - supplemental](#section_84cefe3ea6884232b997ac5d9993f5eb) 86

D

Data model

abstract

[client](#section_d31e5664a6af4a9698522a1c30eb444f) 229

[server](#section_814e12f610374b648d288f1b899dc57f) 99

[SamContextHandle](#section_cc2dc6cee45f481aa8ed5d0e273336b3) 125

Data model - abstract

[client](#section_d31e5664a6af4a9698522a1c30eb444f) 229

[server](#section_814e12f610374b648d288f1b899dc57f) 99

Data types

[alias query/set](#section_67e7d6a2ec3642e5b6c30903d7611f22) 54

[basic](#section_414e52d7fa31486d8f38234f661ad157) 42

[common - overview](#section_1639f1a09cb74e4093548dd7dfa2b4a3) 29

[constant value definitions](#section_3dd38771546240d285ae641d3cdf7383) 29

[domain query/set](#section_a485032f403b4272929696916bd4f254) 44

[group query/set](#section_d869319f27c14913ac88d3731c1e743b) 51

[protocol-specific types](#section_7c7a5edf2d5f4553b6020f9ade89f20d) 71

[SamrValidatePassword](#section_218e3a7a042f4c8097ce3d46c4efa3fe) 81

[selective enumerate associated structures](#section_6e1afbe517b34e37b409a996f4e02c6b) 76

[supplemental credentials structures](#section_84cefe3ea6884232b997ac5d9993f5eb) 86

[user query/set](#section_b7c69df54fe848658d265340c023f91e) 56

[Default access](#section_82fbaa9eefc64a66a51d489878be3f0a) 132

[Default accounts](#section_2c0c5129c259452f81cf792f1ba2c265) 132

[DELETE](#section_15b9ebf7161d4c83a672dceb2ac8c448) 29

[Delete pattern](#section_7cf72c9694b545a39ae34bff96541946) 185

[Delete Pattern method](#section_7cf72c9694b545a39ae34bff96541946) 185

[Directory service schema elements](#section_6c5ed06fba804980a6469cca57266a91) 98

Domain

[fields](#section_d275ab1910b040e094bb45b7fc130025) 44

[query/set data types](#section_a485032f403b4272929696916bd4f254) 44

[DOMAIN\_ADMINISTER\_SERVER](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

[DOMAIN\_ALIAS\_RID\_ADMINS](#section_565a658430614edea531f5c53826504b) 40

[DOMAIN\_ALL\_ACCESS](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

[DOMAIN\_CREATE\_ALIAS](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

[DOMAIN\_CREATE\_GROUP](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

[DOMAIN\_CREATE\_USER](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

[DOMAIN\_DISPLAY\_INFORMATION enumeration](#section_929775e003394f0b81616d3699c18185) 80

[DOMAIN\_EXECUTE](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

[DOMAIN\_GET\_ALIAS\_MEMBERSHIP](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

[DOMAIN\_GROUP\_RID\_COMPUTERS](#section_565a658430614edea531f5c53826504b) 40

[DOMAIN\_GROUP\_RID\_CONTROLLERS](#section_565a658430614edea531f5c53826504b) 40

[DOMAIN\_GROUP\_RID\_READONLY\_CONTROLLERS](#section_565a658430614edea531f5c53826504b) 40

[DOMAIN\_GROUP\_RID\_USERS](#section_565a658430614edea531f5c53826504b) 40

[DOMAIN\_INFORMATION\_CLASS enumeration](#section_3e8738b25df6499f907dac2471bf0281) 50

[DOMAIN\_LIST\_ACCOUNTS](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

[DOMAIN\_LOGOFF\_INFORMATION structure](#section_6fb0bbea888c4353b5f875e7862344be) 47

[DOMAIN\_LOOKUP](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

[DOMAIN\_MODIFIED\_INFORMATION structure](#section_e1da96808968423b98c0fbdcf1535ef9) 47

[DOMAIN\_MODIFIED\_INFORMATION2 structure](#section_47eea81b5fee4925b5c1fc594dcc8dff) 48

[DOMAIN\_PASSWORD\_INFORMATION structure](#section_0ae356d8c2204706846eebbdc6fabdcb) 47

[DOMAIN\_READ](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

[DOMAIN\_READ\_OTHER\_PARAMETERS](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

[DOMAIN\_READ\_PASSWORD\_PARAMETERS](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

[DOMAIN\_SERVER\_ENABLE\_STATE enumeration](#section_5b1a45c9937e4129bec4593f64c56d48) 46

[DOMAIN\_SERVER\_ROLE enumeration](#section_abb07f2eb9e648ec91187caeb03cca35) 47

[DOMAIN\_SERVER\_ROLE\_INFORMATION structure](#section_cb0e586a29c849b28cedc273a7476c22) 47

[DOMAIN\_STATE\_INFORMATION structure](#section_f224edcf8d4e4294b0c3b0eda384c402) 46

[DOMAIN\_USER\_RID\_ADMIN](#section_565a658430614edea531f5c53826504b) 40

[DOMAIN\_USER\_RID\_GUEST](#section_565a658430614edea531f5c53826504b) 40

[DOMAIN\_USER\_RID\_KRBTGT](#section_565a658430614edea531f5c53826504b) 40

[DOMAIN\_WRITE](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

[DOMAIN\_WRITE\_OTHER\_PARAMETERS](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

[DOMAIN\_WRITE\_PASSWORD\_PARAMS](#section_aef23495f6aa48e9aebc22e022a2b4eb) 30

E

[Elements - directory service schema](#section_6c5ed06fba804980a6469cca57266a91) 98

[Enabling a user account example](#section_bf8cfb7624f742dea95fe5b9ec7435d0) 234

[Enabling user account example](#section_bf8cfb7624f742dea95fe5b9ec7435d0) 234

[ENCRYPTED\_LM\_OWF\_PASSWORD structure](#section_ce061fef6d4f4802bd5d26b11f14f4a6) 71

[ENCRYPTED\_NT\_OWF\_PASSWORD](#section_ce061fef6d4f4802bd5d26b11f14f4a6) 71

[Encrypting an nt or lm hash example](#section_8c5c143818074d199f4e66290f214a63) 236

[Encrypting NT or LM hash example](#section_8c5c143818074d199f4e66290f214a63) 236

[Enumerate pattern](#section_a14d7cf8e908468497879d6c4492957a) 151

[Enumerate Pattern method](#section_a14d7cf8e908468497879d6c4492957a) 151

Events

[local - client](#section_77e451fa33c4470496f7191499c97f75) 231

[timer - client](#section_c0a840d832ce4eb98bc4c33bb39be76e) 231

[timer - server](#section_cedb6f40179d408293e3e55e6c4facb5) 228

Examples

[creating a user account](#section_3d8e23d8d9df481f83b39175f980294c) 232

[creating user account example](#section_3d8e23d8d9df481f83b39175f980294c) 232

[enabling a user account](#section_bf8cfb7624f742dea95fe5b9ec7435d0) 234

[enabling user account example](#section_bf8cfb7624f742dea95fe5b9ec7435d0) 234

[encrypting an nt or lm hash](#section_8c5c143818074d199f4e66290f214a63) 236

[encrypting NT or LM hash](#section_8c5c143818074d199f4e66290f214a63) 236

F

Fields

[alias](#section_e60b8ed7aeb64627a19a7ac37a74bc10) 54

[domain](#section_d275ab1910b040e094bb45b7fc130025) 44

[group](#section_2c93edfeae774f49b849b6435ff63444) 52

[selective enumerate](#section_4e123dbfe1014ade9ffff2247ed25a11) 76

[user](#section_899d68a981454334a52f2fbd92bec94f) 56

[vendor-extensible](#section_574a4d8666ff448cad6ff81e70b795c8) 27

[Fields - vendor-extensible](#section_574a4d8666ff448cad6ff81e70b795c8) 27

[Full IDL](#section_1cd138b9cc1b4706b11549e53189e32e) 240

G

[GENERIC\_ALL](#section_262970b7cd4a41f48c4d5a27f0092aaa) 29

[GENERIC\_EXECUTE](#section_262970b7cd4a41f48c4d5a27f0092aaa) 29

[GENERIC\_READ](#section_262970b7cd4a41f48c4d5a27f0092aaa) 29

[GENERIC\_WRITE](#section_262970b7cd4a41f48c4d5a27f0092aaa) 29

[Glossary](#section_7b2aeb2792fc41f68437deb65d950921) 11

Group

[creating](#section_8df397babdda4f638f004502375f9cf6) 161

[fields](#section_2c93edfeae774f49b849b6435ff63444) 52

[query/set data types](#section_d869319f27c14913ac88d3731c1e743b) 51

[GROUP\_ADD\_MEMBER](#section_f24f9fa8798d4e7da110a5eda6900f41) 31

[GROUP\_ALL\_ACCESS](#section_f24f9fa8798d4e7da110a5eda6900f41) 31

[GROUP\_ATTRIBUTE\_INFORMATION structure](#section_cb80061b78014082bbe720d88b118eaa) 52

[GROUP\_EXECUTE](#section_f24f9fa8798d4e7da110a5eda6900f41) 31

[GROUP\_INFORMATION\_CLASS enumeration](#section_46e6300dcb8a402f918701b41b13919a) 53

[GROUP\_LIST\_MEMBERS](#section_f24f9fa8798d4e7da110a5eda6900f41) 31

[GROUP\_MEMBERSHIP structure](#section_dc0d27ac521847099d1bcab6f6d90b10) 74

[GROUP\_READ](#section_f24f9fa8798d4e7da110a5eda6900f41) 31

[GROUP\_READ\_INFORMATION](#section_f24f9fa8798d4e7da110a5eda6900f41) 31

[GROUP\_REMOVE\_MEMBER](#section_f24f9fa8798d4e7da110a5eda6900f41) 31

[GROUP\_TYPE\_ACCOUNT\_GROUP](#section_1f8d7ea1fcc14833839af94d67c08fcd) 37

[GROUP\_TYPE\_RESOURCE\_GROUP](#section_1f8d7ea1fcc14833839af94d67c08fcd) 37

[GROUP\_TYPE\_SECURITY\_ACCOUNT](#section_1f8d7ea1fcc14833839af94d67c08fcd) 37

[GROUP\_TYPE\_SECURITY\_ENABLED](#section_1f8d7ea1fcc14833839af94d67c08fcd) 37

[GROUP\_TYPE\_SECURITY\_RESOURCE](#section_1f8d7ea1fcc14833839af94d67c08fcd) 37

[GROUP\_TYPE\_SECURITY\_UNIVERSAL](#section_1f8d7ea1fcc14833839af94d67c08fcd) 37

[GROUP\_TYPE\_UNIVERSAL\_GROUP](#section_1f8d7ea1fcc14833839af94d67c08fcd) 37

[GROUP\_WRITE](#section_f24f9fa8798d4e7da110a5eda6900f41) 31

[GROUP\_WRITE\_ACCOUNT](#section_f24f9fa8798d4e7da110a5eda6900f41) 31

H

[Handle-based access checks](#section_87bacbd07b8b429fabc64b3d895d4e90) 126

[Handling strings](#section_5c4b2e83044349318cef011d05257a61) 100

I

[IDL](#section_1cd138b9cc1b4706b11549e53189e32e) 240

[Implementer - security considerations](#section_801e68e3b26d4a65b95becce817becd9) 239

[Index of security parameters](#section_d9bd6ce0787948d0ab29f4cf3a84e99e) 239

[Information levels - methods](#section_978d8841a24f40d1947f4b27e399e50e) 27

[Informative references](#section_d39954f3a9b24c0e92eebf99a7676872) 17

Initialization

[client](#section_91cddc62b6fb4a65a76e9fb1018e5a0b) 231

[server](#section_c6c15f12dc63441f865bf78c552e0c73) 131

[Introduction](#section_969524111d174fe4879cd5b48a264314) 11

K

[KERB\_KEY\_DATA packet](#section_9921c3edf4704f369edf7afe24c468ec) 92

[KERB\_KEY\_DATA\_NEW packet](#section_447520a5e1cc48cc8fdcb90db57f7eac) 95

[KERB\_STORED\_CREDENTIAL packet](#section_866f018086944e14be7edfd91d91694f) 91

[KERB\_STORED\_CREDENTIAL\_NEW packet](#section_08cb3ca7954b45e3902e77512fe3ba8e) 93

[Kerberos encryption algorithm identifiers](#section_1355fa6bd0974ecc8d5e75b3a6533e04) 96

L

[Listing attributes](#section_2a79a0a80d564b5bbee5c8df393cde57) 101

[LM hash - encrypting - example](#section_8c5c143818074d199f4e66290f214a63) 236

Local events

[client](#section_77e451fa33c4470496f7191499c97f75) 231

server

[domain join processing](#section_a2cb29f5f0c54a7886b06ba9029a1bd7) 229

[domain unjoin processing](#section_ea1feef87b4244c38446646fc6bfda01) 229

[Lookup pattern](#section_ae07443cf5864fd7b0b361cf300d06b7) 202

[Lookup Pattern method](#section_ae07443cf5864fd7b0b361cf300d06b7) 202

M

[Matching strings](#section_3e92c4e9cc5440bf9627994dcee92857) 100

[MAXIMUM\_ALLOWED](#section_15b9ebf7161d4c83a672dceb2ac8c448) 29

[MD5 usage](#section_d1310f9523af4c2d92c0aa5166da8450) 230

[Membership pattern](#section_e8205d2c9ebb4845b9270aca7cbc1f2c) 187

[Membership Pattern method](#section_e8205d2c9ebb4845b9270aca7cbc1f2c) 187

[Membership-of pattern](#section_95a94bcc822c48be81c39697917b3633) 193

[Membership-Of Pattern method](#section_95a94bcc822c48be81c39697917b3633) 193

Message processing

[client](#section_4d35b12bbdbe42afa6e946c85ee8d1d1) 231

[server](#section_3f15814e46004647abfd3890f5f3570c) 134

[supplemental - server](#section_98bef3a3cf604854a933274f21dd59a8) 223

Messages

[common data types](#section_1639f1a09cb74e4093548dd7dfa2b4a3) 29

[data types](#section_1639f1a09cb74e4093548dd7dfa2b4a3) 29

[transport](#section_084da2e70ba044fc8f17e8a200c69eb5) 28

[Method-based perspective](#section_d7b625964a46455692dc3aba6d517907) 21

Methods

[Change Password Pattern](#section_41d7ca60909f4d0db85ac9a35b5f2aaa) 194

[Create Pattern](#section_2214fd297c2d450fb68d6a0e97ebe48f) 161

[Delete Pattern](#section_7cf72c9694b545a39ae34bff96541946) 185

[Enumerate Pattern](#section_a14d7cf8e908468497879d6c4492957a) 151

[information levels](#section_978d8841a24f40d1947f4b27e399e50e) 27

[Lookup Pattern](#section_ae07443cf5864fd7b0b361cf300d06b7) 202

[Membership Pattern](#section_e8205d2c9ebb4845b9270aca7cbc1f2c) 187

[Membership-Of Pattern](#section_95a94bcc822c48be81c39697917b3633) 193

[Miscellaneous](#section_8fd836b683f64d14a138ef532ffbec4e) 214

[Open Pattern](#section_6d92e4991d164c3596e6c564dfd2972b) 140

[overview](#section_6b137f7899074cdbaab6764e8fb62db4) 27

[Query Pattern](#section_97c29e0795d14273b7dfe8fe4e0f7592) 167

[Security Pattern](#section_34a12061d0ea4f0eb8f536107794062a) 206

[Selective Enumerate Pattern](#section_f1970f00e3d54ec485bc2caf0bf18f6d) 156

[Set Pattern](#section_8e86fe5140e2489992e9676466d4e6d3) 175

[versioning](#section_b1e11eccf41a40aca38b4b894e119cb1) 27

[Miscellaneous method](#section_8fd836b683f64d14a138ef532ffbec4e) 214

[Miscellaneous patterns](#section_8fd836b683f64d14a138ef532ffbec4e) 214

N

[Normative references](#section_8626acfed4f3409b89763ae772b5120b) 15

[NT hash - encrypting - example](#section_8c5c143818074d199f4e66290f214a63) 236

O

[Object class list](#section_a4ebabe2e00e49e7b8315a5eb345eb94) 103

[Object-based perspective](#section_8aaff2f71edd41a0ab584807ac6124c5) 18

[OLD\_LARGE\_INTEGER structure](#section_68d6dc19bedf4a90b939b2e4e04b7cf6) 43

[Open pattern](#section_6d92e4991d164c3596e6c564dfd2972b) 140

[Open Pattern method](#section_6d92e4991d164c3596e6c564dfd2972b) 140

[Overview (synopsis)](#section_d036091033dd4e3cb5a226daf95f5ffa) 18

P

[Parameter index - security](#section_d9bd6ce0787948d0ab29f4cf3a84e99e) 239

[Parameters - security index](#section_d9bd6ce0787948d0ab29f4cf3a84e99e) 239

[Password settings - attributes](#section_6da47028e678475eb015a5a15ee98212) 103

[PASSWORD\_POLICY\_VALIDATION\_TYPE enumeration](#section_8bef882e1e2c4e18a91f75191d70b39c) 85

Pattern

[change password](#section_41d7ca60909f4d0db85ac9a35b5f2aaa) 194

[create](#section_2214fd297c2d450fb68d6a0e97ebe48f) 161

[delete](#section_7cf72c9694b545a39ae34bff96541946) 185

[enumerate](#section_a14d7cf8e908468497879d6c4492957a) 151

[lookup](#section_ae07443cf5864fd7b0b361cf300d06b7) 202

[membership](#section_e8205d2c9ebb4845b9270aca7cbc1f2c) 187

[membership-of](#section_95a94bcc822c48be81c39697917b3633) 193

[miscellaneous](#section_8fd836b683f64d14a138ef532ffbec4e) 214

[open](#section_6d92e4991d164c3596e6c564dfd2972b) 140

[query](#section_97c29e0795d14273b7dfe8fe4e0f7592) 167

[security](#section_34a12061d0ea4f0eb8f536107794062a) 206

[selective enumerate](#section_f1970f00e3d54ec485bc2caf0bf18f6d) 156

[set](#section_8e86fe5140e2489992e9676466d4e6d3) 175

[PDOMAIN\_LOGOFF\_INFORMATION](#section_6fb0bbea888c4353b5f875e7862344be) 47

[PDOMAIN\_MODIFIED\_INFORMATION](#section_e1da96808968423b98c0fbdcf1535ef9) 47

[PDOMAIN\_MODIFIED\_INFORMATION2](#section_47eea81b5fee4925b5c1fc594dcc8dff) 48

[PDOMAIN\_PASSWORD\_INFORMATION](#section_0ae356d8c2204706846eebbdc6fabdcb) 47

[PDOMAIN\_SERVER\_ROLE\_INFORMATION](#section_cb0e586a29c849b28cedc273a7476c22) 47

[PDOMAIN\_STATE\_INFORMATION](#section_f224edcf8d4e4294b0c3b0eda384c402) 46

[PENCRYPTED\_LM\_OWF\_PASSWORD](#section_ce061fef6d4f4802bd5d26b11f14f4a6) 71

[PENCRYPTED\_NT\_OWF\_PASSWORD](#section_ce061fef6d4f4802bd5d26b11f14f4a6) 71

[PGROUP\_ATTRIBUTE\_INFORMATION](#section_cb80061b78014082bbe720d88b118eaa) 52

[PGROUP\_MEMBERSHIP](#section_dc0d27ac521847099d1bcab6f6d90b10) 74

[POLD\_LARGE\_INTEGER](#section_68d6dc19bedf4a90b939b2e4e04b7cf6) 43

[Preconditions](#section_1080b153835e44f0966daa9ad474e203) 26

[Prerequisites](#section_1080b153835e44f0966daa9ad474e203) 26

[Processing for group and alias creation](#section_8df397babdda4f638f004502375f9cf6) 161

[Product behavior](#section_fa61e5fcf8fb4d5b9695c724af0c3829) 261

[Protocol-specific data types](#section_7c7a5edf2d5f4553b6020f9ade89f20d) 71

[PRPC\_SHORT\_BLOB](#section_77dbfdbb66274871ab125333929347dc) 44

[PRPC\_STRING](#section_28330caf78c449e091c146bbfeed633b) 43

[PSAM\_VALIDATE\_AUTHENTICATION\_INPUT\_ARG](#section_5a666b2ccd214615819b6dbfe7a2b13f) 83

[PSAM\_VALIDATE\_PASSWORD\_CHANGE\_INPUT\_ARG](#section_c2efbd2771664305a33de2519a1dc2cf) 83

[PSAM\_VALIDATE\_PASSWORD\_HASH](#section_295b336b27c84dfebc18216c13bb6ebc) 81

[PSAM\_VALIDATE\_PASSWORD\_RESET\_INPUT\_ARG](#section_74ba81886b2447d9837e7210f6dd70a1) 84

[PSAM\_VALIDATE\_PERSISTED\_FIELDS](#section_e0b2d21d0b1c4fc08f4a895bef6ffc4e) 81

[PSAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG](#section_37717c35b4a44febb7eb7e0da42a511f) 83

[PSAMPR\_ALIAS\_ADM\_COMMENT\_INFORMATION](#section_0968f7a5d87c4fec85c45287a7be6cf4) 55

[PSAMPR\_ALIAS\_GENERAL\_INFORMATION](#section_025890e9c48347e08761632caf30be3f) 54

[PSAMPR\_ALIAS\_NAME\_INFORMATION](#section_e32831e5429e493d8cd2297236f0792c) 55

[PSAMPR\_DOMAIN\_DISPLAY\_GROUP](#section_3dd24be0ed1342f28fba8af4445b4fdc) 77

[PSAMPR\_DOMAIN\_DISPLAY\_GROUP\_BUFFER](#section_ce9c4119c5e446b5a71aec16daf8f457) 79

[PSAMPR\_DOMAIN\_DISPLAY\_MACHINE](#section_353bfb6107304aee848988c249b1a9e1) 77

[PSAMPR\_DOMAIN\_DISPLAY\_MACHINE\_BUFFER](#section_18aa8d70157e482eb2e44e6f4dbb76fe) 78

[PSAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP](#section_5392e27d9f924803b6b70a9f02497443) 78

[PSAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP\_BUFFER](#section_a549f5f16b2b4eef87122d43d94410d2) 79

[PSAMPR\_DOMAIN\_DISPLAY\_OEM\_USER](#section_c8e910bdfeb549a4939e2262fe17ad2b) 78

[PSAMPR\_DOMAIN\_DISPLAY\_OEM\_USER\_BUFFER](#section_0e9f25a9d30540b68b9fa50e58430124) 79

[PSAMPR\_DOMAIN\_DISPLAY\_USER](#section_4f5a1a08c75346ee888b39d360022745) 77

[PSAMPR\_DOMAIN\_DISPLAY\_USER\_BUFFER](#section_453c355ea84d421ebdbcdd7fbfc148a4) 78

[PSAMPR\_DOMAIN\_GENERAL\_INFORMATION](#section_85973e1c96f24c808135b24d74ad7794) 48

[PSAMPR\_DOMAIN\_GENERAL\_INFORMATION2](#section_9a663cf209234959b2c52e25c19735ff) 49

[PSAMPR\_DOMAIN\_LOCKOUT\_INFORMATION](#section_c9d789edc54a4450be56251e627e1f52) 50

[PSAMPR\_DOMAIN\_NAME\_INFORMATION](#section_5131d2c004c74c1b8fd50b0b6cfa6c24) 49

[PSAMPR\_DOMAIN\_OEM\_INFORMATION](#section_7cbb7ff0e593440d8341a3435195cdf1) 49

[PSAMPR\_DOMAIN\_REPLICATION\_INFORMATION](#section_c9293797e11d4098be12bf9e1de91f20) 49

[PSAMPR\_ENCRYPTED\_USER\_PASSWORD](#section_23f9ef4ccf3e43309287ea4799b03201) 63

[PSAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW](#section_112ecc941cbe41cdb669377402c20786) 64

[PSAMPR\_ENUMERATION\_BUFFER](#section_c53161a438e84a28a33e0d378fce03dd) 73

[PSAMPR\_GET\_GROUPS\_BUFFER](#section_31879676cc954cf18f75c09ddcef8750) 74

[PSAMPR\_GET\_MEMBERS\_BUFFER](#section_225147b145b74fdea5bfbf420e18fa08) 74

[PSAMPR\_GROUP\_ADM\_COMMENT\_INFORMATION](#section_84b322dd06214502ac311f50af943abb) 53

[PSAMPR\_GROUP\_GENERAL\_INFORMATION](#section_fe821071d3334eef862af45408343e4e) 52

[PSAMPR\_GROUP\_NAME\_INFORMATION](#section_75b2f39b46fd44d980bda78123dddbdb) 52

[PSAMPR\_LOGON\_HOURS](#section_d83c356b7dda409682705c581f84a4d9) 58

[PSAMPR\_PSID\_ARRAY](#section_0c138399f6634039b4e7b3c9f82bff65) 72

[PSAMPR\_PSID\_ARRAY\_OUT](#section_a44c2dc18a3a4eb89c4d2e9f6eb07f78) 72

[PSAMPR\_RETURNED\_USTRING\_ARRAY](#section_029f91321f574a80906dfb9e5bc0480e) 73

[PSAMPR\_REVISION\_INFO\_V1](#section_963e60b592334669b8a885bf4f0806dc) 75

[PSAMPR\_RID\_ENUMERATION](#section_5c94a35ae7f24675af34741f5a8ee1a2) 73

[PSAMPR\_SID\_INFORMATION](#section_1d40622e52e44aaabc77aa626089f116) 72

[PSAMPR\_SR\_SECURITY\_DESCRIPTOR](#section_675e37d9bb974f14bba2be081c87cd5d) 74

[PSAMPR\_ULONG\_ARRAY](#section_2feb38064db245b790d286c8336a31ba) 72

[PSAMPR\_USER\_A\_NAME\_INFORMATION](#section_89a4133172c74814af3979bdba0d822f) 62

[PSAMPR\_USER\_ACCOUNT\_INFORMATION](#section_7aef58f2474a46abaa2a54f7eb8625e9) 61

[PSAMPR\_USER\_ADMIN\_COMMENT\_INFORMATION](#section_fd4e8928c3704d0cb0fb5c781128cccb) 63

[PSAMPR\_USER\_ALL\_INFORMATION](#section_dc966b81da274daea28cec16534f1cb9) 59

[PSAMPR\_USER\_F\_NAME\_INFORMATION](#section_22f073f55e9340098c9a8569e910f379) 62

[PSAMPR\_USER\_GENERAL\_INFORMATION](#section_d74231bd81e242299e82ce6d3713cc62) 60

[PSAMPR\_USER\_HOME\_INFORMATION](#section_c42b2b9d301d40b7ad4c257f31904602) 62

[PSAMPR\_USER\_INTERNAL1\_INFORMATION](#section_50d17755c6b840bd8cacbd6cfa31adf2) 65

[PSAMPR\_USER\_INTERNAL4\_INFORMATION](#section_e28bf420898944fb8b08f5a7c2f2e33c) 65

[PSAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW](#section_b2f614b90312421aabed10ee002ef780) 66

[PSAMPR\_USER\_INTERNAL5\_INFORMATION](#section_1d2be36a754e46b18697d8aaa62bc450) 66

[PSAMPR\_USER\_INTERNAL5\_INFORMATION\_NEW](#section_563cc0495e4649a1b0e086efe4bc6e19) 66

[PSAMPR\_USER\_LOGON\_HOURS\_INFORMATION](#section_094647addb3647b6956de672b13ec3e6) 63

[PSAMPR\_USER\_LOGON\_INFORMATION](#section_29b54f06896143fd8ecb4b2a8020d474) 61

[PSAMPR\_USER\_NAME\_INFORMATION](#section_400d937e66e544af929d13dfab550d46) 62

[PSAMPR\_USER\_PARAMETERS\_INFORMATION](#section_05f1ac12c90f4aed8821d211a9c834b3) 60

[PSAMPR\_USER\_PREFERENCES\_INFORMATION](#section_551365bdd06d4e0ca72a41567e008f1e) 60

[PSAMPR\_USER\_PROFILE\_INFORMATION](#section_53883066b32941938a9fcbf53927804d) 63

[PSAMPR\_USER\_SCRIPT\_INFORMATION](#section_4e55e798c5874e0cbcfd944e7480dc22) 62

[PSAMPR\_USER\_WORKSTATIONS\_INFORMATION](#section_f991ad1bf9b043ac853b4d900c7618f0) 63

[PUSER\_CONTROL\_INFORMATION](#section_eb5f1508ede14ff1be8255f3e2ef1633) 58

[PUSER\_DOMAIN\_PASSWORD\_INFORMATION](#section_07bd97943db0458db327bbb2d9e799cf) 76

[PUSER\_EXPIRES\_INFORMATION](#section_8c1308e0723e4170817045a6a723989a) 58

[PUSER\_PRIMARY\_GROUP\_INFORMATION](#section_3fd86f83f5ad4bb4814a6554f5797b34) 58

Q

[Query pattern](#section_97c29e0795d14273b7dfe8fe4e0f7592) 167

[Query Pattern method](#section_97c29e0795d14273b7dfe8fe4e0f7592) 167

R

[RC4 cipher usage](#section_5fe3c4c4e71b440db2fd8448bfaf6e04) 230

[READ\_CONTROL](#section_15b9ebf7161d4c83a672dceb2ac8c448) 29

[References](#section_7a3554d46cce41eca95d7d56f53e59e7) 15

[informative](#section_d39954f3a9b24c0e92eebf99a7676872) 17

[normative](#section_8626acfed4f3409b89763ae772b5120b) 15

[Relationship to other protocols](#section_c483654f84ed46af8f4ab2a28cedca5d) 25

[RPC\_S\_PROCNUM\_OUT\_OF\_RANGE](#section_775e581d03f34d2dafa45e9ad3604fee) 41

[RPC\_SHORT\_BLOB structure](#section_77dbfdbb66274871ab125333929347dc) 44

[RPC\_STRING structure](#section_28330caf78c449e091c146bbfeed633b) 43

S

[SAM\_ALIAS\_OBJECT](#section_e742be45665d4576b8720bc99d1e1fbe) 36

[SAM\_APP\_BASIC\_GROUP](#section_e742be45665d4576b8720bc99d1e1fbe) 36

[SAM\_APP\_QUERY\_GROUP](#section_e742be45665d4576b8720bc99d1e1fbe) 36

[SAM\_DOMAIN\_OBJECT](#section_e742be45665d4576b8720bc99d1e1fbe) 36

[SAM\_GROUP\_OBJECT](#section_e742be45665d4576b8720bc99d1e1fbe) 36

[SAM\_MACHINE\_ACCOUNT](#section_e742be45665d4576b8720bc99d1e1fbe) 36

[SAM\_NON\_SECURITY\_ALIAS\_OBJECT](#section_e742be45665d4576b8720bc99d1e1fbe) 36

[SAM\_NON\_SECURITY\_GROUP\_OBJECT](#section_e742be45665d4576b8720bc99d1e1fbe) 36

[SAM\_SERVER\_ALL\_ACCESS](#section_e8afb15ec0534984b84b66877236e141) 30

[SAM\_SERVER\_CONNECT](#section_e8afb15ec0534984b84b66877236e141) 30

[SAM\_SERVER\_CREATE\_DOMAIN](#section_e8afb15ec0534984b84b66877236e141) 30

[SAM\_SERVER\_ENUMERATE\_DOMAINS](#section_e8afb15ec0534984b84b66877236e141) 30

[SAM\_SERVER\_EXECUTE](#section_e8afb15ec0534984b84b66877236e141) 30

[SAM\_SERVER\_INITIALIZE](#section_e8afb15ec0534984b84b66877236e141) 30

[SAM\_SERVER\_LOOKUP\_DOMAIN](#section_e8afb15ec0534984b84b66877236e141) 30

[SAM\_SERVER\_READ](#section_e8afb15ec0534984b84b66877236e141) 30

[SAM\_SERVER\_SHUTDOWN](#section_e8afb15ec0534984b84b66877236e141) 30

[SAM\_SERVER\_WRITE](#section_e8afb15ec0534984b84b66877236e141) 30

[SAM\_TRUST\_ACCOUNT](#section_e742be45665d4576b8720bc99d1e1fbe) 36

[SAM\_USER\_OBJECT](#section_e742be45665d4576b8720bc99d1e1fbe) 36

[SAM\_VALIDATE\_AUTHENTICATION\_INPUT\_ARG structure](#section_5a666b2ccd214615819b6dbfe7a2b13f) 83

[SAM\_VALIDATE\_PASSWORD\_CHANGE\_INPUT\_ARG structure](#section_c2efbd2771664305a33de2519a1dc2cf) 83

[SAM\_VALIDATE\_PASSWORD\_HASH structure](#section_295b336b27c84dfebc18216c13bb6ebc) 81

[SAM\_VALIDATE\_PASSWORD\_RESET\_INPUT\_ARG structure](#section_74ba81886b2447d9837e7210f6dd70a1) 84

[SAM\_VALIDATE\_PERSISTED\_FIELDS structure](#section_e0b2d21d0b1c4fc08f4a895bef6ffc4e) 81

[SAM\_VALIDATE\_STANDARD\_OUTPUT\_ARG structure](#section_37717c35b4a44febb7eb7e0da42a511f) 83

[SAM\_VALIDATE\_VALIDATION\_STATUS enumeration](#section_e0b65a04ca2a458b8e32e5dcb569c919) 82

[SamContextHandle data model](#section_cc2dc6cee45f481aa8ed5d0e273336b3) 125

[SAMPR\_ALIAS\_ADM\_COMMENT\_INFORMATION structure](#section_0968f7a5d87c4fec85c45287a7be6cf4) 55

[SAMPR\_ALIAS\_GENERAL\_INFORMATION structure](#section_025890e9c48347e08761632caf30be3f) 54

[SAMPR\_ALIAS\_NAME\_INFORMATION structure](#section_e32831e5429e493d8cd2297236f0792c) 55

[SAMPR\_DOMAIN\_DISPLAY\_GROUP structure](#section_3dd24be0ed1342f28fba8af4445b4fdc) 77

[SAMPR\_DOMAIN\_DISPLAY\_GROUP\_BUFFER structure](#section_ce9c4119c5e446b5a71aec16daf8f457) 79

[SAMPR\_DOMAIN\_DISPLAY\_MACHINE structure](#section_353bfb6107304aee848988c249b1a9e1) 77

[SAMPR\_DOMAIN\_DISPLAY\_MACHINE\_BUFFER structure](#section_18aa8d70157e482eb2e44e6f4dbb76fe) 78

[SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP structure](#section_5392e27d9f924803b6b70a9f02497443) 78

[SAMPR\_DOMAIN\_DISPLAY\_OEM\_GROUP\_BUFFER structure](#section_a549f5f16b2b4eef87122d43d94410d2) 79

[SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER structure](#section_c8e910bdfeb549a4939e2262fe17ad2b) 78

[SAMPR\_DOMAIN\_DISPLAY\_OEM\_USER\_BUFFER structure](#section_0e9f25a9d30540b68b9fa50e58430124) 79

[SAMPR\_DOMAIN\_DISPLAY\_USER structure](#section_4f5a1a08c75346ee888b39d360022745) 77

[SAMPR\_DOMAIN\_DISPLAY\_USER\_BUFFER structure](#section_453c355ea84d421ebdbcdd7fbfc148a4) 78

[SAMPR\_DOMAIN\_GENERAL\_INFORMATION structure](#section_85973e1c96f24c808135b24d74ad7794) 48

[SAMPR\_DOMAIN\_GENERAL\_INFORMATION2 structure](#section_9a663cf209234959b2c52e25c19735ff) 49

[SAMPR\_DOMAIN\_LOCKOUT\_INFORMATION structure](#section_c9d789edc54a4450be56251e627e1f52) 50

[SAMPR\_DOMAIN\_NAME\_INFORMATION structure](#section_5131d2c004c74c1b8fd50b0b6cfa6c24) 49

[SAMPR\_DOMAIN\_OEM\_INFORMATION structure](#section_7cbb7ff0e593440d8341a3435195cdf1) 49

[SAMPR\_DOMAIN\_REPLICATION\_INFORMATION structure](#section_c9293797e11d4098be12bf9e1de91f20) 49

[SAMPR\_ENCRYPTED\_USER\_PASSWORD structure](#section_23f9ef4ccf3e43309287ea4799b03201) 63

[SAMPR\_ENCRYPTED\_USER\_PASSWORD\_NEW structure](#section_112ecc941cbe41cdb669377402c20786) 64

[SAMPR\_ENUMERATION\_BUFFER structure](#section_c53161a438e84a28a33e0d378fce03dd) 73

[SAMPR\_GET\_GROUPS\_BUFFER structure](#section_31879676cc954cf18f75c09ddcef8750) 74

[SAMPR\_GET\_MEMBERS\_BUFFER structure](#section_225147b145b74fdea5bfbf420e18fa08) 74

[SAMPR\_GROUP\_ADM\_COMMENT\_INFORMATION structure](#section_84b322dd06214502ac311f50af943abb) 53

[SAMPR\_GROUP\_GENERAL\_INFORMATION structure](#section_fe821071d3334eef862af45408343e4e) 52

[SAMPR\_GROUP\_NAME\_INFORMATION structure](#section_75b2f39b46fd44d980bda78123dddbdb) 52

[SAMPR\_LOGON\_HOURS structure](#section_d83c356b7dda409682705c581f84a4d9) 58

[SAMPR\_PSID\_ARRAY structure](#section_0c138399f6634039b4e7b3c9f82bff65) 72

[SAMPR\_PSID\_ARRAY\_OUT structure](#section_a44c2dc18a3a4eb89c4d2e9f6eb07f78) 72

[SAMPR\_RETURNED\_USTRING\_ARRAY structure](#section_029f91321f574a80906dfb9e5bc0480e) 73

[SAMPR\_REVISION\_INFO\_V1 structure](#section_963e60b592334669b8a885bf4f0806dc) 75

[SAMPR\_RID\_ENUMERATION structure](#section_5c94a35ae7f24675af34741f5a8ee1a2) 73

[SAMPR\_SID\_INFORMATION structure](#section_1d40622e52e44aaabc77aa626089f116) 72

[SAMPR\_SR\_SECURITY\_DESCRIPTOR structure](#section_675e37d9bb974f14bba2be081c87cd5d) 74

[SAMPR\_ULONG\_ARRAY structure](#section_2feb38064db245b790d286c8336a31ba) 72

[SAMPR\_USER\_A\_NAME\_INFORMATION structure](#section_89a4133172c74814af3979bdba0d822f) 62

[SAMPR\_USER\_ACCOUNT\_INFORMATION structure](#section_7aef58f2474a46abaa2a54f7eb8625e9) 61

[SAMPR\_USER\_ADMIN\_COMMENT\_INFORMATION structure](#section_fd4e8928c3704d0cb0fb5c781128cccb) 63

[SAMPR\_USER\_ALL\_INFORMATION structure](#section_dc966b81da274daea28cec16534f1cb9) 59

[SAMPR\_USER\_F\_NAME\_INFORMATION structure](#section_22f073f55e9340098c9a8569e910f379) 62

[SAMPR\_USER\_GENERAL\_INFORMATION structure](#section_d74231bd81e242299e82ce6d3713cc62) 60

[SAMPR\_USER\_HOME\_INFORMATION structure](#section_c42b2b9d301d40b7ad4c257f31904602) 62

[SAMPR\_USER\_INTERNAL1\_INFORMATION structure](#section_50d17755c6b840bd8cacbd6cfa31adf2) 65

[SAMPR\_USER\_INTERNAL4\_INFORMATION structure](#section_e28bf420898944fb8b08f5a7c2f2e33c) 65

[SAMPR\_USER\_INTERNAL4\_INFORMATION\_NEW structure](#section_b2f614b90312421aabed10ee002ef780) 66

[SAMPR\_USER\_INTERNAL5\_INFORMATION structure](#section_1d2be36a754e46b18697d8aaa62bc450) 66

[SAMPR\_USER\_INTERNAL5\_INFORMATION\_NEW structure](#section_563cc0495e4649a1b0e086efe4bc6e19) 66

[SAMPR\_USER\_LOGON\_HOURS\_INFORMATION structure](#section_094647addb3647b6956de672b13ec3e6) 63

[SAMPR\_USER\_LOGON\_INFORMATION structure](#section_29b54f06896143fd8ecb4b2a8020d474) 61

[SAMPR\_USER\_NAME\_INFORMATION structure](#section_400d937e66e544af929d13dfab550d46) 62

[SAMPR\_USER\_PARAMETERS\_INFORMATION structure](#section_05f1ac12c90f4aed8821d211a9c834b3) 60

[SAMPR\_USER\_PREFERENCES\_INFORMATION structure](#section_551365bdd06d4e0ca72a41567e008f1e) 60

[SAMPR\_USER\_PROFILE\_INFORMATION structure](#section_53883066b32941938a9fcbf53927804d) 63

[SAMPR\_USER\_SCRIPT\_INFORMATION structure](#section_4e55e798c5874e0cbcfd944e7480dc22) 62

[SAMPR\_USER\_WORKSTATIONS\_INFORMATION structure](#section_f991ad1bf9b043ac853b4d900c7618f0) 63

[SamrAddMemberToAlias method](#section_9a5d2c35e84b4e59b7b096c6fa0fc8d7) 189

[SamrAddMemberToGroup method](#section_3c70fec36a2e48efbd4eaa2b3a1cd96a) 187

[SamrAddMultipleMembersToAlias method](#section_6d981290a203408094b04a26de22d8c5) 192

[SamrChangePasswordUser method](#section_9699d8cae1a4433ca8c3d7bebeb01476) 195

[SamrCloseHandle method](#section_55d134dfe25748ad8afacb2ca45cd3cc) 214

[SamrConnect method](#section_defe20910a614dfabe9a2c1206d53a1f) 143

[SamrConnect2 method](#section_1076eb2a4f514c5aa7c7a78323b06198) 142

[SamrConnect4 method](#section_c1b14e670f4647a69fa70689d284afc9) 142

[SamrConnect5 method](#section_c842a8970a424ca5a6072afd05271dae) 140

[SamrCreateAliasInDomain method](#section_7e505875e44f4f9d922b7ff3871c752d) 163

[SamrCreateGroupInDomain method](#section_175c1cf94fa248379e5bbb1f0f950bee) 162

[SamrCreateUser2InDomain method](#section_a98d7fbb17354fbfb41aef363c899002) 163

[SamrCreateUserInDomain method](#section_cd0efce013fb410687db4bef73bb4b1b) 166

[SamrDeleteAlias method](#section_ecb88df184d148c8b49cd3f444943100) 186

[SamrDeleteGroup method](#section_664ed55f198a4775b9d4398131dba577) 185

[SamrDeleteUser method](#section_4643a57956ec4c66a1ef4ab78dd21d73) 186

[SamrEnumerateAliasesInDomain method](#section_ce340cffedef4356ace033d2874d306b) 154

[SamrEnumerateDomainsInSamServer method](#section_2142fd2d085442c1a9fb2fe964e381ce) 151

[SamrEnumerateGroupsInDomain method](#section_e0b7a4b7ecfc405f9d7d32b3cd2cd6c8) 153

[SamrEnumerateUsersInDomain method](#section_6bdc92c0c6924ffb9de765858b68da75) 155

[SamrGetAliasMembership method](#section_0318404522084c02b38bef955d6dc3ef) 194

[SamrGetDisplayEnumerationIndex method](#section_af95abf82e1142c394aaba72f8f96c09) 161

[SamrGetDisplayEnumerationIndex2 method](#section_bd429624f2d547178aa2659952c3e209) 159

[SamrGetDomainPasswordInformation method](#section_1a8841487e904a93b4c47a0006f59026) 215

[SamrGetGroupsForUser method](#section_a4adbf20040f4416a960e5b7917fdae7) 193

[SamrGetMembersInAlias method](#section_5909af7f858b4cc7a2e4f852070d479d) 191

[SamrGetMembersInGroup method](#section_3ed5030d88a342caa6e08c12aa2fdfbd) 189

[SamrGetUserDomainPasswordInformation method](#section_db4bedfe465147439381d74c3ad8c41c) 215

[SamrLookupDomainInSamServer method](#section_47492d59e0954398b03e8a062b989123) 202

[SamrLookupIdsInDomain method](#section_c870951c74b347149857224595ffc61a) 204

[SamrLookupNamesInDomain method](#section_d91271c67b2e419499278fabfa429f90) 203

[SamrOemChangePasswordUser2 method](#section_8d0bf63efa5f4c75be22558c52075842) 198

[SamrOpenAlias method](#section_c3696b7620054eba820d91fdded3422d) 148

[SamrOpenDomain method](#section_ba710c905b1242f89e5ad4aacc1329fa) 144

[SamrOpenGroup method](#section_d396e6c9d04a4729b0d8f50f2748f3c8) 147

[SamrOpenUser method](#section_0aee1c31ec404633bb560cf8429093c0) 149

[SamrQueryDisplayInformation method](#section_c1458942f2d54317a888abd27abad504) 159

[SamrQueryDisplayInformation2 method](#section_9aa37aa4f77e49b199935c92ba97cd62) 158

[SamrQueryDisplayInformation3 method](#section_7cd1c6d088194a5897215be588c1d4ad) 156

[SamrQueryInformationAlias method](#section_599d6624d93b46de827a58eb0e7527bd) 171

[SamrQueryInformationDomain method](#section_5d6a2817caa941caa269fd13ecbb4fa8) 169

[SamrQueryInformationDomain2 method](#section_fbdbbd7401064ca2941e36ded0be0c62) 167

[SamrQueryInformationGroup method](#section_8fd0003f807640ae872f9cdb6576b3c7) 170

[SamrQueryInformationUser method](#section_4ad8d54c0d5a4d5a9e9a1bc9ee008d47) 174

[SamrQueryInformationUser2 method](#section_29ab27f661da4c7d863ce228ee798f4d) 172

[SamrQuerySecurityObject method](#section_0ecf8fecd17e4a88b7f1e0f0f66790db) 210

[SamrRemoveMemberFromAlias method](#section_7e99402967a244fd84397640fe5a376b) 190

[SamrRemoveMemberFromForeignDomain method](#section_03afc843584d473b834a3f5a1ac86cce) 191

[SamrRemoveMemberFromGroup method](#section_613ff6635863443089238f1edc05310b) 188

[SamrRemoveMultipleMembersFromAlias method](#section_7b2455d23f1344a7996950c240562c42) 192

[SamrRidToSid method](#section_00ff8192a4f645ba9f65917e46b6a693) 216

[SamrSetDSRMPassword method](#section_9bcad7d2b8e14d28a03386040b7b3ce9) 217

[SamrSetInformationAlias method](#section_ba787e4e3a4b47a2aca3c3ac2d2c511e) 177

[SamrSetInformationDomain method](#section_9b7ae0b0bd1141339c62fba7095aee12) 175

[SamrSetInformationGroup method](#section_e66db19f600a481bbc4e23953433255d) 177

[SamrSetInformationUser method](#section_538222f71b894811949a0eac62e38dce) 185

[SamrSetInformationUser2 method](#section_99ee9f3943e84bbaac3a82e0c0e0699e) 178

[SamrSetMemberAttributesOfGroup method](#section_a4db0c2755404d2fb11108db1ee0b5a4) 214

[SamrSetSecurityObject method](#section_6666a06658cf4118bf4bdd54ed55ecf0) 206

[SamrUnicodeChangePasswordUser2 method](#section_acb3204ada8b478e91391ea589edb880) 199

[SamrValidatePassword data types](#section_218e3a7a042f4c8097ce3d46c4efa3fe) 81

[SamrValidatePassword method](#section_c78a7239f8fc4a42bb71321e897dc046) 217

[Schema elements - directory service](#section_6c5ed06fba804980a6469cca57266a91) 98

[SE\_GROUP\_ENABLED](#section_9e093bd2e4514dd5970097b977d7ebb2) 36

[SE\_GROUP\_ENABLED\_BY\_DEFAULT](#section_9e093bd2e4514dd5970097b977d7ebb2) 36

[SE\_GROUP\_MANDATORY](#section_9e093bd2e4514dd5970097b977d7ebb2) 36

Security

[implementer considerations](#section_801e68e3b26d4a65b95becce817becd9) 239

[parameter index](#section_d9bd6ce0787948d0ab29f4cf3a84e99e) 239

Security model

[client](#section_a533f08da1664274a6e8e59125afa433) 230

[server](#section_5e3f7f39df374f9cbb99a27d79cc957f) 125

[Security pattern](#section_34a12061d0ea4f0eb8f536107794062a) 206

[Security Pattern method](#section_34a12061d0ea4f0eb8f536107794062a) 206

[Selective enumerate associated structures](#section_6e1afbe517b34e37b409a996f4e02c6b) 76

[Selective enumerate fields](#section_4e123dbfe1014ade9ffff2247ed25a11) 76

[Selective enumerate pattern](#section_f1970f00e3d54ec485bc2caf0bf18f6d) 156

[Selective Enumerate Pattern method](#section_f1970f00e3d54ec485bc2caf0bf18f6d) 156

Sequencing rules

[client](#section_4d35b12bbdbe42afa6e946c85ee8d1d1) 231

[server](#section_3f15814e46004647abfd3890f5f3570c) 134

Server

[abstract data model](#section_814e12f610374b648d288f1b899dc57f) 99

[Change Password Pattern method](#section_41d7ca60909f4d0db85ac9a35b5f2aaa) 194

[Create Pattern method](#section_2214fd297c2d450fb68d6a0e97ebe48f) 161

[Delete Pattern method](#section_7cf72c9694b545a39ae34bff96541946) 185

[Enumerate Pattern method](#section_a14d7cf8e908468497879d6c4492957a) 151

[initialization](#section_c6c15f12dc63441f865bf78c552e0c73) 131

local events

[domain join processing](#section_a2cb29f5f0c54a7886b06ba9029a1bd7) 229

[domain unjoin processing](#section_ea1feef87b4244c38446646fc6bfda01) 229

[Lookup Pattern method](#section_ae07443cf5864fd7b0b361cf300d06b7) 202

[Membership Pattern method](#section_e8205d2c9ebb4845b9270aca7cbc1f2c) 187

[Membership-Of Pattern method](#section_95a94bcc822c48be81c39697917b3633) 193

[message processing](#section_3f15814e46004647abfd3890f5f3570c) 134

[Miscellaneous method](#section_8fd836b683f64d14a138ef532ffbec4e) 214

[Open Pattern method](#section_6d92e4991d164c3596e6c564dfd2972b) 140

[overview](#section_8adc992116ba4ba1becf67cb7dd7b205) 99

[Query Pattern method](#section_97c29e0795d14273b7dfe8fe4e0f7592) 167

[security model](#section_5e3f7f39df374f9cbb99a27d79cc957f) 125

[Security Pattern method](#section_34a12061d0ea4f0eb8f536107794062a) 206

[Selective Enumerate Pattern method](#section_f1970f00e3d54ec485bc2caf0bf18f6d) 156

[sequencing rules](#section_3f15814e46004647abfd3890f5f3570c) 134

[Set Pattern method](#section_8e86fe5140e2489992e9676466d4e6d3) 175

[supplemental message processing](#section_98bef3a3cf604854a933274f21dd59a8) 223

[timer events](#section_cedb6f40179d408293e3e55e6c4facb5) 228

[timers](#section_d736ccbad40b49c09adec8840696549c) 131

[Set pattern](#section_8e86fe5140e2489992e9676466d4e6d3) 175

[Set Pattern method](#section_8e86fe5140e2489992e9676466d4e6d3) 175

[SID\_NAME\_USE enumeration](#section_312aea80d6c9410da8544c58a0a53ac3) 43

[Standards assignments](#section_57f09cdb51c3429a894dfd10d0407b2c) 27

[STATUS\_ACCESS\_DENIED](#section_7440cfac6052492584e4c32e417de300) 41

[STATUS\_ACCOUNT\_LOCKED\_OUT](#section_7440cfac6052492584e4c32e417de300) 41

[STATUS\_GROUP\_EXISTS](#section_7440cfac6052492584e4c32e417de300) 41

[STATUS\_LM\_CROSS\_ENCRYPTION\_REQUIRED](#section_7440cfac6052492584e4c32e417de300) 41

[STATUS\_MORE\_ENTRIES](#section_7440cfac6052492584e4c32e417de300) 41

[STATUS\_NO\_MORE\_ENTRIES](#section_7440cfac6052492584e4c32e417de300) 41

[STATUS\_NONE\_MAPPED](#section_7440cfac6052492584e4c32e417de300) 41

[STATUS\_NT\_CROSS\_ENCRYPTION\_REQUIRED](#section_7440cfac6052492584e4c32e417de300) 41

[STATUS\_SOME\_NOT\_MAPPED](#section_7440cfac6052492584e4c32e417de300) 41

[STATUS\_USER\_EXISTS](#section_7440cfac6052492584e4c32e417de300) 41

[STATUS\_WRONG\_PASSWORD](#section_7440cfac6052492584e4c32e417de300) 41

String

[handling](#section_5c4b2e83044349318cef011d05257a61) 100

[matching](#section_3e92c4e9cc5440bf9627994dcee92857) 100

T

Timer events

[client](#section_c0a840d832ce4eb98bc4c33bb39be76e) 231

[server](#section_cedb6f40179d408293e3e55e6c4facb5) 228

Timers

[client](#section_2e602eb4fafb47f28bee64da82b060f2) 231

[server](#section_d736ccbad40b49c09adec8840696549c) 131

[Tracking changes](#section_fc6206af03894af584bf28713d70b17c) 275

[Transport](#section_084da2e70ba044fc8f17e8a200c69eb5) 28

Triggers

[attribute - originating updates](#section_90f7a608d9f14831b2feb9a50cf26aec) 112

[referenced from other constraints or triggers](#section_8dc2359a1fc146ce826ecf9150911b31) 123

U

[UF\_ACCOUNTDISABLE](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_DONT\_EXPIRE\_PASSWD](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_DONT\_REQUIRE\_PREAUTH](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_ENCRYPTED\_TEXT\_PASSWORD\_ALLOWED](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_HOMEDIR\_REQUIRED](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_INTERDOMAIN\_TRUST\_ACCOUNT](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_LOCKOUT](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_MNS\_LOGON\_ACCOUNT](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_NO\_AUTH\_DATA\_REQUIRED](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_NORMAL\_ACCOUNT](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_NOT\_DELEGATED](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_PARTIAL\_SECRETS\_ACCOUNT](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_PASSWD\_CANT\_CHANGE](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_PASSWD\_NOTREQD](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_PASSWORD\_EXPIRED](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_SCRIPT](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_SERVER\_TRUST\_ACCOUNT](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_SMARTCARD\_REQUIRED](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_TEMP\_DUPLICATE\_ACCOUNT](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_TRUSTED\_FOR\_DELEGATION](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_TRUSTED\_TO\_AUTHENTICATE\_FOR\_DELEGATION](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_USE\_AES\_KEYS](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_USE\_DES\_KEY\_ONLY](#section_10bf6c8e34af4cf98dff6b6330922863) 39

[UF\_WORKSTATION\_TRUST\_ACCOUNT](#section_10bf6c8e34af4cf98dff6b6330922863) 39

Update constraints

[additional triggers](#section_8dc2359a1fc146ce826ecf9150911b31) 123

[attribute triggers](#section_90f7a608d9f14831b2feb9a50cf26aec) 112

attributes ([section 3.1.1.5](#section_6da47028e678475eb015a5a15ee98212) 103, [section 3.1.1.6](#section_e270cd0a529541be9e892c3dc3a39536) 104)

[referenced from other constraints or triggers](#section_946bee91429f425c88042a48912688db) 108

User

[fields](#section_899d68a981454334a52f2fbd92bec94f) 56

[query/set data types](#section_b7c69df54fe848658d265340c023f91e) 56

User account

[creating - example](#section_3d8e23d8d9df481f83b39175f980294c) 232

[enabling - example](#section_bf8cfb7624f742dea95fe5b9ec7435d0) 234

[USER\_ACCOUNT\_AUTO\_LOCKED](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_ACCOUNT\_DISABLED](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_ALL\_ACCESS](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

[USER\_ALL\_ACCOUNTEXPIRES](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_ADMINCOMMENT](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_BADPASSWORDCOUNT](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_CODEPAGE](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_COUNTRYCODE](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_FULLNAME](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_HOMEDIRECTORY](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_HOMEDIRECTORYDRIVE](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_LASTLOGOFF](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_LASTLOGON](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_LMPASSWORDPRESENT](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_LOGONCOUNT](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_LOGONHOURS](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_NTPASSWORDPRESENT](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_PARAMETERS](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_PASSWORDCANCHANGE](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_PASSWORDEXPIRED](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_PASSWORDLASTSET](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_PASSWORDMUSTCHANGE](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_PRIMARYGROUPID](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_PRIVATEDATA](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_PROFILEPATH](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_SCRIPTPATH](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_SECURITYDESCRIPTOR](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_UNDEFINED\_MASK](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_USERACCOUNTCONTROL](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_USERCOMMENT](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_USERID](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_USERNAME](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_ALL\_WORKSTATIONS](#section_2675c17672e04ac9ae6dcdd87b8ba520) 34

[USER\_CHANGE\_PASSWORD](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

[USER\_CONTROL\_INFORMATION structure](#section_eb5f1508ede14ff1be8255f3e2ef1633) 58

[USER\_DOMAIN\_PASSWORD\_INFORMATION structure](#section_07bd97943db0458db327bbb2d9e799cf) 76

[USER\_DONT\_EXPIRE\_PASSWORD](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_DONT\_REQUIRE\_PREAUTH](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_ENCRYPTED\_TEXT\_PASSWORD\_ALLOWED](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_EXECUTE](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

[USER\_EXPIRES\_INFORMATION structure](#section_8c1308e0723e4170817045a6a723989a) 58

[USER\_FORCE\_PASSWORD\_CHANGE](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

[USER\_HOME\_DIRECTORY\_REQUIRED](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_INFORMATION\_CLASS enumeration](#section_6b0dff905ac0429a93aa150334adabf6) 67

[USER\_INTERDOMAIN\_TRUST\_ACCOUNT](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_LIST\_GROUPS](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

[USER\_MNS\_LOGON\_ACCOUNT](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_NO\_AUTH\_DATA\_REQUIRED](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_NORMAL\_ACCOUNT](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_NOT\_DELEGATED](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_PARTIAL\_SECRETS\_ACCOUNT](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_PASSWORD\_EXPIRED](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_PASSWORD\_NOT\_REQUIRED](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_PRIMARY\_GROUP\_INFORMATION structure](#section_3fd86f83f5ad4bb4814a6554f5797b34) 58

[USER\_PROPERTIES packet](#section_8263e7ababa943d28a363a9cb2dd3dad) 86

[USER\_PROPERTY packet](#section_7c0f2eca1783450bb5a0754cf11f22c9) 87

[USER\_READ](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

[USER\_READ\_ACCOUNT](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

[USER\_READ\_GENERAL](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

[USER\_READ\_GROUP\_INFORMATION](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

[USER\_READ\_LOGON](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

[USER\_READ\_PREFERENCES](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

[USER\_SERVER\_TRUST\_ACCOUNT](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_SMARTCARD\_REQUIRED](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_TEMP\_DUPLICATE\_ACCOUNT](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_TRUSTED\_FOR\_DELEGATION](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_TRUSTED\_TO\_AUTHENTICATE\_FOR\_DELEGATION](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_USE\_AES\_KEYS](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_USE\_DES\_KEY\_ONLY](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_WORKSTATION\_TRUST\_ACCOUNT](#section_b10cfda1f24f441b8f4380cb93e786ec) 37

[USER\_WRITE](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

[USER\_WRITE\_ACCOUNT](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

[USER\_WRITE\_GROUP\_INFORMATION](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

[USER\_WRITE\_PREFERENCES](#section_c0be3f43bcf943eeb0273d02ab372c53) 33

V

[Vendor-extensible fields](#section_574a4d8666ff448cad6ff81e70b795c8) 27

[Versioning](#section_4e7d249a98c04c44ab177c63b67d337e) 27

W

[WDIGEST\_CREDENTIALS packet](#section_830b39623a0042cf94dfc192dcfd4803) 87

[WRITE\_DAC](#section_15b9ebf7161d4c83a672dceb2ac8c448) 29

[WRITE\_OWNER](#section_15b9ebf7161d4c83a672dceb2ac8c448) 29