

# EVIDEN

Identity and Access Management

## DirX Directory

### Syntaxes and Attributes

Version 9.1, Edition June 2025



All product names quoted are trademarks or registered trademarks of the manufacturers concerned.

© 2025 Eviden

All Rights Reserved

Distribution and reproduction not permitted without the consent of Eviden.

# Table of Contents

|                                                                  |    |
|------------------------------------------------------------------|----|
| Copyright .....                                                  | ii |
| Preface .....                                                    | 1  |
| DirX Directory Documentation Set.....                            | 2  |
| Notation Conventions.....                                        | 3  |
| 1. DirX Directory String Representation for DAP Binds .....      | 4  |
| 1.1. Reserved Characters.....                                    | 4  |
| 1.2. Simple and Structured Attributes.....                       | 5  |
| 1.2.1. Attribute Types .....                                     | 5  |
| 1.2.2. Simple Attribute Values.....                              | 6  |
| 1.2.3. Structured Attribute Values.....                          | 6  |
| 1.2.4. Attribute Lists for Simple and Structured Attributes..... | 6  |
| 1.2.5. Attribute Values in a File .....                          | 7  |
| 1.2.6. Base-64 Encoded Attribute Values .....                    | 8  |
| 1.3. Distinguished Names .....                                   | 9  |
| 1.4. Search Filters.....                                         | 9  |
| 1.4.1. Search Filter Expression Example .....                    | 10 |
| 1.4.2. Search Operator Order of Precedence.....                  | 11 |
| 1.4.3. Search Filter Expressions and Reserved Characters .....   | 11 |
| 1.4.4. Search Filter Expressions and Matching Rules .....        | 11 |
| 1.5. Attribute Syntax .....                                      | 12 |
| 1.5.1. Overriding Default Attribute Syntax .....                 | 12 |
| 1.5.2. Syntax Abbreviations .....                                | 13 |
| 1.5.3. Undefined Types and Abbreviations .....                   | 14 |
| 1.5.4. Attribute Syntax and Equality Matching Rule Syntax .....  | 14 |
| 1.5.5. Attribute Syntax vs. Equality Matching Rule Syntax .....  | 15 |
| 1.6. String Representations for Simple Attribute Syntaxes .....  | 18 |
| 1.6.1. Attribute Type Syntax.....                                | 18 |
| 1.6.2. Bit String Syntax .....                                   | 18 |
| 1.6.3. Boolean Syntax.....                                       | 18 |
| 1.6.4. Country Name Syntax.....                                  | 18 |
| 1.6.5. Destination Indicator Syntax .....                        | 18 |
| 1.6.6. Directory String Syntax.....                              | 19 |
| 1.6.7. DSE Type Syntax .....                                     | 19 |
| 1.6.8. Generalized Time Syntax .....                             | 21 |
| 1.6.9. IA5 String Syntax .....                                   | 21 |
| 1.6.10. Integer String Syntax.....                               | 21 |
| 1.6.11. International ISDN Number Syntax .....                   | 21 |
| 1.6.12. ISO646 String Syntax .....                               | 21 |
| 1.6.13. Local Scope Policy Syntax .....                          | 21 |



|                                                                     |    |
|---------------------------------------------------------------------|----|
| 1.6.14. Numeric String Syntax .....                                 | 22 |
| 1.6.15. Object ID Syntax .....                                      | 22 |
| 1.6.16. Octet String Syntax .....                                   | 22 |
| 1.6.17. Preferred Delivery Method Syntax .....                      | 23 |
| 1.6.18. Printable String Syntax .....                               | 23 |
| 1.6.19. SDSE Type Syntax .....                                      | 24 |
| 1.6.20. TTX String Syntax .....                                     | 24 |
| 1.6.21. Unique Identifier Syntax .....                              | 24 |
| 1.6.22. Universal String Syntax .....                               | 24 |
| 1.6.23. UTC Time Syntax .....                                       | 24 |
| 1.6.24. X121 Address Syntax .....                                   | 24 |
| 1.7. String Representations for Structured Attribute Syntaxes ..... | 24 |
| 1.7.1. Access-Point .....                                           | 26 |
| 1.7.1.1. Synopsis .....                                             | 26 |
| 1.7.1.2. Attribute Type .....                                       | 26 |
| 1.7.1.3. Components .....                                           | 26 |
| 1.7.1.4. Description .....                                          | 26 |
| 1.7.1.5. Example .....                                              | 27 |
| 1.7.2. Consumer-Information .....                                   | 27 |
| 1.7.2.1. Synopsis .....                                             | 27 |
| 1.7.2.2. Attribute Type .....                                       | 27 |
| 1.7.2.3. Components .....                                           | 27 |
| 1.7.2.4. Description .....                                          | 28 |
| 1.7.2.5. Examples .....                                             | 28 |
| 1.7.3. Master-And-Shadow-Access-Points .....                        | 28 |
| 1.7.3.1. Synopsis .....                                             | 28 |
| 1.7.3.2. Attribute Type .....                                       | 28 |
| 1.7.3.3. Components .....                                           | 28 |
| 1.7.3.4. Description .....                                          | 29 |
| 1.7.3.5. Examples .....                                             | 29 |
| 1.7.4. Supplier-Information .....                                   | 29 |
| 1.7.4.1. Synopsis .....                                             | 30 |
| 1.7.4.2. Attribute Type .....                                       | 30 |
| 1.7.4.3. Components .....                                           | 30 |
| 1.7.4.4. Description .....                                          | 31 |
| 1.7.4.5. Examples .....                                             | 31 |
| 1.7.5. Supplier-And-Consumers .....                                 | 31 |
| 1.7.5.1. Synopsis .....                                             | 31 |
| 1.7.5.2. Attribute Type .....                                       | 32 |
| 1.7.5.3. Components .....                                           | 32 |
| 1.7.5.4. Description .....                                          | 32 |
| 1.7.5.5. Examples .....                                             | 32 |



|                                      |    |
|--------------------------------------|----|
| 1.7.6. Certificate (Signed) .....    | 32 |
| 1.7.6.1. Synopsis .....              | 32 |
| 1.7.6.2. Attribute Type .....        | 33 |
| 1.7.6.3. Components .....            | 33 |
| 1.7.6.4. Description .....           | 34 |
| 1.7.6.5. Examples .....              | 34 |
| 1.7.7. Certificate (Unsigned) .....  | 34 |
| 1.7.7.1. Synopsis .....              | 34 |
| 1.7.7.2. Components .....            | 35 |
| 1.7.7.3. Description .....           | 37 |
| 1.7.7.4. Examples .....              | 38 |
| 1.7.8. Certificate-Pair .....        | 38 |
| 1.7.8.1. Synopsis .....              | 38 |
| 1.7.8.2. Attribute Type .....        | 38 |
| 1.7.8.3. Components .....            | 38 |
| 1.7.8.4. Description .....           | 39 |
| 1.7.8.5. Example .....               | 39 |
| 1.7.9. Cert-List (Signed) .....      | 40 |
| 1.7.9.1. Synopsis .....              | 40 |
| 1.7.9.2. Attribute Type .....        | 41 |
| 1.7.9.3. Components .....            | 41 |
| 1.7.9.4. Description .....           | 42 |
| 1.7.9.5. Example .....               | 42 |
| 1.7.10. Cert-List (Unsigned) .....   | 42 |
| 1.7.10.1. Synopsis .....             | 42 |
| 1.7.10.2. Components .....           | 42 |
| 1.7.10.3. Description .....          | 45 |
| 1.7.10.4. Example .....              | 45 |
| 1.7.11. ACI-Item .....               | 45 |
| 1.7.11.1. Synopsis .....             | 46 |
| 1.7.11.2. Attribute Type .....       | 46 |
| 1.7.11.3. Components .....           | 46 |
| 1.7.11.4. Description .....          | 52 |
| 1.7.11.5. Example .....              | 53 |
| 1.7.12. Cooperating-DSA .....        | 54 |
| 1.7.12.1. Synopsis .....             | 54 |
| 1.7.12.2. Attribute Type .....       | 54 |
| 1.7.12.3. Components .....           | 54 |
| 1.7.12.4. Description .....          | 55 |
| 1.7.12.5. Examples .....             | 56 |
| 1.7.13. Cooperating-DSA-Status ..... | 57 |
| 1.7.13.1. Synopsis .....             | 57 |

|                                          |    |
|------------------------------------------|----|
| 1.7.13.2. Components .....               | 57 |
| 1.7.13.3. Description .....              | 58 |
| 1.7.13.4. Examples .....                 | 58 |
| 1.7.14. DSA-Policies .....               | 60 |
| 1.7.14.1. Synopsis .....                 | 60 |
| 1.7.14.2. Attribute Type .....           | 60 |
| 1.7.14.3. Components .....               | 60 |
| 1.7.14.4. Description .....              | 63 |
| 1.7.14.5. Examples .....                 | 63 |
| 1.7.15. Paging-Policies .....            | 65 |
| 1.7.15.1. Synopsis .....                 | 65 |
| 1.7.15.2. Attribute Type .....           | 65 |
| 1.7.15.3. Components .....               | 65 |
| 1.7.15.4. Description .....              | 66 |
| 1.7.15.5. Examples .....                 | 66 |
| 1.7.16. Shadowing-Agreement-Info .....   | 66 |
| 1.7.16.1. Synopsis .....                 | 66 |
| 1.7.16.2. Components .....               | 67 |
| 1.7.16.3. Description .....              | 70 |
| 1.7.16.4. Example .....                  | 71 |
| 1.7.17. SOB-Policies .....               | 72 |
| 1.7.17.1. Synopsis .....                 | 72 |
| 1.7.17.2. Components .....               | 72 |
| 1.7.17.3. Description .....              | 82 |
| 1.7.17.4. Example .....                  | 83 |
| 1.7.18. Update-Status .....              | 84 |
| 1.7.18.1. Synopsis .....                 | 84 |
| 1.7.18.2. Components .....               | 84 |
| 1.7.18.3. Description .....              | 88 |
| 1.7.18.4. Examples .....                 | 88 |
| 1.7.19. User-Policies .....              | 88 |
| 1.7.19.1. Synopsis .....                 | 88 |
| 1.7.19.2. Attribute Type .....           | 88 |
| 1.7.19.3. Components .....               | 88 |
| 1.7.19.4. Description .....              | 91 |
| 1.7.19.5. Example .....                  | 92 |
| 1.7.20. Continuation-Reference .....     | 92 |
| 1.7.20.1. Synopsis .....                 | 92 |
| 1.7.20.2. Components .....               | 93 |
| 1.7.20.3. Description .....              | 94 |
| 1.7.20.4. Examples .....                 | 94 |
| 1.7.21. Facsimile-Telephone-Number ..... | 95 |

|                                           |     |
|-------------------------------------------|-----|
| 1.7.21.1. Synopsis .....                  | 95  |
| 1.7.21.2. Attribute Type .....            | 95  |
| 1.7.21.3. Components .....                | 95  |
| 1.7.21.4. Description .....               | 95  |
| 1.7.21.5. Example .....                   | 95  |
| 1.7.22. Name-And-Optional-UID .....       | 95  |
| 1.7.22.1. Synopsis .....                  | 96  |
| 1.7.22.2. Attribute Type .....            | 96  |
| 1.7.22.3. Components .....                | 96  |
| 1.7.22.4. Description .....               | 96  |
| 1.7.22.5. Examples .....                  | 96  |
| 1.7.23. Partial-Outcome-Qualifier .....   | 96  |
| 1.7.23.1. Synopsis .....                  | 97  |
| 1.7.23.2. Components .....                | 97  |
| 1.7.23.3. Description .....               | 97  |
| 1.7.23.4. Examples .....                  | 97  |
| 1.7.24. Postal-Address .....              | 98  |
| 1.7.24.1. Synopsis .....                  | 98  |
| 1.7.24.2. Attribute Type .....            | 98  |
| 1.7.24.3. Components .....                | 98  |
| 1.7.24.4. Description .....               | 98  |
| 1.7.24.5. Example .....                   | 99  |
| 1.7.25. Presentation-Address .....        | 99  |
| 1.7.25.1. Synopsis .....                  | 99  |
| 1.7.25.2. Attribute Type .....            | 99  |
| 1.7.25.3. Components .....                | 99  |
| 1.7.25.4. Description .....               | 101 |
| 1.7.25.5. Examples .....                  | 102 |
| 1.7.26. Protocol-Information .....        | 103 |
| 1.7.26.1. Synopsis .....                  | 103 |
| 1.7.26.2. Attribute Type .....            | 103 |
| 1.7.26.3. Components .....                | 104 |
| 1.7.26.4. Description .....               | 104 |
| 1.7.26.5. Examples .....                  | 104 |
| 1.7.27. Subtree-Specification .....       | 104 |
| 1.7.27.1. Synopsis .....                  | 104 |
| 1.7.27.2. Attribute Type .....            | 104 |
| 1.7.27.3. Components .....                | 105 |
| 1.7.27.4. Description .....               | 106 |
| 1.7.27.5. Examples .....                  | 106 |
| 1.7.28. Teletex-Terminal-Identifier ..... | 107 |
| 1.7.28.1. Synopsis .....                  | 107 |



|                                                                                                     |     |
|-----------------------------------------------------------------------------------------------------|-----|
| 1.7.28.2. Attribute Type .....                                                                      | 107 |
| 1.7.28.3. Components.....                                                                           | 107 |
| 1.7.28.4. Description .....                                                                         | 108 |
| 1.7.28.5. Examples .....                                                                            | 108 |
| 1.7.29. Telex-Number .....                                                                          | 108 |
| 1.7.29.1. Synopsis .....                                                                            | 108 |
| 1.7.29.2. Attribute Type .....                                                                      | 109 |
| 1.7.29.3. Components.....                                                                           | 109 |
| 1.7.29.4. Description .....                                                                         | 109 |
| 1.7.29.5. Example .....                                                                             | 109 |
| 1.7.30. DirX Directory-Member-Url .....                                                             | 109 |
| 1.7.30.1. Synopsis .....                                                                            | 109 |
| 1.7.30.2. Attribute Type .....                                                                      | 109 |
| 1.7.30.3. Components .....                                                                          | 110 |
| 1.7.30.4. Description .....                                                                         | 110 |
| 1.7.30.5. Example.....                                                                              | 110 |
| 2. DirX Directory String Representation for LDAP Binds .....                                        | 111 |
| 2.1. Reserved Characters .....                                                                      | 111 |
| 2.2. Simple and Structured Attributes .....                                                         | 111 |
| 2.2.1. Attribute Types .....                                                                        | 112 |
| 2.2.2. Simple Attribute Values .....                                                                | 112 |
| 2.2.3. Structured Attribute Values .....                                                            | 113 |
| 2.2.4. Attribute Lists for Simple and Structured Attributes .....                                   | 113 |
| 2.2.5. Binary Attribute Values.....                                                                 | 113 |
| 2.2.6. Attribute Values in a File .....                                                             | 114 |
| 2.3. Distinguished Names .....                                                                      | 116 |
| 2.4. Search Filters.....                                                                            | 117 |
| 2.4.1. Search Filter Expression Example .....                                                       | 118 |
| 2.4.2. Certificates in Search Filters .....                                                         | 118 |
| 2.4.2.1. The certificateExactMatching Rule.....                                                     | 119 |
| 2.4.2.1.1. Example.....                                                                             | 120 |
| 2.4.2.2. The certificateMatching Rule.....                                                          | 120 |
| 2.4.2.2.1. Searching for Certificates with a specific keyUsage: .....                               | 120 |
| 2.4.2.2.2. Example .....                                                                            | 120 |
| 2.4.2.2.3. Searching for Certificates issued by a specific CA (issuer): .....                       | 121 |
| 2.4.2.2.4. Example .....                                                                            | 121 |
| 2.4.2.2.5. Searching for Certificates that are valid (certificateValid):.....                       | 121 |
| 2.4.2.2.6. Examples .....                                                                           | 122 |
| 2.4.2.2.7. Searching for Certificates that expires at a specific time<br>(certificateExpired):..... | 122 |
| 2.4.2.2.8. Example .....                                                                            | 123 |
| 2.4.2.2.9. Searching for Certificates that are not yet valid                                        |     |

|                                                                                    |     |
|------------------------------------------------------------------------------------|-----|
| (certificateNotYetValid): .....                                                    | 123 |
| 2.4.2.2.10. Example .....                                                          | 123 |
| 2.4.2.2.11. Searching for Certificates with a combined extended LDAP filter: ..... | 123 |
| 2.4.2.2.12. Example .....                                                          | 123 |
| 2.4.3. LDAP Matched Value Only Filter Control .....                                | 124 |
| 2.5. Attribute Syntax .....                                                        | 126 |
| 2.5.1. Undefined Types .....                                                       | 126 |
| 2.6. String Representations for Simple Attribute Syntaxes .....                    | 126 |
| 2.6.1. Attribute Type Syntax .....                                                 | 126 |
| 2.6.2. Bit String Syntax .....                                                     | 126 |
| 2.6.3. Boolean Syntax .....                                                        | 126 |
| 2.6.4. Object ID Syntax .....                                                      | 127 |
| 2.6.5. Generalized Time Syntax .....                                               | 127 |
| 2.6.6. IA5 String Syntax .....                                                     | 127 |
| 2.6.7. Integer String Syntax .....                                                 | 127 |
| 2.6.8. Numeric String Syntax .....                                                 | 127 |
| 2.6.9. Preferred Delivery Method Syntax .....                                      | 127 |
| 2.6.10. Printable String Syntax .....                                              | 128 |
| 2.6.11. UTC Time Syntax .....                                                      | 128 |
| 2.7. String Representations for Structured Attribute Syntaxes .....                | 128 |
| 2.7.1. Attribute-Type-Description .....                                            | 128 |
| 2.7.1.1. Synopsis .....                                                            | 128 |
| 2.7.1.2. Attribute Type .....                                                      | 129 |
| 2.7.1.3. Components .....                                                          | 129 |
| 2.7.1.4. Description .....                                                         | 131 |
| 2.7.1.5. Examples .....                                                            | 131 |
| 2.7.2. Object-Class-Description .....                                              | 132 |
| 2.7.2.1. Synopsis .....                                                            | 132 |
| 2.7.2.2. Attribute Type .....                                                      | 132 |
| 2.7.2.3. Components .....                                                          | 132 |
| 2.7.2.4. Description .....                                                         | 133 |
| 2.7.2.5. Examples .....                                                            | 134 |
| 2.7.3. Facsimile-Telephone-Number .....                                            | 134 |
| 2.7.3.1. Synopsis .....                                                            | 134 |
| 2.7.3.2. Attribute Type .....                                                      | 134 |
| 2.7.3.3. Components .....                                                          | 134 |
| 2.7.3.4. Description .....                                                         | 135 |
| 2.7.3.5. Example .....                                                             | 135 |
| 2.7.4. Name-And-Optional-UID .....                                                 | 135 |
| 2.7.4.1. Synopsis .....                                                            | 135 |
| 2.7.4.2. Attribute Type .....                                                      | 135 |
| 2.7.4.3. Components .....                                                          | 135 |

|                                              |     |
|----------------------------------------------|-----|
| 2.7.4.4. Description .....                   | 136 |
| 2.7.4.5. Examples .....                      | 136 |
| 2.7.5. Postal-Address .....                  | 136 |
| 2.7.5.1. Synopsis .....                      | 136 |
| 2.7.5.2. Attribute Type .....                | 136 |
| 2.7.5.3. Components .....                    | 136 |
| 2.7.5.4. Description .....                   | 137 |
| 2.7.5.5. Example .....                       | 137 |
| 2.7.6. Teletex-Terminal-Identifier .....     | 137 |
| 2.7.6.1. Synopsis .....                      | 137 |
| 2.7.6.2. Attribute Type .....                | 137 |
| 2.7.6.3. Components .....                    | 137 |
| 2.7.6.4. Description .....                   | 138 |
| 2.7.6.5. Examples .....                      | 138 |
| 2.7.7. Telex-Number .....                    | 138 |
| 2.7.7.1. Synopsis .....                      | 138 |
| 2.7.7.2. Attribute Type .....                | 138 |
| 2.7.7.3. Components .....                    | 138 |
| 2.7.7.4. Description .....                   | 139 |
| 2.7.7.5. Example .....                       | 139 |
| 3. DirX Directory Attributes .....           | 140 |
| 3.1. X.500 User Application Attributes ..... | 140 |
| 3.1.1. Naming Attributes .....               | 140 |
| 3.1.1.1. Common Name .....                   | 140 |
| 3.1.1.1.1. Abbreviation .....                | 140 |
| 3.1.1.1.2. LDAP Name(s) .....                | 141 |
| 3.1.1.1.3. Syntax (DAP) .....                | 141 |
| 3.1.1.1.4. Example (DAP) .....               | 141 |
| 3.1.1.1.5. Example (LDAP) .....              | 141 |
| 3.1.1.2. Country Name .....                  | 141 |
| 3.1.1.2.1. Abbreviation .....                | 141 |
| 3.1.1.2.2. LDAP Name(s) .....                | 141 |
| 3.1.1.2.3. Syntax (DAP) .....                | 141 |
| 3.1.1.2.4. Syntax (LDAP) .....               | 141 |
| 3.1.1.2.5. Example (DAP) .....               | 141 |
| 3.1.1.3. Locality Name .....                 | 142 |
| 3.1.1.3.1. Abbreviation .....                | 142 |
| 3.1.1.3.2. LDAP Name(s) .....                | 142 |
| 3.1.1.3.3. Syntax (DAP) .....                | 142 |
| 3.1.1.3.4. Example (DAP) .....               | 142 |
| 3.1.1.4. Organization Name .....             | 142 |
| 3.1.1.4.1. Abbreviation .....                | 142 |

|                                         |     |
|-----------------------------------------|-----|
| 3.1.1.4.2. LDAP Name(s) .....           | 142 |
| 3.1.1.4.3. Syntax (DAP) .....           | 143 |
| 3.1.1.4.4. Example (DAP) .....          | 143 |
| 3.1.1.4.5. Example (LDAP) .....         | 143 |
| 3.1.1.5. Organizational Unit Name ..... | 143 |
| 3.1.1.5.1. Abbreviation .....           | 143 |
| 3.1.1.5.2. LDAP Name(s) .....           | 143 |
| 3.1.1.5.3. Syntax (DAP) .....           | 143 |
| 3.1.1.5.4. Example (DAP) .....          | 143 |
| 3.1.1.5.5. Example (LDAP) .....         | 143 |
| 3.1.1.6. State or Province Name .....   | 144 |
| 3.1.1.6.1. Abbreviation .....           | 144 |
| 3.1.1.6.2. LDAP Name(s) .....           | 144 |
| 3.1.1.6.3. Syntax (DAP) .....           | 144 |
| 3.1.1.6.4. Example (DAP) .....          | 144 |
| 3.1.1.7. Street Address .....           | 144 |
| 3.1.1.7.1. Abbreviation .....           | 144 |
| 3.1.1.7.2. LDAP Name(s) .....           | 144 |
| 3.1.1.7.3. Syntax (DAP) .....           | 144 |
| 3.1.1.7.4. Example (DAP) .....          | 144 |
| 3.1.1.7.5. Example (LDAP) .....         | 145 |
| 3.1.2. Names in General .....           | 145 |
| 3.1.2.1. DMD Name .....                 | 145 |
| 3.1.2.1.1. Abbreviation .....           | 145 |
| 3.1.2.1.2. LDAP Name(s) .....           | 145 |
| 3.1.2.1.3. Syntax (DAP) .....           | 145 |
| 3.1.2.1.4. Example (DAP) .....          | 145 |
| 3.1.2.1.5. Example (LDAP) .....         | 145 |
| 3.1.2.2. DN Qualifier .....             | 145 |
| 3.1.2.2.1. Abbreviation .....           | 145 |
| 3.1.2.2.2. LDAP Name(s) .....           | 145 |
| 3.1.2.2.3. Syntax (DAP) .....           | 146 |
| 3.1.2.2.4. Syntax (LDAP) .....          | 146 |
| 3.1.2.2.5. Example (DAP) .....          | 146 |
| 3.1.2.2.6. Example (LDAP) .....         | 146 |
| 3.1.2.3. Generation Qualifier .....     | 146 |
| 3.1.2.3.1. Abbreviation .....           | 146 |
| 3.1.2.3.2. LDAP Name(s) .....           | 146 |
| 3.1.2.3.3. Syntax (DAP) .....           | 146 |
| 3.1.2.3.4. Example (DAP) .....          | 146 |
| 3.1.2.4. Given Name .....               | 147 |
| 3.1.2.4.1. Abbreviation .....           | 147 |

|                                        |     |
|----------------------------------------|-----|
| 3.1.2.4.2. LDAP Name(s) .....          | 147 |
| 3.1.2.4.3. Syntax (DAP) .....          | 147 |
| 3.1.2.4.4. Example (DAP) .....         | 147 |
| 3.1.2.4.5. Example (LDAP) .....        | 147 |
| 3.1.2.5. Initials .....                | 147 |
| 3.1.2.5.1. Abbreviation .....          | 147 |
| 3.1.2.5.2. LDAP Name(s) .....          | 147 |
| 3.1.2.5.3. Syntax (DAP) .....          | 147 |
| 3.1.2.5.4. Example (DAP) .....         | 147 |
| 3.1.2.5.5. Example (LDAP) .....        | 147 |
| 3.1.2.6. Name .....                    | 148 |
| 3.1.2.6.1. Abbreviation .....          | 148 |
| 3.1.2.6.2. LDAP Name(s) .....          | 148 |
| 3.1.2.6.3. Syntax (DAP) .....          | 148 |
| 3.1.2.6.4. Example (DAP) .....         | 148 |
| 3.1.2.6.5. Example (LDAP) .....        | 148 |
| 3.1.2.7. Organization Identifier ..... | 148 |
| 3.1.2.7.1. Abbreviation .....          | 148 |
| 3.1.2.7.2. LDAP Name(s) .....          | 148 |
| 3.1.2.7.3. Syntax (DAP) .....          | 148 |
| 3.1.2.7.4. Example (DAP) .....         | 148 |
| 3.1.2.7.5. Example (LDAP) .....        | 149 |
| 3.1.2.8. Pseudonym .....               | 149 |
| 3.1.2.8.1. Abbreviation .....          | 149 |
| 3.1.2.8.2. LDAP Name(s) .....          | 149 |
| 3.1.2.8.3. Syntax (DAP) .....          | 149 |
| 3.1.2.8.4. Example (DAP) .....         | 149 |
| 3.1.2.8.5. Example (LDAP) .....        | 149 |
| 3.1.2.9. Serial Number .....           | 149 |
| 3.1.2.9.1. Abbreviation .....          | 149 |
| 3.1.2.9.2. LDAP Name(s) .....          | 149 |
| 3.1.2.9.3. Syntax (DAP) .....          | 150 |
| 3.1.2.9.4. Syntax (LDAP) .....         | 150 |
| 3.1.2.9.5. Example (DAP) .....         | 150 |
| 3.1.2.9.6. Example (LDAP) .....        | 150 |
| 3.1.2.10. Surname .....                | 150 |
| 3.1.2.10.1. Abbreviation .....         | 150 |
| 3.1.2.10.2. LDAP Name(s) .....         | 150 |
| 3.1.2.10.3. Syntax (DAP) .....         | 150 |
| 3.1.2.10.4. Example (DAP) .....        | 150 |
| 3.1.2.10.5. Example (LDAP) .....       | 150 |
| 3.1.2.11. Unique Identifier .....      | 150 |

|                                            |     |
|--------------------------------------------|-----|
| 3.1.2.11.1. Abbreviation .....             | 151 |
| 3.1.2.11.2. LDAP Name(s) .....             | 151 |
| 3.1.2.11.3. Syntax (DAP) .....             | 151 |
| 3.1.2.11.4. Syntax (LDAP) .....            | 151 |
| 3.1.2.11.5. Example (DAP) .....            | 151 |
| 3.1.2.11.6. Example (LDAP) .....           | 151 |
| 3.1.3. Attributes for Authentication ..... | 151 |
| 3.1.3.1. Authority Revocation List .....   | 151 |
| 3.1.3.1.1. Abbreviation .....              | 151 |
| 3.1.3.1.2. LDAP Name(s) .....              | 151 |
| 3.1.3.1.3. Syntax (DAP) .....              | 151 |
| 3.1.3.1.4. Syntax (LDAP) .....             | 151 |
| 3.1.3.1.5. Example (DAP) .....             | 152 |
| 3.1.3.1.6. Example (LDAP) .....            | 152 |
| 3.1.3.2. CA-Certificate .....              | 152 |
| 3.1.3.2.1. Abbreviation .....              | 152 |
| 3.1.3.2.2. LDAP Name(s) .....              | 152 |
| 3.1.3.2.3. Syntax (DAP) .....              | 152 |
| 3.1.3.2.4. Syntax (LDAP) .....             | 153 |
| 3.1.3.2.5. Example (DAP) .....             | 153 |
| 3.1.3.3. Certificate Revocation List ..... | 153 |
| 3.1.3.3.1. Abbreviation .....              | 153 |
| 3.1.3.3.2. LDAP Name(s) .....              | 153 |
| 3.1.3.3.3. Syntax (DAP) .....              | 154 |
| 3.1.3.3.4. Syntax (LDAP) .....             | 154 |
| 3.1.3.3.5. Example (DAP) .....             | 154 |
| 3.1.3.3.6. Example (LDAP) .....            | 154 |
| 3.1.3.4. Cross-Certificate-Pair .....      | 154 |
| 3.1.3.4.1. Abbreviation .....              | 154 |
| 3.1.3.4.2. LDAP Name(s) .....              | 155 |
| 3.1.3.4.3. Syntax (DAP) .....              | 155 |
| 3.1.3.4.4. Syntax (LDAP) .....             | 155 |
| 3.1.3.4.5. Example (DAP) .....             | 155 |
| 3.1.3.5. Delta Revocation List .....       | 156 |
| 3.1.3.5.1. Abbreviation .....              | 156 |
| 3.1.3.5.2. LDAP Name(s) .....              | 156 |
| 3.1.3.5.3. Syntax (DAP) .....              | 156 |
| 3.1.3.5.4. Syntax (LDAP) .....             | 156 |
| 3.1.3.5.5. Example (DAP) .....             | 156 |
| 3.1.3.5.6. Example (LDAP) .....            | 157 |
| 3.1.3.6. NTS Security Identifier .....     | 157 |
| 3.1.3.6.1. Abbreviation .....              | 157 |



|                                              |     |
|----------------------------------------------|-----|
| 3.1.3.6.2. LDAP Name(s) .....                | 157 |
| 3.1.3.6.3. Syntax (DAP) .....                | 157 |
| 3.1.3.6.4. Example (DAP) .....               | 157 |
| 3.1.3.6.5. Example (LDAP) .....              | 158 |
| 3.1.3.7. User Certificate .....              | 158 |
| 3.1.3.7.1. Abbreviation .....                | 158 |
| 3.1.3.7.2. LDAP Name(s) .....                | 158 |
| 3.1.3.7.3. Syntax (DAP) .....                | 158 |
| 3.1.3.7.4. Syntax (LDAP) .....               | 158 |
| 3.1.3.7.5. Example (DAP) .....               | 158 |
| 3.1.3.8. User Password .....                 | 159 |
| 3.1.3.8.1. Password Encryption .....         | 159 |
| 3.1.3.8.2. Abbreviation .....                | 162 |
| 3.1.3.8.3. LDAP Name(s) .....                | 162 |
| 3.1.3.8.4. Syntax (DAP) .....                | 162 |
| 3.1.3.8.5. Example (DAP) .....               | 162 |
| 3.1.3.8.6. Example (LDAP) .....              | 162 |
| 3.1.3.9. DirX TOTP Secret .....              | 162 |
| 3.1.3.9.1. Abbreviation .....                | 163 |
| 3.1.3.9.2. LDAP Name(s) .....                | 163 |
| 3.1.3.9.3. Syntax (DAP) .....                | 163 |
| 3.1.3.9.4. Example (DAP) .....               | 163 |
| 3.1.3.9.5. Example (LDAP) .....              | 164 |
| 3.1.4. Explanatory Attributes .....          | 164 |
| 3.1.4.1. Business Category .....             | 164 |
| 3.1.4.1.1. Abbreviation .....                | 164 |
| 3.1.4.1.2. LDAP Name(s) .....                | 164 |
| 3.1.4.1.3. Syntax (DAP) .....                | 164 |
| 3.1.4.1.4. Example (DAP) .....               | 164 |
| 3.1.4.1.5. Example (LDAP) .....              | 164 |
| 3.1.4.2. Description .....                   | 164 |
| 3.1.4.2.1. Abbreviation .....                | 165 |
| 3.1.4.2.2. LDAP Name(s) .....                | 165 |
| 3.1.4.2.3. Syntax (DAP) .....                | 165 |
| 3.1.4.2.4. Example (DAP) .....               | 165 |
| 3.1.4.2.5. Example (LDAP) .....              | 165 |
| 3.1.5. Postal Addressing .....               | 165 |
| 3.1.5.1. Physical Delivery Office Name ..... | 165 |
| 3.1.5.1.1. Abbreviation .....                | 165 |
| 3.1.5.1.2. LDAP Name(s) .....                | 165 |
| 3.1.5.1.3. Syntax (DAP) .....                | 165 |
| 3.1.5.1.4. Example (DAP) .....               | 165 |

|                                          |     |
|------------------------------------------|-----|
| 3.1.5.1.5. Example (LDAP) .....          | 165 |
| 3.1.5.2. Postal Address .....            | 166 |
| 3.1.5.2.1. Abbreviation .....            | 166 |
| 3.1.5.2.2. LDAP Name(s) .....            | 166 |
| 3.1.5.2.3. Syntax (DAP) .....            | 166 |
| 3.1.5.2.4. Syntax (LDAP) .....           | 166 |
| 3.1.5.2.5. Example (DAP) .....           | 166 |
| 3.1.5.2.6. Example (LDAP) .....          | 166 |
| 3.1.5.3. Postal Code .....               | 166 |
| 3.1.5.3.1. Abbreviation .....            | 166 |
| 3.1.5.3.2. LDAP Name(s) .....            | 166 |
| 3.1.5.3.3. Syntax (DAP) .....            | 166 |
| 3.1.5.3.4. Example (DAP) .....           | 167 |
| 3.1.5.3.5. Example (LDAP) .....          | 167 |
| 3.1.5.4. Post Office Box .....           | 167 |
| 3.1.5.4.1. Abbreviation .....            | 167 |
| 3.1.5.4.2. LDAP Name(s) .....            | 167 |
| 3.1.5.4.3. Syntax (DAP) .....            | 167 |
| 3.1.5.4.4. Example (DAP) .....           | 167 |
| 3.1.5.4.5. Example (LDAP) .....          | 167 |
| 3.1.6. Telecommunication Addresses ..... | 167 |
| 3.1.6.1. Destination Indicator .....     | 167 |
| 3.1.6.1.1. Abbreviation .....            | 167 |
| 3.1.6.1.2. LDAP Name(s) .....            | 168 |
| 3.1.6.1.3. Syntax (DAP) .....            | 168 |
| 3.1.6.1.4. Syntax (LDAP) .....           | 168 |
| 3.1.6.1.5. Example (DAP) .....           | 168 |
| 3.1.6.1.6. Example (LDAP) .....          | 168 |
| 3.1.6.2. Fax Telephone Number .....      | 168 |
| 3.1.6.2.1. Abbreviation .....            | 168 |
| 3.1.6.2.2. LDAP Name(s) .....            | 168 |
| 3.1.6.2.3. Syntax (DAP) .....            | 168 |
| 3.1.6.2.4. Syntax (LDAP) .....           | 168 |
| 3.1.6.2.5. Example (DAP) .....           | 168 |
| 3.1.6.2.6. Example (LDAP) .....          | 168 |
| 3.1.6.3. International ISDN Number ..... | 169 |
| 3.1.6.3.1. Abbreviation .....            | 169 |
| 3.1.6.3.2. LDAP Name(s) .....            | 169 |
| 3.1.6.3.3. Syntax (DAP) .....            | 169 |
| 3.1.6.3.4. Syntax (LDAP) .....           | 169 |
| 3.1.6.3.5. Example (DAP) .....           | 169 |
| 3.1.6.3.6. Example (LDAP) .....          | 169 |

|                                            |     |
|--------------------------------------------|-----|
| 3.1.6.4. Registered Address .....          | 169 |
| 3.1.6.4.1. Abbreviation .....              | 169 |
| 3.1.6.4.2. LDAP Name(s) .....              | 169 |
| 3.1.6.4.3. Syntax (DAP) .....              | 170 |
| 3.1.6.4.4. Syntax (LDAP) .....             | 170 |
| 3.1.6.4.5. Example (DAP) .....             | 170 |
| 3.1.6.4.6. Example (LDAP) .....            | 170 |
| 3.1.6.5. Telephone Number .....            | 170 |
| 3.1.6.5.1. Abbreviation .....              | 170 |
| 3.1.6.5.2. LDAP Name(s) .....              | 170 |
| 3.1.6.5.3. Syntax (DAP) .....              | 170 |
| 3.1.6.5.4. Syntax (LDAP) .....             | 170 |
| 3.1.6.5.5. Example (DAP) .....             | 170 |
| 3.1.6.5.6. Example (LDAP) .....            | 170 |
| 3.1.6.6. Teletex Terminal Identifier ..... | 171 |
| 3.1.6.6.1. Abbreviation .....              | 171 |
| 3.1.6.6.2. LDAP Name(s) .....              | 171 |
| 3.1.6.6.3. Syntax (DAP) .....              | 171 |
| 3.1.6.6.4. Syntax (LDAP) .....             | 171 |
| 3.1.6.6.5. Example (DAP) .....             | 171 |
| 3.1.6.6.6. Example (LDAP) .....            | 171 |
| 3.1.6.7. Telex Number .....                | 171 |
| 3.1.6.7.1. Abbreviation .....              | 171 |
| 3.1.6.7.2. LDAP Name(s) .....              | 171 |
| 3.1.6.7.3. Syntax (DAP) .....              | 171 |
| 3.1.6.7.4. Syntax (LDAP) .....             | 171 |
| 3.1.6.7.5. Example (DAP) .....             | 172 |
| 3.1.6.7.6. Example (LDAP) .....            | 172 |
| 3.1.6.8. X.121 Address .....               | 172 |
| 3.1.6.8.1. Abbreviation .....              | 172 |
| 3.1.6.8.2. LDAP Name(s) .....              | 172 |
| 3.1.6.8.3. Syntax (DAP) .....              | 172 |
| 3.1.6.8.4. Syntax (LDAP) .....             | 172 |
| 3.1.6.8.5. Example (DAP) .....             | 172 |
| 3.1.6.8.6. Example (LDAP) .....            | 172 |
| 3.1.7. Preference Attributes .....         | 172 |
| 3.1.7.1. Preferred Delivery Method .....   | 172 |
| 3.1.7.1.1. Abbreviation .....              | 173 |
| 3.1.7.1.2. LDAP Name(s) .....              | 173 |
| 3.1.7.1.3. Syntax (DAP) .....              | 173 |
| 3.1.7.1.4. Syntax (LDAP) .....             | 173 |
| 3.1.7.1.5. Example (DAP) .....             | 173 |

|                                              |     |
|----------------------------------------------|-----|
| 3.1.7.1.6. Example (LDAP) .....              | 173 |
| 3.1.8. OSI Applications .....                | 174 |
| 3.1.8.1. Application Entity .....            | 174 |
| 3.1.8.1.1. Abbreviation .....                | 174 |
| 3.1.8.1.2. LDAP Name(s) .....                | 174 |
| 3.1.8.1.3. Syntax (DAP) .....                | 174 |
| 3.1.8.1.4. Syntax (LDAP) .....               | 174 |
| 3.1.8.1.5. Example (DAP) .....               | 174 |
| 3.1.8.1.6. Example (LDAP) .....              | 174 |
| 3.1.8.2. Presentation Address .....          | 174 |
| 3.1.8.2.1. Abbreviation .....                | 174 |
| 3.1.8.2.2. Syntax (DAP) .....                | 174 |
| 3.1.8.2.3. Example (DAP) .....               | 174 |
| 3.1.8.3. Protocol Information .....          | 175 |
| 3.1.8.3.1. Abbreviation .....                | 175 |
| 3.1.8.3.2. Syntax (DAP) .....                | 175 |
| 3.1.8.3.3. Example (DAP) .....               | 175 |
| 3.1.8.4. Supported Application Context ..... | 175 |
| 3.1.8.4.1. Abbreviation .....                | 176 |
| 3.1.8.4.2. Syntax .....                      | 176 |
| 3.1.8.4.3. Example (dirxadm) .....           | 176 |
| 3.1.8.5. Transfer Syntaxes Supported .....   | 176 |
| 3.1.8.5.1. Abbreviation .....                | 176 |
| 3.1.8.5.2. Syntax .....                      | 176 |
| 3.1.8.5.3. Example (dirxadm) .....           | 177 |
| 3.1.8.6. Upper Layer Profile Info .....      | 177 |
| 3.1.8.6.1. Abbreviation .....                | 177 |
| 3.1.8.6.2. LDAP Name(s) .....                | 177 |
| 3.1.8.6.3. Syntax (DAP) .....                | 177 |
| 3.1.8.6.4. Syntax (LDAP) .....               | 177 |
| 3.1.8.6.5. Example (DAP) .....               | 177 |
| 3.1.8.6.6. Example (LDAP) .....              | 177 |
| 3.1.9. Relational Attributes .....           | 177 |
| 3.1.9.1. Aliased Object Name .....           | 177 |
| 3.1.9.1.1. Abbreviation .....                | 178 |
| 3.1.9.1.2. LDAP Name(s) .....                | 178 |
| 3.1.9.1.3. Syntax (DAP) .....                | 178 |
| 3.1.9.1.4. Syntax (LDAP) .....               | 178 |
| 3.1.9.1.5. Example (DAP) .....               | 178 |
| 3.1.9.1.6. Example (LDAP) .....              | 178 |
| 3.1.9.2. DirX Directory ImportFrom .....     | 178 |
| 3.1.9.2.1. Abbreviation .....                | 178 |

|                                         |     |
|-----------------------------------------|-----|
| 3.1.9.2.2. LDAP Name(s) .....           | 178 |
| 3.1.9.2.3. Syntax (DAP) .....           | 178 |
| 3.1.9.2.4. Syntax (LDAP) .....          | 178 |
| 3.1.9.2.5. Example (DAP) .....          | 178 |
| 3.1.9.2.6. Example (LDAP) .....         | 179 |
| 3.1.9.3. DirX Directory MemberUrl ..... | 179 |
| 3.1.9.3.1. Abbreviation .....           | 179 |
| 3.1.9.3.2. LDAP Name(s) .....           | 179 |
| 3.1.9.3.3. Syntax (DAP) .....           | 179 |
| 3.1.9.3.4. Syntax (LDAP) .....          | 179 |
| 3.1.9.3.5. Example (DAP) .....          | 179 |
| 3.1.9.3.6. Example (LDAP) .....         | 179 |
| 3.1.9.4. DN .....                       | 179 |
| 3.1.9.4.1. Abbreviation .....           | 179 |
| 3.1.9.4.2. LDAP Name(s) .....           | 180 |
| 3.1.9.4.3. Syntax (DAP) .....           | 180 |
| 3.1.9.4.4. Syntax (LDAP) .....          | 180 |
| 3.1.9.4.5. Example (DAP) .....          | 180 |
| 3.1.9.4.6. Example (LDAP) .....         | 180 |
| 3.1.9.5. Member .....                   | 180 |
| 3.1.9.5.1. Abbreviation .....           | 180 |
| 3.1.9.5.2. LDAP Name(s) .....           | 180 |
| 3.1.9.5.3. Syntax (DAP) .....           | 180 |
| 3.1.9.5.4. Syntax (LDAP) .....          | 180 |
| 3.1.9.5.5. Example (DAP) .....          | 180 |
| 3.1.9.5.6. Example (LDAP) .....         | 180 |
| 3.1.9.6. Owner .....                    | 181 |
| 3.1.9.6.1. Abbreviation .....           | 181 |
| 3.1.9.6.2. LDAP Name(s) .....           | 181 |
| 3.1.9.6.3. Syntax (DAP) .....           | 181 |
| 3.1.9.6.4. Syntax (LDAP) .....          | 181 |
| 3.1.9.6.5. Example (DAP) .....          | 181 |
| 3.1.9.6.6. Example (LDAP) .....         | 181 |
| 3.1.9.7. Role Occupant .....            | 181 |
| 3.1.9.7.1. Abbreviation .....           | 181 |
| 3.1.9.7.2. LDAP Name(s) .....           | 181 |
| 3.1.9.7.3. Syntax (DAP) .....           | 181 |
| 3.1.9.7.4. Syntax (LDAP) .....          | 182 |
| 3.1.9.7.5. Example (DAP) .....          | 182 |
| 3.1.9.7.6. Example (LDAP) .....         | 182 |
| 3.1.9.8. See Also .....                 | 182 |
| 3.1.9.8.1. Abbreviation .....           | 182 |

|                                        |     |
|----------------------------------------|-----|
| 3.1.9.8.2. LDAP Name(s) .....          | 182 |
| 3.1.9.8.3. Syntax (DAP) .....          | 182 |
| 3.1.9.8.4. Syntax (LDAP) .....         | 182 |
| 3.1.9.8.5. Example (DAP) .....         | 182 |
| 3.1.9.8.6. Example (LDAP) .....        | 182 |
| 3.1.9.9. Unique Member .....           | 182 |
| 3.1.9.9.1. Abbreviation .....          | 183 |
| 3.1.9.9.2. LDAP Name(s) .....          | 183 |
| 3.1.9.9.3. Syntax (DAP) .....          | 183 |
| 3.1.9.9.4. Syntax (LDAP) .....         | 183 |
| 3.1.9.9.5. Example (DAP) .....         | 183 |
| 3.1.9.9.6. Example (LDAP) .....        | 183 |
| 3.1.10. Miscellaneous Attributes ..... | 183 |
| 3.1.10.1. House Identifier .....       | 183 |
| 3.1.10.1.1. Abbreviation .....         | 183 |
| 3.1.10.1.2. LDAP Name(s) .....         | 183 |
| 3.1.10.1.3. Syntax (DAP) .....         | 183 |
| 3.1.10.1.4. Example (DAP) .....        | 184 |
| 3.1.10.1.5. Example (LDAP) .....       | 184 |
| 3.1.10.2. Knowledge Information .....  | 184 |
| 3.1.10.2.1. Abbreviation .....         | 184 |
| 3.1.10.2.2. LDAP Name(s) .....         | 184 |
| 3.1.10.2.3. Syntax (DAP) .....         | 184 |
| 3.1.10.2.4. Example (DAP) .....        | 184 |
| 3.1.10.2.5. Example (LDAP) .....       | 184 |
| 3.1.10.3. Labeled URI .....            | 184 |
| 3.1.10.3.1. Abbreviation .....         | 184 |
| 3.1.10.3.2. LDAP Name(s) .....         | 185 |
| 3.1.10.3.3. Syntax (DAP) .....         | 185 |
| 3.1.10.3.4. Example (DAP) .....        | 185 |
| 3.1.10.3.5. Example (LDAP) .....       | 185 |
| 3.1.10.4. Object Class .....           | 185 |
| 3.1.10.4.1. Abbreviation .....         | 185 |
| 3.1.10.4.2. LDAP Name(s) .....         | 186 |
| 3.1.10.4.3. Syntax (DAP) .....         | 186 |
| 3.1.10.4.4. Syntax (LDAP) .....        | 186 |
| 3.1.10.4.5. Example (DAP) .....        | 186 |
| 3.1.10.4.6. Example (LDAP) .....       | 186 |
| 3.1.10.5. RFC822 Mailbox .....         | 186 |
| 3.1.10.5.1. Abbreviation .....         | 186 |
| 3.1.10.5.2. LDAP Name(s) .....         | 186 |
| 3.1.10.5.3. Syntax (DAP) .....         | 186 |

|                                                      |     |
|------------------------------------------------------|-----|
| 3.1.10.5.4. Example (DAP) .....                      | 186 |
| 3.1.10.5.5. Example (LDAP) .....                     | 186 |
| 3.1.10.6. Title .....                                | 187 |
| 3.1.10.6.1. Abbreviation .....                       | 187 |
| 3.1.10.6.2. LDAP Name(s) .....                       | 187 |
| 3.1.10.6.3. Syntax (DAP) .....                       | 187 |
| 3.1.10.6.4. Example (DAP) .....                      | 187 |
| 3.1.10.6.5. Example (LDAP) .....                     | 187 |
| 3.1.11. Collective Attributes .....                  | 187 |
| 3.1.11.1. Collective Fax Telephone Number .....      | 187 |
| 3.1.11.1.1. Abbreviation .....                       | 187 |
| 3.1.11.1.2. LDAP Name(s) .....                       | 188 |
| 3.1.11.1.3. Syntax (DAP) .....                       | 188 |
| 3.1.11.1.4. Syntax (LDAP) .....                      | 188 |
| 3.1.11.1.5. Example (DAP) .....                      | 188 |
| 3.1.11.1.6. Example (LDAP) .....                     | 188 |
| 3.1.11.2. Collective International ISDN Number ..... | 188 |
| 3.1.11.2.1. Abbreviation .....                       | 188 |
| 3.1.11.2.2. LDAP Name(s) .....                       | 188 |
| 3.1.11.2.3. Syntax (DAP) .....                       | 188 |
| 3.1.11.2.4. Syntax (LDAP) .....                      | 188 |
| 3.1.11.2.5. Example (DAP) .....                      | 188 |
| 3.1.11.2.6. Example (LDAP) .....                     | 188 |
| 3.1.11.3. Collective Locality Name .....             | 189 |
| 3.1.11.3.1. Abbreviation .....                       | 189 |
| 3.1.11.3.2. LDAP Name(s) .....                       | 189 |
| 3.1.11.3.3. Syntax (DAP) .....                       | 189 |
| 3.1.11.3.4. Example (DAP) .....                      | 189 |
| 3.1.11.3.5. Example (LDAP) .....                     | 189 |
| 3.1.11.4. Collective Organization Name .....         | 189 |
| 3.1.11.4.1. Abbreviation .....                       | 189 |
| 3.1.11.4.2. LDAP Name(s) .....                       | 189 |
| 3.1.11.4.3. Syntax (DAP) .....                       | 189 |
| 3.1.11.4.4. Example (DAP) .....                      | 189 |
| 3.1.11.4.5. Example (LDAP) .....                     | 190 |
| 3.1.11.5. Collective Organizational Unit Name .....  | 190 |
| 3.1.11.5.1. Abbreviation .....                       | 190 |
| 3.1.11.5.2. LDAP Name(s) .....                       | 190 |
| 3.1.11.5.3. Syntax (DAP) .....                       | 190 |
| 3.1.11.5.4. Example (DAP) .....                      | 190 |
| 3.1.11.5.5. Example (LDAP) .....                     | 190 |
| 3.1.11.6. Collective Postal Address .....            | 190 |

|                                                          |     |
|----------------------------------------------------------|-----|
| 3.1.11.6.1. Abbreviation .....                           | 190 |
| 3.1.11.6.2. LDAP Name(s) .....                           | 190 |
| 3.1.11.6.3. Syntax (DAP) .....                           | 191 |
| 3.1.11.6.4. Syntax (LDAP) .....                          | 191 |
| 3.1.11.6.5. Example (DAP) .....                          | 191 |
| 3.1.11.6.6. Example (LDAP) .....                         | 191 |
| 3.1.11.7. Collective Postal Code .....                   | 191 |
| 3.1.11.7.1. Abbreviation .....                           | 191 |
| 3.1.11.7.2. LDAP Name(s) .....                           | 191 |
| 3.1.11.7.3. Syntax (DAP) .....                           | 191 |
| 3.1.11.7.4. Example (DAP) .....                          | 191 |
| 3.1.11.7.5. Example (LDAP) .....                         | 191 |
| 3.1.11.8. Collective Physical Delivery Office Name ..... | 191 |
| 3.1.11.8.1. Abbreviation .....                           | 192 |
| 3.1.11.8.2. LDAP Name(s) .....                           | 192 |
| 3.1.11.8.3. Syntax (DAP) .....                           | 192 |
| 3.1.11.8.4. Example (DAP) .....                          | 192 |
| 3.1.11.8.5. Example (LDAP) .....                         | 192 |
| 3.1.11.9. Collective Post Office Box .....               | 192 |
| 3.1.11.9.1. Abbreviation .....                           | 192 |
| 3.1.11.9.2. LDAP Name(s) .....                           | 192 |
| 3.1.11.9.3. Syntax (DAP) .....                           | 192 |
| 3.1.11.9.4. Example (DAP) .....                          | 192 |
| 3.1.11.9.5. Example (LDAP) .....                         | 192 |
| 3.1.11.10. Collective State or Province Name .....       | 193 |
| 3.1.11.10.1. Abbreviation .....                          | 193 |
| 3.1.11.10.2. LDAP Name(s) .....                          | 193 |
| 3.1.11.10.3. Syntax (DAP) .....                          | 193 |
| 3.1.11.10.4. Example (DAP) .....                         | 193 |
| 3.1.11.10.5. Example (LDAP) .....                        | 193 |
| 3.1.11.11. Collective Street Address .....               | 193 |
| 3.1.11.11.1. Abbreviation .....                          | 193 |
| 3.1.11.11.2. LDAP Name(s) .....                          | 193 |
| 3.1.11.11.3. Syntax (DAP) .....                          | 193 |
| 3.1.11.11.4. Example (DAP) .....                         | 193 |
| 3.1.11.11.5. Example (LDAP) .....                        | 193 |
| 3.1.11.12. Collective Telephone Number .....             | 194 |
| 3.1.11.12.1. Abbreviation .....                          | 194 |
| 3.1.11.12.2. LDAP Name(s) .....                          | 194 |
| 3.1.11.12.3. Syntax (DAP) .....                          | 194 |
| 3.1.11.12.4. Syntax (LDAP) .....                         | 194 |
| 3.1.11.12.5. Example (DAP) .....                         | 194 |



|                                                     |     |
|-----------------------------------------------------|-----|
| 3.1.11.12.6. Example (LDAP) .....                   | 194 |
| 3.1.11.13. Collective TTX Terminal Identifier ..... | 194 |
| 3.1.11.13.1. Abbreviation .....                     | 194 |
| 3.1.11.13.2. LDAP Name(s) .....                     | 194 |
| 3.1.11.13.3. Syntax (DAP) .....                     | 194 |
| 3.1.11.13.4. Syntax (LDAP) .....                    | 195 |
| 3.1.11.13.5. Example (DAP) .....                    | 195 |
| 3.1.11.13.6. Example (LDAP) .....                   | 195 |
| 3.1.11.14. Collective Telex Number .....            | 195 |
| 3.1.11.14.1. Abbreviation .....                     | 195 |
| 3.1.11.14.2. LDAP Name(s) .....                     | 195 |
| 3.1.11.14.3. Syntax (DAP) .....                     | 195 |
| 3.1.11.14.4. Syntax (LDAP) .....                    | 195 |
| 3.1.11.14.5. Example (DAP) .....                    | 195 |
| 3.1.11.14.6. Example (LDAP) .....                   | 195 |
| 3.1.12. Attributes of the LDAP Root Subentry .....  | 195 |
| 3.1.12.1. LDAP Naming Contexts .....                | 196 |
| 3.1.12.1.1. Abbreviation .....                      | 196 |
| 3.1.12.1.2. LDAP Name(s) .....                      | 196 |
| 3.1.12.1.3. Syntax (DAP) .....                      | 196 |
| 3.1.12.1.4. Syntax (LDAP) .....                     | 196 |
| 3.1.12.1.5. Example (DAP) .....                     | 196 |
| 3.1.12.1.6. Example (LDAP) .....                    | 196 |
| 3.1.12.2. Alternate Servers .....                   | 197 |
| 3.1.12.2.1. Abbreviation .....                      | 197 |
| 3.1.12.2.2. LDAP Name(s) .....                      | 197 |
| 3.1.12.2.3. Syntax (DAP) .....                      | 197 |
| 3.1.12.2.4. Example (DAP) .....                     | 197 |
| 3.1.12.2.5. Example (LDAP) .....                    | 197 |
| 3.1.12.3. Supported Extensions .....                | 197 |
| 3.1.12.3.1. Abbreviation .....                      | 198 |
| 3.1.12.3.2. LDAP Name(s) .....                      | 198 |
| 3.1.12.3.3. Syntax (DAP) .....                      | 198 |
| 3.1.12.3.4. Syntax (LDAP) .....                     | 198 |
| 3.1.12.3.5. Example (DAP) .....                     | 198 |
| 3.1.12.3.6. Example (LDAP) .....                    | 198 |
| 3.1.12.4. Supported Control .....                   | 198 |
| 3.1.12.4.1. Abbreviation .....                      | 201 |
| 3.1.12.4.2. LDAP Name(s) .....                      | 201 |
| 3.1.12.4.3. Syntax (DAP) .....                      | 201 |
| 3.1.12.4.4. Syntax (LDAP) .....                     | 201 |
| 3.1.12.4.5. Example (DAP) .....                     | 201 |

|                                          |     |
|------------------------------------------|-----|
| 3.1.12.4.6. Example (LDAP) .....         | 201 |
| 3.1.12.5. Supported SASL Mechanism ..... | 202 |
| 3.1.12.5.1. Abbreviation .....           | 202 |
| 3.1.12.5.2. LDAP Name(s) .....           | 202 |
| 3.1.12.5.3. Syntax (DAP) .....           | 202 |
| 3.1.12.5.4. Example (DAP) .....          | 202 |
| 3.1.12.5.5. Example (LDAP) .....         | 202 |
| 3.1.12.6. Supported LDAP Versions .....  | 202 |
| 3.1.12.6.1. Abbreviation .....           | 202 |
| 3.1.12.6.2. LDAP Name(s) .....           | 202 |
| 3.1.12.6.3. Syntax (DAP) .....           | 202 |
| 3.1.12.6.4. Syntax (LDAP) .....          | 203 |
| 3.1.12.6.5. Example (DAP) .....          | 203 |
| 3.1.12.6.6. Example (LDAP) .....         | 203 |
| 3.1.12.7. Og Supported Profile .....     | 203 |
| 3.1.12.7.1. Abbreviation .....           | 203 |
| 3.1.12.7.2. LDAP Name(s) .....           | 203 |
| 3.1.12.7.3. Syntax (DAP) .....           | 203 |
| 3.1.12.7.4. Syntax (LDAP) .....          | 203 |
| 3.1.12.7.5. Example (DAP) .....          | 203 |
| 3.1.12.7.6. Example (LDAP) .....         | 203 |
| 3.1.12.8. Vendor Name .....              | 204 |
| 3.1.12.8.1. Abbreviation .....           | 204 |
| 3.1.12.8.2. LDAP Name(s) .....           | 204 |
| 3.1.12.8.3. Syntax (DAP) .....           | 204 |
| 3.1.12.8.4. Syntax (LDAP) .....          | 204 |
| 3.1.12.8.5. Example (DAP) .....          | 204 |
| 3.1.12.8.6. Example (LDAP) .....         | 204 |
| 3.1.12.9. Vendor Version .....           | 204 |
| 3.1.12.9.1. Abbreviation .....           | 204 |
| 3.1.12.9.2. LDAP Name(s) .....           | 204 |
| 3.1.12.9.3. Syntax (DAP) .....           | 204 |
| 3.1.12.9.4. Syntax (LDAP) .....          | 204 |
| 3.1.12.9.5. Example (DAP) .....          | 205 |
| 3.1.12.9.6. Example (LDAP) .....         | 205 |
| 3.1.12.10. LDAP Subschema Subentry ..... | 205 |
| 3.1.12.10.1. Abbreviation .....          | 205 |
| 3.1.12.10.2. LDAP Name(s) .....          | 205 |
| 3.1.12.10.3. Syntax (DAP) .....          | 205 |
| 3.1.12.10.4. Syntax (LDAP) .....         | 205 |
| 3.1.12.10.5. Example (DAP) .....         | 205 |
| 3.1.12.10.6. Example (LDAP) .....        | 205 |

|                                                        |     |
|--------------------------------------------------------|-----|
| 3.1.13. Attributes of the Schema Subentry .....        | 205 |
| 3.1.13.1. LDAP AT .....                                | 206 |
| 3.1.13.1.1. Abbreviation .....                         | 206 |
| 3.1.13.1.2. LDAP Name(s) .....                         | 207 |
| 3.1.13.1.3. Syntax (LDAP) .....                        | 207 |
| 3.1.13.1.4. Example (LDAP) .....                       | 207 |
| 3.1.13.2. LDAP OC .....                                | 207 |
| 3.1.13.2.1. Abbreviation .....                         | 207 |
| 3.1.13.2.2. LDAP Name(s) .....                         | 207 |
| 3.1.13.2.3. Syntax (LDAP) .....                        | 207 |
| 3.1.13.2.4. Example (LDAP) .....                       | 207 |
| 3.1.14. Attributes for LDAP Server Configuration ..... | 208 |
| 3.1.14.1. LDAP Search Service Controls .....           | 210 |
| 3.1.14.1.1. Abbreviation .....                         | 211 |
| 3.1.14.1.2. LDAP Name(s) .....                         | 212 |
| 3.1.14.1.3. Syntax (DAP) .....                         | 212 |
| 3.1.14.1.4. Example (DAP) .....                        | 212 |
| 3.1.14.1.5. Example (LDAP) .....                       | 212 |
| 3.1.14.2. LDAP Compare Service Controls .....          | 212 |
| 3.1.14.2.1. Abbreviation .....                         | 212 |
| 3.1.14.2.2. LDAP Name(s) .....                         | 212 |
| 3.1.14.2.3. Syntax (DAP) .....                         | 212 |
| 3.1.14.2.4. Example (DAP) .....                        | 212 |
| 3.1.14.2.5. Example (LDAP) .....                       | 212 |
| 3.1.14.3. LDAP Add Service Controls .....              | 213 |
| 3.1.14.3.1. Abbreviation .....                         | 213 |
| 3.1.14.3.2. LDAP Name(s) .....                         | 213 |
| 3.1.14.3.3. Syntax (DAP) .....                         | 213 |
| 3.1.14.3.4. Example (DAP) .....                        | 213 |
| 3.1.14.3.5. Example (LDAP) .....                       | 213 |
| 3.1.14.4. LDAP Remove Service Controls .....           | 213 |
| 3.1.14.4.1. Abbreviation .....                         | 213 |
| 3.1.14.4.2. LDAP Name(s) .....                         | 213 |
| 3.1.14.4.3. Syntax (DAP) .....                         | 213 |
| 3.1.14.4.4. Example (DAP) .....                        | 213 |
| 3.1.14.4.5. Example (LDAP) .....                       | 214 |
| 3.1.14.5. LDAP Modify Service Controls .....           | 214 |
| 3.1.14.5.1. Abbreviation .....                         | 214 |
| 3.1.14.5.2. LDAP Name(s) .....                         | 214 |
| 3.1.14.5.3. Syntax (DAP) .....                         | 214 |
| 3.1.14.5.4. Example (DAP) .....                        | 214 |
| 3.1.14.5.5. Example (LDAP) .....                       | 214 |

|                                                     |     |
|-----------------------------------------------------|-----|
| 3.1.14.6. LDAP ModifyDN Service Controls .....      | 214 |
| 3.1.14.6.1. Abbreviation .....                      | 214 |
| 3.1.14.6.2. LDAP Name(s) .....                      | 214 |
| 3.1.14.6.3. Syntax (DAP) .....                      | 215 |
| 3.1.14.6.4. Example (DAP) .....                     | 215 |
| 3.1.14.6.5. Example (LDAP) .....                    | 215 |
| 3.1.14.7. LDAP Maximum Requested Attributes .....   | 215 |
| 3.1.14.7.1. Abbreviation .....                      | 215 |
| 3.1.14.7.2. LDAP Name(s) .....                      | 215 |
| 3.1.14.7.3. Syntax (DAP) .....                      | 215 |
| 3.1.14.7.4. Example (DAP) .....                     | 215 |
| 3.1.14.7.5. Example (LDAP) .....                    | 216 |
| 3.1.14.8. LDAP Maximum Filter Items .....           | 216 |
| 3.1.14.8.1. Abbreviation .....                      | 216 |
| 3.1.14.8.2. LDAP Name(s) .....                      | 216 |
| 3.1.14.8.3. Syntax (DAP) .....                      | 216 |
| 3.1.14.8.4. Example (DAP) .....                     | 216 |
| 3.1.14.8.5. Example (LDAP) .....                    | 216 |
| 3.1.14.9. LDAP Port Number .....                    | 216 |
| 3.1.14.9.1. Abbreviation .....                      | 217 |
| 3.1.14.9.2. LDAP Name(s) .....                      | 217 |
| 3.1.14.9.3. Syntax (DAP) .....                      | 217 |
| 3.1.14.9.4. Syntax (LDAP) .....                     | 217 |
| 3.1.14.9.5. Example (DAP) .....                     | 217 |
| 3.1.14.9.6. Example (LDAP) .....                    | 217 |
| 3.1.14.10. LDAP Secure Port Number .....            | 217 |
| 3.1.14.10.1. Abbreviation .....                     | 217 |
| 3.1.14.10.2. LDAP Name(s) .....                     | 217 |
| 3.1.14.10.3. Syntax (DAP) .....                     | 217 |
| 3.1.14.10.4. Syntax (LDAP) .....                    | 218 |
| 3.1.14.10.5. Example (DAP) .....                    | 218 |
| 3.1.14.10.6. Example (LDAP) .....                   | 218 |
| 3.1.14.11. LDAP SSL Configuration Subentry CN ..... | 218 |
| 3.1.14.11.1. Abbreviation .....                     | 218 |
| 3.1.14.11.2. LDAP Name(s) .....                     | 218 |
| 3.1.14.11.3. Syntax (DAP) .....                     | 218 |
| 3.1.14.11.4. Syntax (LDAP) .....                    | 218 |
| 3.1.14.11.5. Example (DAP) .....                    | 218 |
| 3.1.14.11.6. Example (LDAP) .....                   | 218 |
| 3.1.14.12. LDAP Start TLS Enabled .....             | 219 |
| 3.1.14.12.1. Abbreviation .....                     | 219 |
| 3.1.14.12.2. LDAP Name(s) .....                     | 219 |

|                                               |     |
|-----------------------------------------------|-----|
| 3.1.14.12.3. Syntax (DAP) .....               | 219 |
| 3.1.14.12.4. Syntax (LDAP) .....              | 219 |
| 3.1.14.12.5. Example (DAP) .....              | 219 |
| 3.1.14.12.6. Example (LDAP) .....             | 219 |
| 3.1.14.13. LDAP Max Connections .....         | 220 |
| 3.1.14.13.1. Abbreviation .....               | 220 |
| 3.1.14.13.2. LDAP Name(s) .....               | 220 |
| 3.1.14.13.3. Syntax (DAP) .....               | 220 |
| 3.1.14.13.4. Syntax (LDAP) .....              | 220 |
| 3.1.14.13.5. Example (DAP) .....              | 220 |
| 3.1.14.13.6. Example (LDAP) .....             | 220 |
| 3.1.14.14. LDAP Anonym DAP Pool Size .....    | 221 |
| 3.1.14.14.1. Abbreviation .....               | 221 |
| 3.1.14.14.2. LDAP Name(s) .....               | 221 |
| 3.1.14.14.3. Syntax (DAP) .....               | 221 |
| 3.1.14.14.4. Syntax (LDAP) .....              | 221 |
| 3.1.14.14.5. Example (DAP) .....              | 221 |
| 3.1.14.14.6. Example (LDAP) .....             | 221 |
| 3.1.14.15. LDAP Connection Idle Time .....    | 221 |
| 3.1.14.15.1. Abbreviation .....               | 221 |
| 3.1.14.15.2. LDAP Name(s) .....               | 222 |
| 3.1.14.15.3. Syntax (DAP) .....               | 222 |
| 3.1.14.15.4. Syntax (LDAP) .....              | 222 |
| 3.1.14.15.5. Example (DAP) .....              | 222 |
| 3.1.14.15.6. Example (LDAP) .....             | 222 |
| 3.1.14.16. LDAP Unbind Delay Time .....       | 222 |
| 3.1.14.16.1. Abbreviation .....               | 223 |
| 3.1.14.16.2. LDAP Name(s) .....               | 223 |
| 3.1.14.16.3. Syntax (DAP) .....               | 223 |
| 3.1.14.16.4. Syntax (LDAP) .....              | 223 |
| 3.1.14.16.5. Example (DAP) .....              | 223 |
| 3.1.14.16.6. Example (LDAP) .....             | 223 |
| 3.1.14.17. LDAP Backend Sharing .....         | 223 |
| 3.1.14.17.1. Abbreviation .....               | 224 |
| 3.1.14.17.2. LDAP Name(s) .....               | 224 |
| 3.1.14.17.3. Syntax (DAP) .....               | 224 |
| 3.1.14.17.4. Syntax (LDAP) .....              | 224 |
| 3.1.14.17.5. Example (DAP) .....              | 224 |
| 3.1.14.17.6. Example (LDAP) .....             | 224 |
| 3.1.14.18. LDAP Maximum DAP Share Count ..... | 224 |
| 3.1.14.18.1. Abbreviation .....               | 225 |
| 3.1.14.18.2. LDAP Name(s) .....               | 225 |

|                                             |     |
|---------------------------------------------|-----|
| 3.1.14.18.3. Syntax (DAP) .....             | 225 |
| 3.1.14.18.4. Syntax (LDAP) .....            | 225 |
| 3.1.14.18.5. Example (DAP) .....            | 225 |
| 3.1.14.18.6. Example (LDAP) .....           | 225 |
| 3.1.14.19. LDAP Allow IP List .....         | 225 |
| 3.1.14.19.1. Abbreviation .....             | 226 |
| 3.1.14.19.2. LDAP Name(s) .....             | 226 |
| 3.1.14.19.3. Syntax (DAP) .....             | 226 |
| 3.1.14.19.4. Example (DAP) .....            | 226 |
| 3.1.14.19.5. Example (LDAP) .....           | 226 |
| 3.1.14.20. LDAP Deny IP List .....          | 226 |
| 3.1.14.20.1. Abbreviation .....             | 227 |
| 3.1.14.20.2. LDAP Name(s) .....             | 227 |
| 3.1.14.20.3. Syntax (DAP) .....             | 227 |
| 3.1.14.20.4. Example (DAP) .....            | 227 |
| 3.1.14.20.5. Example (LDAP) .....           | 227 |
| 3.1.14.21. LDAP Allowed Users List .....    | 227 |
| 3.1.14.21.1. Abbreviation .....             | 228 |
| 3.1.14.21.2. LDAP Name(s) .....             | 228 |
| 3.1.14.21.3. Syntax (DAP) .....             | 228 |
| 3.1.14.21.4. Example (DAP) .....            | 229 |
| 3.1.14.21.5. Example (LDAP) .....           | 229 |
| 3.1.14.22. LDAP Allowed Groups .....        | 229 |
| 3.1.14.22.1. Abbreviation .....             | 230 |
| 3.1.14.22.2. LDAP Name(s) .....             | 230 |
| 3.1.14.22.3. Syntax (DAP) .....             | 230 |
| 3.1.14.22.4. Example (DAP) .....            | 230 |
| 3.1.14.22.5. Example (LDAP) .....           | 230 |
| 3.1.14.23. LDAP Denied Users List .....     | 230 |
| 3.1.14.23.1. Abbreviation .....             | 231 |
| 3.1.14.23.2. LDAP Name(s) .....             | 231 |
| 3.1.14.23.3. Syntax (DAP) .....             | 231 |
| 3.1.14.23.4. Example (DAP) .....            | 231 |
| 3.1.14.23.5. Example (LDAP) .....           | 232 |
| 3.1.14.24. LDAP Denied Groups .....         | 232 |
| 3.1.14.24.1. Abbreviation .....             | 232 |
| 3.1.14.24.2. LDAP Name(s) .....             | 232 |
| 3.1.14.24.3. Syntax (DAP) .....             | 232 |
| 3.1.14.24.4. Example (DAP) .....            | 233 |
| 3.1.14.24.5. Example (LDAP) .....           | 233 |
| 3.1.14.25. LDAP Deny Anonymous Access ..... | 233 |
| 3.1.14.25.1. Abbreviation .....             | 233 |

|                                                   |     |
|---------------------------------------------------|-----|
| 3.1.14.25.2. LDAP Name(s) .....                   | 233 |
| 3.1.14.25.3. Syntax (DAP) .....                   | 233 |
| 3.1.14.25.4. Syntax (LDAP) .....                  | 234 |
| 3.1.14.25.5. Example (DAP) .....                  | 234 |
| 3.1.14.25.6. Example (LDAP) .....                 | 234 |
| 3.1.14.26. LDAP Only Allow Anonymous Access ..... | 234 |
| 3.1.14.26.1. Abbreviation .....                   | 234 |
| 3.1.14.26.2. LDAP Name(s) .....                   | 234 |
| 3.1.14.26.3. Syntax (DAP) .....                   | 234 |
| 3.1.14.26.4. Syntax (LDAP) .....                  | 235 |
| 3.1.14.26.5. Example (DAP) .....                  | 235 |
| 3.1.14.26.6. Example (LDAP) .....                 | 235 |
| 3.1.14.27. LDAP User Policies .....               | 235 |
| 3.1.14.27.1. Abbreviation .....                   | 241 |
| 3.1.14.27.2. LDAP Name(s) .....                   | 241 |
| 3.1.14.27.3. Syntax (DAP) .....                   | 241 |
| 3.1.14.27.4. Example (DAP) .....                  | 241 |
| 3.1.14.27.5. Example (LDAP) .....                 | 242 |
| 3.1.14.28. LDAP Group Policies .....              | 242 |
| 3.1.14.28.1. Abbreviation .....                   | 242 |
| 3.1.14.28.2. LDAP Name(s) .....                   | 242 |
| 3.1.14.28.3. Syntax (DAP) .....                   | 242 |
| 3.1.14.28.4. Example (DAP) .....                  | 242 |
| 3.1.14.28.5. Example (LDAP) .....                 | 243 |
| 3.1.14.29. LDAP Listen IP List .....              | 243 |
| 3.1.14.29.1. Abbreviation .....                   | 243 |
| 3.1.14.29.2. LDAP Name(s) .....                   | 243 |
| 3.1.14.29.3. Syntax (DAP) .....                   | 243 |
| 3.1.14.29.4. Syntax (LDAP) .....                  | 243 |
| 3.1.14.29.5. Example (DAP) .....                  | 244 |
| 3.1.14.29.6. Example (LDAP) .....                 | 244 |
| 3.1.14.30. LDAP ASN1 Header .....                 | 244 |
| 3.1.14.30.1. Abbreviation .....                   | 244 |
| 3.1.14.30.2. LDAP Name(s) .....                   | 244 |
| 3.1.14.30.3. Syntax (DAP) .....                   | 244 |
| 3.1.14.30.4. Syntax (LDAP) .....                  | 244 |
| 3.1.14.30.5. Example (DAP) .....                  | 244 |
| 3.1.14.30.6. Example (LDAP) .....                 | 244 |
| 3.1.14.31. LDAP Charset Conv Request .....        | 244 |
| 3.1.14.31.1. Abbreviation .....                   | 245 |
| 3.1.14.31.2. LDAP Name(s) .....                   | 245 |
| 3.1.14.31.3. Syntax (DAP) .....                   | 245 |

|                                            |     |
|--------------------------------------------|-----|
| 3.1.14.31.4. Example (DAP) .....           | 245 |
| 3.1.14.31.5. Example (LDAP) .....          | 245 |
| 3.1.14.32. LDAP Charset Conv Result .....  | 245 |
| 3.1.14.32.1. Abbreviation .....            | 246 |
| 3.1.14.32.2. LDAP Name(s) .....            | 246 |
| 3.1.14.32.3. Syntax (DAP) .....            | 246 |
| 3.1.14.32.4. Example (DAP) .....           | 246 |
| 3.1.14.32.5. Example (LDAP) .....          | 246 |
| 3.1.14.33. LDAP Read Only Server .....     | 246 |
| 3.1.14.33.1. Abbreviation .....            | 246 |
| 3.1.14.33.2. LDAP Name(s) .....            | 247 |
| 3.1.14.33.3. Syntax (DAP) .....            | 247 |
| 3.1.14.33.4. Syntax (LDAP) .....           | 247 |
| 3.1.14.33.5. Example (DAP) .....           | 247 |
| 3.1.14.33.6. Example (LDAP) .....          | 247 |
| 3.1.14.34. LDAP Thread Pool Size .....     | 247 |
| 3.1.14.34.1. Abbreviation .....            | 247 |
| 3.1.14.34.2. LDAP Name(s) .....            | 247 |
| 3.1.14.34.3. Syntax (DAP) .....            | 248 |
| 3.1.14.34.4. Syntax (LDAP) .....           | 248 |
| 3.1.14.34.5. Example (DAP) .....           | 248 |
| 3.1.14.34.6. Example (LDAP) .....          | 248 |
| 3.1.14.35. LDAP Socket Keep Alive .....    | 248 |
| 3.1.14.35.1. Abbreviation .....            | 248 |
| 3.1.14.35.2. LDAP Name(s) .....            | 248 |
| 3.1.14.35.3. Syntax (DAP) .....            | 248 |
| 3.1.14.35.4. Syntax (LDAP) .....           | 248 |
| 3.1.14.35.5. Example (DAP) .....           | 249 |
| 3.1.14.35.6. Example (LDAP) .....          | 249 |
| 3.1.14.36. LDAP Asynchronous Sockets ..... | 249 |
| 3.1.14.36.1. Abbreviation .....            | 249 |
| 3.1.14.36.2. LDAP Name(s) .....            | 249 |
| 3.1.14.36.3. Syntax (DAP) .....            | 249 |
| 3.1.14.36.4. Syntax (LDAP) .....           | 249 |
| 3.1.14.36.5. Example (DAP) .....           | 249 |
| 3.1.14.36.6. Example (LDAP) .....          | 250 |
| 3.1.14.37. LDAP Max Send Timeout .....     | 250 |
| 3.1.14.37.1. Abbreviation .....            | 250 |
| 3.1.14.37.2. LDAP Name(s) .....            | 250 |
| 3.1.14.37.3. Syntax (DAP) .....            | 250 |
| 3.1.14.37.4. Syntax (LDAP) .....           | 250 |
| 3.1.14.37.5. Example (DAP) .....           | 250 |



|                                                                |     |
|----------------------------------------------------------------|-----|
| 3.1.14.37.6. Example (LDAP) .....                              | 250 |
| 3.1.14.38. LDAP Max Recv Timeout .....                         | 251 |
| 3.1.14.38.1. Abbreviation .....                                | 251 |
| 3.1.14.38.2. LDAP Name(s) .....                                | 251 |
| 3.1.14.38.3. Syntax (DAP) .....                                | 251 |
| 3.1.14.38.4. Syntax (LDAP) .....                               | 251 |
| 3.1.14.38.5. Example (DAP) .....                               | 251 |
| 3.1.14.38.6. Example (LDAP) .....                              | 251 |
| 3.1.14.39. LDAP Max Incomplete Operations Per Connection ..... | 251 |
| 3.1.14.39.1. Abbreviation .....                                | 252 |
| 3.1.14.39.2. LDAP Name(s) .....                                | 252 |
| 3.1.14.39.3. Syntax (DAP) .....                                | 252 |
| 3.1.14.39.4. Syntax (LDAP) .....                               | 252 |
| 3.1.14.39.5. Example (DAP) .....                               | 252 |
| 3.1.14.39.6. Example (LDAP) .....                              | 252 |
| 3.1.14.40. LDAP OpStack Size Limit .....                       | 252 |
| 3.1.14.40.1. Abbreviation .....                                | 253 |
| 3.1.14.40.2. LDAP Name(s) .....                                | 253 |
| 3.1.14.40.3. Syntax (DAP) .....                                | 253 |
| 3.1.14.40.4. Syntax (LDAP) .....                               | 253 |
| 3.1.14.40.5. Example (DAP) .....                               | 253 |
| 3.1.14.40.6. Example (LDAP) .....                              | 253 |
| 3.1.14.41. LDAP Overflow Threads .....                         | 253 |
| 3.1.14.41.1. Abbreviation .....                                | 254 |
| 3.1.14.41.2. LDAP Name(s) .....                                | 254 |
| 3.1.14.41.3. Syntax (DAP) .....                                | 254 |
| 3.1.14.41.4. Syntax (LDAP) .....                               | 254 |
| 3.1.14.41.5. Example (DAP) .....                               | 254 |
| 3.1.14.41.6. Example (LDAP) .....                              | 254 |
| 3.1.14.42. LDAP Cache .....                                    | 254 |
| 3.1.14.42.1. Abbreviation .....                                | 255 |
| 3.1.14.42.2. LDAP Name(s) .....                                | 255 |
| 3.1.14.42.3. Syntax (DAP) .....                                | 255 |
| 3.1.14.42.4. Syntax (LDAP) .....                               | 255 |
| 3.1.14.42.5. Example (DAP) .....                               | 255 |
| 3.1.14.42.6. Example (LDAP) .....                              | 255 |
| 3.1.14.43. LDAP Cache Max Cached Elements .....                | 255 |
| 3.1.14.43.1. Abbreviation .....                                | 256 |
| 3.1.14.43.2. LDAP Name(s) .....                                | 256 |
| 3.1.14.43.3. Syntax (DAP) .....                                | 256 |
| 3.1.14.43.4. Syntax (LDAP) .....                               | 256 |
| 3.1.14.43.5. Example (DAP) .....                               | 256 |

|                                           |     |
|-------------------------------------------|-----|
| 3.1.14.43.6. Example (LDAP)               | 256 |
| 3.1.14.44. LDAP Cache Maximum Memory Size | 256 |
| 3.1.14.44.1. Abbreviation                 | 257 |
| 3.1.14.44.2. LDAP Name(s)                 | 257 |
| 3.1.14.44.3. Syntax (DAP)                 | 257 |
| 3.1.14.44.4. Syntax (LDAP)                | 257 |
| 3.1.14.44.5. Example (DAP)                | 257 |
| 3.1.14.44.6. Example (LDAP)               | 257 |
| 3.1.14.45. LDAP Cache Min Cache Time      | 257 |
| 3.1.14.45.1. Abbreviation                 | 257 |
| 3.1.14.45.2. LDAP Name(s)                 | 257 |
| 3.1.14.45.3. Syntax (DAP)                 | 257 |
| 3.1.14.45.4. Syntax (LDAP)                | 258 |
| 3.1.14.45.5. Example (DAP)                | 258 |
| 3.1.14.45.6. Example (LDAP)               | 258 |
| 3.1.14.46. LDAP Cache Max Cache Time      | 258 |
| 3.1.14.46.1. Abbreviation                 | 258 |
| 3.1.14.46.2. LDAP Name(s)                 | 258 |
| 3.1.14.46.3. Syntax (DAP)                 | 258 |
| 3.1.14.46.4. Syntax (LDAP)                | 258 |
| 3.1.14.46.5. Example (DAP)                | 258 |
| 3.1.14.46.6. Example (LDAP)               | 258 |
| 3.1.14.47. LDAP Cache Min Cache Entries   | 259 |
| 3.1.14.47.1. Abbreviation                 | 259 |
| 3.1.14.47.2. LDAP Name(s)                 | 259 |
| 3.1.14.47.3. Syntax (DAP)                 | 259 |
| 3.1.14.47.4. Syntax (LDAP)                | 259 |
| 3.1.14.47.5. Example (DAP)                | 259 |
| 3.1.14.47.6. Example (LDAP)               | 259 |
| 3.1.14.48. LDAP Cache Max Cache Entries   | 259 |
| 3.1.14.48.1. Abbreviation                 | 259 |
| 3.1.14.48.2. LDAP Name(s)                 | 260 |
| 3.1.14.48.3. Syntax (DAP)                 | 260 |
| 3.1.14.48.4. Syntax (LDAP)                | 260 |
| 3.1.14.48.5. Example (DAP)                | 260 |
| 3.1.14.48.6. Example (LDAP)               | 260 |
| 3.1.14.49. LDAP Cache Min Attributes      | 260 |
| 3.1.14.49.1. Abbreviation                 | 260 |
| 3.1.14.49.2. LDAP Name(s)                 | 260 |
| 3.1.14.49.3. Syntax (DAP)                 | 260 |
| 3.1.14.49.4. Syntax (LDAP)                | 260 |
| 3.1.14.49.5. Example (DAP)                | 260 |

|                                                       |     |
|-------------------------------------------------------|-----|
| 3.1.14.49.6. Example (LDAP) .....                     | 261 |
| 3.1.14.50. LDAP Cache Max Attributes .....            | 261 |
| 3.1.14.50.1. Abbreviation .....                       | 261 |
| 3.1.14.50.2. LDAP Name(s) .....                       | 261 |
| 3.1.14.50.3. Syntax (DAP) .....                       | 261 |
| 3.1.14.50.4. Syntax (LDAP) .....                      | 261 |
| 3.1.14.50.5. Example (DAP) .....                      | 261 |
| 3.1.14.50.6. Example (LDAP) .....                     | 261 |
| 3.1.14.51. LDAP Cache Min Values .....                | 261 |
| 3.1.14.51.1. Abbreviation .....                       | 262 |
| 3.1.14.51.2. LDAP Name(s) .....                       | 262 |
| 3.1.14.51.3. Syntax (DAP) .....                       | 262 |
| 3.1.14.51.4. Syntax (LDAP) .....                      | 262 |
| 3.1.14.51.5. Example (DAP) .....                      | 262 |
| 3.1.14.51.6. Example (LDAP) .....                     | 262 |
| 3.1.14.52. LDAP Cache Max Values .....                | 262 |
| 3.1.14.52.1. Abbreviation .....                       | 262 |
| 3.1.14.52.2. LDAP Name(s) .....                       | 262 |
| 3.1.14.52.3. Syntax (DAP) .....                       | 263 |
| 3.1.14.52.4. Syntax (LDAP) .....                      | 263 |
| 3.1.14.52.5. Example (DAP) .....                      | 263 |
| 3.1.14.52.6. Example (LDAP) .....                     | 263 |
| 3.1.14.53. LDAP Cache Table Size .....                | 263 |
| 3.1.14.53.1. Abbreviation .....                       | 263 |
| 3.1.14.53.2. LDAP Name(s) .....                       | 263 |
| 3.1.14.53.3. Syntax (DAP) .....                       | 263 |
| 3.1.14.53.4. Syntax (LDAP) .....                      | 263 |
| 3.1.14.53.5. Example (DAP) .....                      | 263 |
| 3.1.14.53.6. Example (LDAP) .....                     | 264 |
| 3.1.14.54. LDAP Cache Update Strategy .....           | 264 |
| 3.1.14.54.1. Abbreviation .....                       | 264 |
| 3.1.14.54.2. LDAP Name(s) .....                       | 264 |
| 3.1.14.54.3. Syntax (DAP) .....                       | 264 |
| 3.1.14.54.4. Syntax (LDAP) .....                      | 264 |
| 3.1.14.54.5. Example (DAP) .....                      | 265 |
| 3.1.14.54.6. Example (LDAP) .....                     | 265 |
| 3.1.14.55. LDAP Audit Configuration Subentry CN ..... | 265 |
| 3.1.14.55.1. Abbreviation .....                       | 265 |
| 3.1.14.55.2. LDAP Name(s) .....                       | 265 |
| 3.1.14.55.3. Syntax (DAP) .....                       | 265 |
| 3.1.14.55.4. Syntax (LDAP) .....                      | 265 |
| 3.1.14.55.5. Example (DAP) .....                      | 265 |

|                                                                  |     |
|------------------------------------------------------------------|-----|
| 3.1.14.55.6. Example (LDAP) .....                                | 265 |
| 3.1.14.56. Attributes Controlling LDAP Extended Operations ..... | 265 |
| 3.1.14.56.1. LDAP Extended Operations Admins .....               | 267 |
| 3.1.14.56.2. Abbreviation .....                                  | 267 |
| 3.1.14.56.3. LDAP Name(s) .....                                  | 268 |
| 3.1.14.56.4. Syntax (DAP) .....                                  | 268 |
| 3.1.14.56.5. Syntax (LDAP) .....                                 | 268 |
| 3.1.14.56.6. Example (DAP) .....                                 | 268 |
| 3.1.14.56.7. Example (LDAP) .....                                | 268 |
| 3.1.14.56.8. LDAP Extended Operations Admin Groups .....         | 268 |
| 3.1.14.56.9. Abbreviation .....                                  | 269 |
| 3.1.14.56.10. LDAP Name(s) .....                                 | 269 |
| 3.1.14.56.11. Syntax (DAP) .....                                 | 269 |
| 3.1.14.56.12. Syntax (LDAP) .....                                | 269 |
| 3.1.14.56.13. Example (DAP) .....                                | 269 |
| 3.1.14.56.14. Example (LDAP) .....                               | 269 |
| 3.1.14.56.15. LDAP Extended Operations Read Users .....          | 269 |
| 3.1.14.56.16. Abbreviation .....                                 | 270 |
| 3.1.14.56.17. LDAP Name(s) .....                                 | 270 |
| 3.1.14.56.18. Syntax (DAP) .....                                 | 270 |
| 3.1.14.56.19. Syntax (LDAP) .....                                | 270 |
| 3.1.14.56.20. Example (DAP) .....                                | 270 |
| 3.1.14.56.21. Example (LDAP) .....                               | 270 |
| 3.1.14.56.22. LDAP Extended Operations Read Groups .....         | 270 |
| 3.1.14.56.23. Abbreviation .....                                 | 271 |
| 3.1.14.56.24. LDAP Name(s) .....                                 | 271 |
| 3.1.14.56.25. Syntax (DAP) .....                                 | 271 |
| 3.1.14.56.26. Syntax (LDAP) .....                                | 271 |
| 3.1.14.56.27. Example (DAP) .....                                | 271 |
| 3.1.14.56.28. Example (LDAP) .....                               | 271 |
| 3.1.14.56.29. LDAP Extended Operations Execute Users .....       | 271 |
| 3.1.14.56.30. Abbreviation .....                                 | 271 |
| 3.1.14.56.31. LDAP Name(s) .....                                 | 271 |
| 3.1.14.56.32. Syntax (DAP) .....                                 | 272 |
| 3.1.14.56.33. Syntax (LDAP) .....                                | 272 |
| 3.1.14.56.34. Example (DAP) .....                                | 272 |
| 3.1.14.56.35. Example (LDAP) .....                               | 272 |
| 3.1.14.56.36. LDAP Extended Operations Execute Groups .....      | 272 |
| 3.1.14.56.37. Abbreviation .....                                 | 272 |
| 3.1.14.56.38. LDAP Name(s) .....                                 | 272 |
| 3.1.14.56.39. Syntax (DAP) .....                                 | 273 |
| 3.1.14.56.40. Syntax (LDAP) .....                                | 273 |

|                                                                |     |
|----------------------------------------------------------------|-----|
| 3.1.14.56.41. Example (DAP) .....                              | 273 |
| 3.1.14.56.42. Example (LDAP) .....                             | 273 |
| 3.1.14.56.43. LDAP Extended Operations Monitoring Users .....  | 273 |
| 3.1.14.56.44. Abbreviation .....                               | 273 |
| 3.1.14.56.45. LDAP Name(s) .....                               | 273 |
| 3.1.14.56.46. Syntax (DAP) .....                               | 273 |
| 3.1.14.56.47. Syntax (LDAP) .....                              | 273 |
| 3.1.14.56.48. Example (DAP) .....                              | 274 |
| 3.1.14.56.49. Example (LDAP) .....                             | 274 |
| 3.1.14.56.50. LDAP Extended Operations Monitoring Groups ..... | 274 |
| 3.1.14.56.51. Abbreviation .....                               | 274 |
| 3.1.14.56.52. LDAP Name(s) .....                               | 274 |
| 3.1.14.56.53. Syntax (DAP) .....                               | 274 |
| 3.1.14.56.54. Syntax (LDAP) .....                              | 275 |
| 3.1.14.56.55. Example (DAP) .....                              | 275 |
| 3.1.14.56.56. Example (LDAP) .....                             | 275 |
| 3.1.14.56.57. LDAP Extended Read Operations .....              | 275 |
| 3.1.14.56.58. Abbreviation .....                               | 277 |
| 3.1.14.56.59. LDAP Name(s) .....                               | 277 |
| 3.1.14.56.60. Syntax (DAP) .....                               | 277 |
| 3.1.14.56.61. Syntax (LDAP) .....                              | 277 |
| 3.1.14.56.62. Example (DAP) .....                              | 277 |
| 3.1.14.56.63. Example (LDAP) .....                             | 277 |
| 3.1.14.56.64. LDAP Extended Execute Operations .....           | 277 |
| 3.1.14.56.65. Abbreviation .....                               | 278 |
| 3.1.14.56.66. LDAP Name(s) .....                               | 278 |
| 3.1.14.56.67. Syntax (DAP) .....                               | 278 |
| 3.1.14.56.68. Syntax (LDAP) .....                              | 278 |
| 3.1.14.56.69. Example (DAP) .....                              | 279 |
| 3.1.14.56.70. Example (LDAP) .....                             | 279 |
| 3.1.14.56.71. LDAP Extended Monitoring Operations .....        | 279 |
| 3.1.14.56.72. Abbreviation .....                               | 279 |
| 3.1.14.56.73. LDAP Name(s) .....                               | 280 |
| 3.1.14.56.74. Syntax (DAP) .....                               | 280 |
| 3.1.14.56.75. Syntax (LDAP) .....                              | 280 |
| 3.1.14.56.76. Example (DAP) .....                              | 280 |
| 3.1.14.56.77. Example (LDAP) .....                             | 280 |
| 3.1.15. Attributes for LDAP Server SSL Configuration .....     | 280 |
| 3.1.15.1. LDAP Security Level A .....                          | 280 |
| 3.1.15.1.1. Abbreviation .....                                 | 282 |
| 3.1.15.1.2. LDAP Name(s) .....                                 | 282 |
| 3.1.15.1.3. Syntax (DAP) .....                                 | 282 |

|                                                             |     |
|-------------------------------------------------------------|-----|
| 3.1.15.1.4. Example (DAP) .....                             | 282 |
| 3.1.15.1.5. Example (LDAP) .....                            | 282 |
| 3.1.15.2. LDAP Security Level B .....                       | 282 |
| 3.1.15.2.1. Abbreviation .....                              | 282 |
| 3.1.15.2.2. LDAP Name(s) .....                              | 282 |
| 3.1.15.2.3. Syntax (DAP) .....                              | 283 |
| 3.1.15.2.4. Example (DAP) .....                             | 283 |
| 3.1.15.2.5. Example (LDAP) .....                            | 283 |
| 3.1.15.3. LDAP Supported Encryption Strength .....          | 283 |
| 3.1.15.3.1. Abbreviation .....                              | 283 |
| 3.1.15.3.2. LDAP Name(s) .....                              | 283 |
| 3.1.15.3.3. Syntax (DAP) .....                              | 283 |
| 3.1.15.3.4. Example (DAP) .....                             | 283 |
| 3.1.15.3.5. Example (LDAP) .....                            | 284 |
| 3.1.15.4. LDAP Supported Encryption Strength Extended ..... | 284 |
| 3.1.15.4.1. Abbreviation .....                              | 284 |
| 3.1.15.4.2. LDAP Name(s) .....                              | 284 |
| 3.1.15.4.3. Syntax (DAP) .....                              | 284 |
| 3.1.15.4.4. Example (DAP) .....                             | 285 |
| 3.1.15.4.5. Example (LDAP) .....                            | 285 |
| 3.1.15.5. LDAP Supported Security Protocols .....           | 285 |
| 3.1.15.5.1. Abbreviation .....                              | 285 |
| 3.1.15.5.2. LDAP Name(s) .....                              | 285 |
| 3.1.15.5.3. Syntax (DAP) .....                              | 285 |
| 3.1.15.5.4. Example (DAP) .....                             | 285 |
| 3.1.15.5.5. Example (LDAP) .....                            | 285 |
| 3.1.15.6. LDAP SSL Trace File .....                         | 286 |
| 3.1.15.6.1. Abbreviation .....                              | 286 |
| 3.1.15.6.2. LDAP Name(s) .....                              | 286 |
| 3.1.15.6.3. Syntax (DAP) .....                              | 286 |
| 3.1.15.6.4. Example (DAP) .....                             | 286 |
| 3.1.15.6.5. Example (LDAP) .....                            | 286 |
| 3.1.15.7. LDAP SSL Trace Level .....                        | 287 |
| 3.1.15.7.1. Abbreviation .....                              | 287 |
| 3.1.15.7.2. LDAP Name(s) .....                              | 287 |
| 3.1.15.7.3. Syntax (DAP) .....                              | 287 |
| 3.1.15.7.4. Syntax (LDAP) .....                             | 287 |
| 3.1.15.7.5. Example (DAP) .....                             | 287 |
| 3.1.15.7.6. Example (LDAP) .....                            | 287 |
| 3.1.15.8. LDAP SSL Client Authentication Required .....     | 287 |
| 3.1.15.8.1. Abbreviation .....                              | 288 |
| 3.1.15.8.2. LDAP Name(s) .....                              | 288 |

|                                                |     |
|------------------------------------------------|-----|
| 3.1.15.8.3. Syntax (DAP) .....                 | 288 |
| 3.1.15.8.4. Syntax (LDAP) .....                | 288 |
| 3.1.15.8.5. Example (DAP) .....                | 288 |
| 3.1.15.8.6. Example (LDAP) .....               | 289 |
| 3.1.15.9. LDAP SASL Authz Id Mapping .....     | 289 |
| 3.1.15.9.1. Abbreviation .....                 | 291 |
| 3.1.15.9.2. LDAP Name(s) .....                 | 291 |
| 3.1.15.9.3. Syntax (DAP) .....                 | 291 |
| 3.1.15.9.4. Syntax (LDAP) .....                | 291 |
| 3.1.15.9.5. Example (DAP) .....                | 291 |
| 3.1.15.9.6. Example (LDAP) .....               | 291 |
| 3.1.15.10. LDAP SSL CRL Checking .....         | 291 |
| 3.1.15.10.1. Abbreviation .....                | 292 |
| 3.1.15.10.2. LDAP Name(s) .....                | 292 |
| 3.1.15.10.3. Syntax (DAP) .....                | 292 |
| 3.1.15.10.4. Syntax (LDAP) .....               | 292 |
| 3.1.15.10.5. Example (DAP) .....               | 292 |
| 3.1.15.10.6. Example (LDAP) .....              | 292 |
| 3.1.15.11. LDAP SSL CRL Filenames .....        | 292 |
| 3.1.15.11.1. Abbreviation .....                | 293 |
| 3.1.15.11.2. LDAP Name(s) .....                | 294 |
| 3.1.15.11.3. Syntax (DAP) .....                | 294 |
| 3.1.15.11.4. Syntax (LDAP) .....               | 294 |
| 3.1.15.11.5. Example (DAP) .....               | 294 |
| 3.1.15.11.6. Example (LDAP) .....              | 294 |
| 3.1.15.12. LDAP SSL Tolerate Missing CRL ..... | 294 |
| 3.1.15.12.1. Abbreviation .....                | 294 |
| 3.1.15.12.2. LDAP Name(s) .....                | 295 |
| 3.1.15.12.3. Syntax (DAP) .....                | 295 |
| 3.1.15.12.4. Syntax (LDAP) .....               | 295 |
| 3.1.15.12.5. Example (DAP) .....               | 295 |
| 3.1.15.12.6. Example (LDAP) .....              | 295 |
| 3.1.15.13. LDAP SSL Allow NYV CRL .....        | 295 |
| 3.1.15.13.1. Abbreviation .....                | 295 |
| 3.1.15.13.2. LDAP Name(s) .....                | 295 |
| 3.1.15.13.3. Syntax (DAP) .....                | 295 |
| 3.1.15.13.4. Syntax (LDAP) .....               | 296 |
| 3.1.15.13.5. Example (DAP) .....               | 296 |
| 3.1.15.13.6. Example (LDAP) .....              | 296 |
| 3.1.15.14. LDAP SSL Allow Expired CRL .....    | 296 |
| 3.1.15.14.1. Abbreviation .....                | 296 |
| 3.1.15.14.2. LDAP Name(s) .....                | 296 |

|                                                                                  |     |
|----------------------------------------------------------------------------------|-----|
| 3.1.15.14.3. Syntax (DAP) .....                                                  | 296 |
| 3.1.15.14.4. Syntax (LDAP) .....                                                 | 296 |
| 3.1.15.14.5. Example (DAP) .....                                                 | 296 |
| 3.1.15.14.6. Example (LDAP) .....                                                | 297 |
| 3.1.15.15. Attributes of the LDAP SSL Configuration Subentry for the LDAP Server |     |
| Personal Security Environment (PSE) .....                                        | 297 |
| 3.1.15.15.1. LDAP Own Keymaterial PEM .....                                      | 297 |
| 3.1.15.15.2. Abbreviation .....                                                  | 298 |
| 3.1.15.15.3. LDAP Name(s) .....                                                  | 298 |
| 3.1.15.15.4. Syntax (DAP) .....                                                  | 298 |
| 3.1.15.15.5. Example (DAP) .....                                                 | 298 |
| 3.1.15.15.6. Example (LDAP) .....                                                | 298 |
| 3.1.15.15.7. LDAP Trusted CA Certs .....                                         | 299 |
| 3.1.15.15.8. Abbreviation .....                                                  | 299 |
| 3.1.15.15.9. LDAP Name(s) .....                                                  | 299 |
| 3.1.15.15.10. Syntax (DAP) .....                                                 | 299 |
| 3.1.15.15.11. Example (DAP) .....                                                | 299 |
| 3.1.15.15.12. Example (LDAP) .....                                               | 299 |
| 3.1.15.15.13. LDAP PKCS12 Password File .....                                    | 299 |
| 3.1.15.15.14. Abbreviation .....                                                 | 300 |
| 3.1.15.15.15. LDAP Name(s) .....                                                 | 300 |
| 3.1.15.15.16. Syntax (DAP) .....                                                 | 300 |
| 3.1.15.15.17. Example (DAP) .....                                                | 300 |
| 3.1.15.15.18. Example (LDAP) .....                                               | 300 |
| 3.1.15.16. Obsolete Attributes of the LDAP SSL Configuration Subentry for the    |     |
| LDAP Server PSE .....                                                            | 301 |
| 3.1.16. Attributes for LDAP Server Audit Configuration .....                     | 301 |
| 3.1.16.1. LDAP Audit On .....                                                    | 301 |
| 3.1.16.1.1. Abbreviation .....                                                   | 302 |
| 3.1.16.1.2. LDAP Name(s) .....                                                   | 302 |
| 3.1.16.1.3. Syntax (DAP) .....                                                   | 302 |
| 3.1.16.1.4. Syntax (LDAP) .....                                                  | 302 |
| 3.1.16.1.5. Example (DAP) .....                                                  | 302 |
| 3.1.16.1.6. Example (LDAP) .....                                                 | 302 |
| 3.1.16.2. LDAP Audit Destination .....                                           | 302 |
| 3.1.16.2.1. Abbreviation .....                                                   | 303 |
| 3.1.16.2.2. LDAP Name(s) .....                                                   | 303 |
| 3.1.16.2.3. Syntax (DAP) .....                                                   | 303 |
| 3.1.16.2.4. Example (DAP) .....                                                  | 303 |
| 3.1.16.2.5. Example (LDAP) .....                                                 | 303 |
| 3.1.16.3. LDAP Audit Size Limit .....                                            | 304 |
| 3.1.16.3.1. Abbreviation .....                                                   | 304 |



|                                                    |     |
|----------------------------------------------------|-----|
| 3.1.16.3.2. LDAP Name(s) .....                     | 304 |
| 3.1.16.3.3. Syntax (DAP) .....                     | 305 |
| 3.1.16.3.4. Syntax (LDAP) .....                    | 305 |
| 3.1.16.3.5. Example (DAP) .....                    | 305 |
| 3.1.16.3.6. Example (LDAP) .....                   | 305 |
| 3.1.16.4. LDAP Audit Max File Size .....           | 305 |
| 3.1.16.4.1. Abbreviation .....                     | 305 |
| 3.1.16.4.2. LDAP Name(s) .....                     | 305 |
| 3.1.16.4.3. Syntax (DAP) .....                     | 306 |
| 3.1.16.4.4. Syntax (LDAP) .....                    | 306 |
| 3.1.16.4.5. Example (DAP) .....                    | 306 |
| 3.1.16.4.6. Example (LDAP) .....                   | 306 |
| 3.1.16.5. LDAP Audit Overflow .....                | 306 |
| 3.1.16.5.1. Abbreviation .....                     | 306 |
| 3.1.16.5.2. LDAP Name(s) .....                     | 306 |
| 3.1.16.5.3. Syntax (DAP) .....                     | 307 |
| 3.1.16.5.4. Example (DAP) .....                    | 307 |
| 3.1.16.5.5. Example (LDAP) .....                   | 307 |
| 3.1.16.6. LDAP Audit Number Wraparound Files ..... | 307 |
| 3.1.16.6.1. Abbreviation .....                     | 308 |
| 3.1.16.6.2. LDAP Name(s) .....                     | 308 |
| 3.1.16.6.3. Syntax (DAP) .....                     | 308 |
| 3.1.16.6.4. Syntax (LDAP) .....                    | 308 |
| 3.1.16.6.5. Example (DAP) .....                    | 308 |
| 3.1.16.6.6. Example (LDAP) .....                   | 308 |
| 3.1.16.7. LDAP Audit Level .....                   | 309 |
| 3.1.16.7.1. Abbreviation .....                     | 310 |
| 3.1.16.7.2. LDAP Name(s) .....                     | 310 |
| 3.1.16.7.3. Syntax (DAP) .....                     | 310 |
| 3.1.16.7.4. Example (DAP) .....                    | 310 |
| 3.1.16.7.5. Example (LDAP) .....                   | 312 |
| 3.1.16.8. LDAP Audit Value Limit .....             | 312 |
| 3.1.16.8.1. Abbreviation .....                     | 312 |
| 3.1.16.8.2. LDAP Name(s) .....                     | 312 |
| 3.1.16.8.3. Syntax (DAP) .....                     | 312 |
| 3.1.16.8.4. Syntax (LDAP) .....                    | 313 |
| 3.1.16.8.5. Example (DAP) .....                    | 313 |
| 3.1.16.8.6. Example (LDAP) .....                   | 313 |
| 3.1.16.9. LDAP Audit Op Selection .....            | 313 |
| 3.1.16.9.1. Abbreviation .....                     | 314 |
| 3.1.16.9.2. LDAP Name(s) .....                     | 314 |
| 3.1.16.9.3. Syntax (DAP) .....                     | 314 |

|                                                  |     |
|--------------------------------------------------|-----|
| 3.1.16.9.4. Example (DAP) .....                  | 314 |
| 3.1.16.9.5. Example (LDAP) .....                 | 314 |
| 3.1.16.10. LDAP Audit Erroneous Operations ..... | 314 |
| 3.1.16.10.1. Abbreviation .....                  | 315 |
| 3.1.16.10.2. LDAP Name(s) .....                  | 315 |
| 3.1.16.10.3. Syntax (DAP) .....                  | 315 |
| 3.1.16.10.4. Syntax (LDAP) .....                 | 315 |
| 3.1.16.10.5. Example (DAP) .....                 | 315 |
| 3.1.16.10.6. Example (LDAP) .....                | 315 |
| 3.1.16.11. LDAP Audit Internal Buffer Size ..... | 315 |
| 3.1.16.11.1. Abbreviation .....                  | 315 |
| 3.1.16.11.2. LDAP Name(s) .....                  | 315 |
| 3.1.16.11.3. Syntax (DAP) .....                  | 316 |
| 3.1.16.11.4. Syntax (LDAP) .....                 | 316 |
| 3.1.16.11.5. Example (DAP) .....                 | 316 |
| 3.1.16.11.6. Example (LDAP) .....                | 316 |
| 3.1.16.12. LDAP Audit Encryption .....           | 316 |
| 3.1.16.12.1. Abbreviation .....                  | 316 |
| 3.1.16.12.2. LDAP Name(s) .....                  | 316 |
| 3.1.16.12.3. Syntax (DAP) .....                  | 316 |
| 3.1.16.12.4. Example (DAP) .....                 | 317 |
| 3.1.16.12.5. Example (LDAP) .....                | 317 |
| 3.1.16.13. LDAP Audit Encryption Key File .....  | 317 |
| 3.1.16.13.1. Abbreviation .....                  | 317 |
| 3.1.16.13.2. LDAP Name(s) .....                  | 317 |
| 3.1.16.13.3. Syntax (DAP) .....                  | 317 |
| 3.1.16.13.4. Example (DAP) .....                 | 317 |
| 3.1.16.13.5. Example (LDAP) .....                | 318 |
| 3.1.16.14. LDAP Audit Start Time .....           | 318 |
| 3.1.16.14.1. Abbreviation .....                  | 319 |
| 3.1.16.14.2. LDAP Name(s) .....                  | 319 |
| 3.1.16.14.3. Syntax (DAP) .....                  | 319 |
| 3.1.16.14.4. Example (DAP) .....                 | 319 |
| 3.1.16.14.5. Example (LDAP) .....                | 319 |
| 3.1.16.15. LDAP Audit Stop Time .....            | 319 |
| 3.1.16.15.1. Abbreviation .....                  | 320 |
| 3.1.16.15.2. LDAP Name(s) .....                  | 320 |
| 3.1.16.15.3. Syntax (DAP) .....                  | 320 |
| 3.1.16.15.4. Example (DAP) .....                 | 320 |
| 3.1.16.15.5. Example (LDAP) .....                | 320 |
| 3.1.16.16. LDAP Audit Cron Job Start Time .....  | 320 |
| 3.1.16.16.1. Abbreviation .....                  | 321 |

|                                                          |     |
|----------------------------------------------------------|-----|
| 3.1.16.16.2. LDAP Name(s) .....                          | 321 |
| 3.1.16.16.3. Syntax (DAP) .....                          | 321 |
| 3.1.16.16.4. Syntax (LDAP) .....                         | 321 |
| 3.1.16.16.5. Example (DAP) .....                         | 321 |
| 3.1.16.16.6. Example (LDAP) .....                        | 321 |
| 3.1.16.17. LDAP Audit Cron Job Interval .....            | 321 |
| 3.1.16.17.1. Abbreviation .....                          | 323 |
| 3.1.16.17.2. LDAP Name(s) .....                          | 323 |
| 3.1.16.17.3. Syntax (DAP) .....                          | 323 |
| 3.1.16.17.4. Syntax (LDAP) .....                         | 323 |
| 3.1.16.17.5. Example (DAP) .....                         | 323 |
| 3.1.16.17.6. Example (LDAP) .....                        | 323 |
| 3.1.16.18. LDAP Audit Session Tracking Enabled .....     | 323 |
| 3.1.16.18.1. Abbreviation .....                          | 324 |
| 3.1.16.18.2. LDAP Name(s) .....                          | 324 |
| 3.1.16.18.3. Syntax (DAP) .....                          | 324 |
| 3.1.16.18.4. Syntax (LDAP) .....                         | 324 |
| 3.1.16.18.5. Example (DAP) .....                         | 324 |
| 3.1.16.18.6. Example (LDAP) .....                        | 324 |
| 3.1.16.19. LDAP Audit Session Tracking InfoLen .....     | 324 |
| 3.1.16.19.1. Abbreviation .....                          | 324 |
| 3.1.16.19.2. LDAP Name(s) .....                          | 324 |
| 3.1.16.19.3. Syntax (DAP) .....                          | 324 |
| 3.1.16.19.4. Syntax (LDAP) .....                         | 325 |
| 3.1.16.19.5. Example (DAP) .....                         | 325 |
| 3.1.16.19.6. Example (LDAP) .....                        | 325 |
| 3.1.17. Attributes of the Password Policy Subentry ..... | 325 |
| 3.1.17.1. Password Check Syntax .....                    | 327 |
| 3.1.17.1.1. Abbreviation .....                           | 328 |
| 3.1.17.1.2. LDAP Name(s) .....                           | 328 |
| 3.1.17.1.3. Syntax (DAP) .....                           | 328 |
| 3.1.17.1.4. Syntax (LDAP) .....                          | 328 |
| 3.1.17.1.5. Example (DAP) .....                          | 328 |
| 3.1.17.1.6. Example (LDAP) .....                         | 328 |
| 3.1.17.2. Password Minimum Length .....                  | 328 |
| 3.1.17.2.1. Abbreviation .....                           | 328 |
| 3.1.17.2.2. LDAP Name(s) .....                           | 328 |
| 3.1.17.2.3. Syntax (DAP) .....                           | 329 |
| 3.1.17.2.4. Syntax (LDAP) .....                          | 329 |
| 3.1.17.2.5. Example (DAP) .....                          | 329 |
| 3.1.17.2.6. Example (LDAP) .....                         | 329 |
| 3.1.17.3. Password Maximum Length .....                  | 329 |

|                                                       |     |
|-------------------------------------------------------|-----|
| 3.1.17.3.1. Abbreviation .....                        | 329 |
| 3.1.17.3.2. LDAP Name(s) .....                        | 329 |
| 3.1.17.3.3. Syntax (DAP) .....                        | 329 |
| 3.1.17.3.4. Syntax (LDAP) .....                       | 329 |
| 3.1.17.3.5. Example (DAP) .....                       | 330 |
| 3.1.17.3.6. Example (LDAP) .....                      | 330 |
| 3.1.17.4. Password Minimum Lower Case Character ..... | 330 |
| 3.1.17.4.1. Abbreviation .....                        | 330 |
| 3.1.17.4.2. LDAP Name(s) .....                        | 330 |
| 3.1.17.4.3. Syntax (DAP) .....                        | 330 |
| 3.1.17.4.4. Syntax (LDAP) .....                       | 330 |
| 3.1.17.4.5. Example (DAP) .....                       | 330 |
| 3.1.17.4.6. Example (LDAP) .....                      | 330 |
| 3.1.17.5. Password Minimum Upper Case Character ..... | 330 |
| 3.1.17.5.1. Abbreviation .....                        | 331 |
| 3.1.17.5.2. LDAP Name(s) .....                        | 331 |
| 3.1.17.5.3. Syntax (DAP) .....                        | 331 |
| 3.1.17.5.4. Syntax (LDAP) .....                       | 331 |
| 3.1.17.5.5. Example (DAP) .....                       | 331 |
| 3.1.17.5.6. Example (LDAP) .....                      | 331 |
| 3.1.17.6. Password Minimum Digit .....                | 331 |
| 3.1.17.6.1. Abbreviation .....                        | 331 |
| 3.1.17.6.2. LDAP Name(s) .....                        | 331 |
| 3.1.17.6.3. Syntax (DAP) .....                        | 331 |
| 3.1.17.6.4. Syntax (LDAP) .....                       | 332 |
| 3.1.17.6.5. Example (DAP) .....                       | 332 |
| 3.1.17.6.6. Example (LDAP) .....                      | 332 |
| 3.1.17.7. Password Minimum Special Character .....    | 332 |
| 3.1.17.7.1. Abbreviation .....                        | 332 |
| 3.1.17.7.2. LDAP Name(s) .....                        | 332 |
| 3.1.17.7.3. Syntax (DAP) .....                        | 332 |
| 3.1.17.7.4. Syntax (LDAP) .....                       | 332 |
| 3.1.17.7.5. Example (DAP) .....                       | 332 |
| 3.1.17.7.6. Example (LDAP) .....                      | 332 |
| 3.1.17.8. Password Exclude Name .....                 | 333 |
| 3.1.17.8.1. Abbreviation .....                        | 333 |
| 3.1.17.8.2. LDAP Name(s) .....                        | 333 |
| 3.1.17.8.3. pwdExcludeName .....                      | 333 |
| 3.1.17.8.4. Syntax (DAP) .....                        | 333 |
| 3.1.17.8.5. Syntax (LDAP) .....                       | 333 |
| 3.1.17.8.6. Example (DAP) .....                       | 333 |
| 3.1.17.8.7. Example (LDAP) .....                      | 333 |

|                                             |     |
|---------------------------------------------|-----|
| 3.1.17.9. Password In History .....         | 334 |
| 3.1.17.9.1. Abbreviation .....              | 334 |
| 3.1.17.9.2. LDAP Name(s) .....              | 334 |
| 3.1.17.9.3. Syntax (DAP) .....              | 334 |
| 3.1.17.9.4. Syntax (LDAP) .....             | 334 |
| 3.1.17.9.5. Example (DAP) .....             | 334 |
| 3.1.17.9.6. Example (LDAP) .....            | 334 |
| 3.1.17.10. Password Minimum Age .....       | 334 |
| 3.1.17.10.1. Abbreviation .....             | 334 |
| 3.1.17.10.2. LDAP Name(s) .....             | 335 |
| 3.1.17.10.3. Syntax (DAP) .....             | 335 |
| 3.1.17.10.4. Syntax (LDAP) .....            | 335 |
| 3.1.17.10.5. Example (DAP) .....            | 335 |
| 3.1.17.10.6. Example (LDAP) .....           | 335 |
| 3.1.17.11. Password Maximum Age .....       | 335 |
| 3.1.17.11.1. Abbreviation .....             | 335 |
| 3.1.17.11.2. LDAP Name(s) .....             | 335 |
| 3.1.17.11.3. Syntax (DAP) .....             | 335 |
| 3.1.17.11.4. Syntax (LDAP) .....            | 335 |
| 3.1.17.11.5. Example (DAP) .....            | 336 |
| 3.1.17.11.6. Example (LDAP) .....           | 336 |
| 3.1.17.12. Password Expire Warning .....    | 336 |
| 3.1.17.12.1. Abbreviation .....             | 336 |
| 3.1.17.12.2. LDAP Name(s) .....             | 336 |
| 3.1.17.12.3. Syntax (DAP) .....             | 336 |
| 3.1.17.12.4. Syntax (LDAP) .....            | 336 |
| 3.1.17.12.5. Example (DAP) .....            | 336 |
| 3.1.17.12.6. Example (LDAP) .....           | 336 |
| 3.1.17.13. Password Grace Login Limit ..... | 337 |
| 3.1.17.13.1. Abbreviation .....             | 337 |
| 3.1.17.13.2. LDAP Name(s) .....             | 337 |
| 3.1.17.13.3. Syntax (DAP) .....             | 337 |
| 3.1.17.13.4. Syntax (LDAP) .....            | 337 |
| 3.1.17.13.5. Example (DAP) .....            | 337 |
| 3.1.17.13.6. Example (LDAP) .....           | 337 |
| 3.1.17.14. Password Lockout .....           | 337 |
| 3.1.17.14.1. Abbreviation .....             | 338 |
| 3.1.17.14.2. LDAP Name(s) .....             | 338 |
| 3.1.17.14.3. Syntax (DAP) .....             | 338 |
| 3.1.17.14.4. Syntax (LDAP) .....            | 338 |
| 3.1.17.14.5. Example (DAP) .....            | 338 |
| 3.1.17.14.6. Example (LDAP) .....           | 338 |

|                                                    |     |
|----------------------------------------------------|-----|
| 3.1.17.15. Password Lockout Duration .....         | 338 |
| 3.1.17.15.1. Abbreviation .....                    | 339 |
| 3.1.17.15.2. LDAP Name(s) .....                    | 339 |
| 3.1.17.15.3. Syntax (DAP) .....                    | 339 |
| 3.1.17.15.4. Syntax (LDAP) .....                   | 339 |
| 3.1.17.15.5. Example (DAP) .....                   | 339 |
| 3.1.17.15.6. Example (LDAP) .....                  | 339 |
| 3.1.17.16. Password Maximum Failure .....          | 339 |
| 3.1.17.16.1. Abbreviation .....                    | 339 |
| 3.1.17.16.2. LDAP Name(s) .....                    | 339 |
| 3.1.17.16.3. Syntax (DAP) .....                    | 339 |
| 3.1.17.16.4. Syntax (LDAP) .....                   | 339 |
| 3.1.17.16.5. Example (DAP) .....                   | 340 |
| 3.1.17.16.6. Example (LDAP) .....                  | 340 |
| 3.1.17.17. Password Failure Counter Interval ..... | 340 |
| 3.1.17.17.1. Abbreviation .....                    | 340 |
| 3.1.17.17.2. LDAP Name(s) .....                    | 340 |
| 3.1.17.17.3. Syntax (DAP) .....                    | 340 |
| 3.1.17.17.4. Syntax (LDAP) .....                   | 340 |
| 3.1.17.17.5. Example (DAP) .....                   | 340 |
| 3.1.17.17.6. Example (LDAP) .....                  | 340 |
| 3.1.17.18. Password Must Change .....              | 341 |
| 3.1.17.18.1. Abbreviation .....                    | 341 |
| 3.1.17.18.2. LDAP Name(s) .....                    | 341 |
| 3.1.17.18.3. Syntax (DAP) .....                    | 341 |
| 3.1.17.18.4. Syntax (LDAP) .....                   | 341 |
| 3.1.17.18.5. Example (DAP) .....                   | 341 |
| 3.1.17.18.6. Example (LDAP) .....                  | 341 |
| 3.1.17.19. Password Store Login Info .....         | 341 |
| 3.1.17.19.1. Abbreviation .....                    | 342 |
| 3.1.17.19.2. LDAP Name .....                       | 342 |
| 3.1.17.19.3. Syntax (DAP) .....                    | 342 |
| 3.1.17.19.4. Syntax (LDAP) .....                   | 342 |
| 3.1.17.19.5. Example (DAP) .....                   | 342 |
| 3.1.17.19.6. Example (LDAP) .....                  | 342 |
| 3.1.17.20. Password Storage Scheme .....           | 342 |
| 3.1.17.20.1. Abbreviation .....                    | 343 |
| 3.1.17.20.2. LDAP Name(s) .....                    | 343 |
| 3.1.17.20.3. Syntax (DAP) .....                    | 343 |
| 3.1.17.20.4. Syntax (LDAP) .....                   | 343 |
| 3.1.17.20.5. Example (DAP) .....                   | 343 |
| 3.1.17.20.6. Example (LDAP) .....                  | 344 |

|                                                    |     |
|----------------------------------------------------|-----|
| 3.1.17.21. Password Storage Scheme Algorithm ..... | 344 |
| 3.1.17.21.1. Abbreviation .....                    | 344 |
| 3.1.17.21.2. LDAP Name(s) .....                    | 344 |
| 3.1.17.21.3. Syntax (DAP) .....                    | 344 |
| 3.1.17.21.4. Syntax (LDAP) .....                   | 344 |
| 3.1.17.21.5. Example (DAP) .....                   | 344 |
| 3.1.17.21.6. Example (LDAP) .....                  | 345 |
| 3.1.17.22. Password Storage Scheme Level .....     | 345 |
| 3.1.17.22.1. Abbreviation .....                    | 345 |
| 3.1.17.22.2. LDAP Name(s) .....                    | 345 |
| 3.1.17.22.3. Syntax (DAP) .....                    | 345 |
| 3.1.17.22.4. Syntax (LDAP) .....                   | 345 |
| 3.1.17.22.5. Example (DAP) .....                   | 345 |
| 3.1.17.22.6. Example (LDAP) .....                  | 346 |
| 3.1.17.23. Password Attribute Type .....           | 346 |
| 3.1.17.23.1. Abbreviation .....                    | 346 |
| 3.1.17.23.2. LDAP Name(s) .....                    | 346 |
| 3.1.17.23.3. Syntax (DAP) .....                    | 346 |
| 3.1.17.23.4. Syntax (LDAP) .....                   | 346 |
| 3.1.17.23.5. Example (DAP) .....                   | 346 |
| 3.1.17.23.6. Example (LDAP) .....                  | 346 |
| 3.1.17.24. Password Allow User Change .....        | 346 |
| 3.1.17.24.1. Abbreviation .....                    | 347 |
| 3.1.17.24.2. LDAP Name(s) .....                    | 347 |
| 3.1.17.24.3. Syntax (DAP) .....                    | 347 |
| 3.1.17.24.4. Syntax (LDAP) .....                   | 347 |
| 3.1.17.24.5. Example (DAP) .....                   | 347 |
| 3.1.17.24.6. Example (LDAP) .....                  | 347 |
| 3.1.17.25. Password Safe Modify .....              | 347 |
| 3.1.17.25.1. Abbreviation .....                    | 347 |
| 3.1.17.25.2. LDAP Name(s) .....                    | 347 |
| 3.1.17.25.3. Syntax (DAP) .....                    | 347 |
| 3.1.17.25.4. Syntax (LDAP) .....                   | 348 |
| 3.1.17.25.5. Example (DAP) .....                   | 348 |
| 3.1.17.25.6. Example (LDAP) .....                  | 348 |
| 3.1.17.26. Password Policy Exclusions .....        | 348 |
| 3.1.17.26.1. Abbreviation .....                    | 348 |
| 3.1.17.26.2. LDAP Name(s) .....                    | 348 |
| 3.1.17.26.3. Syntax (DAP) .....                    | 348 |
| 3.1.17.26.4. Syntax (LDAP) .....                   | 348 |
| 3.1.17.26.5. Example (DAP) .....                   | 348 |
| 3.1.17.26.6. Example (LDAP) .....                  | 349 |

|                                                                        |     |
|------------------------------------------------------------------------|-----|
| 3.1.18. Attributes of the Proxied Authorization Control Subentry ..... | 349 |
| 3.1.18.1. Proxy Control Owner .....                                    | 350 |
| 3.1.18.1.1. Abbreviation .....                                         | 350 |
| 3.1.18.1.2. LDAP Name .....                                            | 350 |
| 3.1.18.1.3. Syntax (DAP) .....                                         | 351 |
| 3.1.18.1.4. Syntax (LDAP) .....                                        | 351 |
| 3.1.18.1.5. Example (DAP) .....                                        | 351 |
| 3.1.18.1.6. Example (LDAP) .....                                       | 351 |
| 3.1.18.2. Subtree Specification .....                                  | 351 |
| 3.1.18.2.1. Examples (DAP) .....                                       | 351 |
| 3.2. X.500 Directory Operational Attributes .....                      | 352 |
| 3.2.1. Attributes for Access Control .....                             | 352 |
| 3.2.1.1. Access Control Scheme .....                                   | 352 |
| 3.2.1.1.1. Abbreviation .....                                          | 352 |
| 3.2.1.1.2. Syntax (DAP) .....                                          | 352 |
| 3.2.1.1.3. Example (DAP) .....                                         | 352 |
| 3.2.1.2. Entry ACI .....                                               | 353 |
| 3.2.1.2.1. Abbreviation .....                                          | 353 |
| 3.2.1.2.2. Syntax (DAP) .....                                          | 353 |
| 3.2.1.2.3. Example (DAP) .....                                         | 353 |
| 3.2.1.3. Prescriptive ACI .....                                        | 353 |
| 3.2.1.3.1. Abbreviation .....                                          | 353 |
| 3.2.1.3.2. Syntax (DAP) .....                                          | 353 |
| 3.2.1.3.3. Example (DAP) .....                                         | 353 |
| 3.2.1.4. Subentry ACI .....                                            | 354 |
| 3.2.1.4.1. Abbreviation .....                                          | 354 |
| 3.2.1.4.2. Syntax (DAP) .....                                          | 355 |
| 3.2.1.4.3. Example (DAP) .....                                         | 355 |
| 3.2.2. Attributes of Information Framework .....                       | 355 |
| 3.2.2.1. Administrative Role .....                                     | 355 |
| 3.2.2.1.1. Abbreviation .....                                          | 355 |
| 3.2.2.1.2. Syntax (DAP) .....                                          | 356 |
| 3.2.2.1.3. Example (DAP) .....                                         | 356 |
| 3.2.2.2. Collective Exclusions .....                                   | 356 |
| 3.2.2.2.1. Abbreviation .....                                          | 356 |
| 3.2.2.2.2. Syntax (DAP) .....                                          | 356 |
| 3.2.2.2.3. Example (DAP) .....                                         | 356 |
| 3.2.2.3. Creators Name .....                                           | 356 |
| 3.2.2.3.1. Abbreviation .....                                          | 356 |
| 3.2.2.3.2. Syntax (DAP) .....                                          | 356 |
| 3.2.2.3.3. Example (DAP) .....                                         | 356 |
| 3.2.2.4. Creation Time .....                                           | 357 |



|                                                    |     |
|----------------------------------------------------|-----|
| 3.2.2.4.1. Abbreviation .....                      | 357 |
| 3.2.2.4.2. Syntax (DAP) .....                      | 357 |
| 3.2.2.4.3. Example (DAP) .....                     | 357 |
| 3.2.2.5. DirX Directory Entry UUID .....           | 357 |
| 3.2.2.5.1. Abbreviation .....                      | 357 |
| 3.2.2.5.2. Syntax (DAP) .....                      | 357 |
| 3.2.2.5.3. Example (DAP) .....                     | 357 |
| 3.2.2.6. Modifiers Name .....                      | 357 |
| 3.2.2.6.1. Abbreviation .....                      | 357 |
| 3.2.2.6.2. Syntax (DAP) .....                      | 357 |
| 3.2.2.6.3. Example (DAP) .....                     | 357 |
| 3.2.2.7. Modification Time .....                   | 358 |
| 3.2.2.7.1. Abbreviation .....                      | 358 |
| 3.2.2.7.2. Syntax (DAP) .....                      | 358 |
| 3.2.2.7.3. Example (DAP) .....                     | 358 |
| 3.2.2.8. Number of Subordinates .....              | 358 |
| 3.2.2.8.1. Abbreviation .....                      | 358 |
| 3.2.2.8.2. LDAP Name(s) .....                      | 358 |
| 3.2.2.8.3. Syntax (DAP) .....                      | 358 |
| 3.2.2.8.4. Example (DAP) .....                     | 358 |
| 3.2.2.8.5. Example (LDAP) .....                    | 358 |
| 3.2.2.9. Number of All Subordinates .....          | 359 |
| 3.2.2.9.1. Abbreviation .....                      | 359 |
| 3.2.2.9.2. LDAP Name(s) .....                      | 359 |
| 3.2.2.9.3. Syntax (DAP) .....                      | 359 |
| 3.2.2.9.4. Example (DAP) .....                     | 359 |
| 3.2.2.9.5. Example (LDAP) .....                    | 359 |
| 3.2.2.10. Subtree Specification .....              | 359 |
| 3.2.2.10.1. Abbreviation .....                     | 359 |
| 3.2.2.10.2. Syntax (DAP) .....                     | 359 |
| 3.2.2.10.3. Example (DAP) .....                    | 359 |
| 3.2.3. Attributes for Schema Administration .....  | 360 |
| 3.2.3.1. Governing Structure Rule .....            | 360 |
| 3.2.3.1.1. Abbreviation .....                      | 360 |
| 3.2.3.1.2. Syntax (DAP) .....                      | 360 |
| 3.2.3.1.3. Example (DAP) .....                     | 360 |
| 3.2.3.2. Structural Object Class .....             | 360 |
| 3.2.3.2.1. Abbreviation .....                      | 360 |
| 3.2.3.2.2. Syntax (DAP) .....                      | 360 |
| 3.2.3.2.3. Example (DAP) .....                     | 360 |
| 3.2.4. Attributes for Database Configuration ..... | 360 |
| 3.2.4.1. Attribute Index .....                     | 360 |

|                                                         |     |
|---------------------------------------------------------|-----|
| 3.2.4.1.1. Abbreviation .....                           | 361 |
| 3.2.4.1.2. Syntax (DAP) .....                           | 361 |
| 3.2.4.1.3. Example (DAP) .....                          | 361 |
| 3.2.4.2. Attribute Outsourcing .....                    | 361 |
| 3.2.4.2.1. Abbreviation .....                           | 361 |
| 3.2.4.2.2. Syntax (DAP) .....                           | 361 |
| 3.2.4.2.3. Example (DAP) .....                          | 361 |
| 3.2.5. Password Policy Attributes .....                 | 361 |
| 3.2.5.1. Attributes Supporting Password Checking .....  | 362 |
| 3.2.5.1.1. Password History Attribute .....             | 362 |
| 3.2.5.1.2. Abbreviation .....                           | 362 |
| 3.2.5.1.3. LDAP Name(s) .....                           | 362 |
| 3.2.5.1.4. Syntax (DAP) .....                           | 362 |
| 3.2.5.1.5. Example (DAP) .....                          | 362 |
| 3.2.5.1.6. Example (LDAP) .....                         | 362 |
| 3.2.5.2. Attributes Supporting Password Aging .....     | 362 |
| 3.2.5.2.1. Password Changed Time Attribute .....        | 362 |
| 3.2.5.2.2. Abbreviation .....                           | 363 |
| 3.2.5.2.3. LDAP Name(s) .....                           | 363 |
| 3.2.5.2.4. Syntax (DAP) .....                           | 363 |
| 3.2.5.2.5. Syntax (LDAP) .....                          | 363 |
| 3.2.5.2.6. Example (DAP) .....                          | 363 |
| 3.2.5.2.7. Example (LDAP) .....                         | 363 |
| 3.2.5.2.8. Password Expiration Warned .....             | 363 |
| 3.2.5.2.9. Abbreviation .....                           | 364 |
| 3.2.5.2.10. LDAP Name(s) .....                          | 364 |
| 3.2.5.2.11. Syntax (DAP) .....                          | 364 |
| 3.2.5.2.12. Syntax (LDAP) .....                         | 364 |
| 3.2.5.2.13. Example (DAP) .....                         | 364 |
| 3.2.5.2.14. Example (LDAP) .....                        | 364 |
| 3.2.5.2.15. Password Grace Use Time .....               | 364 |
| 3.2.5.2.16. Abbreviation .....                          | 364 |
| 3.2.5.2.17. LDAP Name(s) .....                          | 365 |
| 3.2.5.2.18. Syntax (DAP) .....                          | 365 |
| 3.2.5.2.19. Syntax (LDAP) .....                         | 365 |
| 3.2.5.2.20. Example (DAP) .....                         | 365 |
| 3.2.5.2.21. Example (LDAP) .....                        | 365 |
| 3.2.5.3. Attributes Supporting Account Lockout .....    | 365 |
| 3.2.5.3.1. Password Account Locked Time Attribute ..... | 365 |
| 3.2.5.3.2. Abbreviation .....                           | 366 |
| 3.2.5.3.3. LDAP Name(s) .....                           | 366 |
| 3.2.5.3.4. Syntax (DAP) .....                           | 366 |

|                                                                   |     |
|-------------------------------------------------------------------|-----|
| 3.2.5.3.5. Syntax (LDAP) .....                                    | 366 |
| 3.2.5.3.6. Example (DAP) .....                                    | 366 |
| 3.2.5.3.7. Example (LDAP) .....                                   | 366 |
| 3.2.5.3.8. Password Failure Time .....                            | 366 |
| 3.2.5.3.9. Abbreviation .....                                     | 367 |
| 3.2.5.3.10. LDAP Name(s) .....                                    | 367 |
| 3.2.5.3.11. Syntax (DAP) .....                                    | 367 |
| 3.2.5.3.12. Syntax (LDAP) .....                                   | 367 |
| 3.2.5.3.13. Example (DAP) .....                                   | 367 |
| 3.2.5.3.14. Example (LDAP) .....                                  | 367 |
| 3.2.5.4. Attributes Supporting Password Administration .....      | 367 |
| 3.2.5.4.1. Password Reset Attribute .....                         | 367 |
| 3.2.5.4.2. Abbreviation .....                                     | 368 |
| 3.2.5.4.3. LDAP Name(s) .....                                     | 368 |
| 3.2.5.4.4. Syntax (DAP) .....                                     | 368 |
| 3.2.5.4.5. Syntax (LDAP) .....                                    | 368 |
| 3.2.5.4.6. Example (DAP) .....                                    | 368 |
| 3.2.5.4.7. Example (LDAP) .....                                   | 368 |
| 3.2.5.5. Attributes Supporting Storing of Login Information ..... | 369 |
| 3.2.5.5.1. Password Number of Failed Logins Attribute .....       | 369 |
| 3.2.5.5.2. Abbreviation .....                                     | 369 |
| 3.2.5.5.3. LDAP Name(s) .....                                     | 369 |
| 3.2.5.5.4. Syntax (DAP) .....                                     | 369 |
| 3.2.5.5.5. Syntax (LDAP) .....                                    | 369 |
| 3.2.5.5.6. Example (DAP) .....                                    | 369 |
| 3.2.5.5.7. Example (LDAP) .....                                   | 369 |
| 3.2.5.5.8. Password Last Login Time Attribute .....               | 370 |
| 3.2.5.5.9. Abbreviation .....                                     | 370 |
| 3.2.5.5.10. LDAP Name(s) .....                                    | 370 |
| 3.2.5.5.11. Syntax (DAP) .....                                    | 370 |
| 3.2.5.5.12. Syntax (LDAP) .....                                   | 370 |
| 3.2.5.5.13. Example (DAP) .....                                   | 370 |
| 3.2.5.5.14. Example (LDAP) .....                                  | 370 |
| 3.2.6. Attributes for Synchronous Shadowing .....                 | 370 |
| 3.2.6.1. DirX Directory Backup Type .....                         | 371 |
| 3.2.6.1.1. Abbreviation .....                                     | 371 |
| 3.2.6.1.2. Name (LDAP) .....                                      | 371 |
| 3.2.6.1.3. Syntax .....                                           | 371 |
| 3.2.6.1.4. Example (DAP) .....                                    | 371 |
| 3.2.6.1.5. Example (LDAP) .....                                   | 371 |
| 3.2.6.2. DirX Directory Backup Time .....                         | 371 |
| 3.2.6.2.1. Abbreviation .....                                     | 372 |

|                                                     |     |
|-----------------------------------------------------|-----|
| 3.2.6.2.2. Name (LDAP) .....                        | 372 |
| 3.2.6.2.3. Syntax .....                             | 372 |
| 3.2.6.2.4. Example (DAP) .....                      | 372 |
| 3.2.6.2.5. Example (LDAP) .....                     | 372 |
| 3.2.6.3. DirX Directory Backup MSN .....            | 372 |
| 3.2.6.3.1. Abbreviation .....                       | 372 |
| 3.2.6.3.2. Name (LDAP) .....                        | 372 |
| 3.2.6.3.3. Syntax .....                             | 372 |
| 3.2.6.3.4. Example (DAP) .....                      | 372 |
| 3.2.6.3.5. Example (LDAP) .....                     | 373 |
| 3.2.6.4. DirX Directory Backup DSAName .....        | 373 |
| 3.2.6.4.1. Abbreviation .....                       | 373 |
| 3.2.6.4.2. Name (LDAP) .....                        | 373 |
| 3.2.6.4.3. Syntax .....                             | 373 |
| 3.2.6.4.4. Example (DAP) .....                      | 373 |
| 3.2.6.4.5. Example (LDAP) .....                     | 373 |
| 3.2.6.5. DirX Directory Entry MSN .....             | 373 |
| 3.2.6.5.1. Abbreviation .....                       | 373 |
| 3.2.6.5.2. Name (LDAP) .....                        | 374 |
| 3.2.6.5.3. Syntax (LDAP) .....                      | 374 |
| 3.2.6.5.4. Example (LDAP) .....                     | 374 |
| 3.2.6.6. DirX Directory In Sync .....               | 374 |
| 3.2.6.6.1. Abbreviation .....                       | 374 |
| 3.2.6.6.2. Name (LDAP) .....                        | 374 |
| 3.2.6.6.3. Syntax (LDAP) .....                      | 374 |
| 3.2.6.6.4. Example (LDAP) .....                     | 374 |
| 3.2.6.7. DirX Directory Recent DN .....             | 374 |
| 3.2.6.7.1. Abbreviation .....                       | 374 |
| 3.2.6.7.2. Name (LDAP) .....                        | 375 |
| 3.2.6.7.3. Syntax (LDAP) .....                      | 375 |
| 3.2.6.7.4. Example (LDAP) .....                     | 375 |
| 3.2.6.8. DirX Directory Recent MSN .....            | 375 |
| 3.2.6.8.1. Abbreviation .....                       | 375 |
| 3.2.6.8.2. Name (LDAP) .....                        | 375 |
| 3.2.6.8.3. Syntax (LDAP) .....                      | 375 |
| 3.2.6.8.4. Example (LDAP) .....                     | 375 |
| 3.2.6.9. DirX Directory Recent MSN Time Stamp ..... | 375 |
| 3.2.6.9.1. Abbreviation .....                       | 375 |
| 3.2.6.9.2. Name (LDAP) .....                        | 376 |
| 3.2.6.9.3. Syntax (LDAP) .....                      | 376 |
| 3.2.6.9.4. Example (LDAP) .....                     | 376 |
| 3.2.6.10. DirX Directory Recent Operation .....     | 376 |

|                                                       |     |
|-------------------------------------------------------|-----|
| 3.2.6.10.1. Abbreviation .....                        | 376 |
| 3.2.6.10.2. Name (LDAP) .....                         | 376 |
| 3.2.6.10.3. Syntax (LDAP) .....                       | 376 |
| 3.2.6.10.4. Example (LDAP) .....                      | 376 |
| 3.2.7. Attributes for Nested and Dynamic Groups ..... | 376 |
| 3.2.7.1. DirX Directory Member .....                  | 376 |
| 3.2.7.1.1. Abbreviation .....                         | 377 |
| 3.2.7.1.2. LDAP Name(s) .....                         | 377 |
| 3.2.7.1.3. Syntax (DAP) .....                         | 377 |
| 3.2.7.1.4. Syntax (LDAP) .....                        | 377 |
| 3.2.7.1.5. Example (DAP) .....                        | 377 |
| 3.2.7.1.6. Example (LDAP) .....                       | 377 |
| 3.2.7.2. DirX Directory Member Of .....               | 377 |
| 3.2.7.2.1. Abbreviation .....                         | 377 |
| 3.2.7.2.2. LDAP Name(s) .....                         | 377 |
| 3.2.7.2.3. Syntax (DAP) .....                         | 377 |
| 3.2.7.2.4. Syntax (LDAP) .....                        | 377 |
| 3.2.7.2.5. Example (DAP) .....                        | 378 |
| 3.2.7.2.6. Example (LDAP) .....                       | 378 |
| 3.2.7.3. DirX Directory Member Url .....              | 378 |
| 3.2.7.3.1. Abbreviation .....                         | 378 |
| 3.2.7.3.2. LDAP Name(s) .....                         | 378 |
| 3.2.7.3.3. Syntax (DAP) .....                         | 378 |
| 3.2.7.3.4. Syntax (LDAP) .....                        | 378 |
| 3.2.7.3.5. Example (DAP) .....                        | 378 |
| 3.2.7.3.6. Example (LDAP) .....                       | 378 |
| 3.2.7.4. DirX Directory Imported Groups .....         | 378 |
| 3.2.7.4.1. Abbreviation .....                         | 379 |
| 3.2.7.4.2. LDAP Name(s) .....                         | 379 |
| 3.2.7.4.3. Syntax (DAP) .....                         | 379 |
| 3.2.7.4.4. Syntax (LDAP) .....                        | 379 |
| 3.2.7.4.5. Example (DAP) .....                        | 379 |
| 3.2.7.4.6. Example (LDAP) .....                       | 379 |
| 3.2.7.5. DirX Directory Export To .....               | 379 |
| 3.2.7.5.1. Abbreviation .....                         | 379 |
| 3.2.7.5.2. LDAP Name(s) .....                         | 379 |
| 3.2.7.5.3. Syntax (DAP) .....                         | 379 |
| 3.2.7.5.4. Syntax (LDAP) .....                        | 380 |
| 3.2.7.5.5. Example (DAP) .....                        | 380 |
| 3.2.7.5.6. Example (LDAP) .....                       | 380 |
| 3.3. X.500 Distributed Operational Attributes .....   | 380 |
| 3.3.1. Knowledge Reference Attributes .....           | 380 |

|                                                     |     |
|-----------------------------------------------------|-----|
| 3.3.1.1. Non Specific Knowledge .....               | 380 |
| 3.3.1.1.1. Abbreviation.....                        | 380 |
| 3.3.1.1.2. Syntax (DAP) .....                       | 380 |
| 3.3.1.1.3. Example (DAP) .....                      | 381 |
| 3.3.1.2. Specific Knowledge .....                   | 381 |
| 3.3.1.2.1. Abbreviation.....                        | 381 |
| 3.3.1.2.2. Syntax (DAP) .....                       | 381 |
| 3.3.1.2.3. Example (DAP) .....                      | 381 |
| 3.4. DirX Directory DSA Operational Attributes..... | 382 |
| 3.4.1. Authentication .....                         | 382 |
| 3.4.1.1. DirX Directory-Administrators .....        | 382 |
| 3.4.1.1.1. Abbreviation.....                        | 382 |
| 3.4.1.1.2. Syntax.....                              | 382 |
| 3.4.1.1.3. Example (dirxadm).....                   | 382 |
| 3.4.2. Knowledge Attributes.....                    | 382 |
| 3.4.2.1. Consumer Knowledge.....                    | 383 |
| 3.4.2.1.1. Abbreviation .....                       | 383 |
| 3.4.2.1.2. Syntax (DAP) .....                       | 383 |
| 3.4.2.1.3. Example (DAP) .....                      | 383 |
| 3.4.2.2. Superior Knowledge .....                   | 383 |
| 3.4.2.2.1. Abbreviation.....                        | 383 |
| 3.4.2.2.2. Syntax.....                              | 383 |
| 3.4.2.2.3. Example (dirxadm).....                   | 384 |
| 3.4.2.3. Supplier Knowledge .....                   | 384 |
| 3.4.2.3.1. Abbreviation .....                       | 384 |
| 3.4.2.3.2. Syntax (DAP) .....                       | 384 |
| 3.4.2.3.3. Example (DAP).....                       | 384 |
| 3.4.2.4. Secondary Shadows .....                    | 385 |
| 3.4.2.4.1. Abbreviation .....                       | 385 |
| 3.4.2.4.2. Syntax (DAP).....                        | 385 |
| 3.4.2.4.3. Example (DAP).....                       | 385 |
| 3.4.3. Policy Attributes .....                      | 385 |
| 3.4.3.1. Audit Policy.....                          | 385 |
| 3.4.3.1.1. Abbreviation .....                       | 386 |
| 3.4.3.1.2. Syntax .....                             | 386 |
| 3.4.3.1.3. Example (dirxadm) .....                  | 386 |
| 3.4.3.2. DSA Policy .....                           | 386 |
| 3.4.3.2.1. Abbreviation.....                        | 386 |
| 3.4.3.2.2. Syntax.....                              | 386 |
| 3.4.3.2.3. Example (dirxadm).....                   | 386 |
| 3.4.3.3. Local Scope Policy .....                   | 386 |
| 3.4.3.3.1. Abbreviation.....                        | 387 |

|                                               |     |
|-----------------------------------------------|-----|
| 3.4.3.3.2. Syntax .....                       | 387 |
| 3.4.3.3.3. Example (dirxadm) .....            | 387 |
| 3.4.3.4. Paging Policy .....                  | 387 |
| 3.4.3.4.1. Abbreviation .....                 | 387 |
| 3.4.3.4.2. Syntax .....                       | 387 |
| 3.4.3.4.3. Example (dirxadm) .....            | 388 |
| 3.4.3.5. Partial Entry Shadowing Policy ..... | 388 |
| 3.4.3.5.1. Abbreviation .....                 | 389 |
| 3.4.3.5.2. Syntax .....                       | 389 |
| 3.4.3.5.3. Example (dirxadm) .....            | 389 |
| 3.4.3.6. Search Base Policy .....             | 389 |
| 3.4.3.6.1. Abbreviation .....                 | 389 |
| 3.4.3.6.2. Syntax .....                       | 389 |
| 3.4.3.6.3. Example (dirxadm) .....            | 389 |
| 3.4.3.7. Time Limit Policy .....              | 389 |
| 3.4.3.7.1. Abbreviation .....                 | 390 |
| 3.4.3.7.2. Syntax .....                       | 390 |
| 3.4.3.7.3. Example (dirxadm) .....            | 390 |
| 3.4.3.8. User Policy .....                    | 390 |
| 3.4.3.8.1. Abbreviation .....                 | 390 |
| 3.4.3.8.2. Syntax .....                       | 390 |
| 3.4.3.8.3. Example (dirxadm) .....            | 390 |
| 3.4.4. Miscellaneous Attributes .....         | 391 |
| 3.4.4.1. Cooperating DSA .....                | 391 |
| 3.4.4.1.1. Abbreviation .....                 | 391 |
| 3.4.4.1.2. Syntax .....                       | 391 |
| 3.4.4.1.3. Example (dirxadm) .....            | 391 |
| 3.4.4.2. DSE Type .....                       | 393 |
| 3.4.4.2.1. Abbreviation .....                 | 394 |
| 3.4.4.2.2. Syntax .....                       | 394 |
| 3.4.4.2.3. Example (dirxadm) .....            | 394 |
| 3.4.4.3. Enable Unique Index .....            | 394 |
| 3.4.4.3.1. Abbreviation .....                 | 395 |
| 3.4.4.3.2. LDAP Name .....                    | 395 |
| 3.4.4.3.3. Syntax .....                       | 395 |
| 3.4.4.3.4. Example (dirxadm) .....            | 395 |
| Appendix A: Country Codes .....               | 396 |
| Legal Remarks .....                           | 405 |

# Preface

This manual is reference for the DirX Directory (DirX). It consists of the following sections:

- [Chapter 1](#) describes the string representations and use of DirX Directory attribute syntaxes for DAP binds and contains a reference page for simple syntaxes and reference pages for each of the structured syntaxes.
- [Chapter 2](#) describes the string representations and use of DirX Directory attribute syntaxes for LDAP binds and contains a reference page for simple syntaxes and reference pages for each of the structured syntaxes.
- [Chapter 3](#) describes the DirX Directory attributes defined in the DirX Directory default DSA schema.
- [Appendix A](#) lists countries and their 2-letter ISO 3166 country code.



# DirX Directory Documentation Set

DirX Directory provides a powerful set of documentation that helps you configure your directory server and its applications.

The DirX Directory document set consists of the following manuals:

- [DirX Directory Introduction](#). Use this book to obtain a description of the concepts of DirX Directory.
- [DirX Directory Administration Guide](#). Use this book to understand the basic DirX Directory administration tasks and how to perform them with the DirX Directory administration tools.
- [DirX Directory Administration Reference](#). Use this book to obtain reference information about DirX Directory administration tools and their command syntax, configuration files, environment variables and file locations of the DirX Directory installation.
- [DirX Directory Syntaxes and Attributes](#). Use this book to obtain reference information about DirX Directory syntaxes and attributes.
- [DirX Directory LDAP Extended Operations](#). Use this book to obtain reference information about DirX Directory LDAP Extended Operations.
- [DirX Directory External Authentication](#). Use this book to obtain reference information about external authentication.
- [DirX Directory Supervisor](#). Use this book to obtain reference information about the DirX Directory supervisor.
- [DirX Directory Plugins for Nagios](#). Use this book to obtain reference information about DirX Directory plugins for Nagios.
- [DirX Directory Disc Dimensioning Guide](#). Use this book to understand how to calculate and organize necessary disc space for initial database configuration and enhancing existing configurations.
- [DirX Directory Guide for CSP Administrators](#). Use this book to obtain information about installing, configuring and managing DirX Directory in the context of a Certificate Provisioning Service operating in accordance with regulations like the German "Signaturgesetz".
- [DirX Directory Release Notes](#). Use this book to install DirX Directory and to understand the features and limitations of the current release.

# Notation Conventions

## **Boldface type**

In command syntax, bold words and characters represent commands or keywords that must be entered exactly as shown.

In examples, bold words and characters represent user input.

## *Italic type*

In command syntax, italic words and characters represent placeholders for information that you must supply.

## [ ]

In command syntax, square braces enclose optional items.

## { }

In command syntax, braces enclose a list from which you must choose one item.

In Tcl syntax, you must actually type in the braces, which will appear in boldface type.

## |

In command syntax, the vertical bar separates items in a list of choices.

## ...

In command syntax, ellipses indicate that the previous item can be repeated.

## *install\_path*

The exact name of the root of the directory where DirX Identity programs and files are installed. The default installation directory is *userID\_home\_directory\*/DirX Identity\** on UNIX systems and **C:\Program Files\DirX\Identity** on Windows systems. During installation the installation directory can be specified. In this manual, the installation-specific portion of pathnames is represented by the notation *install\_path*.

# 1. DirX Directory String Representation for DAP Binds

This chapter describes the DAP-style string representations of simple and structured DirX Directory attributes, search filters and distinguished names. This string representation is used with the administration programs **dirxcp** for DAP binds and **dirxadm** to enter and display directory information and, if you are creating new attribute types that extend your directory schema, as the syntax for those new attribute types. (See section **Bind Types and Bind IDs** of chapter **DirX Directory Commands** in the *DirX Directory Administration Reference* for details.)

This chapter provides:

- A table of reserved characters for attributes
- An overview of the elements and format of simple and structured attributes
- An overview of elements and format of distinguished names
- An overview of elements and format of search filters
- A description of attribute syntax, including how to override it and how to use defined and undefined abbreviations
- A description of the string representations of each DirX Directory simple attribute syntax
- A reference page for each DirX Directory structured attribute that describes its string representation

## 1.1. Reserved Characters

The following table describes reserved characters for simple and structured attributes and distinguished names.

| Character | Purpose                                                                                                                                                                                                                                                                                                                         |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| =         | Associates the attribute type and its value(s) or component(s).                                                                                                                                                                                                                                                                 |
| "         | <b>For simple attributes:</b> Indicates the start and end of an attribute value. Used to indicate that white space is part of an attribute value and should not be ignored. (By default, white space is ignored.) For example: ).<br><br><b>CN='Henry Mueller'</b><br><br>Curly braces ( <b>{}</b> ) perform the same function. |

| Character    | Purpose                                                                                                                                                                                                                                                                                                                                                                                                                    |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| { }          | <p><b>For structured attributes:</b> Indicates the start and end of a structured attribute component.</p> <p><b>For simple attributes and distinguished names:</b> Encloses the entire attribute (type and value) or distinguished name to indicate that white space is part of an attribute value. For example, <b>{CN=Henry Mueller}</b> or <b>/O=My-Company</b></p> <p>Single quotes (") perform the same function.</p> |
| ,            | <p><b>For structured attributes:</b> Separates components.</p> <p><b>For distinguished names:</b> Separates multiple naming attribute value assertions (multiple AVAs).</p>                                                                                                                                                                                                                                                |
| ;            | <b>For simple and structured attributes:</b> Separates multiple values.                                                                                                                                                                                                                                                                                                                                                    |
| /            | <b>For distinguished names:</b> Separates RDNs in distinguished name syntax.                                                                                                                                                                                                                                                                                                                                               |
| \ <i>xnn</i> | Specifies hexadecimal data where <i>nn</i> represents the hexadecimal characters. For example:<br><b>\xab\xff\xcd</b>                                                                                                                                                                                                                                                                                                      |
| \            | Escapes a reserved character.                                                                                                                                                                                                                                                                                                                                                                                              |

## 1.2. Simple and Structured Attributes

Simple attributes are in the form:

*type=simple\_value[;simple\_value ...]*

Structured attributes are in the form:

*type=structured\_value[;structured\_value ...]*

Each of the elements in simple and structured attributes is described in the following sections.

### 1.2.1. Attribute Types

For both simple and structured attributes, the *type* parameter identifies the attribute.

Internally attribute types are identified by OID (object identifiers), a unique series of integers separated by the period (.) character. For example, **2.5.4.3** is the OID for the Common-Name attribute type. You can identify attributes by OID, but to make it easier to specify attributes, the **dirxadm** and **dirxcp** commands allow you to identify them by attribute type abbreviations in command lines.

An abbreviation is an abbreviation of the attribute name. Abbreviations are specified in the DirX Directory abbreviation files. For example, the abbreviation **C** represents the Country attribute type, the abbreviation **CN** represents the Common-Name attribute type. Abbreviations are case-insensitive; for example, **cn**, **Cn**, and **CN** are all valid ways to specify

the abbreviation for the Common-Name attribute type. When you display attributes, **dirxcp** and **dirxadm** use attribute type abbreviations by default. Use the **-pretty** option to display attribute names.

To display an attribute's name, type abbreviations and OID, either use the **dirxadm** and **dirxcp abbr show** commands or list the contents of the abbreviation files, which contains them.

### 1.2.2. Simple Attribute Values

For simple attributes, the *simple\_value* parameter is the value assigned to the attribute. *simple\_value* can be only a simple value, not another attribute. To enter more than one *simple\_value*, separate each with a semicolon as shown in the following example:

```
{TN=+1 964 123 456}  
{2.5.4.20=+1 964 123 456}  
{TN=+1 964 123 456;+1 965 234 543}
```

### 1.2.3. Structured Attribute Values

For structured attributes, the *structured\_value* parameter is the value assigned to the attribute. *structured\_value* is a comma-separated sequence of component type/value pairs. In other words, *structured\_value* has the form:

```
structured_value={comp_type=comp_value[,comp_type=comp_value]...}
```

*comp\_value* can either be a simple or a structured value. To enter more than one *structured\_value*, separate each with a semicolon (;).

Some sample structured attributes follow:

```
FTN={PN=63875}  
2.5.4.23={PN=63875};{PN=12345}  
PACI={ID=Admin-ACI,PR=10,AL={BL={L=SIMPLE}}}
```



Component types (*comp\_type*) can only be specified with their abbreviations. They do not have OIDs assigned.

### 1.2.4. Attribute Lists for Simple and Structured Attributes

Most **dirxadm** and **dirxcp** operations allow you to specify more than one simple or structured attribute on the command line. To specify more than one attribute, separate each attribute type and value string with white space. The following **-attribute** option specifies an OCL, DSC and TN simple attributes, all separated by a space. Note that the DSC and TN attributes are enclosed in braces ({} ) because they contain white space for readability.

```
-attribute OCL=OU {DSC=Engineering Department} {TN=+1 964 123 4567}
```

## 1.2.5. Attribute Values in a File

Single attribute values can also be specified in a file or written to a file. The syntax is as follows:

*attribute*[\_**ANY**]**\_FILE**=*filename*

where *attribute* specifies the attribute type and *filename* the name of the file containing the attribute value.

If the keyword **\_ANY** is specified, the value file contains the valid ASN.1 encoded value of the attribute regardless of the syntax for the attribute that is defined in the directory schema. If **\_ANY** is not used, the attribute syntax OCTET STRING is assumed, and the value file contains the binary representation of the attribute value. In this case, an error is returned if the syntax for the attribute specified in the schema is not OCTET STRING.

Multiple attribute values cannot be specified in a file.

Specifying the attribute value in a file is possible for the following **dirxcp** operations:

- **obj create** (-**attribute** option)
- **obj modify** (-**addattr**, -**changeattr**, -**removeattr** options)
- **obj compare** (-**attribute** option)
- **obj show** (-**attribute** option)

### Examples:

1. The following example adds a picture to the JPT attribute. The file "/tmp/john.jpg" contains the binary representation of the picture.

```
modify /O=my-company/OU=sales/CN=Huber \  
-addattr JPT_FILE=/tmp/john.jpg
```

2. Compare the JPT attribute. The file "pict\_value.jpg" contains the value of the JPT attribute in binary representation.

```
compare /O=my-company/OU=asw/CN=zapf \  
-attr JPT_FILE=pict_value.jpg
```

3. Modify the value of the JPT attribute. The file "pict\_old.jpg" contains the old value and the file "pict\_new.jpg" contains the new value.

```
modify /O=my-company/OU=sales/CN=Mayer \  
- changeattr JPT_FILE=pict_old.jpg JPT_FILE=pict_new.jpg
```

4. Create an entry. The file "cert.cer" contains the ASN.1 encoded value of the user certificate attribute.

```
create /O=my-company/OU=sales/CN=Huber -attr {OCL=per;orp;sau} \
      SN=Huber UC_ANY_FILE=cert.cer
```

5. Show an entry. The ASN.1 encoded value of the user certificate attribute is written to the file "huber\_certificate.cer".

```
show /O=my-company/OU=sales/CN=Huber \
     -attr UC_ANY_FILE=huber_certificate.cer
```

### 1.2.6. Base-64 Encoded Attribute Values

Single attribute values can also be specified Base-64 encoded or be read Base-64 encoded. The syntax is as follows:

*attribute***[\_ANY]\_B64**[= '*attribute\_value*']

where *attribute* specifies the attribute type and *attribute\_value* specifies the Base-64 encoded attribute value when creating or modifying this value.

If the keyword **\_ANY** is specified, the value contains the valid ASN.1 encoded value of the attribute regardless of the syntax for the attribute defined in the directory schema. If **\_ANY** is not used, the syntax of the attribute according to the schema is assumed, and the value contains the Base 64-encoded representation of the attribute value.

Structured attributes can only be specified with the keyword **\_ANY**. Otherwise an error is returned.

When the syntax of the attribute is not OCTET STRING null bytes in the value are not allowed.

Base-64 encoded attribute values must be enclosed in single quotes ( '*attribute value*' ).

Specifying the attribute value Base-64 encoded is possible for the following **dirxcp** operations:

- **obj create** (-**attribute** option)
- **obj modify** (-**addattr**, -**changeattr**, -**removeattr** options)
- **obj compare** (-**attribute** option)
- **obj search** (-**attribute** option)
- **obj show** (-**attribute** option)

**Examples:**

The following example displays the Street Address attribute (**STA**) of the person **/o=My-Company/ou=Development/cn=Digger** Base-64 encoded:

```
show {/o=my-company/ou=Development/cn=Digger} -attr sta_b64 -p
```

The output of the sample command as follows:

```
1) /O=My-Company/OU=Development/CN=Digger
   Street-Address : 'MjQgRG91Z2FuIFN0cmVldA=='
```

## 1.3. Distinguished Names

A distinguished name consists of a list of one or more relative distinguished names (RDNs), each beginning with a forward slash (/). The list of relative distinguished names starts from the root entry. For example:

```
/O=my-company/OU=dap11/CN=schmid, OU=ap11
```

Each RDN consists of one or more naming attributes in the following format:

```
type=value[, type=value]...
```

where *type* is an abbreviation or an OID that corresponds to a naming attribute type and *value* is the string representation that corresponds to the attribute syntax assigned to the attribute type. For example:

```
c=de
```

or

```
2.5.6.2=de
```

or

```
OU=dap11, L=munich
```

The maximum length of an ASN.1 encoded RDN is 470 bytes. The value length of the naming attribute is affected by

- the length of the object identifier
- the number of 2-byte UTF8 characters contained in the attribute value
- the number of attribute value assertions (AVAs) the RDN consists of.

## 1.4. Search Filters

Use search filter expressions to specify a filter in a **dirxcp search** operation. A search filter is composed of one or more simple attributes, structured attributes, or distinguished name strings separated by search filter operators. The following virtual attributes must not be



used in search filters:

- numSubordinates
- numAllSubordinates
- structuralObjectClass

If the listed virtual attributes are used, an **Unwilling to Perform** error will be returned.

Recall from section **Tcl Command Language** in chapter **DirX Directory Commands** that you must enclose the filter expression in curly brackets (**{}**) or quotation marks (**"**) if attribute values contain a space character.

The operators in the following table can be used in search filter expressions.

| Operator | Meaning                                                             |
|----------|---------------------------------------------------------------------|
| &&       | To “logically AND” two specified conditions                         |
|          | To “logically OR” two specified conditions                          |
| !        | To “logically NEGATE” a specified condition                         |
| =        | To specify equality                                                 |
| ~=       | To specify phonetic matching                                        |
| >        | To match values that are greater than or equal to a specified value |
| >=       | To match values that are greater than or equal to a specified value |
| <        | To match values that are less than or equal to a specified value    |
| <=       | To match values that are less than or equal to a specified value    |
| *        | To specify substrings                                               |
| ( )      | To nest search filter expressions                                   |

White space in a search filter string expression is ignored.

### 1.4.1. Search Filter Expression Example

The following sample search filter string

```
(cn~=schmid)&&(ocl=orp|ocl=rep)&&!(sn=ronnie)
```

directs **dirxcp** to search for names that meet all the following criteria:

- Have an object class attribute value of **Organizational-Person** or **Residential-Person**
- Have a **Common-Name** attribute value that approximately matches **schmid**
- Do not have a Surname attribute value of **ronnie**.

The following search filter string test for the presence of the **Common-Name** attribute type:

```
(c=de)&&(cn)
```

## 1.4.2. Search Operator Order of Precedence

Search filter operators have the following precedence when combined in expressions within a search filter string:

1. The filter nesting operators () are evaluated first.
2. The “logical NEGATE” operator is evaluated next.
3. The “logical AND” operators are evaluated next.
4. The “logical OR” operator is evaluated last.

## 1.4.3. Search Filter Expressions and Reserved Characters

You can use any of the reserved characters (described in section **Reserved Characters** above in this chapter) in a search filter string. For example, use the quote characters to enclose an attribute that uses white space as data within a search filter string:

```
{ou=sni && cn='Henri Mueller' && tn=89989}
```



The filter expression is enclosed in curly brackets because the cn attribute value contains the space character.

## 1.4.4. Search Filter Expressions and Matching Rules

Some structured attributes have special matching rules that require that they must be specified with special syntax when they are used in filters. For example, the structured attribute type **Protocol-Information** is normally specified as follows:

```
PI={NA=network_address, PRO=profiles}
```

However, because it has special matching rules, this attribute must be specified as an octet string without the components when used in a search filter string:

```
PI_OCT=octet_string
```

For each attribute type default matching rules are defined, for example a matching rule for equality matching, a matching rule for substring matching, etc.

The extensible match filter condition allows to specify a different matching rule than the default one. It also allows to extend the search on the attributes in the distinguished name that are not stored as attributes in an entry itself. The syntax of the extensible match filter condition is specified as follows:

```
EM={MR=matching_rule [, AT=attribute_type], MV=matching_value, DNA=boolean}
```

The **MR**=*matching\_rule* component and the **AT**=*attribute\_type* component can be specified either as object identifiers or as abbreviations. The **MV**=*matching\_value* component specifies the searched attribute value. The **DNA**=*boolean* component can have the value **1**, **TRUE**, **0**, or **FALSE**. The value **1** or **TRUE** indicates that the attributes of the distinguished name of the entry are to be used in addition to those of the entry in

evaluating the match.

The following filter tests for the presence of the **Organizational Unit** attribute with value **Sales**, using the matching rule **Case-Ignore-Match**, also evaluating the attributes of the distinguished name:

```
EM={AT=OU, MV=Sales, DNA=TRUE, MR=CIM}
```

To search for entries where the **Organizational Unit** attribute has a value starting with **S** specify the extensible match filter condition as follows:

```
EM={AT=OU, MV=S, DNA=TRUE, MR=CISM}
```

## 1.5. Attribute Syntax

Both simple and structured attributes are assigned default attribute syntaxes that determine the form in which you must supply the attribute values and the allowed character set. For example, the Common-Name simple attribute has as its default syntax Directory-String, which means the value must be a string of the first 256 Latin-1 characters of the Latin-1 (ISO 8859-1) character set. If you supply this attribute to the **dirxadm** or **dirxcp** commands, you must supply its value in the Latin-1 character set. Note that for the Directory-String syntax and a number of other syntaxes you can also supply strings of T61 characters as values. This is described in Chapter 1 in the **dirxcp** and **dirxadm** reference pages.

Attribute syntaxes are identified by name or abbreviation. For example, the abbreviation **DST** represents the Directory-String simple attribute syntax, and the abbreviation **ACIITEM** represents the Access-Control-Information-Item structured attribute syntax.

To see the names of attribute syntaxes, use the **dirxadm** and **dirxcp abbr show** commands or list the contents of the abbreviation files.

### 1.5.1. Overriding Default Attribute Syntax

The abbreviation files define the default syntax associated with each attribute type. When you specify an attribute, the **dirxcp** and **dirxadm** programs consult the abbreviation files to determine the attribute's legal formats and its character encoding methods.

You can override the default attribute syntax by appending an underscore and the attribute syntax abbreviation to the specified attribute type, as follows:

*attribute-type\_attribute-syntax-abbreviation*

where *attribute-type* specifies the attribute type and *attribute-syntax-abbreviation* is the attribute syntax abbreviation (which can be found by examining the abbreviation files).

For example, the attribute type CN has an attribute syntax of Directory-String. To encode the attribute type/value association **CN=naik** with a Printable-String syntax, append the abbreviation **\_PRI** for printable string, as shown below:

```
CN_PRI=naik
```

This causes the value to use Printable-String syntax instead of Directory-String.

If you use an attribute syntax abbreviation to override an attribute's default syntax, the **dirxcp** and **dirxadm show** and **search** operations return the attribute type and selected attribute syntax abbreviation to indicate that it is different from the default. For example, if you specify **cn\_pri=naik**, the programs return **cn\_pri=naik** in their output.

If the default syntax is overridden with the syntax ANY, the ASN.1 decoding is suppressed. This may be useful to write an ASN.1 encoded value of an attribute to a file. (See chapter titled **Attribute Values in a File** for details). Note that the **dirxcp** and **dirxadm** commands are not able to display ASN.1 encoded values.

## 1.5.2. Syntax Abbreviations

The following is a list of attribute syntax abbreviations which can be used to override default attribute syntaxes. Attribute syntax abbreviations are case-insensitive.

| Syntax           | Abbreviation                            |
|------------------|-----------------------------------------|
| ANY              | ANY                                     |
| ATTR_TYPE        | ATT                                     |
| BIT_STR          | BIT                                     |
| BMP_STR          | BMP                                     |
| BOOLEAN          | BOO                                     |
| DIR_STR          | DSTR                                    |
| DIR_STR_T61      | DT61<br>(Note that the default is BMP.) |
| DIR_STR_PRI      | DPRI                                    |
| DIR_STR_BMP      | DBMP                                    |
| DIR_STR_UNI      | DUNI                                    |
| GENERALIZED_TIME | GET                                     |
| IA5_STR          | IA5                                     |
| INTEGER          | INT                                     |
| NUMERIC_STR      | NUM                                     |
| OBJECT_ID        | OID                                     |
| OCTET_STR        | OCT                                     |
| PRINTABLE_STR    | PRI                                     |
| TTX_STR          | TTX                                     |
| T61_STR          | T61                                     |
| UNIVERSAL_STR    | UNI                                     |
| UTC_TIME         | UTC                                     |

### 1.5.3. Undefined Types and Abbreviations

To specify an attribute type that has not been defined in an abbreviation file, use the attribute type OID and append to it the abbreviation of the attribute syntax, for example, **1.2.325.67890.4.2\_PRI**.

In the output, the **dirxcp** and **dirxadm** programs return a string in the form:

*oid\_attribute-syntax-abbreviation=attribute\_value*

For example:

**1.2.325.67890.4.2\_PRI=xyz**

which indicates that the attribute with the object identifier 1.2.325.67890.4.2 and the attribute syntax Printable-String has a value of xyz, which is to be encoded as a printable string.

### 1.5.4. Attribute Syntax and Equality Matching Rule Syntax

In some instances an attribute's Equality-Matching Rule syntax differs from its attribute syntax. These attributes, their abbreviations, attribute syntax and Equality Matching Rule syntax are listed in the following table.

| Abbr. | Attribute-Name         | Attribute Syntax           | Equality-Match-Rule              |
|-------|------------------------|----------------------------|----------------------------------|
| AT    | Attribute-Type         | ATTR_TYPE_DESC             | ATTR_TYPE                        |
| CDSA  | Cooperating-DSA        | COOPERATING_DSA            | DN                               |
| CK    | Consumer-Knowledge     | CONS_INFO                  | SUPPL_OR_CONS_INFO_MATCH_SET     |
| DCR   | DIT-Content-Rules      | DITCONTENT_RULE_DESCR      | OBJECT_ID                        |
| DSAP  | DSA-Policy             | DSA_POLICIES               | DN                               |
| DSR   | DIT-Structure-Rules    | DITSTRUCT_RULE_DESCR       | INTEGER                          |
| EACI  | Entry-ACI              | ACIITEM                    | DIR_STR                          |
| MAC   | My-Access-Point        | ACC_POINT                  | DN                               |
| MR    | Matching-Rules         | MRULE_DESCR                | OBJECT_ID                        |
| MRU   | Matching-Rule-Use      | MRULE_USE_DESCR            | OBJECT_ID                        |
| NF    | Name-Forms             | NAME_FORM_DESCR            | OBJECT_ID                        |
| NSK   | Non-Specific-Knowledge | MASTER_AND_SHAD_ACC_POINTS | MASTER_AND_SHAD_ACC_POINTS_MATCH |
| OC    | Object-Classes         | OBJECT_CLASS_DESCR         | OBJECT_ID                        |
| PACI  | Prescriptive-ACI       | ACIITEM                    | DIR_STR                          |
| PI    | Protocol-Information   | PROTO_INFO                 | OCTET_STR                        |
| SACI  | Subentry-ACI           | ACIITEM                    | DIR_STR                          |

| Abbr. | Attribute-Name     | Attribute Syntax           | Equality-Match-Rule              |
|-------|--------------------|----------------------------|----------------------------------|
| SES   | Secondary-Shadows  | SUPPL_AND_CONSS            | DN                               |
| SK    | Specific-Knowledge | MASTER_AND_SHAD_ACC_POINTS | MASTER_AND_SHAD_ACC_POINTS_MATCH |
| SPK   | Superior-Knowledge | ACC_POINT                  | DN                               |
| SUK   | Supplier-Knowledge | SUPPL_INFO                 | SUPPL_OR_CONS_INFO_MATCH_SET     |
| USP   | User-Policy        | USR_POLICIES               | DN                               |

### 1.5.5. Attribute Syntax vs. Equality Matching Rule Syntax

For those attributes for which the Equality Matching Rule syntax differs from the attribute syntax, you must specify the attribute to **dirxcp** and **dirxadm** according to the following conditions

- Use the attribute syntax with the:
  - attribute** option for **create** operations
  - addattr** option for **modify** operations
  - changeattr** option for **modify** operations (but only for the new attribute value)
- Use the Equality Matching Rule syntax in the:
  - filter** option for **search** operations, if EQUALITY matching has been chosen
  - removeattr** option for **modify** operations, if an attribute with it's value has been entered (e.g., TN=41098)
  - changeattr** option for **modify** operations (but only for the OLD attribute value)
  - attribute** option for **compare** operations

Each of the attributes for which the syntaxes differ is described in the following table.

| Syntax                                                                                      | Notes                                                                                                                                                                                                                                                                                                                    |
|---------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attribute-Syntax:</b><br>ATTR_TYPE_DESC<br><b>Equality Matching Syntax:</b><br>ATTR_TYPE | The Equality Matching-Rule is objectIdentifierFirstComponentMatch, which means that only the component ID of the ATTR_TYPE_DESCR should match. Enter the attribute as: <b>AT=CN</b> or <b>AT=2.5.4.3</b> . This Equality Matching Rule is a simple one and only requires the attribute value without any component name. |

| Syntax                                                                                                    | Notes                                                                                                                                                                                                                                                                                                                                |
|-----------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attribute-Syntax:</b><br>COOPERATING_DSA<br><b>Equality Matching Syntax:</b><br>DN                     | The Equality Matching-Rule is DistinguishedNameFirstComponentMatch, which means that only the component DSA of the COOPERATING_DSA should match. Enter the attribute as: <b>CDSA={/O=my-company/OU=asw/CN=dsa/CN=asw-dsa}</b> . This matching rule is a simple one and only requires the attribute value without any component name. |
| <b>Attribute-Syntax:</b><br>CONS_INFO<br><b>Equality Matching Syntax:</b><br>SUPPL_OR_CONS_INFO_MATCH_SET | The Equality-Matching-Rule is that the component AE of the Access-Point subcomponent CON of the CONS_INFO and additionally the component AGR should match. Enter the attribute as <b>CK={AE={/O=my-company/OU=asw/CN=dsa/CN=asw-dsa}, AGR=10}</b> .                                                                                  |
| <b>Attribute-Syntax:</b><br>DITCONTENT_RULE_DESCR<br><b>Equality-Matching-Syntax:</b><br>OBJECT_ID        | The Equality-Matching-Rule is objectIdentifierFirstComponentMatch, which means that only the component SOC of the DITCONTENT_RULE_DESCR should match. Enter the attribute as <b>DCR=ORP</b> or <b>DCR=2.5.6.7</b> . This matching rule is a simple one and only requires the attribute value without any component name.             |
| <b>Attribute-Syntax:</b><br>DSA_POLICIES<br><b>Equality-Matching-Syntax:</b><br>DN                        | The Equality-Matching-Rule is DistinguishedNameFirstComponentMatch, which means that only the component DSA of the DSA_POLICIES should match. Enter the attribute as <b>DSAP={/O=my-company/OU=asw/CN=dsa/CN=asw-dsa}</b> . This matching rule is a simple one and only requires the attribute value without any component name.     |
| <b>Attribute-Syntax:</b><br>DITSTRUCT_RULE_DESCR<br><b>Equality-Matching-Syntax:</b><br>INTEGER           | The Equality-Matching-Rule is integerFirstComponentMatch, which means that only the component ID of the DITSTRUCT_RULE_DESCR should match. Enter the attribute as <b>DSR=7</b> . This matching rule is a simple one and only requires the attribute value without any component name.                                                |
| <b>Attribute-Syntax:</b><br>ACIITEM<br><b>Equality-Matching-Syntax:</b><br>DIR_STR                        | The Equality-Matching-Rule is directoryStringFirstComponentMatch, which means that only the component ID of the ACIITEM should match. Enter the attribute as <b>PACI=Administrator-ACI</b> . This matching rule is a simple one and only requires the attribute value without any component name.                                    |

| Syntax                                                                                                                         | Notes                                                                                                                                                                                                                                                                                                              |
|--------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attribute-Syntax:</b><br>ACC_POINT<br><b>Equality-Matching-Syntax:</b><br>DN                                                | The Equality-Matching-Rule is that the component AE of the ACC_POINT should match. Enter the attribute as <b>MAC={/O=my-company/OU=asw/CN=dsa/CN=dsa1}</b> . This matching rule is a simple one and only requires the attribute value without any component name.                                                  |
| <b>Attribute-Syntax:</b><br>MRULE_DESCR<br><b>Equality-Matching-Syntax:</b><br>OBJECT_ID                                       | The Equality-Matching-Rule is objectIdentifierFirstComponentMatch, which means that only the component ID of the MRULE_DESCR should match. Enter the attribute as <b>MR=CEM</b> or <b>MR=2.5.13.5</b> . This matching rule is a simple one and only requires the attribute value without any component name.       |
| <b>Attribute-Syntax:</b><br>MRULE_USE_DESCR<br><b>Equality-Matching-Syntax:</b><br>OBJECT_ID                                   | The Equality-Matching-Rule is objectIdentifierFirstComponentMatch, which means that only the component ID of the MRULE_USE_DESCR should match. Enter the attribute as <b>MRU=CEM</b> or <b>MRU=2.5.13.5</b> . This matching rule is a simple one and only requires the attribute value without any component name. |
| <b>Attribute-Syntax:</b><br>NAME_FORM_DESCR<br><b>Equality-Matching-Syntax:</b><br>OBJECT_ID                                   | The Equality-Matching-Rule is objectIdentifierFirstComponentMatch, which means that only the component ID of the NAME_FORM_DESCR should match. Enter the attribute as <b>NF=LNF</b> or <b>NF=2.5.15.1</b> . This matching rule is a simple one and only requires the attribute value without any component name.   |
| <b>Attribute-Syntax:</b><br>MASTER_AND_SHAD_ACC_POINTS<br><b>Equality-Matching-Syntax:</b><br>MASTER_AND_SHAD_ACC_POINTS_MATCH | The Equality-Matching-Rule is that the component AE of all Access-Points should match. Enter the attribute as <b>NSK={AE={/O=my-company/OU=asw/CN=dsa/CN=dsa1}; /O=my-company/OU=asw/CN=dsa/CN=dsa2}}</b>                                                                                                          |
| <b>Attribute-Syntax:</b><br>OBJECT_CLASS_DESCR<br><b>Equality-Matching-Syntax:</b><br>OBJECT_ID                                | The Equality-Matching-Rule is objectIdentifierFirstComponentMatch, which means that only the component ID of the OBJECT_CLASS_DESCR should match. Enter the attribute as <b>OC=ORP</b> or <b>OC=2.5.6.7</b> . This matching rule is a simple one and only requires the attribute value without any component name. |
| <b>Attribute-Syntax:</b><br>PROTO_INFO<br><b>Equality-Matching-Syntax:</b><br>OCTET_STR                                        | The Equality-Matching-Rule is that the component NA of the PROTO_INFO should match. Enter the attribute as <b>PI=\x0a\x0b</b> . This matching rule is a simple one and only requires the attribute value without any component name.                                                                               |



| Syntax                                                                                | Notes                                                                                                                                                                                                                                                                                                         |
|---------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Attribute-Syntax:</b><br>SUPPL_AND_CONSS<br><b>Equality-Matching-Syntax:</b><br>DN | The Equality-Matching-Rule is that the component AE of the Access-Point subcomponent SUP of the SUPPL_AND_CONSS should match. Enter the attribute as <b>SES={/O=my-company/OU=asw}</b> . This matching rule is a simple one and only requires the attribute value without any component name.                 |
| <b>Attribute-Syntax:</b><br>USR_POLICIES<br><b>Equality-Matching-Syntax:</b><br>DN    | The Equality-Matching-Rule is DistinguishedNameFirstComponentMatch, which means that only the component USN of the USR_POLICIES should match. Enter the attribute as <b>USP={/O=my-company/OU=asw}</b> . This matching rule is a simple one and only requires the attribute value without any component name. |

## 1.6. String Representations for Simple Attribute Syntaxes

The following section describes the string representations for simple attribute syntaxes.

### 1.6.1. Attribute Type Syntax

Specify an attribute type as an abbreviation (defined in the Attribute Definition Block in an abbreviation file) or the attribute type's object identifier (OID) in dotted notation (for example, 1.2.5.6).

### 1.6.2. Bit String Syntax

Specify a bit string as a sequence of 1's and 0's (for example, 11110100100001001101101).

### 1.6.3. Boolean Syntax

Specify a boolean as the integer **1** (true) or **0** (false) or the keyword **TRUE** or **FALSE**.

### 1.6.4. Country Name Syntax

Specify a country name as a two-letter code according to the definitions in ISO 3166. Appendix A lists the countries and their two-letter name codes.

### 1.6.5. Destination Indicator Syntax

Specify the country and city needed to provide the public telegram service according to CCITT Rec. F.1 and Rec. F.31. It is a printable string (alphabetical characters only) of maximum length 128.

## 1.6.6. Directory String Syntax

Specify a directory string using any of the following four syntaxes.

- **T61 String Syntax** - If the Tcl variable `_t61` is set to **TRUE**, you must use the T.61 character set for the directory string (for example, `R\xF9m\xF9`, which has the Latin-1 representation `Rømø`). If the Tcl variable `_t61` is set to **FALSE**, you must use the BMP string syntax. See the description of BMP string syntax below in this section.
- **Printable Syntax** - Specify the directory string as a sequence of characters. Valid characters are A to Z, a to z, 0 to 9, the space character, and the special characters: ' (apostrophe), ( (left parentheses), ) (right parentheses), + (plus sign), , (comma), - (hyphen), . (period), / (slash), : (colon), = (equal sign), and ? (question mark) (for example, South Africa).
- **BMP String Syntax** - Specify the directory string as a BMP string. A BMP (Basic Multilingual Plane) string is the internal representation of strings used by X.500 protocols and DSAs. Specify a BMP string depending on the value of the TCL variables `_t61` and `_localcode`. (See section **TCL Variables** in the *DirX Directory Administration Reference* for details.) When `_t61=TRUE` T61 format is used as described above (for example, `Locality_T61="M\x8uncchen"`, T61 format of the value München). `_t61=FALSE` species a range of Latin-1 (ISO 8859-1) characters, with Hex values ranging from 20 to 7E and A0 to FF (for example, `Locality_BMP="M\xfcnchen"`, Latin-1 format of the value München). (See also ISO 10646 Part 1: Architecture and Basic Multilingual Plane.)
- **Universal String Syntax** - Specify the directory string using BMP string syntax. See the description of BMP string syntax above in this section.
- **UTF-8 String Syntax** - Specify the directory string using UTF-8 string syntax. A UTF-8 string is a transformation format of the Universal Character Set (UCS) as defined in ISO 10646. See RFC 2279 for details.

## 1.6.7. DSE Type Syntax

DSE Type syntax is a bit string syntax that represents the role that the DSE plays in regard to the entry and its distribution. Note that DSEs can have multiple DSE types. For example, a DSE representing a subordinate reference can also represent a shadowed context prefix that is an administrative point. However, some combinations of DSE types are meaningless.) (Every DSE in a DSA has a DSA-specific attribute of this syntax; the attribute is accessible using **dirxadm** only.)

Specify DSE type syntax in the following format:

`DSE_TYPE = DSE_Type_combination`

**where:**

**dse\_type**

Is the abbreviation or OID that corresponds to an attribute with the DSE Type syntax. (The DSET attribute is the sole example of such an attribute. The DSET attribute is not part of a normal Directory entry, visible using the Directory service. Instead it is an attribute of the DSE (DSA Specific Entry) that represents aspects of entries in particular DSAs.)

## DSE\_Type\_combination

Is a list of one or more keywords separated by the + (plus sign) that define the role or roles of the DSE. The possible keywords are:

- **ROOT** - Used exclusively by the DSE that represents the root of the DIT in a particular DSA.
- **GLUE** - A DSE that represents a name only. A DSA uses glue DSEs to hold the names of the superiors of the context prefixes and cross references where there is no other knowledge associated with them. You cannot assign any other DSE types to a DSE typed as GLUE.
- **CP** - A DSE that represents the context prefix of a naming context.
- **ENTRY** - A DSE that represents the entry itself, normally including the user attributes. If the SHADOW bit is not set, the DSE represents the master copy of the entry and holds all the attributes associated with the entry.
- **ALIAS** - A DSE that represents an alias entry.
- **SUBR** - A DSE that represents a specific subordinate reference and holds a corresponding knowledge attribute.
- **NSSR** - A DSE that represents a non-specific subordinate reference, and holds a corresponding knowledge attribute.
- **SUPR** - Used to mark a root DSE that holds a knowledge attribute that represents a specific superior reference. (See the **ROOT** keyword.)
- **XR** - A DSE that represents a cross-reference, and holds a corresponding knowledge attribute.
- **ADM\_POINT** - A DSE that represents an administrative point.
- **SUBENTRY** - A DSE that represents a subentry.
- **SHADOW** - A DSE that represents shadowed information.
- **IMM\_SUPR** - A DSE that represents an immediate superior reference and holds a corresponding knowledge attribute.
- **RHOB** - A DSE that holds policy information (either administrative point or subentry information), which has been received from a superior DSA as the result of a hierarchical operational binding.
- **SA** - An additional characteristic of a subordinate reference DSE (i.e., of type **SUBR**), which is used to indicate that the context prefix pointed to by the reference is an alias and not a normal entry.

The following lists the most commonly used types and combinations in the absence of shadowing:

- **ROOT+SUPR**
- **GLUE** (must be used alone)
- **ENTRY +CP+ADM\_POINT**
- **ALIAS +CP**

- **SUBENTRY** (used alone)
- **IMM\_SUPR** (used alone)

An example of an attribute in DSE Type syntax follows:

```
DSET=ENTRY+ADM_POINT
DSET=GLUE
```

### 1.6.8. Generalized Time Syntax

Specify generalized time as a simple string. The value of the string is the concatenation of the 8 year-month-day digits (YYYYMMDD), plus the six hour-minute-second digits (HHMMSS), plus a time zone difference of **Z** (designating GMT), +HHMM, or -HHMM. The three possible forms, then, are:

- YYYYMMDDHHMMSSZ
- YYYYMMDDHHMMSS+HHMM
- YYYYMMDDHHMMSS-HHMM

For example, 19970101123000Z specifies Jan 1, 1997, 12:30:00 GMT.



The only difference between generalized time and UTC time is that the year is specified with four digits instead of two.

### 1.6.9. IA5 String Syntax

Specify an IA5 string using 7-bit ASCII characters, with valid Hex values in the range 20 to 7E (for example, [smith@my-company.de](#)).

### 1.6.10. Integer String Syntax

Specify as an integer in the range 0 to  $2^{32} - 1$  (4,294,967,295). For example, 65535.

### 1.6.11. International ISDN Number Syntax

Specify as a numeric string (see **Numeric String Syntax**) of maximum length 16 which complies with the internationally agreed format for ISDN addresses given in CCITT Rec. E.164.

### 1.6.12. ISO646 String Syntax

Specify an ISO 646 string using 7-bit ASCII characters, with valid Hex values in the range 20 to 7E (for example, Washington D.C.).

### 1.6.13. Local Scope Policy Syntax

Local Scope Policy syntax is used for DSA configuration attributes that specify the interpretation of the local-scope service control.

Specify Local Scope Policy syntax in the following format:

*LocalScopePolicy*=**DSA** | **LOCAL-DOMAIN** | **DMD**

**where:**

### **LocalScopePolicy**

Is an abbreviation or OID that corresponds to an attribute with the Local Scope Policy syntax. (The local-scope-policy (LSP) attribute, which specifies the interpretation of the local-scope service control, is an example of such an attribute. It is a single-valued attribute of the root DSE, and is accessible using **dirxadm** only.)

### **LSP=local\_scope\_interpretation**

Defines the local scope for the DSA. *local\_scope\_interpretation* takes one of the following values:

- **DSA** - The local scope is the scope of the DSA. The DSA responds on the basis of information that it holds itself, and does not chain or create referrals or continuation references
- **LOCAL-DOMAIN** - The local scope is the scope of the local domain of the DSA. The DSA chains or creates referrals or continuation references only to the group of DSAs in the local domain. A DSA belongs to the same local domain as the local DSA if a DSA policy exists for this DSA with the value **OPT=LOCAL-DOMAIN**
- **DMD** - The local scope is the scope of the Directory Management Domain (DMD) of the DSA. The DSA will chain or create referrals or continuation references only within the group of DSAs configured within the DMD. A DSA belongs to the same DMD as the local DSA if a DSA policy exists for this DSA with the value **OPT=DMD**

The default value is **DSA**.

An example of a Local Scope Policy Attribute follows:

LSP=DMD

## **1.6.14. Numeric String Syntax**

Specify as a sequence of digits (0 to 9) and space (for example, 4711 13).

## **1.6.15. Object ID Syntax**

Specify an OID as an abbreviation (defined in the OID Definition Block in an abbreviation file) or a dotted notation. For example, Organizational-Person could be specified as ORP or 2.5.6.7.

## **1.6.16. Octet String Syntax**

Specify an octet string using single-byte Hex values (00 to FF) or their Latin-1 character representation (for example \x01\x02ABC).

### 1.6.17. Preferred Delivery Method Syntax

Preferred-delivery-method syntax is a syntax for single-valued attributes that document the order of preference for message delivery methods. Specify this syntax in the following format:

*preferredDeliveryMethod=option [+option...]*

*option* is one or more of the following keywords that describe the delivery methods and the order of preference:

- **ANY** - Any method of delivery
- **MHS** - Message handling system delivery
- **PHYSICAL** - Physical delivery
- **TELEX** - Telex delivery
- **TELETEX** - Teletex delivery
- **G3\_FAX** - G3 FAX delivery
- **G4\_FAX** - G4 FAX delivery
- **IA5** - IA5 terminal delivery
- **VIDEOTEX** - Videotex delivery
- **TELEPHONE** - Telephone delivery

The keywords are specified diminishing order of preference, with the most preferred method first in the list. Separate multiple keywords with a plus sign (+). For example:

**PDM=MHS+TELETEX+TELEPHONE**

### 1.6.18. Printable String Syntax

Specify a printable string as a sequence of characters. Valid characters are

A to Z,  
a to z,  
0 to 9,  
the space character,

and the special characters:

' (apostrophe),  
( (left parentheses),  
) (right parentheses),  
+ (plus sign),  
, (comma),  
- (hyphen),  
. (period),  
/ (slash),  
: (colon),

= (equal sign),  
? (question mark).

An example of Printable String Syntax follows:

Smith/My-Company .

### 1.6.19. SDSE Type Syntax

Specify a SDSE (shadow DSE) type the same as a DSE type (described earlier in this section).

### 1.6.20. TTX String Syntax

Specify a Teletex (TTX) string the same as a directory T61 string (described earlier in this section).

### 1.6.21. Unique Identifier Syntax

Specify a unique identifier the same as a bit string. See the description of bit string syntax earlier in this section.

### 1.6.22. Universal String Syntax

Specify a universal string the same as a BMP string. See the description of BMP string syntax in the **Directory String Syntax** section of this chapter.

### 1.6.23. UTC Time Syntax

Specify UTC time as a simple string. The value of the string is the concatenation of the six year-month-day digits (YYMMDD), plus the six hour-minute-second digits (HHMMSS), plus a time zone difference of **Z** (designating GMT), **+HHMM**, or **-HHMM**. The three possible forms, then, are:

- YYMMDDHHMMSSZ
- YYMMDDHHMMSS+HHMM
- YYMMDDHHMMSS-HHMM

For example, 970101123000Z specifies Jan 1, 1997, 12:30:00 GMT.

### 1.6.24. X121 Address Syntax

Specify as a numeric string (see **Numeric String Syntax**) of maximum length 15 which complies with an address as defined by CCITT Rec.X.121 associated with an object.

## 1.7. String Representations for Structured Attribute Syntaxes

This section describes the following structured attribute syntaxes supported by DirX

## Directory.

- Syntaxes for knowledge attribute types
  - Access-Point
  - Consumer-Information
  - Master-And-Shadow-Access-Points
  - Supplier-Information
  - Suppliers-And-Consumers
- Syntaxes for certificate attribute types and certificate subcomponents
  - Certificate (Signed)
  - Certificate (Unsigned)
  - Certificate Pair
  - Cert-List (Signed)
  - Cert-List (Unsigned)
- Syntaxes for operational attribute types and subcomponents
  - ACI-Item
  - Cooperating-DSA
  - DSA-Policies
  - Paging-Policies
  - Shadowing-Agreement-Info
  - SOB-Policies
  - Update-Status
  - User-Policies
- Syntaxes for miscellaneous attribute types and subcomponents
  - Continuation-Reference
  - Facsimile-Telephone-Number
  - Name-And-Optional-UID
  - Partial-Outcome-Qualifier
  - Postal-Address
  - Presentation-Address
  - Protocol-Information
  - Subtree-Specification
  - Teletex-Terminal-Identifier
  - Telex-Number



## 1.7.1. Access-Point

An attribute syntax for knowledge attributes that specify DSA access points (that is, the names and addresses of DSAs that can interact with the holding DSA using a Directory protocol).

### 1.7.1.1. Synopsis

```
AccessPoint = {AE=application_entity_title,  
               PSAP=presentation_address  
               [,PRINFO=protocol_info]  
               [,DISP-KIND=disp_kind]}
```

### 1.7.1.2. Attribute Type

#### AccessPoint

The abbreviation or OID that corresponds to a structured attribute with the Access-Point (ACC\_POINT) syntax. SPK (Superior Knowledge) is an example of such an attribute.

### 1.7.1.3. Components

#### **AE**=*application\_entity\_title*

The distinguished name of the DSA.

#### **PSAP**=*presentation\_address*

The presentation address of the DSA. Specify *presentation\_address* as a Presentation-Address attribute. (See **Presentation-Address** for details.)

#### **PRINFO**=*protocol\_info*

A structured component with the Protocol-Information syntax. See **Protocol-Information** in this chapter for a complete description of protocol information format. If it is specified, its NSAP address component must correspond to one of the NSAP addresses given in the presentation address



The **PRINFO** component is not required for TCP/IP communications; it can be omitted.

#### **DISP-KIND**=*disp\_kind*

Whether the DSA participates in the central cooperating DSA table administration (**CENTRALADMIN**) or supports only X.500 compliant shadowing (**X500**).



This value is specified when creating a shadowing agreement. (See the **sob create** operation in the *DirX Directory Administration Reference* for details.) Perform a **sob show** operation to display the value.

### 1.7.1.4. Description

The Access-Point specifies distinguished names and addresses of DSAs that can interact with the holding DSA using a Directory protocol. The DSA uses, for example, the SPK

(Superior Knowledge) attribute information when constructing a continuation reference returned in a DAP or DSP referral or when performing chaining.

#### 1.7.1.5. Example

```
SPK={AE={/O=My-Company/CN=dsa},  
PSAP={TS=Server19,NA='TCP/IP!internet=136.30.74.3+port=102'}}
```

### 1.7.2. Consumer-Information

An attribute syntax for DSA-specific attributes of context prefixes. The Consumer Knowledge attribute (CK) is the only such attribute in the default DSA schema. It describes consumer DSAs that shadow all or part of the corresponding naming context. It specifies the access points of each DSA together with the numeric identifier for the corresponding operational binding.

#### 1.7.2.1. Synopsis

```
ConsumerInformation = {CON=access_point_of_consumer,  
AGR=operational_binding_ID}
```

#### 1.7.2.2. Attribute Type

##### **ConsumerInformation**

The abbreviation or OID that corresponds to a structured attribute with the Consumer-Information (CONS\_INFO) syntax. CK (Consumer Knowledge) is such an attribute. The CK attribute is an attribute of the context prefixes of shadowed naming contexts in supplier DSAs, and is accessible with **dirxadm** only.

#### 1.7.2.3. Components

##### **CON**=*access\_point\_of\_consumer*

The access point of a consumer DSA which shadows all or part of the naming context corresponding to the context prefix holding this attribute. Specify *access\_point\_of\_consumer* as an Access-Point syntax. (See **Access-Point** for details.)

##### **AGR**=*operational\_binding\_ID*

The identifier of the operational binding for the shadow agreement. Specify *operational\_binding\_ID* in the following format:

```
{ID=operational_binding_identifier,  
VERS=version_number}
```

where:

##### **ID**=*operational\_binding\_identifier*

Is an integer that identifies the shadow agreement binding. The ID is assigned by the holding DSA as part of the establishment of the operational binding, and is unique within the scope of the two DSAs.

### **VERS=version\_number**

Is an integer that identifies the version of the shadow agreement binding. This integer increases when the operational binding is modified.

#### **1.7.2.4. Description**

Information about shadow consumers for a particular naming context held by a shadow supplier DSA are represented by a multi-valued Consumer Knowledge operational attribute associated with the context prefix of the naming context. The information identifies the access points of the consumer DSAs and also the identifiers for the corresponding shadowing agreements. Note that there can be several shadowing agreements involving this naming context with a specific consumer DSA.

#### **1.7.2.5. Examples**

```
CK={CON={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa},
  PSAP={TS=Server19,NA='TCP/IP!internet=123.45.67.89+port=102'},
  AGR={ID=10,VERS=20}
}
```

### **1.7.3. Master-And-Shadow-Access-Points**

An attribute syntax for distributed-operation attributes that specify access points for partner DSAs within a distributed DIT. The SK attribute (Specific Knowledge) is an example of such an attribute in the default DSA schema.

#### **1.7.3.1. Synopsis**

*MasterAndShadowAccPoints*={**MOS**=MasterOrShadowAccPoint}



The value is a set of components of type MasterOrShadowAccPoint

#### **1.7.3.2. Attribute Type**

##### **MasterAndShadowAccPoints**

The abbreviation or OID that corresponds to a structured attribute with the Master-And-Shadow-Acc-Points (MASTER\_AND\_SHAD\_ACC\_POINTS) syntax. The SK (Specific Knowledge) attribute is an operational attribute that represent knowledge references, and is accessible using **dirxadm** only. The SK attribute is used for DSEs of type subordinate reference, cross-reference and immediate superior reference.

#### **1.7.3.3. Components**

##### **MOS=MasterOrShadowAccPoint**

The access for the master or a shadow DSA. Specify *MasterOrShadowAccPoint* in the following format:

```
{MSAP=access_point  
[,CAT=category]}
```

where:

#### **MSAP=access\_point**

Is the access point for the master or a shadow DSA. *access\_point* is an **AccessPoint** component. See **AccessPoint** for details on the component format.

#### **CAT=category**

Specifies whether the access point is for the master or a shadow DSA. Specify *category* as one of the following keywords:

- **MASTER**
- **SHADOW**

The default is **MASTER**. Only one MasterOrShadowAccPoint value can be assigned a category of MASTER.

### **1.7.3.4. Description**

This syntax must be used for the administration of subordinate references, cross-references and immediate superior references. Access points for the corresponding master and shadow DSAs are represented with the MasterAndShadowAccPoints (MASTER\_AND\_SHAD\_ACC\_POINTS) syntax. The access points are specified with an **AccessPoint** component and the category of the DSA (master or shadow) is also specified.

### **1.7.3.5. Examples**

```
SK={MOS={MSAP={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa1},  
                PSAP={TS=Server19,  
                      NA='TCP/IP!internet=123.45.67.89+port=102' }},  
      CAT=MASTER};  
{MSAP={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa2},  
      PSAP={TS=Server19,  
            NA='TCP/IP!internet=139.88.99.1+port=102' }},  
      CAT=SHADOW}  
}
```

### **1.7.4. Supplier-Information**

An attribute syntax for attributes that specify information known to a shadow consumer DSA about the shadow supplier DSA in a shadow agreement. There is only one such attribute in the default DSA schema: **SUK** (Supplier-Knowledge).

#### 1.7.4.1. Synopsis

```
SupplierInformation={SUP=access_point,  
  AGR=agreement_ID,  
  ISM=is_master  
  [,NSM=access_point]}
```

#### 1.7.4.2. Attribute Type

##### SupplierInformation

The abbreviation or OID that corresponds to a structured attribute with the Supplier-Information (SUPPL\_INFO) syntax. For example, SUK is an attribute of a context prefix of a shadowed naming context.

#### 1.7.4.3. Components

##### **SUP=access\_point**

The access point for the shadow supplier DSA that shadows all or part of the naming context covered by the shadow agreement specified with the **AGR** subcomponent. The shadow-supplier DSA can be the master DSA only if the agreement is not a secondary shadowing agreement. *access\_point* is an AccessPoint subcomponent. See **AccessPoint** for details on the attribute format.

##### **AGR=agreement\_ID**

The identifier of the operational binding for the shadow agreement. Specify *operational\_binding\_ID* in the following format:

```
{ID=operational_binding_identifier,  
  VERS=version_number}
```

where:

##### **ID=operational\_binding\_identifier**

Is an integer that identifies the shadow agreement binding. The ID is assigned by the holding DSA as part of the establishment of the operational binding, and is unique within the scope of the two DSAs.

##### **VERS=version\_number**

Is an integer that identifies the version of the shadow agreement binding. This integer increases when the operational binding is modified.

##### **ISM=is\_master**

A Boolean value that specifies whether the supplier holds the master copy of the naming context. The value is represented using BOOLEAN attribute syntax. (See the **Simple Attribute Syntax** for a description of BOOLEAN syntax representation.) The default value is **TRUE** (the supplier holds the master copies). If the value is **FALSE**; the shadowing agreement is a secondary shadowing agreement.

##### **NSM=access\_point**

If the shadow supplier in the shadow agreement specified by the **ARG** component does

not hold the master copy of the naming context, the **NSM** component defines the master DSA's access point. The **NSM** subcomponent is optional but should be used to optimize operations (for example, update operations) that are passed to the DSA that holds the master naming context. *access\_point* is an AccessPoint subcomponent. See **AccessPoint** for details on the subcomponent format.

#### 1.7.4.4. Description

Information about a shadow supplier is represented with the SupplierInformation (SUPPL\_INFO) syntax. This information known by a shadow consumer DSA is hold in the SUK attribute of the shadowed context prefix and consists of the ID and version number that identify the shadow agreement between the supplier and the consumer, the access point for the supplier of the copy of the naming context, and an indication of whether this supplier holds the master copy of the naming context. In addition, the information can optionally include the access point for the master if the supplier does not hold the master copy.

#### 1.7.4.5. Examples

```
SUK={SUP={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa1},
        PSAP={TS=Server19,
              NA='TCP/IP!internet=123.45.67.89+port=102' }},
  AGR={ID=1,
      VERS=1},
  ISM=TRUE,
  NSM={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa2},
      PSAP={TS=Server19,
            NA='TCP/IP!internet=139.88.99.1+port=102' }},
    {AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa3},
      PSAP={TS=Server19,
            NA='TCP/IP!internet=139.88.99.2+port=102' }}
}
```

### 1.7.5. Supplier-And-Consumers

An attribute syntax for attributes that document the suppliers of secondary shadows and their consumers. In the default DSA schema, this is used only in the SES (Secondary-Shadows) operational attribute.

#### 1.7.5.1. Synopsis

*SupplierAndConsumers*\*={**SUP**=\_access\_point,  
**CON**=access\_point[:\_access\_point\_]...}

### 1.7.5.2. Attribute Type

#### SupplierAndConsumers

The abbreviation or OID that corresponds to a structured attribute with the Supplier-And-Consumers (SUPPL\_AND\_CONSS) syntax. For example, SES (Secondary-Shadows) is an attribute of a context prefix.

### 1.7.5.3. Components

#### **SUP=access\_point**

The access point for each shadow supplier that has a secondary shadow. Specify *access\_point* as an Access-Point subcomponent. See **Access-Point** for more information.

#### **CON=access\_point**

The access point for each shadow consumer of the suppliers named with the **SUP** component. Specify *access\_point* as an AccessPoint subcomponent. See **Access-Point** for more information.

### 1.7.5.4. Description

The suppliers of secondary shadows and their consumers are represented with the Supplier-And-Consumers attribute syntax. The syntax uses an Access-Point attribute to identify the access points.

### 1.7.5.5. Examples

```
SES={SUP={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa1},
        PSAP={TS=Server19,
              NA='TCP/IP!internet=123.45.67.89+port=102' } } },
CON={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa2},
    PSAP={TS=Server19,
          NA='TCP/IP!internet=139.88.99.2+port=102' } } };
{AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa3},
  PSAP={TS=Server19,
        NA='TCP/IP!internet=139.88.99.3+port=102' } } }
}
```

## 1.7.6. Certificate (Signed)

An attribute syntax for schema attributes that specify authentication certificates that store the public key of a user and the public key of the user's certification authority.

### 1.7.6.1. Synopsis

```
Cert = {CU=certificate_unsigned,
       ALG=algorithm,
```

**EV**=*encrypted\_value*}

### 1.7.6.2. Attribute Type

#### **Cert**

The abbreviation or OID that corresponds to a structured attribute with the Cert (Cert) syntax. Two attributes in the default schema have this syntax:

- **UC (user-certificate)**, an attribute of entries with object class strong-authentication-user
- **CAC (CA-Certificate)**, an attribute of entries representing Certification Authorities

### 1.7.6.3. Components

#### **CU**=certificate\_unsigned

The unsigned user certificate. The *certificate\_unsigned* parameter is in the form of a Certificate (Unsigned) component. See **Certificate (Unsigned)** for details on its format.

#### **ALG**=\_algorithm

The OID that identifies the signature algorithm used to sign (i.e., hash the data and sign it) the certificate and any parameters used by that algorithm. Specify *algorithm* in the following format:

{**AL**=*object\_identifier*[,**ALP**=*algorithm\_parameters*]}

where:

##### **AL**=object\_identifier

Is the OID or abbreviation that identifies the algorithm used to sign the certificate. This parameter represents the combination of the method used to hash the certificate information and the public-key cryptography system employed.

##### **ALP**=algorithm\_parameters

Are parameters used by the algorithm for the signature. The form depends on the object identifier specified by the **AL** component, but is generally absent (no additional parameters needed).

#### **EV**=encrypted\_value

The encrypted value. This value is the result of the following procedure:

1. The unencrypted certificate list is encoded using ASN.1 under the constraints documented in clause 9 of X.509.
2. The results are hashed using the hashing algorithm specified by the object identifier
3. The hash results are encrypted using the encryption algorithm specified by the object identifier, and making use of the private key of the signing entity
4. The result is expressed as a bit string



#### 1.7.6.4. Description

Certificates that store either a user's public key or the public key of the user's certification authority (known as CA certificates) are represented with the Certificate attribute syntax. Both forms of certificates are unsigned until a signature is provided by the relevant certification authority. The signature represents the encrypted hash of an encoding of the certificate.

#### 1.7.6.5. Examples

```
CAC={CU={VER=V3,
        SN=10,
        SIG={AL=2.5.8.1.1,
              ALP=1024
            },
        IS={/O=my-company/OU=asw},
        VAL={VNA=9412011200Z,
              VNB=9412011500Z
            },
        SUB={/O=my-company/OU=asw/CN=wythenshawe},
        SUBK={AL={AL=2.5.8.1.1,
                    ALP=1024
                  },
              },
        PK=0011000100110010001100111100100110000001100010000001
          },
        IUID=0011000100110010001100,
        SUID=1111010100101001011110
      },
      ALG={AL=2.5.8.1.1,
            ALP=1024
          },
      EV=0011000100110010001100}
```

#### 1.7.7. Certificate (Unsigned)

A syntax for a component of an unsigned certificate.

##### 1.7.7.1. Synopsis

```
CU = {[VER=version_number]
      ,SN=serial_number
      ,SIG=signature_algorithm
      ,IS=DN_of_issuer
      ,VAL=validity}
```

,**SUB**=*DN\_of\_subject*  
,**SUBK**=*public\_key\_for\_subject*  
[,**IUID**=*unique\_ID\_of\_issuer*]  
[,**SUID**=*subject\_unique\_id*]  
[,**EXT**=*extensions;extensions;...}}*

#### 1.7.7.2. Components

##### **VER=version\_number**

The version number of the user certificate. Up to 3 versions are supported. Specify the version number as one of the following keywords.

- **V1** - Includes neither unique-identifiers or extensions (The default.)
- **V2** - Optionally includes unique-identifiers, but does not include extensions
- **V3** - Optionally includes unique-identifiers and extensions

##### **SN=serial\_number**

An integer that specifies the serial number of the user certificate. The number is unique for the Certification Authority.

##### **SIG=signature\_algorithm**

The OID that identifies the algorithm used to sign the certificate and any parameters used by that algorithm. Specify *signature\_algorithm* in the following format:

{**AL**=*object\_identifier*  
[,**ALP**=*algorithm\_parameters*]}

where:

##### **AL=object\_identifier**

Is the OID or abbreviation that identifies the algorithm used to sign the certificate. This parameter represents the combination of the method used to hash the certificate information and the public-key cryptography system employed.

##### **ALP=algorithm\_parameters**

Are parameters used by the algorithm for the signature. The form depends on the object identifier specified by the **AL** component, but is generally absent (no additional parameters needed).

##### **IS=DN\_of\_issuer**

The distinguished name of the certification authority that issued the certificate for the user.

##### **VAL=validity**

The period of time during which the certificate is valid. Specify *validity* as a range of dates in the following format:

{**VNB**=*valid\_not\_before*,  
**VNA**=*valid\_not\_after*}

where:

**VNB=valid\_not\_before**

Is the time in UTC format that begins the range.

**VNA=valid\_not\_after**

Is the time in UTC format that terminates the range.

**SUB=DN\_of\_subject**

The distinguished name of the user for whom the certificate was created.

**SUBK=public\_key\_for\_subject**

The user's public key. Specify *public\_key\_for\_subject* in the following format:

{**AL**=*algorithm\_for\_subject*,  
**PK**=*public\_key*}

where:

**AL=algorithm\_for\_subject**

Is the OID that identifies the algorithm used to encrypt the user's public key and any parameters used by the algorithm. Specify *algorithm\_for\_subject* in the following format:

{**AL**=*object\_identifier*[,**ALP**=*algorithm\_parameters*]}

where:

**AL=object\_identifier**

Is the OID of the algorithm used to encrypt the user's public key.

**ALP=algorithm\_parameters**

Are parameters used by the algorithm for the encryption.

**PK=public\_key**

The public key for the certificate user. The format depends on the algorithm specified by the AL component. It typically consists of an ASN.1 encoding of the key components.

**IUID=unique\_ID\_of\_issuer**

The UID (a bit string) of the certification authority that issued the certificate. This component is optional. When present, it distinguishes between two objects that possess the same distinguished name at different times. (They cannot both possess the same name at the same time.)

**SUID=subject\_unique\_id**

The UID (a bit string) of the user for whom the certificate was created.

**\_EXT=extensions**

Extensions provide an open-ended method for holding additional information. Each extension consists of three components:

- An identifier for the extension
- A critical extension flag indicating whether the extension can be ignored if not understood
- A value which represents the data corresponding to the extension

Specify **extensions** in the following format:

```
{ID=oid
[,CRIT=boolean value,]
EV=OCTET_STRING}
```

where:

**ID=oid**

Is the OID specifying the extension type.

**CRIT=boolean value**

Is an indication of whether or not the extension can be ignored by a user that does not understand the extension. If **TRUE**, the extension is critical, and the user must be able to understand the extension to use the certificate. (See **Simple Attribute Syntax** for a description of BOOLEAN syntax.)

**EV=octet\_string**

Is the DER encoded extension information represented within an octet string. The format of the information is dependent on the value of the OID used to identify the extension.

### 1.7.7.3. Description

The unsigned components of certificates that store user and certificate authority keys are represented with the Certificate (Unsigned) attribute syntax. The certificates, which are issued by a certificate authority, store users' certified public keys along with other pertinent information. They are used to authenticate the identity of the user.

Specifically the certificates hold the following information provided by the signing certificate authority:

- **A version number**, a number ranging from V1 to V3 that is assigned by the certificate authority, and a version number also assigned by the certificate authority
- **An ID** that identifies the algorithm to be used to sign the certificate with other parameters if needed
- **The distinguished name** of the certification authority that issued the certificate, extended, if necessary with a unique UID
- **A validity period**, specified by a begin date and an end date, that defines the lifetime of the certificate
- **The name of the user of the private key** for which the certificate publishes the public key, extended, if necessary with a unique UID

- **An encoding of the public key** components
- **Optionally, an open-ended set of extensions**, which are additional information that can be used to define the usage or validity of the certificate

#### 1.7.7.4. Examples

```
CU={VER=V3,
    SN=10,
    SIG={AL=1.3.14.3.2.2,
        ALP=1024
        },
    IS={/O=my-company/OU=asw},
    VAL={VNA=9412011200Z,
        VNB=9412011500Z
        },
    SUB={/O=my-company/OU=asw/CN=wythenshawe},
    SUBK={AL={AL=1.2.840.113549.1.1.1,
        ALP=1024},
        PK=001100010011001000110011110010011000001100010000001
    },
    IUID=0000110101001011010100,
    SUID=1011010101101101011000
}
```

### 1.7.8. Certificate-Pair

An attribute syntax for attributes that specify cross-certificate pairs.

#### 1.7.8.1. Synopsis

*Cert-Pair* = {[**F**=certificate]  
[,**R**=certificate]}

#### 1.7.8.2. Attribute Type

##### Cert-Pair

The abbreviation or OID that corresponds to a structured attribute with the Certificate-Pair (CERT\_PAIR) syntax. For example, CCP (Cross-Certificate-Pair) is such an attribute. The CCP attribute is an attribute normally associated with Certification Authorities.

#### 1.7.8.3. Components

##### **F**=certificate

An unsigned certificate that stores the public key of a “forward” member of the cross-

certificate pair. Specify *certificate* as a Certificate (Signed) attribute. See **Certificate (Signed)** for the attribute format.

#### **R=certificate**

An unsigned certificate that stores the public key of a “reverse” member of the cross-certificate pair. Specify *certificate* as a Certificate (Signed) attribute. See **Certificate (Signed)** for the attribute format.

#### **1.7.8.4. Description**

Certificate pairs are used within cross-certificate-pair attributes to cross-connect CAs (e.g., CAs associated with different organizations). Cross-connect CAs are represented with the Certificate-Pair attribute syntax. Certificate pairs are certificates for two CAs that mutually trust each other.

When combined with other CA certificates in a certificate path, the certificate pair makes it possible to create a trusted certification path that progresses from a certificate issued by a trusted CA to the certificate of the object whose signature is to be checked.

A cross-certificate-pair attribute can supply pairs of certificates for which the issuer of one is the subject of the other. It can alternatively supply just one of the pair (typically the forward component). The set of forward certificates in a cross-certificate-pair should represent a single unbroken directed graph (i.e., the subject of each certificate shall be the issuer of another forward certificate or of the originator’s certificate).

#### **1.7.8.5. Example**

```
CCP={F={CU={VER=V3,
            SN=10,
            SIG={AL=2.5.8.1.1,
                ALP=1024
            },
            IS={/O=my-company/OU=asw},
            VAL={VNA=9412011200Z,
                VNB=9412011500Z
            },
            SUB={/C=ie/O=sse},
            SUBK={AL={AL=2.5.8.1.1,
                ALP=512
            },
                PK=0011000100110010001100111100100110000011000100
            },
            IUID=00101010110100101001,
            SUID=00111010110100111001
        },
```

```

    ALG={AL=2.5.8.1.1,
        ALP=1024
    },
    EV=0011000100110010001100
},
R={CU={VER=V3,
        SN=10,
        SIG={AL=2.5.8.1.1,
            ALP=1024
        },
        IS={/C=ie/O=sse},
        VAL={VNA=9412011200Z,
            VNB=9412011500Z
        },
        SUB={/O=my-company/OU=asw},
        SUBK={AL={AL=2.5.8.1.1,
            ALP=512
        },
            PK=00110001001100100011001111001001100000110001000
        },
        IUID=10101010110100101001,
        SUID=00111010110000101001
    },
    ALG={AL=2.5.8.1.1,
        ALP=1024
    },
    EV=0011000100110010001100
}
}

```

### 1.7.9. Cert-List (Signed)

An attribute syntax for signed certificate lists, used solely by Certification Authorities to hold revocation lists (lists of identifiers for certificates that have been revoked, typically because they have become compromised, or are otherwise no longer valid).

#### 1.7.9.1. Synopsis

```

CertList = {CLU=certificate_list_unsigned,
    ALG=algorithm,
    EV=*encrypted_value}*

```

### 1.7.9.2. Attribute Type

#### CertList

The abbreviation or OID that corresponds to a structured attribute with the Cert-List (CERT\_LIST) syntax. Attributes with this syntax have two uses:

- **For CRL (Certificate-Revocation-List) information**, to revoke user's certificates
- **For ARL (Authority-Revocation-List) information**, to revoke CA's certificates

Both are types of information are attributes of Certification Authorities.

### 1.7.9.3. Components

#### CLU=certificate\_list\_unsigned

An unsigned certification list. Specify *certificate\_list\_unsigned* in the form of a Cert-List (Unsigned) component. See **Cert-List (Unsigned)** for a description of this component format.

#### ALG=algorithm

The OID that identifies the algorithm used to sign the certificate list together with any parameters used by that algorithm. Specify *algorithm* in the following format:

**{AL=\*object\_identifier[,ALP=algorithm\_parameters]}\***

where:

#### AL=object\_identifier

Is the OID or abbreviation that identifies the combination of hashing and encryption algorithms used to sign the certificate-list. This component is required by any algorithm that checks the signature.

#### ALP=algorithm\_parameters

Are parameters used by the algorithm for the signature. The format is dependent on the value of the **AL** component, but in most cases, there will be no additional parameters.

#### EV=encrypted\_value

The encrypted value is a result of the following procedure:

1. The encrypted certificate list is encoded using ASN.1 under the constraints documented in clause 9 of X.509
2. The encryption results are hashed using the hashing algorithm specified by the **AL** component
3. The resulting hash is encrypted the encryption algorithm specified by the **AL** and the private key of the signing entity
4. The result is expressed as a bit string



#### 1.7.9.4. Description

Signed certificate lists are represented with the Cert-List (Signed) attribute syntax. The certificate list portion of the Cert-List (Signed) attribute is represented by an Cert-List (Unsigned) component (described in the **Cert-List (Unsigned)**). The signature portion consists of the digital signature in the form of a bit string, together with an object identifier that specifies the procedure by which the signature was created and (where necessary) any additional parameters for the signature process.

#### 1.7.9.5. Example

```
ARL={CLU={VER=V2,
          SIG={AL=1.3.14.3.2.3},
          IS={/O=My-Company/OU=Corporate CA },
          TU=9612011200Z,
          NU=9612011300Z,
          RC={SN=25,
              RD=9603011500Z};
        {SN=75,
          RD=9608231005Z}}},
ALG={AL=1.3.14.3.2.3},
EV=0011000100110010001100}
```

#### 1.7.10. Cert-List (Unsigned)

A syntax for unsigned certificate lists. This syntax is never used as an attribute syntax. It specifies the syntax of information that, with a digital signature, comprises a certificate revocation list. The signed version of this information is defined in the **Cert-List (Signed)** attribute syntax, which describes the syntax of the values that used as actual attribute values.

##### 1.7.10.1. Synopsis

```
CLU= {[VER=version_number]
      ,SIG=signature_algorithm
      ,IS=DN_of_issuer
      ,TU=this_update
      [,NU=next_update]
      [,RC=revoked_certificates;revoked_certificates;...]
      [,EXT=extensions;extensions;...]}
```

##### 1.7.10.2. Components

###### **VER=version\_number**

The version number of the certificate list. Up to two versions are supported. Specify the version number as one of the following keywords.

- **V1** (Version 1 lists do not have a corresponding `version_number` element; for version 1 lists, this argument should not be specified)
- **V2** (Version 2 optionally includes extensions)

### **SIG=signature\_algorithm**

The OID that identifies the algorithm used to sign the certificate list together with any parameters used by that algorithm. Specify *signature\_algorithm* in the following format:

```
{AL=object_identifier[,ALP=algorithm_parameters]}
```

where:

#### **AL=object\_identifier**

Is the OID or abbreviation that identifies the combination of hashing and encryption algorithms used to sign the certificate-list. This is required by any algorithm that checks the signature. It duplicates the information provided with the signature itself. (See the **AL** component in Cert-List (Signed) attribute syntax.)

#### **ALP=algorithm\_parameters**

Are parameters used by the algorithm for the signature. The format depends on the value of the **AL** component, but in most cases, there will be no additional parameters. This information, if present, duplicates the information provided with the signature itself. (See the **ALP** component in Cert-List (Signed) attribute.)

### **IS=DN\_of\_issuer**

The distinguished name of the certification authority that issued the certificate list.

### **TU=this\_update**

The date and time of issue of this certificate list, in UTC time format. The information is used to help assess the validity of the list.

### **NU=next\_update**

The date and time of the next planned update to the certificate list, in UTC time format. If this date and time have passed, the current list is obsolete. The CA will check current list for any certificates whose validity is in question. If the current list is unavailable, the CA will consider the certificates to be suspect.

### **RC=revoked\_certificates**

The list of certificates that have been invoked and are no longer valid. Specify *revoked\_certificates* in the following format:

```
{SN=serial_number,  
  RD=revocation_date  
  [,EXT=extensions;extensions;...]}
```

where:

#### **SN=\*\*\*serial\_number**

Is an integer that specifies the serial number of the revoked certificate.

**RD=revocation\_date**

Is the date and time the certificate was revoked in UTC time format.

**EXT=extensions;extensions;...**

Specifies extensions that apply to individual certificates. It cannot be supplied for V1 lists.

Extensions in certificates and certificate lists provide an open-ended method for holding additional information. Each extension consists of three components:

- An identifier for the extension
- A critical extension flag indicating whether the extension can be ignored if not understood
- A value which represents the data corresponding to the extension

Specify **extensions** in the following format:

```
{ID=oid  
[,CRIT=boolean value,]  
EV=octet_string}
```

where:

**ID=oid**

Is the OID for the form of the extension information.

**CRIT=boolean value**

Indicates whether or not the extension can be ignored by a user of this information that does not understand the extension . If **TRUE**, the extension is critical, and the user must be able to understand the extension to use the certificate. (See **Simple Attribute Syntax** for a description of BOOLEAN syntax representation.)

**EV=octet\_string**

Is the DER encoded extension information represented within an octet string, for example,

```
EV=\x02\x01\x00
```

The format of the information is dependent on the value of the OID used to identify the extension.

**EXT=extensions**

Extensions are used here to represents extensions for the list as a whole. Extensions cannot be specified for V1 lists.

Specify **extensions** in the following format:

```
{ID=oid[,CRIT=boolean_value,] EV=octet_string}
```

See the previous *\*RC=\*revoked\_certificates* section for further details.

### 1.7.10.3. Description

Certificate revocation lists (CRLs) are represented with the Certificate-List (Unsigned) (CertListUnsigned) attribute syntax. A CRL is issued regularly by each issuing certification authority (CA), and contains a list of those certificates issued by the CA that have been declared invalid although their validity period has not expired. A separate list is provided for user certificates and CA-certificates issued by a particular CA.

Revoked certificates are invalid, usually because the user's or CA's private key has been compromised. Note that a signature signed by the owner of a revoked certificate may still be valid (just as transactions may be valid for a lost credit-card if they predate the time of loss).

Version 2 certificates contain the means of augmenting the information contained in the CRL in two ways:

- Extension information can be associated with any certificate identifier
- Extension information can be associated with the CRL as a whole.

The purpose of a CRL is to ensure that certificates can be trusted if their identifiers are *not* on the most recent CRL. For this to work, a CRL must be up to date, and users must be able to determine that the CRL that they have obtained is the latest one.

### 1.7.10.4. Example

```
CLU={VER=V2,
      SIG={AL=1.3.14.3.2.3},
      IS={/O=My-Company/OU=Corporate CA },
      TU=9612011200Z,
      NU=9612011300Z,
      RC={SN=25,
          RD=9603011500Z};
      {SN=75,
        RD=9608231005Z}
}
```

### 1.7.11. ACI-Item

An attribute syntax for attributes that specify one of the following:

- Access control policies (for prescriptive ACI or subentry-ACI operational attributes)
- Local access control requirements applicable to a single entry when used in the entry ACI operational attribute.

### 1.7.11.1. Synopsis

*ACII* = {**ID**=*id\_tag*,  
      **PR**=*integer*,  
      **AL**=*{authentication\_level}*,  
      **IF**=*{item\_first}* | **UF**=*{user\_first}*}

### 1.7.11.2. Attribute Type

#### **ACII**

The abbreviation or OID that corresponds to a structured attribute with the ACI-Item (ACIITEM) syntax. The attributes with ACIITEM attribute syntax are:

- **Prescriptive-ACI (PACI)** - An operational attribute of a subentry that contains access control information applicable to entries within a subentry's scope.
- **Entry-ACI (EACI)** - An operational attribute of an entry that contains access control information applicable to the entry in which it appears and to that entry's contents.
- **Subentry-ACI (SACI)** - An operational attribute of an administrative point that contains access control information that applies to each of the subentries associated with the administrative point.

### 1.7.11.3. Components

#### **ID**=*id\_tag*

A string (of up to 64 characters long) that identifies the ACIITEM using a string that can be meaningful to an administrator. Specify the string in DIR\_STR format. (See **Simple Attribute Syntax** for a description of DIR\_STR format.)

#### **PR**=*integer*

An integer that specifies the precedence of the ACIITEM relative to other ACIITEMs. If the ACI evaluation encounters a grant and a denial that relate to the same user and action on the same protected object, and the grant has a higher precedence than the denial, the grant wins.

If both have the same precedence, the rules of implicit preference are followed:

- The ACIITEM with the more specifically defined user wins
- If there is no more specifically defined user, the ACIITEM with the more specifically defined protected-object wins, or the denial wins.

Specify integer as a value from 0 to 255, with 0 corresponding to a default and 255 to a mandatory policy.

#### **AL**=*{authentication\_level}*

The minimum authentication level required to access the entry. Specify *authentication\_level* in the format:

**BL**=*{L=\*level}*\*

where *level* is one of the keywords:

- **NONE** - Requires user identification
- **SIMPLE** - Requires user identification and password authentication
- **STRONG** - Requires user authentication using public key cryptography. An ldap sasl bind operation with mechanism **EXTERNAL** is mapped onto the authentication level **STRONG**.

#### ***IF={item\_first}***

The protected items and their permissions. This component can help provide a variety of access capabilities to different users and user classes through a simply expressed set of protected items.

Specify *item\_first* in the format:

```
PI={protected_items},
IP={item_permission[];{item_permission ...}}
```

where:

#### ***PI={protected\_items\_}***

Is a structured component that represents a set of protected items. Each component is optional and defines a protected item. Specify *protected\_items* in the format:

```
[E=boolean]
[,AUAT=boolean]
[,AT=attribute_list]
[,AAV=attribute_list]
[,AUATV=boolean]
```

where:

#### ***E=boolean***

Determines whether the permissions apply to the entry. Further permissions are usually needed to access the information in the entry. However, there is one exception: the entry can be deleted without regard to the access rights to the attributes. The value is represented using BOOLEAN attribute syntax. (See **Simple Attribute Syntax** for a description of BOOLEAN syntax representation.)

#### ***AUAT=boolean***

Determines whether the permissions apply to all user attributes as a whole. Further permissions are usually needed to access the information in the entry. However, there is one exception: the entry can be deleted without regard to the access rights to the attributes.

This component cannot be used to define permissions for operational attributes. The value is represented using BOOLEAN attribute syntax. (See **Simple Attribute Syntax** for a description of BOOLEAN syntax representation.)

#### ***AT=attribute\_list***

Specifies that the permissions apply to the attribute types identified by abbreviation or OID in *attribute\_list*. Consult the attribute definition block section

of the abbreviation files or use the **dirxadm** or **dirxcp show abbr** operation to obtain the list of attribute type abbreviations and OIDs.

Additional permissions are usually needed to access the information in the entry with one exception: the entry can be deleted without regard to the access rights to the attributes.

This component can be used to define permissions for explicitly identified operational attributes.

#### **AAV=attribute\_list**

Specifies that the permissions apply to the attribute values associated with the attribute types identified by abbreviation or OID in *attribute\_list*. Consult the attribute definition block section of the abbreviation files or use the **dirxadm** or **dirxcp show abbr** operation to obtain the list of attribute type abbreviations and OIDs.

In general, access must be granted first to each attribute before access to the attribute values is granted. However, there is one exception: in a modify-entry operation; attribute values can be added or removed without regard to the access rights on the attribute.

#### **AUATV=boolean**

Determines whether the permissions apply to all user attribute information associated with the entry, including all values of user attributes. The value is represented using BOOLEAN attribute syntax. (See **Simple Attribute Syntax** for a description of BOOLEAN syntax representation.)

#### **IP={item\_permission\_[];{item\_permission ...}}**

*item\_permission* represents one or more permissions to be applied to *protected\_items*. Specify *item\_permission* in the format:

[**PR**=integer,]  
**UC**= {user\_class},  
**GAD**=permissions

where:

#### **PR=integer**

Specifies the precedence of the ACIITEM relative to other ACIITEMs. (See the **PR** component for more information on ACIITEM precedence.) Specify a value from **0** to **255**. This is an optional component. If it is not specified, the precedence level used is the level specified in the **ACIITEM PR** component.

#### **UC={user\_class}**

Specifies the users to which the permissions apply. Specify *user\_class* in the format:

[**AU**=boolean] [,**TH**=boolean] [,**N**= {name\_and\_opt\_uid}] [,**UG**= {user\_group}] [,**SUB**= {subtree}]

where each component is optional and defines a set of users:

#### **AU=boolean**

Determines whether the permissions apply to every directory user that has been authenticated in compliance with the authentication level specified by the **AL** component. The value is represented using BOOLEAN attribute syntax. (See **Simple Attribute Syntax** for a description of BOOLEAN syntax representation.)

#### **TH=boolean**

Determines whether the permissions apply to the authenticated user with the same distinguished name as the entry being accessed. (Authentication is performed using the method defined by the **AL** component.)

#### **N={name\_and\_opt\_uid}**

**\*Specifies that the permissions apply to the user with the specified distinguished name and (optionally) UID. The UID component can only be used with users authenticated with the strong method of authentication. (See the \*AL component for more information.)** See the description of the **Name-And-Optional-UID** structured attribute for information on how to specify the *name\_and\_opt\_uid* argument. Supply the distinguished name of the object absolute (relative to the root of the DIT).

#### **UG={user\_group}**

Specifies the distinguished name and (optionally) UID of a Group-of-Unique-Names object (GOUN), or the distinguished name of a Group-Of-Names object (GON) with nested group support (auxiliary object class IMPGRP), or the distinguished name of a Dynamic-Group object (DYNGRP). The permissions apply to all members of the object. Supply the distinguished name of the object absolute (relative to the root of the DIT).

#### **SUB={subtree}**

Specifies that the permissions apply to the set of users whose distinguished names fall within the definition of the specified subtree. See the description of the **Subtree-Specification** structured attribute for an explanation of its component syntax.

#### **GAD=permissions**

Specifies the access rights to be granted or denied. Specify *permissions* as one of more of the following keywords, separated by the plus sign (+):

|                      |                     |
|----------------------|---------------------|
| grantAdd             | denyAdd             |
| grantBrowse          | denyBrowse          |
| grantCompare         | denyCompare         |
| grantDiscloseOnError | denyDiscloseOnError |
| grantExport          | denyExport          |
| grantFilterMatch     | denyFilterMatch     |



|               |              |
|---------------|--------------|
| grantImport   | denyImport   |
| grantModify   | denyModify   |
| grantRead     | denyRead     |
| grantRemove   | denyRemove   |
| grantRename   | denyRename   |
| grantReturnDN | denyReturnDN |

Do not mix permissions that grant and deny access in a single ACITEM. Grant and denial permissions evaluate authentication levels slightly differently, so that the meaning of the user classes is potentially different. For example, with the authentication level set to simple, a granting access to all users will insist that they have authenticated using simple authentication. Denying access to all users will deny all users whatever the means of authentication.

The meanings of these permissions are described in the following table.

| Permission             | Meaning                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Add</b>             | <b>For entries:</b> controls the ability to create an entry in the DIT (subject to the access controls on all attributes and values to be placed in the new entry). In order to add an entry, permission to add its mandatory attributes and their values must at least be granted. <b>For attributes:</b> controls the ability to add an attribute (subject to the ability to add all specified attribute values). <b>For attribute values:</b> controls the ability to add an attribute value to an existing attribute                                                                               |
| <b>Browse</b>          | <b>For entries:</b> controls whether directory operations (i.e., list and search operations) can access entries that they do not specifically name                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| <b>Compare</b>         | <b>For attributes and their values:</b> controls whether attributes and values can be used in compare operations                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| <b>DiscloseOnError</b> | <b>For entries:</b> controls whether the entry's name can be returned in an error or empty result. If not granted, the Directory returns an error as if the entry did not exist. <b>For attributes:</b> controls whether the attribute's presence can be disclosed in attribute or security errors. If not granted, the Directory returns an error as if the attribute did not exist. <b>For attribute values:</b> controls whether the attribute value's presence can be disclosed in attribute or security errors. If not granted, the Directory returns an error as if the attribute did not exist. |

| Permission         | Meaning                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Export</b>      | <b>For modify DN operations:</b> controls whether an entry can be moved (with its subordinates, if any) from its present position in the DIT to another position in which it has a different superior entry. The <b>Export</b> permission must be held by the old superior entry. If the last RDN of the entry being moved is changed, the entry must also possess the <b>Rename</b> permission.                                   |
| <b>FilterMatch</b> | <b>For attributes and their values only:</b> controls whether the attributes and values can be matched with corresponding values in a search filter. If the cannot, the filter must be evaluated as if the attribute or value was either absent or could not be accessed.                                                                                                                                                          |
| <b>Import</b>      | <b>For modifyDN operations only:</b> applies when an entry is being moved (with its subordinates, if any) to another position in the DIT where it has a different superior entry: The <b>Import</b> permission must be held by the new superior entry. The entry must itself grant <b>Export</b> permissions for success.                                                                                                          |
| <b>Modify</b>      | <b>For entries only:</b> controls whether the information contained in an entry can be modified using a modify operation. Note that appropriate attribute and value permissions must also be granted in order to modify any attributes and values in the entry. (Modifying the RDN is no affected by the Modify permission.)                                                                                                       |
| <b>Read</b>        | <b>For entries:</b> controls whether Directory read or compare operations that specifically name an entry can read that entry. Further permissions are required to access the attributes and their values. <b>For attributes and values:</b> controls whether read or search operations can return attributes and values as entry information.                                                                                     |
| <b>Remove</b>      | <b>For entries:</b> controls the ability to remove an entry from the DIT regardless of controls on attributes or attribute values within the entry. <b>For attributes:</b> controls the ability to remove an attribute and all of its values. <b>For attribute values:</b> controls the ability to remove an attribute value from an existing attribute.                                                                           |
| <b>Rename</b>      | <b>For entries only:</b> controls whether an entry can be renamed with a new RDN subject to the consequential changes to the distinguished names of subordinate entries. (It may not be possible to rename an entry when there is a subordinate reference that is subordinate to the entry.) Note, however, that permissions within subordinate entries are not taken into account when evaluating permissions for this operation. |

| Permission      | Meaning                                                                                                                                                                                                                                                                            |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>ReturnDN</b> | <b>For entries only:</b> controls whether an entry's distinguished name can be returned in an operation result. If this permission is not granted, an alias name can in some cases be returned; if it is not possible to return an alias, information about the entry is withheld. |

#### **UF={user\_first}**

A component that specifies user classes and their permissions This component can help provide a variety of access capabilities to different items through a simple set of user classes.

Specify *user\_first* in the format:

**UC**={*user\_class*},  
**UP**={*user\_permission*[];{*user\_permission ...*}}

where:

#### **UC={user\_class}**

Represents a set of users. Specify *user\_class* in the same format as the **UC** component of the **IF** (*item\_first*) component's **IP** (*item\_permission*) component.

#### **UP={user\_permission[];{user\_permission ...}}**

Represents one or more permissions to be applied to *user\_class*. Specify *user\_permission* in the format:

[**PR**=*integer*,]  
**PI**={ *protected\_items*},  
**GAD**=*permissions*

where:

#### **PR=integer**

Has the same format and meanings as the **PR** component of *item\_permission* (**IP**).

#### **PI=protected\_items**

Has the same format and meaning as the **PI** component of *item\_first* (**IF**).

#### **GAD=permissions**

Has the same format and meaning as the **GAD** component of *item\_permission* (**IP**).

#### **1.7.11.4. Description**

Access control information is represented with the ACIITEM attribute syntax. Each ACIITEM grants or denies defined permissions to specific users and protected items.

The *id\_tag* component provides an identifier for each ACIITEM. You can use it to remove a specific ACIITEM value, or to protect it with a prescriptive-ACI, subentry-ACI or entry-ACI, as

appropriate.

The *item\_first* and *user\_first* components represent different ways to organize access control information. Use *item\_first* if you want to group permissions by protected items (entries). Alternately, you can use *user\_first* to group permissions by user classes.

#### 1.7.11.5. Example

```
PACI={ID=Some authenticated users can Read all entries,
      PR=50,
      AL={BL={L=SIMPLE}}},
      UF={UC={N={DN={/O=My-Company/CN=admin}};
           {DN={/O=My-Company/OU=Sales/CN=Mayer}};
           {DN={/O=My-Company/OU=Manufacturing/CN=Ketzer}}},
          SUB={BAS={/O=My-Company/OU=Development}}},
      UP={PI={E=TRUE},
          GAD=grantRead+grantBrowse+grantReturnDN};
      {PI={AUATV=TRUE},
          GAD=grantRead+grantCompare+grantFilterMatch};
      {PR=51,
          PI={AT=UP},
          GAD=denyRead+denyFilterMatch}}};
{ID=Authenticated users can Modify their passwords,
  PR=100,
  AL={BL={L=SIMPLE}}},
  UF={UC={TH=TRUE},
      UP={PI={E=TRUE},
          GAD=grantModify};
      {PI={AT=UP,
          AAV=UP},
          GAD=grantAdd+grantRemove}}};
{ID=The administrator has additional permissions,
  PR=254,
  AL={BL={L=SIMPLE}}},
  UF={UC={N={DN={/O=My-Company/CN=admin}}},
      UP={PI={E=TRUE},
          GAD=grantAdd+grantRemove+grantModify};
      {PI={AUATV=TRUE},
          GAD=grantAdd+grantRemove};
      {PI={AT=CRN;CRT;MN;MT},
          GAD=grantRead}}
```

```
}
```

## 1.7.12. Cooperating-DSA

An attribute syntax for attributes that hold information about the administered operational bindings with partner DSAs or LDIF agreements. There is only one such attribute in the default schema: CDSA (Cooperating-DSA-Table), which is an attribute of a subentry of the root DSE. The Cooperating-DSA attribute syntax is only used for display purposes; it cannot be used for input.

### 1.7.12.1. Synopsis

```
CooperatingDSA = {SUPP=supplier_access_point  
  [,CONS=consumer_access_point]  
  [,SOB=shadow_operational_binding]}
```

### 1.7.12.2. Attribute Type

#### CooperatingDSA

The abbreviation or OID that corresponds to a structured attribute with the Cooperating-DSA (COOPERATING-DSA) syntax. The DirX Directory specific CDSA (Cooperating-DSA) attribute is such an attribute. The CDSA attribute is a system-maintained attribute of a subentry of the root DSE that can only be displayed with **dirxadm**.

### 1.7.12.3. Components

#### **SUPP=supplier\_access\_point**

Specifies the distinguished name and address of the supplier DSA of a shadowing or LDIF agreement. (See **Access-Point** structured attribute syntax for details.)

#### **CONS=consumer\_access\_point**

Specifies the distinguished name and address of the consumer DSA of a shadowing agreement. This component is not present for LDIF agreements. (See **Access-Point** structured attribute syntax for details.)

#### **SOB=shadow\_operational\_binding**

A structured component that defines the shadow binding. *shadow\_operational\_binding* is displayed in the format:

```
{SAI=shadow_agreement_info  
  [,AGR=agreement_id]  
  [,OBMV=operational_binding_validity]  
  ,OBS=operational_binding_state  
  [,RPOL=relating_sob_policies]}
```

#### **SAI=shadow\_agreement\_info**

A structured component that defines the components of a shadowing agreement. See **Shadowing-Agreement-Info** for a complete description of shadowing agreement

information format.

#### **AGR=agreement\_ID**

A structured component that identifies the operations binding. *agreement\_id* is in the following form:

{**ID**=*integer*,**VERS**=*integer*}

where

**ID**=*integer* specifies the agreement ID and **VERS**=*integer* the binding version.

#### **OBMV=operational\_binding\_validity**

A structured component that defines the range of dates during which the binding is valid. *operational\_binding\_validity* is in the following form:

{[**VF**=*valid\_from*][,**VU**=*valid\_until*]}

where

*valid\_from* is a generalized time string or the keyword **NOW**, which defines the beginning of the range of dates. The default value **NOW** defines the start of the date range as the current date.

*valid\_until* is a generalized time string or the keyword **TERMINATION**, which is the default value and defines the end of the range as never.

#### **OBS=operational\_binding\_state**

Specifies the status of an operational binding. *operational\_binding\_state* is one of the following keywords:

- **COOPERATIVE** - The shadowing or LDIF agreement is active.
- **NONCOOPERATIVE** - The shadowing or LDIF agreement is inactive.

#### **RPOL=relating\_sob\_policies**

A component that identifies the SOB-Policies performed for this shadowing operational binding. See **SOB-Policies** for a detailed description of the *relating\_sob\_policies* format.

### **1.7.12.4. Description**

The Cooperating-DSA attribute is a DSA-generated attribute of a subentry of the root DSE that displays the status of operational bindings between the DSAs participating in shadowing agreements or LDIF agreements. DirX Directory supports only shadowing operational bindings (SOB) that specifies shadowing or LDIF agreements.

This Cooperating-DSA attribute is multi-valued; it contains a value for every DSA involved in a SOB and for every LDIF agreement. The attribute should not be modified by users, and, being a root-DSE subentry attribute, can be accessed by **dirxadm** only. The **dirxadm lob** and **sob** functions can be used to create, modify, or delete the information displayed by this attribute syntax.

### 1.7.12.5. Examples

1. In the following example the CDSA (Cooperating-DSA) attribute displays the attribute values of two LDIF agreements.

```
CDSA={SUPP={AE={/O=My-Company/CN=DSA1},

PSAP={TS=DSA1,NA=TCP/IP_IDM!internet=123.45.67.89+port=21100}},

LDIF agreement 1:
    SOB={SAI={SS={AREA={CP={/O=My-Company},
        RA={BAS={/}}},ATT={DEF=TRUE}},
        UM={SI={S={PS={WS=1000,UI=1}}}},
        AGR={ID=14057,VERS=0},
        OBMV={VF=030401064822Z},
        OBS=NONCOOPERATIVE,
        RPOL={SUPP={MAXLOC=1024},
            LDSUPP={BINO=NONE,MAXROW=200,
                CS=UTF8,CHANGE0=FALSE}}}};

LDIF agreement 2:
    {SAI={SS={AREA={CP={/O=My-Company2}},},
        RA={BAS={/}}},ATT={DEF=TRUE}},
        UM={SI={S={PS={WS=1000,UI=1}}}},
        AGR={ID=16058,VERS=0},
        OBMV={VF=030401064822Z},
        OBS=NONCOOPERATIVE,
        RPOL={LDSUPP={CS=LATIN1,CHANGE0=FALSE}}}}
```

2. In the following example the CDSA (Cooperating-DSA) attribute displays the attribute values of two shadowing agreements.

```
{CDSA={SUPP={AE={/O=My-Company/CN=DSA1}

PSAP={TS=DSA1,NA=TCP/IP_IDM!internet=123.45.67.89+port=21100}},

    CONS={AE={/O=My-Company/CN=DSA2},

PSAP={TS=DSA2,NA=TCP/IP_IDM!internet=123.45.67.88+port=21100}},

agreement 1:
    SOB={SAI={SS={AREA={CP={/O=My-Company},
```

```

RA={BAS={/}}},ATT={DEF=TRUE}},
UM={SI={S={PS={WS=1000,UI=1}}}}},
AGR={ID=0,VERS=0},
OBMV={VF=030401065354Z},
OBS=NONCOOPERATIVE,
RPOL={INI={ATU=TRUE},
SUPP={MAXLOC=100},
CONS={MAXLOC=100,REPLS=TRUE}}};

```

automatically generated agreement to propagate agreement information to consumer DSA:

```

{SAI={SS={AREA={CP={/CN=Cooperating-DSAS-Subentry},
RA={DEF=TRUE}},ATT={DEF=TRUE}},
UM={SI={OC=TRUE}}},
AGR={ID=1,VERS=0},
OBMV={VF=030401065354Z},
OBS=COOPERATIVE};

```

agreement 2:

```

{SAI={SS={AREA={CP={/O=My-Company2},
RA={BAS={/}}},ATT={DEF=TRUE}},
UM={SI={S={PS={WS=1000,UI=1}}}}},
AGR={ID=16058,VERS=0},
OBMV={VF=030401065354Z},
OBS=NONCOOPERATIVE,
RPOL={SUPP={US=TOTAL},CONS={US=TOTAL}}}}

```

### 1.7.13. Cooperating-DSA-Status

An attribute syntax for attributes that hold information about the administered operational bindings with partner DSAs. The Cooperating-DSA-Status attribute is only used for display purposes in **dirxadm sob show** or **lob show** operations. It cannot be used for input.

#### 1.7.13.1. Synopsis

```

CoopDsaStatus = {DSA=cooperating_DSA
[,RMSN=recent_MSN]
[,SYNC=sync_status
[,ASR=async_reason]]}

```

#### 1.7.13.2. Components

**DSA**=cooperating\_DSA

The distinguished name and address of the supplier or consumer DSA in a shadowing



agreement. See the **Access-Point** structured attribute syntax for details.

#### **RMSN=recent\_MSN**

An integer value that identifies the modification sequence number (MSN) of the last confirmed operation. This component is only shown for synchronous shadowing agreements (the *DISP\_kind* value for the **-consumerkind** option is set to **SYNCHRON**). See the **dirxadm sob create** operation in the *DirX Directory Administration Reference* for details.).

#### **SYNC=sync\_status**

A Boolean value that shows the synchronous data status of the consumer DSA. A value of **TRUE** indicates that the consumer DSA's data status is synchronized with the data status of the supplier DSA. This component is only shown for synchronous shadowing agreements (the *DISP\_kind* value for the **-consumerkind** option is set to **SYNCHRON**). See the **dirxadm sob create** operation in the *DirX Directory Administration Reference* for details.).

#### **ASR=async\_reason**

A text value that indicates the reason for a value of **FALSE** in the **SYNC** component. This component is only shown for synchronous shadowing agreements (the *DISP\_kind* value for the **-consumerkind** option is set to **SYNCHRON**). See the **dirxadm sob create** operation in the *DirX Directory Administration Reference* for details.

### **1.7.13.3. Description**

The Cooperating-DSA-Status attribute is maintained by the DSA. It displays the current status of the supplier DSA when the command **dirxadm sob show** is invoked. It displays the current status of all consumer DSAs when the command **dirxadm sob show -supplier dsa\_name** is invoked.

### **1.7.13.4. Examples**

1. The following example shows sample output of the command:

```
dirxadm sob show
```

The sample output shows the status of a supplier DSA in a DirX Directory-compliant/synchronous shadowing configuration:

```
{SUPP={DSA={AE={/CN=DirX-DSA-host1},  
  PSAP={TS=DSA1,NA=TCP/IP_IDM!internet=192.168.0.2+port=21200},  
  KIND=SYNCHRON},RMSN=4,SYNC=TRUE}
```

2. The following example shows sample output of the command:

```
dirxadm sob show
```

The sample output shows the status of a supplier DSA DirX Directory-compliant/asynchronous shadowing configuration:

```
{SUPP={DSA={AE={/CN=DirX-DSA-host2},  
  PSAP={TS=DSA1,NA=TCP/IP_IDM!internet=192.168.0.3+port=21200},  
  KIND=CENTRALADMIN}}}
```

3. The following example shows sample output of the command:

```
dirxadm sob show -supplier {/CN=DirX-DSA-host2}
```

The sample output shows the status of a consumer DSA in a DirX Directory-compliant/asynchronous shadowing configuration:

```
{CONS={DSA={AE={/CN=DirX-DSA-host3},  
  PSAP={TS=DSA1,NA=TCP/IP_IDM!internet=192.168.0.4+port=21200},  
  KIND=CENTRALADMIN}}}
```

4. The following example shows sample output of the command:

```
dirxadm sob show -supplier {/CN=DirX-DSA-host1}
```

The sample output shows the status of a consumer DSA in a DirX Directory-compliant/synchronous shadowing configuration:

```
{CONS={DSA={AE={/CN=DirX-DSA-host4},  
  PSAP={TS=DSA1,NA=TCP/IP_IDM!internet=192.168.0.5+port=21200},  
  KIND=SYNCHRON},RMSN=1,SYNC=TRUE}
```

For more information on the **sob show** and **lob show** operations, see the section on **dirxadm** in the *DirX Directory Administration Reference*. For more information about asynchronous, synchronous, DirX Directory-compliant and X.500-compliant shadowing configurations and agreements, see the section on **dirxadm sob create** in the *DirX Directory Administration Reference* and the chapter “Creating a Shadow DSA” in the *Administration Guide*.

## 1.7.14. DSA-Policies

An attribute syntax for attributes that specify DSA policies. This attribute syntax is used solely by the DSAP (DSA-Policy) attribute, which is an attribute of the root DSE. The DSA-Policy attribute, which is accessible only by using **dirxadm**, defines the bilateral agreements between the present DSA (local DSA) and other remote DSAs known to it.

### 1.7.14.1. Synopsis

```
DsaPolicies = {DSA={DN_string},  
              [,OPT=DSA_policy_options]  
              [,KL=integer]  
              [,TO=integer]  
              [,AP={DSA_authentication_policy}]}
```

### 1.7.14.2. Attribute Type

#### DsaPolicies

The abbreviation or OID that corresponds to a structured attribute with the Dsa-Policies (DSA\_POLICIES) syntax. The DSAP attribute is a DSA-specific operational attribute of the root DSE and is accessible only by using **dirxadm**.

### 1.7.14.3. Components

#### **DSA=DN\_string**

The distinguished name of the DSA to which the policy options apply. This distinguished name is the name of the own DSA or a partner DSA. Specify a value of / (forward slash) to mark the policy as a default for all DSAs for which no specific policy is defined. Note that it is not possible to specify policies for a specific set or group of DSAs.

#### **OPT=DSA\_policy\_options**

Policy options that apply to the interoperation of this DSA with other DSAs. Specify *DSA\_policy\_options* as one or more keywords or the value **0** (zero). Separate multiple keywords with the + (plus sign). The keywords are:

- **CHAINING** - Indicates that the present DSA is willing to send chained operations to this DSA. This keyword does not affect incoming chained operations. (This is the default value.)
- **LOCAL-DOMAIN** - Indicates that the named DSA lies within the same local-domain as the present DSA, for the purpose of the local-scope bit in the service-control options. This option is relevant only if the LSP (Local-Scope-Policy) attribute is set to **LOCAL-DOMAIN** or **DMD**.
- **DMD** - Indicates that the named DSA lies within the same DMD as the present DSA, for the purpose of the scope-of-referral service control. This option is only relevant if the LSP (Local-Scope-Policy) attribute is set to **DMD**.
- **COUNTRY** - Indicates that the named DSA lies within the same country as the present DSA, for the purpose of the scope-of-referral service control.

Supplying the value **0** for *DSA\_policy\_options* means that the local DSA does not chain to

this DSA and that this DSA is neither within the same local domain, DMD nor country as the local DSA.

### **KL=integer**

An *integer* that defines the keep-line policy, a policy that defines the number of seconds a DSA will wait before disconnecting from an idle DSA-to-DSA connection and applies to DSP and DISP protocol. A DSP or DISP association is idle when there are no outstanding operations on it. If you do not specify a keep-line policy, the association will not be terminated after chained or shadowing operations. The default value is **300** seconds (5 minutes).

### **TO=integer**

An *integer* that defines the time-out policy, a policy that defines the number of seconds the DSA will wait before it aborts a DSP connection with a non-responding partner when performing a bind request. This policy is used to detect communication failures during DSP bind operations. The internal shadowing protocol recovery procedure handles communication failures automatically when performing DISP bind operations.

### **AP=DSA\_authentication\_policy**

Authentication policy for the DSA. Specify *DSA\_authentication\_policy* in the following format:

```
[AMB={auth_methods}]
[,AMT={auth_methods}]
[,TD=trust_degree]
[,PWD=any_octet_string]
[,CE=credentials_expiration]
[,CV=credentials_validity]
```

where:

#### **AMB={auth\_methods}**

Specifies the authentication method for binds allowed from the remote DSA. Only these authentication methods will be accepted by the local DSA for incoming DSP and DISP binds from the remote DSA. The local DSA will use the “strongest” authentication method specified for outgoing DSP and DISP binds to the remote DSA. Currently only the following authentication methods are allowed. Specify *auth\_methods* in the following format:

#### **AM=authentication\_methods**

A sequence of the following keywords separated by semicolons:

- **NONE** - No authentication is performed. (Corresponds to an authentication level of **NONE** for access control.)
- **SIMPLE** - Simple authentication is performed, but no password is supplied. (Corresponds to an authentication level of **NONE** for access control.)
- **SIMPLE-WITH-PWD** - Simple authentication with a password is performed. A DN and password are passed in the clear. (Corresponds to an authentication level of **SIMPLE** for access control.)

- **SIMPLE-PROTECTED** - Simple authentication is performed with a secure token. A DN, password, a random number, and a timestamp are passed after being protected with hashing encoding function. (Corresponds to an authentication level of **SIMPLE** for access control.)

The default is **NONE;SIMPLE;SIMPLE-WITH-PWD;SIMPLE-PROTECTED**. A DirX Directory DSA will use as default **SIMPLE-PROTECTED** for outgoing DSP and DISP binds.

#### **AMT={auth\_methods\_}**

Specifies the authentication methods that must be used by the remote DSA in order to be trusted for a chained operation. Specify *auth\_methods* in the same format as the **AMB** component.

If you specify the component **AMT={auth\_methods}**, you should specify the component **TD=trust\_degree**.

The default value is to trust all DSAs independent of the authentication method used in the DSP bind.

#### **TD=trust\_degree**

Specifies the degree of trust allowed to chained operations supplied by the DSA. Specify the *trust\_degree* parameter as one or both of the following keywords:

- **READ** - The DSA can grant access to incoming chained Read operations, subject to normal access control.
- **MODIFY** - The DSA can grant access to incoming chained Modify operations, subject to normal access control.

If you specify both keywords, separate them with a + (plus sign).

If you specify the component **TD=trust\_degree**, you should specify the component **AMT={auth\_methods}**.

The default value is **READ+MODIFY**.

#### **PWD=any\_octet\_string**

The password of the DSA whose name is specified in the **DSA** component of the same attribute value. (This is the own DSA, a possible partner DSA, or a value of / (forward slash - that is all DSAs that are not explicitly listed in one of the DSAP attribute values use this password.)

The DSA uses this password in bind requests and bind results in DSA to DSA communications, for example when performing shadowing (DISP) or chaining (DSP).

The default value is **DirXDSA**. It is strongly recommended to use an individual password committed by a bilateral agreement.

#### **CE=credentials\_expiration**

\_\_Specifies the amount of time in seconds after a bind is initiated that the outgoing

credentials in DSP and DISP binds will expire. This component is used only with simple-protected authentication. The default is 30.

### CV=credentials\_validity

\_\_Specifies the time in seconds after a bind is initiated at which the incoming credentials in DSP and DISP binds will expire. This component is used only with simple-protected authentication. The default is **30**.

#### 1.7.14.4. Description

DSA-Policies is a syntax for the DSAP attribute. This attribute controls a DSA's method of interaction with other DSAs. Specifically, it supplies information about the authentication procedures for DSP and DISP operations. The DSAP attribute is a multivalued attribute. Each value is assigned to a specific DSA or / and contains one or more policy options. The DSAP attribute can be managed only by using **dirxadm**.

#### 1.7.14.5. Examples

1. All DSAs deployed in a common directory service use the same password "dsa\_pass" in their bind operations. All binds are performed using the unprotected credentials that is they send name and password in clear text format. This policy is performed if all DSAs have the following single DSA Policy attribute value stored in their root DSE:

```
DSAP={ DSA={ / },  
      OPT=COUNTRY+DMD,  
      KL=100,  
      TO=60,  
      AP={ AMB={ AM=SIMPLE-WITH-PWD },  
          PWD=dsa_pass }
```

2. A directory service is distributed over three DSAs named **/o=my-company/cn=dsa-sales**, **/o=my-company/cn=dsa-development** and **/o=my-company/cn=dsa-hq**. According to the policy of the company all DSAs perform bind operations with simple protected credentials based on individual DSA-Passwords. All DSAs trust each other to the "READ+MODIFY" degree (which is the default and can therefore be omitted). In order to perform this policy all DSAs must have the following three attribute values for the DSA Policy attribute stored in their root DSE:

```
DSAP={ DSA={ /O=my-company/CN=dsa-hq }, KL=100, TO=60,  
      AP={ AMB={ AM=SIMPLE-PROTECTED },  
          TD=READ+MODIFY,  
          PWD=pwdHQ01 } } ;  
DSAP={ DSA={ /O=my-company/CN=dsa-development }, KL=100, TO=60,  
      AP={ AMB={ AM=SIMPLE-PROTECTED },  
          TD=READ+MODIFY,
```

```
PWD=pwdDEV01}};
DSAP={DSA={/O=my-company/CN=dsa-sales},KL=100,TO=60,
AP={AMB={AM=SIMPLE-PROTECTED},
TD=READ+MODIFY,
PWD=pwdSAL01}};
```

The only component of the own DSAs Policy value necessary to make the DSA policy work is the **PWD** component. For example the DSA named **/o=my-company/cn=dsa-hq** (the first of the three values given above) is:



```
DSAP={DSA={/O=my-company/CN=dsa-hq},AP={PWD=pwdHQ01}};
```

However, if the full policy is specified for all DSAs the behaviour of the DSAs is automatically correct after a backup-media has been used to switch the roles of the DSAs.

3. A directory service is distributed over three DSAs named **/o=my-company/cn=dsa-intern-1**, **/o=my-company/cn=dsa-intern-2** and **/o=my-company/cn=dsa-extern**. According to the policy of the company all DSAs use individual DSA-passwords. The (internal) DSAs named "**dsa-intern-1**" and "**dsa-inter-2**" bind using the simple-protected credential choice and trust each other to perform modification operations. The bind operations performed to the third DSA named "**dsa-extern**" use the simple unprotected credential choice. The degree they trust this third DSA is only "READ". In order to perform this policy the two DSAs named **dsa-intern-1** and **dsa-intern-2** must have the following three attribute values for the DSA Policy attribute stored in their root DSE:

```
DSAP={DSA={/O=my-company/CN=dsa-intern1},KL=100,TO=60,
AP={AMB={AM=SIMPLE-PROTECTED},
TD=READ+MODIFY,
PWD=pwdInt1}};
DSAP={DSA={/O=my-company/CN=dsa-intern2},KL=100,TO=60,
AP={AMB={AM=SIMPLE-PROTECTED},
TD=READ+MODIFY,
PWD=pwdInt2}};
DSAP={DSA={/O=my-company/CN=dsa-extern},KL=100,TO=60,
AP={AMB={AM=SIMPLE-WITH-PWD},
TD=READ,
PWD=pwdExt}};
```

The DSA named "**dsa-extern**" must have the following three attribute values for the DSA Policy attribute stored in its root DSE:

```
DSAP={DSA={/O=my-company/CN=dsa-intern1},KL=100,TO=60,
      AP={AMB={AM=SIMPLE-WITH-PWD},
        PWD=pwdInt1}};
DSAP={DSA={/O=my-company/CN=dsa-intern2},KL=100,TO=60,
      AP={AMB={AM=SIMPLE-WITH-PWD},
        PWD=pwdInt2}};
DSAP={DSA={/O=my-company/CN=dsa-extern},KL=100,TO=60,
      AP={AMB={AM=SIMPLE-WITH-PWD},
        PWD=pwdExt}};
```

## 1.7.15. Paging-Policies

An attribute syntax for attributes that specify paging policies. This attribute syntax is used solely by the PAGP (Paging-Policy) attribute, which is an attribute of the root DSE. The Paging-Policy attribute, which is accessible only by using **dirxadm**, defines the configuration parameters for paged results of search requests.

### 1.7.15.1. Synopsis

```
PagingPolicies = {[PTO=timeout]
                  [,MAXNUM=max_number]
                  [,MAXMEM=max_memory]}
```

### 1.7.15.2. Attribute Type

#### PagingPolicies

The abbreviation or OID that corresponds to a structured attribute with the Paging-Policies syntax. The PAGP attribute is a DSA-specific operational attribute of the root DSE and is accessible only by using **dirxadm**.

### 1.7.15.3. Components

#### **PTO=timeout**

The timeout value in seconds. Search results kept for paged results are released when the specified time period has expired and no subsequent request has been received for the search result. The default value is **300** seconds.

If an LDAP application tries to use an already expired resume/continuation information the paged search operation results in an **unwilling to perform the operation** error and the LDAP message **[DSA]:invalid query ref**.

#### **MAXNUM=max\_number**

The maximum number of results kept for paged results. The default value is **64**.



If *max\_number* is too low, a paged search operation may result in an **unwilling to perform the operation** error and the LDAP message **[DSA]: Paging count *n* exceeded**, where *n* is the currently configured *max\_number* value.

#### **MAXMEM=max\_memory**

The maximum amount of memory in kilobytes that can be used to save one search result for paged results. The default value is **128 MB**.

If *max\_memory* is too low, a paged search operation may result in an **unwilling to perform the operation** error and the LDAP message **[DSA]:Insufficient Paging Memory detected**.

#### **1.7.15.4. Description**

For each search operation that contains a paged result request, the DSA initially keeps resume information in memory for efficient processing of subsequent search operations requesting follow-up pages.

Paging-Policies is a syntax for the PAGP attribute. This attribute controls a DSA's method for handling resume information for search requests. The PAGP attribute is a single-valued attribute. It is valid for all search operations. The PAGP attribute can be managed only by using **dirxadm**.

#### **1.7.15.5. Examples**

In the following example, resume information kept for paged results is released when no subsequent request has been received for a period of 180 seconds.

Search operations are rejected with the error "Unwilling To Perform" when

- a search operation containing a paged result request requires to allocate resume information while resume information is kept already for 15 other search operations
- the resume information exceeds the maximum amount of 2 MB

```
PAGP={PTO=180,  
      MAXNUM=15,  
      MAXMEM=2048}
```

### **1.7.16. Shadowing-Agreement-Info**

A syntax for specifying shadowing or LDIF agreements. This attribute syntax is used in the **dirxadm lob** and **sob create** command for the **-agreement** option and for displaying components of the Cooperating DSA table.

#### **1.7.16.1. Synopsis**

```
ShadowingAgreementInfo={SS={shadow_subject_unit_of_replication}  
  [,CHANGEO=changes_only]  
  [,UM={update_mode}]}
```

[,MAS={*master\_access\_point*}]  
[,SEC={*secondary\_shadows*}]

### 1.7.16.2. Components

#### SS=shadow\_subject\_unit\_of\_replication

A structured component that defines the information (entries, attributes, and knowledge) to be replicated. This information is known as the “shadow subject” or “unit of replication.” Specify *shadow\_subject\_unit\_of\_replication* in the following format:

AREA={*area\_specification*}  
,ATT=*attribute\_selection*  
[,KNO={*knowledge*}]

where:

#### AREA={*area\_specification*}

Identifies the naming context area to be replicated. Specify *area\_specification* in the form:

CP={*context\_prefix\_distinguished\_name*},  
RA={*replication\_area\_subtree\_specification*}

where: where:

#### context\_prefix\_distinguished\_name

Specifies the context prefix of the naming context. This is the distinguished name of the node at the top of the naming context. The value / is root context.

#### replication\_area\_subtree\_specification

Specifies whether the entire naming context should be replicated or only a subtree within the naming context. Specify *replication\_area\_subtree\_specification* in the **Subtree-Specification** attribute syntax format (see **Subtree-Specification** for details).

#### ATT=attribute\_selection

Filters the attributes to be replicated. In addition to the attributes specified here, access control operational attributes, Creation Time (CRT), Modification Time (MT), and object class attribute are always replicated. Further operational attributes, for example, Creators Name (CRN), must be explicitly included. Specify *attribute\_selection* in one of the following formats:

##### • {DEF=TRUE}

This form causes all user attributes and the operational attributes described above to be replicated.

##### • {[ATSCL=*object\_class\_identifier*,]CA={ALL=TRUE} | {INCL=*attribute\_list*} | {EXCL=*attribute\_list*}} [:{[ATSCL=...,]CA={...}};...]

where:

**ATSCL=object\_class\_identifier\_**

Specifies the abbreviation or OID that corresponds to an object class and specifies that the shadow supplier is to perform attribute selection of this class for shadowing. If the class is omitted, the selection applies to all classes.

**CA=...**

Specifies that either all attributes (**{ALL=TRUE}**) of the specified object class or only a part of the attributes are replicated. Include or exclude attributes by specifying the **INCL** or **EXCL** component, where *attribute\_list* specifies one or more abbreviations or OIDs (separated by a semicolon (;)) that correspond to attribute types. **INCL** explicitly includes and **EXCL** explicitly excludes attribute types. When using the **EXCL** component, any attribute contained in an entry, which is not explicitly excluded, is implicitly included in the unit of replication. In case of conflicting specifications, **INCL** has priority over explicitly excluded attributes and **EXCL** has priority over implicitly included attributes. If all attributes (**{ALL=TRUE}**) is not specified, an **INCL** or an **EXCL** component must be specified.

**KNO={knowledge}**

Identifies the knowledge references to be replicated. Specify *knowledge* in the following form:

**KT=\*knowledge\_type** [, EK=\*extended\_knowledge]

where:

**knowledge\_type=MASTER / SHADOW / BOTH**

Is a keyword that specifies the knowledge types to shadow when subordinate references are encountered at the base of the naming context. Specify the knowledge type with one of the following keywords:

- **MASTER** - Shadow only the knowledge reference that references the master DSA for the subordinate naming context
- **SHADOW** - Shadow only the knowledge references that reference commonly used shadows of the subordinate naming context. (In practice, this means shadows of the complete naming context, since less complete shadows are not always reliably useful.)
- **BOTH** - Shadow both master and commonly used shadow references.

The default value (nothing specified) is that no knowledge references are replicated.

**extended\_knowledge=TRUE / FALSE**

Is a Boolean value that specifies whether or not to replicate extended subordinate references, which are subordinate references that are not immediately subordinate to entries that are shadowed.

## **CHANGEO=changes\_only**

A boolean value that specifies whether only updates are saved (**TRUE**) or a total content file is saved (**FALSE**). Specify this component only for shadowing operational bindings. For LDIF agreements this behavior is specified in the **LDSUPP** component of the **SOB-Policies**. The default is **FALSE**. If the DSA supports only X.500 compliant shadowing the default value **FALSE** must be used. For synchronous shadowing agreements, this is the only component that can be specified. (See the **sob create** operation in the *DirX Directory Administration Reference* for details.)

## **UM={update\_mode}**

A structured component that defines when the replicated entries will be updated and whether the consumer or the supplier should initiate the updates. This component is required for LDIF-agreements and consumer-initiated updates. Specify *update\_mode* in either of the following formats:

- **For supplier-initiated updates and LDIF-agreements**, specify *update\_mode* in the format:

**SI={ S={scheduling\_parameters} | OC=on\_change}**

- **For consumer-initiated updates**, specify *update\_mode* in the format:

**CI={scheduling\_parameters}**

For supplier-initiated, consumer-initiated updates and LDIF-agreements, the components

**SI={ S={\*scheduling\_parameters}\* | OC=\*on\_change}\* and CI={\*scheduling\_parameters}\* establish the update schedule. Use the **SI** subcomponent to set up a schedule for supplier-initiated updates and LDIF-agreements or the **CI** parameter to set up a schedule for consumer-initiated updates. When you specify scheduling parameters, all time units are in seconds.**

where:

## **OC=on\_change**

*on\_change\_* is a boolean value. The value **TRUE** means that replicated entries are updated immediately when the master entries are modified or created. The value **FALSE** means that updates will not be performed automatically. For supplier-initiated updates, the default mode is **SI={OC=TRUE}**.

## **scheduling\_parameters**

Specify *scheduling\_parameters* in the following format:

### **PS={periodic\_strategy}**

*periodic\_strategy* is designed to allow shadowing to take place at regular intervals. Specify *periodic\_strategy* in the following format:

**WS=window\_size,**  
**UI=update\_interval**  
**[,BT=begin\_time]**

where:

**WS=window\_size**

Is an integer that defines the time period in seconds during which the shadow consumer will accept updates from the shadow supplier. A shadow window allows updates to occur at anytime within the window time. It is useful if for some reason the updates are unable to start at the exactly scheduled moment or for a retry if some update failure occurs.

**UI=update\_interval**

Is an integer that specifies in seconds the interval in seconds between the opening of the update windows. For example, if **UI** is set to 1800 (30 minutes) and **WS** to 300 (5 minutes), the update cycle is 35 minutes. Updates of up to five minutes in duration can occur every 30 minutes.

**BT=begin\_time**

Is the start-time of the first update window in Generalized Time format. When this start time is after the validity of the shadowing agreement the update cycle begins at this time. (See **-validity** option of **lob** and **sob create** operation in the *DirX Directory Administration Reference* for details.) Once started the update cycle begins until the shadow agreement is terminated. (See **Simple Attribute Syntax** for a description of this format.) The default for this subcomponent is the time the shadowing agreement was activated.

**MAS=master\_access\_point**

An optional structured component that defines the access point of the master DSA for the naming context. (Note that if secondary shadowing is used the master DSA cannot be the shadow supplier, because a shadow consumer acts also as the shadow supplier for the same information.). The value is optional when the shadowing is primary shadowing (i.e. the supplier is also the master of the information).

Specify *master\_access\_point* as an Access-Point attribute. (See **Access-Point** for details.)

**SEC=secondary\_shadows**

A Boolean value that specifies whether the consumer in this agreement can act as a secondary shadow to the shadow supplier. Secondary shadow consumers can shadow the shadowed information on to other DSAs. The value is represented using BOOLEAN attribute syntax. (See **Simple Attribute Syntax** for a description of BOOLEAN syntax representation.). The default is **FALSE** (secondary shadows are not allowed).

### 1.7.16.3. Description

Information pertaining to a shadow agreement between supplier and consumer DSA or an LDIF agreement is represented with the Shadowing-Agreement-Info component syntax. This syntax is only used for specifying the value of the **-agreement** parameter in the **dirxadm lob** and **sob create** command and for displaying components of the Cooperating DSA table. It describes a shadowing agreement, which basically defines:

- What is being replicated, known as the “unit of replication”. (The **SS** subcomponent attribute.)

- When updates of the replicated entries should take place and who initiates the update process. (The **UM** subcomponent attribute.)
- The access point of the master DSA, if necessary. (The **MAS** subcomponent attribute.)
- Whether secondary shadows are allowed. (The **SEC** subcomponent attribute.)

#### 1.7.16.4. Example

```
{SS={AREA={CP={/O=My-Company},
            RA={DEF=TRUE}
          },
  ATT={DEF=TRUE}
},
UM={SI={S={PS={WS=3600,
               UI=86400,
               BT=19970101230000Z
             }
        }
    }
}

{SS={AREA={CP={/O=My-Company},
            RA={BAS={/O=sales},
            EXCL={CHB={/OU=ABC/OU=DEF}};
            MAX=10,
            SF={ITEM=PER}
          },
  KNO={EK=TRUE, KT=MASTER},
  ATT={DEF=TRUE}
}

{SS={AREA={CP={/O=My-Company},
            RA={BAS={/O=sales},
            EXCL={CHB={/OU=ABC/OU=DEF}}; {CHA={/OU=XYZ}};
            MAX=10,
          },
  KNO={EK=TRUE, KT=MASTER},
  ATT={DEF=TRUE}
```

```

}
}

{SS={AREA={CP={/O=My-Company},
            RA={BAS={/OU=Sales}}
        },
  KNO={EK=TRUE,KT=MASTER},
  ATT={CA={INCL=CRN}};      includes operational attribute
                                Creator's Name
    {CA={EXCL=0;L;DSC}};    excludes attributes Organization,
                                Locality, and Description
    {ATSCL=ORP,CA={ALL=TRUE}} includes all attributes of object
                                class Organizational Person
                                (inclusive O, L, and DSC of ORP)
                                and implicitly includes all user
                                attributes that are not
                                explicitly excluded here
  }
}

{SS={AREA={CP={/O=Air.com},RA={DEF=TRUE}},
    ATT={DEF=TRUE}},
  UM={SI={OC=TRUE}}}

```

### 1.7.17. SOB-Policies

A syntax for specifying policies for shadowing operational bindings like shadowing or LDIF agreements. This attribute syntax is used in the **dirxadm lob** and **sob create** command for the **-pol** option.

#### 1.7.17.1. Synopsis

```

SOBPolicies={DEF=TRUE} |
  {[MAS=sob_master_policies]
  [,INI=sob_initiator_policies]
  [,SUPP=sob_supplier_policies]
  [,CONS=sob_consumer_policies]
  [,LDSUPP=sob_ldif_supplier_policies]}

```

#### 1.7.17.2. Components

##### **DEF=TRUE**

Specifies that default policies should be used. See the other components for a description of the default policies.

### **MAS=sob\_master\_policies**

Specifies the policies used for master shadowing agreements. Specify *sob\_master\_policies* in the following format:

```
{[MAXJSZ=max_journal_size]  
[,MAXJFIL=max_journal_files]}
```

where:

#### **MAXJSZ=max\_journal\_size**

Specifies the maximum amount of disk space in kilobytes used by shadowing operational binding journal files. The value of **0** (zero) indicates that an unlimited amount of disk space can be used. The default value is unlimited.

#### **MAXJFIL=max\_journal\_files**

Specifies the maximum number of journal files for the shadowing operational binding. The value of **0** (zero) indicates that an unlimited number of files can be used. The default value is unlimited.

### **INI=sob\_initiator\_policies**

Specifies the policies used for initiator shadowing agreements. These components should only be administered if you want to manipulate the default values of the error recovery. If it is useful, an error recovery is performed automatically; that is, for example, when the server is temporarily shut down. Specify *sob\_initiator\_policies* in the following format:

```
{[RMR=recovery_max_retries]  
[,RTO=recovery_timeout]  
[,RI=recovery_interval]  
[,RD=recovery_divisor]  
[,ATU=automatic_total_update]}
```

where:

#### **RMR=recovery\_max\_retries**

Specifies the maximum number of retries of recovery for shadowing agreements with the DSA specified in the *\*DSA=\*DN\_string* component of a DSAP attribute where the local DSA is the initiator. The default is 20 .

If the **RTO=recovery\_timeout** and the *\*RMR=recovery\_max\_retries* components are equal to 0, recovery is configured as unlimited (that is, there is no limitation by timeout or by number of retries, and recovery never stops).

If either the *\*RTO=recovery\_timeout* or the **RMR=recovery\_max\_retries** component is equal to 0 (and the other one is equal to a non-zero value), no recovery will be done.

If the **RTO=recovery\_timeout** and the *\*RMR=recovery\_max\_retries* components are equal to a non-zero value, recovery will be attempted until both values have been exceeded.



### **RTO=recovery\_timeout**

Specifies the timeout in seconds of a recovery attempt for shadowing agreements with the DSA specified in the *\*DSA=DN\_string* component of a DSAP attribute where the local DSA is the initiator.

The default is 1200 (that is, 20 minutes). The minimum value is 300 (that is, 5 minutes). If the specified value is less than this minimum value the minimum value is used automatically.

If the *\*RTO=recovery\_timeout* and the *\*RMR=recovery\_max\_retries* components are equal to 0, recovery is configured as unlimited (that is there is no limitation by timeout or by number of retries, and recovery never stops).

If either the *\*RTO=recovery\_timeout* or the *\*RMR=recovery\_max\_retries* component is equal to 0 (and the other one is equal to a non-zero value) no recovery will be done.

If the *\*RTO=recovery\_timeout* and the *\*RMR=recovery\_max\_retries* components are equal to a non-zero value, recovery will be attempted until both values have been exceeded.

### **RI=recovery\_interval**

Specifies the time interval in seconds between the end of a recovery attempt and the start of the next attempt for shadowing agreements. This component is relevant only for the shadowing agreements with the DSA specified in the **DSA=DN\_string** component of a DSAP attribute where the local DSA is the initiator and the **UM={update\_mode}** component of the **Shadowing-Agreement-Info** is **OC=TRUE** (*on\_change*). (See **Shadowing-Agreement-Info** for details.)

The default is 240 (that is, 4 minutes). The minimum value is 120 (that is, 2 minutes). If the specified value is less than this minimum value, the minimum value is used automatically.

### **RD=recovery\_divisor**

Specifies the time interval between two recovery attempts for shadowing agreements and is the *window\_size* divided by the *recovery\_divisor*. This component is relevant only for the shadowing agreements with the DSA specified in the **DSA=DN\_string** component of a DSAP attribute where the local DSA is the initiator and the **UM={update\_mode}** component of the **Shadowing-Agreement-Info** is *periodic\_strategy*. (See **Shadowing-Agreement-Info** for details.) The default is 16.

### **ATU=automatic\_total\_update**

A Boolean value that specifies whether the initiator DSA performs a total update automatically if it is required by the cooperating DSA (**TRUE**) or the administrator must trigger off the total update by performing a **sob establish** operation (**FALSE**). The default is **FALSE**. If the DSA supports only X.500-compliant shadowing, the default value **FALSE** must be used. (See the **sob create** operation in the *DirX Directory Administration Reference* for details.)

### **SUPP=sob\_supplier\_policies**

Specifies the policies used for supplier shadowing agreements. Specify

*sob\_supplier\_policies* in the following format:

```
{[US=supplier_update_strategy]  
[,MAXLOC=max_local_memory]  
[,MAXOSK=max_old_segments_kept]  
[,MAXICON=max_incremental_confirmations]}
```

where:

#### **US=supplier\_update\_strategy**

Specifies the standard supplier update strategy that the shadow supplier intends to use for the updates. The first shadowing operation involves a total update and subsequent operations an incremental update of the unit of replication. Specify one of the following keywords:

- **TOTAL-INCR** - the pure X.525 standard strategy. This strategy is recommended for interworking.
- **SEGM-TOTAL-INCR** - a standard conformant DirX Directory extension that overcomes the resource limitations of the standard mechanism by segmenting the messages. This is the default strategy if no value for **US** is specified.

The supplier and the consumer must use the same update strategy for a shadowing operational binding.

+ The use of the following components is recommended to avoid bottlenecks when processing the data to update the unit of replication:

#### **MAXLOC=max\_local\_memory**

Specifies the maximum amount of memory in kilobytes to be used to create the data to update the unit of replication.

The value **0** (zero) indicates that an unlimited amount of memory can be used. (Keep in mind that the limits imposed by **DIRX\_CTX\_LIMIT** always apply. See **DIRX\_CTX\_LIMIT** in section **DirX Directory Environment Variables** in the *DirX Directory Administration Reference* for details.) The default value is 512.

**MAXLOC** should be big enough to take up the largest entry to transfer. The average size of an entry is about 2 KB. To create a replication unit with at least one entry the DSA consumes additional memory for protocol information. For a 2 KB entry the DSA consumes about 6 KB. If the specified value is not big enough to create a replication unit with at least one entry the DSA automatically allocates the necessary size of memory.

Specifying a low value for **MAXLOC** results in a low consumption of temporary memory for the outgoing shadowing operation.

If the update strategy is **TOTAL-INCR**, **MAXLOC** is effective only for incremental updates. For total updates, **MAXLOC** is ignored.

If the update strategy is **SEGM-TOTAL-INCR**, **MAXLOC** is effective for both, incremental and segmented-total updates.

### **MAXOSK=max\_old\_segments\_kept**

Specifies the maximum number of old shadowing operational binding update history segment files saved in case of incremental updates. A value lower or equal **1** indicates that 2 files are saved. The default is **2**.

### **MAXICON=max\_incremental\_confirmations**

Specifies the maximum number of incremental update refreshes sent in a sequence of segmented incrementals without starting a complete new update cycle. A complete new update cycle requires a new bind and coordination before the incremental update can be performed. This component is only used in case of incremental refreshes, the *supplier\_update\_strategy* is **SEGM-TOTAL-INCR**, and the *update\_mode* (see **Shadowing-Agreement-Info** for details) is scheduled **OC=TRUE** (*on\_change*).

This component is useful only for units of replication with a high update frequency. In this case, it can happen that after a single update is confirmed by the consumer, new updates may be ready for transmission on the supplier side.

A value less than or equal to **1** indicates that each transmission is confirmed. The default is **1**.

### **CONS=sob\_consumer\_policies**

Specifies the policies used for consumer shadowing agreements. Specify *sob\_consumer\_policies* in the following format:

```
{[US=consumer_update_strategy]  
[,MAXLOC=max_local_memory]  
[,REPLS=replace_supplier]}
```

where:

#### **US=consumer\_update\_strategy**

Specifies the standard consumer update strategy that the shadow consumer requests. The first shadowing operation involves a total update and subsequent operations an incremental update of the unit of replication. If the database becomes inconsistent, the consumer requests a total refresh of the unit of replication. Specify one of the following keywords:

- **TOTAL-INCR** - the pure X.525 standard strategy. This strategy is recommended for interworking.
- **SEGM-TOTAL-INCR** - a standard-conformant DirX Directory extension that overcomes resource limitations of the standard mechanism by segmenting the messages. This is the default consumer update strategy if no value for **US** is specified.

The supplier and the consumer must use the same update strategy for a shadowing operational binding.

The use of the following component is recommended to avoid bottlenecks when processing the data to update the unit of replication:

**MAXLOC=max\_local\_memory**

Specifies the maximum amount of memory in kilobytes to be used to create the data to update the unit of replication.

This policy has no effect on consumer DSAs.

**REPLS=replace\_supplier**

A boolean value that specifies whether the consumer DSA can become the supplier of the shadowing agreement when performing a **sob switch** operation (**TRUE**) or a **sob switch** operation is always rejected (**FALSE**). The default is **FALSE**. If the DSA supports only X.500 compliant shadowing the default value **FALSE** must be used. (See the **sob create** operation in the *DirX Directory Administration Reference* for details.)

**LDSUP=sob\_ldif\_supplier\_policies**

Specifies the policies used for LDIF agreements. Specify *sob\_ldif\_supplier\_policies* in the following format:

```
{[BAS64=base64_encoding]
[,CHANGEO=changes_only]
[,CS=codeset]
[,CRLF=cr_and_lf_separator]
[,ENTRYO=entry_only]
[,MAXROW=max_rows]
[,{DELAT=delete_attr_info | DELATLIST=attribute_list}]
[,{MODAT=modify_attr_info | MODATLIST=attribute_list}]
[,PROG=program_name]}
```

where:

**BAS64=base64\_encoding**

Specifies a Boolean value that specifies whether Base-64 encoding is performed (**TRUE**) for attribute values and distinguished name strings containing values lower than 32 (SPACE character) or greater than 126 (~) (not printable data) or Base-64 encoding is not performed (**FALSE**). The default is **FALSE**.

**CHANGEO=changes\_only**

Specifies a Boolean value that specifies whether only updates are saved (**TRUE**) or a total content file is saved (**FALSE**). The default is **FALSE**.

**CRLF=cr\_and\_lf\_separator**

Specifies a boolean value that specifies whether carriage-return and line-feed (**TRUE**) or only carriage-return (**FALSE**) is used as a separator. The default is **FALSE**.

**CS=codeset**

Specifies the character set for the encoding in which printable data are saved in the file. Specify one of the following keywords:

- **LATIN1** - The data are saved in Latin1 format as specified in ISO8859-1. If you specify Latin1 format you must ensure that your tools can process the files because Latin1

is a non-standard code set for LDIF files.

- **UTF8** - The data are saved in UTF-8 format.

The default is **UTF8**.

**ENTRYO=entry\_only**

Specifies a Boolean value that specifies whether the entry is saved only with user attributes and the operational attributes **Creation Time** and **Modification Time** (**TRUE**) or all entry information (user attributes, all operational attributes, references and subentries) is saved (**FALSE**). The default is **FALSE**.

**MAXROW=max\_rows**

Specifies the maximum number of characters per line. The default is **80**.

**DELAT=delete\_attr\_info**

Specifies a Boolean value that specifies whether attribute types and values of a deleted entry or attribute are saved (**TRUE**) or not (**FALSE**). The default is **FALSE**.

**DELATLIST=attribute\_list**

Specifies that only a part of attribute types and values of a deleted entry or attribute is to be saved. *attribute\_list* specifies one or more abbreviations or OIDs (separated by a semicolon (;)) that correspond to the attribute types to be saved.

**MODAT=modify\_attr\_info**

Specifies a Boolean value that specifies whether attribute types and values of a modified entry or attribute are saved (**TRUE**) or not (**FALSE**). The attribute values represent the values before the modify operation. Attributes with a distinguished name part are not saved. The default is **FALSE**.

**MODATLIST=attribute\_list**

Specifies that only a part of attribute types and values of a modified entry or attribute is to be saved. *attribute\_list* specifies one or more abbreviations or OIDs (separated by a semicolon (;)) that correspond to the attribute types to be saved. The attribute values represent the values before the modify operation. Attributes with a distinguished name part are not saved.

**PROG=program\_name**

Specifies an IA5 string that specifies a command string to be performed after the creation of the file in LDIF format has been completed. Note that the DSA uses umask 00 and does not change or restrict access rights automatically; that is the files created by procedure will have read and write permission for owner group and everyone unless the called procedure sets umask according to the particular requirements.

When this policy is omitted, see the **lob create** command in the *DirX Directory Administration Reference* for information on how to export the file.

This policy can be used as follows:

- Simple use (only on Linux systems):

The string specifies the name of an executable, for example `/usr/bin/mail`, that is invoked after the creation of the LDIF file has completed. The string must not contain the character `%`. The LDIF file is directed to standard input of the executable. After the executable has completed, the LDIF file is removed and the exit code is evaluated. An exit code of **0** indicates success. An exit code of **1** indicates a temporary error; in this case, the LDIF agreement is scheduled for error recovery. Another exit code indicates a permanent error; in this case, the LDIF agreement is disabled.

- Advanced use:

The string specifies the name of an executable to be invoked after the creation of the LDIF file has completed. The string can contain macros (starting with the character `%`) to be replaced before the executable is invoked. You can specify one or more of the following macros:

**%F** - is replaced by the LDIF file name. This macro is required on Windows systems. When this macro is omitted on Linux systems, the LDIF file is directed to standard input of the executable as described above.

**%U** - is replaced by the current update time. The update time is specified in **Generalized Time** syntax as specified in RFC 2252. (See section **Generalized Time Syntax** details.)

**%L** - is replaced by the last update time. The last update time is specified in **Generalized Time** syntax as specified in RFC 2252. (See section **Generalized Time Syntax** details.) A value of **NONE** indicates that this agreement is processed the first time after establishing the LDIF agreement. (See the **lob establish** command in the *DirX Directory Administration Reference* for details.)

**%S** - indicates the kind of content that was or will be created. This macro is replaced by one of the following keywords:

**CONTENT-FILE** - indicates an LDIF content file (the total content was saved).

**CHANGE-LOG** - indicates a change log file (only updates were saved).

**NO-CHANGE** - indicates an empty change log file (no updates occurred).

**%A** - is replaced by the agreement identifier (without the version number) in numerical string format.

**%%** - is replaced by `%`.

**.\*any** - is replaced by *any* where *any* is a character other than the macros described above, the macro **.\*P** (see coordinated advanced use below for details), and the macro **%V** (see recommendations for troubleshooting below for details).

Other parameters are given to the executable as specified.

After the command has completed, the LDIF file is removed and the exit code is evaluated. An exit code of **0** indicates success. An exit code of **1** indicates a

temporary error; in this case, the LDIF agreement is scheduled for error recovery. Another exit code indicates a permanent error; in this case, the LDIF agreement is disabled.

- Coordinated advanced use:

When the macro **%P** (indicating the processing state) and at least the macro **%U** or **%L** (see advanced use for details) are specified, the executable is invoked before the LDIF agreement is processed. In this case, the **%P** macro is replaced by the keyword **COORDINATE**. The **%F** macro is replaced by the keyword **NO-FILE** and / or no redirection of standard input is performed. The executable may check the last update time, upgrade from a change log file format to a content file format, or discard LDIF files and restart recording of history. After the executable has completed, the exit code is evaluated:

An exit code of **0** indicates success; in this case, an LDIF file is created and the executable is invoked a second time. (The macro **%P** is replaced by the keyword **UPDATE**.)

An exit code of **1** indicates a temporary error; in this case, the LDIF agreement is scheduled for error recovery.

An exit code of **5** indicates that the current (total) content file or change log file should be discarded, but the update history should be recorded and a change log file should be created when processing the LDIF agreement next time. In this case, the executable is invoked a second time without creating an LDIF file. (The macro **%P** is replaced by the keyword **UPDATE**, **%S** is replaced by the keyword **DISCARDED**, and **%F** is replaced by the keyword **NO-FILE**.)

An exit code of **2** indicates that a (total) content file is required, an exit code of **3** indicates that a previous LDIF file is missing, an exit code of **4** indicates that the corresponding LDIF file has already been received. In these cases, the executable is invoked a second time without creating an LDIF file. (The macro **%P** is replaced by the keyword **COORDINATE**.) After the executable completes, the exit code is evaluated. Only exit code of **0** ((total) content file is created) or **5** (content/change file is discarded) are accepted. The executable is invoked again. (The macro **%P** is replaced by the keyword **UPDATE**.)

Any other exit code indicates a permanent error; in this case, the LDIF agreement is disabled.

- Recommendations for troubleshooting:

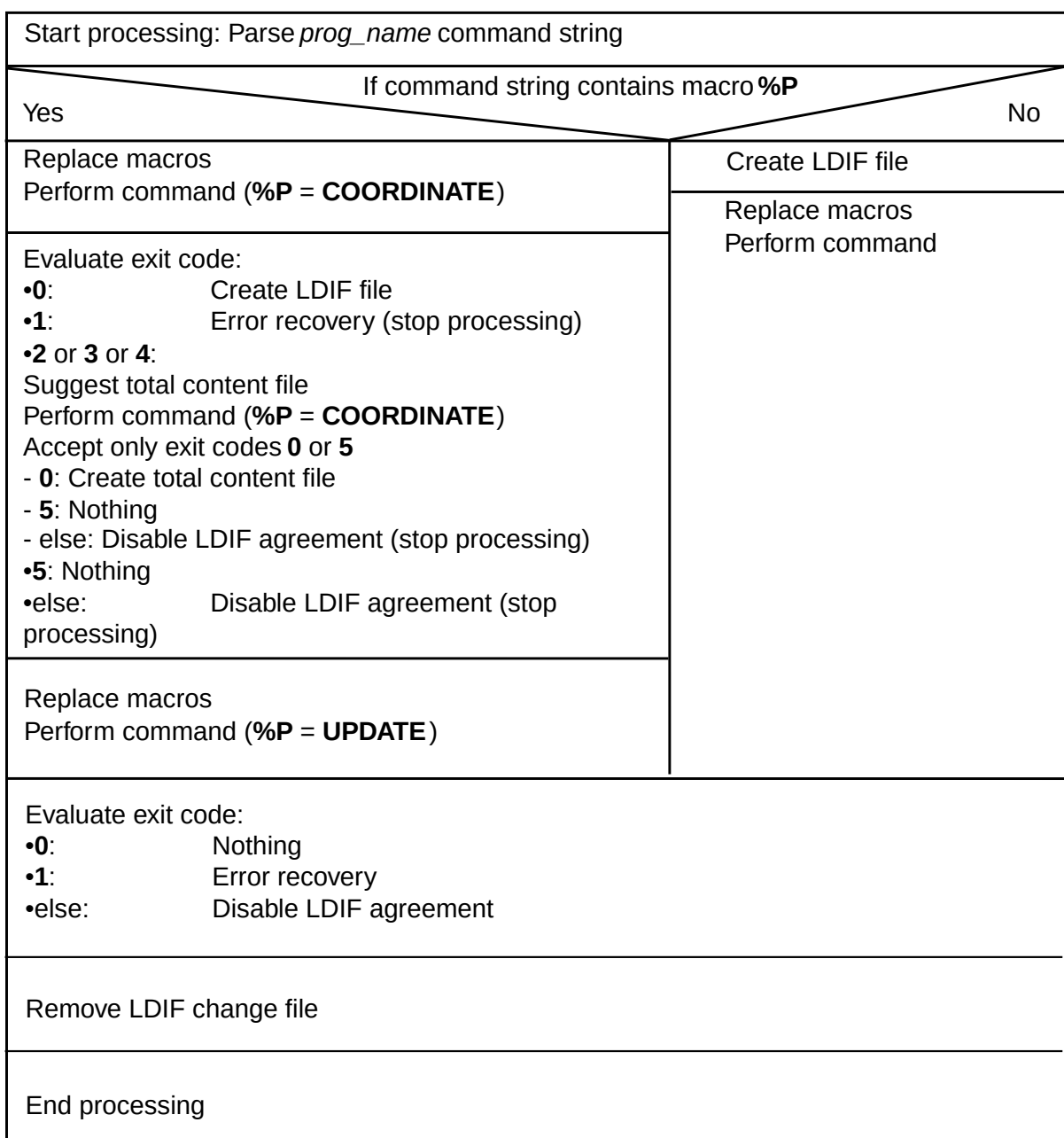
The command string *prog\_name* may contain the name of an existing program (e.g. /usr/bin/mail on Linux systems), a Linux shell script, or a self-written application. These executables should not use the exit codes **-1**, **-2**, and **-3** because they are used internally with the following meanings:

- 1: On Windows systems, create process failed. On Linux systems, fork failed.
- 2: On Windows systems, macro **%F** is missing. On Linux systems, exec failed.
- 3: The DSA was stopped (for example, by performing the **dirxadm** operation **sys stop**) or the LDIF agreement was terminated (for example, by performing the **dirxadm** operation **lob terminate**).

On Windows systems, system information (for example, function return codes of system operations) is saved in the USR log file.

Use the macro **%V** to get additional diagnostic and system information on Linux systems. Using **%V** causes stderr to be directed to a file with the LDIF file name to which an **e** has been appended. (See the **dirxadm lob create** operation in the *DirX Directory Administration Reference* for details on the LDIF filename.) Diagnostic, error and system messages are written to this file. The Linux shell option **-x** can be used to trace the processing of shell scripts into this file. On Windows systems, the macro **%V** is ignored.

The following figure illustrates the general processing while the *prog\_name* policy is performed.





### 1.7.17.3. Description

Information pertaining to policies for a shadow agreement between supplier and consumer DSA or an LDIF agreement is represented with the SOB-Policies attribute syntax. This syntax is only used for specifying the value of the **-pol** parameter in the **dirxadm lob** and **sob create** command and for displaying components of the Cooperating DSA table. It describes the following policies:

- Master-SOB-Policies (the **MAS** component), which defines the use of system resources such as disk space limitations (the **MAXJSZ** subcomponent) and file limitations (the **MAXJFIL** subcomponent).
- Initiator-SOB-Policies (the **INI** component), which defines the error recovery for initiator shadowing agreements.
- Supplier-SOB-Policies (the **SUP** component), which defines the update strategy in use (the **US** subcomponent) and the use of system resources.
- Consumer-SOB-Policies, which defines the update strategy in use (the **US** subcomponent) and the memory limitation (the **MAXLOC** subcomponent).
- LDIF-Supplier-SOB-Policies (the **LDSUPP** component) which basically defines the LDIF file format.

The following restrictions apply for specifying the value of the **-pol** parameter in the **dirxadm lob** and **sob create** command:

- Specify either a Supplier-SOB-Policy or a Consumer-SOB-Policy or an LDIF-Supplier-SOB-Policy according to the agreement created by the **dirxadm lob** and **sob create** command.
- Master-SOB-Policies can only be specified for primary shadowing operational bindings.
- Initiator-SOB-Policies can only be specified for initiator agreements. (The updates are supplier-initiated or consumer-initiated. See also **UM** component of **Shadowing-Agreement-Info** attribute syntax for details.)
- Supplier-SOB-Policies are usually specified for non-LDIF agreements. When specified for LDIF agreements only the **MAXLOC** subcomponent is evaluated. All other subcomponents are ignored.
- Consumer-SOB-Policies can only be specified for non-LDIF agreements.
- LDIF-Supplier-SOB-Policies can only be specified for LDIF agreements.

When entries greater than 512 KB should be saved in an LDIF file a Supplier-SOB-Policy must be specified. The **MAXLOC** subcomponent must be specified accordingly.

If no policy is specified, useful default policies are performed. When specifying the value of the

**-pol** parameter in the **dirxadm lob** and **sob create** command, the policies are performed only for the agreement created by this **dirxadm lob** and **sob create** command.

The **LDSUPP** LDIF agreement policy subcomponents **DELAT**, **DELATLIST**, **MODAT**, and **MODATLIST** add additional non-standard information to generated LDIF files that standard LDIF file parsers cannot process. Use these components only in LDIF synchronization

environments that provide customized LDIF parsing tools to handle these policies.

#### 1.7.17.4. Example

```
{INI={RMR=20,RTO=1200,RI=240,RD=16,ATU=TRUE}}
```

```
{SUPP={US=TOTAL-INCR,MAXLOC=1024}}
```

```
{CONS={US=TOTAL-INCR,MAXLOC=2048,REPLS=TRUE}}
```

```
{LDSUPP={CRLF=TRUE,MAXROW=132}}
```

```
{LDSUPP={CRLF=TRUE,MAXROW=132,DELAT=TRUE,MODAT=TRUE}}
```

```
{LDSUPP={CRLF=TRUE,MAXROW=132,DELATLIST=L;UID;RMB,MODATLIST=UID}}
```

```
{MAS={MAXJSZ=1000,MAXJFIL=20},  
LDSUPP={CRLF=TRUE,MAXROW=132}}
```

```
{LDSUPP={CRLF=TRUE,MAXROW=132,
```

```
    PROG=/usr/bin/mail John.Smith@my-company.net}}
```

On Linux systems, after the creation of the file in LDIF format has completed successfully the file is sent by e-mail to John Smith.

```
{LDSUPP={CRLF=TRUE,MAXROW=132,
```

```
    PROG=ldif-process p1 p2 %A %S %F p6 p7}}
```

After the creation of the file in LDIF format has completed successfully the macros %A, %S, and %F are replaced and then the executable ldif-process is performed. Note that p1, p2, p6, and p7 are also input parameters of ldif-process and are given to the process as specified.

Commandline:

```
ldif-process p1 p2 1001 CHANGE-LOG 0009.04789p p6 p7
```

where 1001 is the LDIF agreement identifier and 0009.04789p is the temporary LDIF file name

```
{LDSUPP={CRLF=TRUE,MAXROW=132,
```

```
    PROG=ldif-process p1 %P %A %S %F %U}}
```

Before starting the creation of the file in LDIF format the macros (starting with a %) are replaced and the

the executable `ldif-process` is invoked. Depending on the exit code of `ldif-process`, the LDIF file is created or not and then the `ldif-process` is invoked a second time.

Commandline (1. call before processing the LDIF agreement):

```
ldif-process p1 COORDINATE 1001 CHANGE-LOG NO-FILE  
19990105060000Z
```

where 1001 is the LDIF agreement identifier and 19990105060000Z is the current update time.

Exit code of `ldif-process` is 0:

The LDIF agreement is processed and an LDIF change log file is created.

Commandline (2. call after the creation of the LDIF file has been finished):

```
ldif-process p1 UPDATE 1001 CHANGE-LOG 0009.04789p  
19990105060000Z
```

where 1001 is the LDIF agreement identifier, 0009.04789p is the temporary LDIF file name 19990105060000Z is the current update time.

```
{LDSUPP={ENTRYO=TRUE,  
PROG="/C:///Multimedia Files///tmp///trigger.bat/" %F %U %L %S }}
```

On Windows systems, after the creation of the file in LDIF format has completed successfully the file is processed by the application `trigger.bat`.

## 1.7.18. Update-Status

An attribute syntax for attributes that hold the update status about the administered operational bindings with partner DSAs or LDIF agreements. The Update-Status attribute syntax is only used for display purposes of the **dirxadm lob** and **sob show** command; it cannot be used for input.

### 1.7.18.1. Synopsis

```
US={[D=disabled]  
US=update_status_info}
```

### 1.7.18.2. Components

#### **D=disabled**

A boolean value that indicates whether the agreement is disabled (**TRUE**), e.g. because errors occurred and the administrator's intervention is necessary before the agreement

can be enabled again, or enabled (**FALSE**). The default value is **FALSE**.

### **US=update\_status\_info**

A structured component that specifies the status of the supplier or consumer update. *update\_status\_info* is displayed in the following format:

```
{[SUS=supplier_update_status] | [,CUS=consumer_update_status]}
```

where:

#### **SUS=supplier\_update\_status**

Specifies the status of supplier updates. *supplier\_update\_status* is in the following format:

```
{AC=area_changes,OU=old_updates,FU=last_failed_update,NUT=next_update_time,  
AU=actual_update}
```

The **AC**=*area\_changes* component is one of the following keywords:

- **NO-CHANGES** - indicates that area changes were not made since last replication was performed.
- **CHANGED** - indicates that changes inside the replicated area were made since last replication was performed or during performing replication, which were not yet successfully shadowed to the consumer DSA.
- **TOUCHED** - indicates that changes inside the replicated area since last replication was performed were made and the replication of this changes is in progress.

The **OU**=*old\_updates* component contains the updates successfully sent by the supplier. *old\_updates* is displayed in the following format:

```
{SID=segment_id,UT=update_time,OPMSN=operation_msn AC=area_changes}[;{...}]
```

The **SID**=*segment\_id* component is an integer value identifying a time segment. It indicates which changes were shadowed within this update. For instance, **SID**=**3** means that all changes done until the end of the time segment 3 were transmitted within this update.

The **UT**=*update\_time* component is a time string in Generalized-Time format. It is the time at which the update was performed.

The **OPMSN**=*operation\_msn* component is an integer value identifying the modification sequence number (MSN) of the last confirmed operation. This component is relevant only for synchronous shadowing agreements (the **-consumerkind** option is set to **SYNCHRON** in the **dirxadm sob create** operation).

The **AC**=*area\_changes* component is a boolean value. A value of **TRUE** indicates that the update contained area changes. The value **FALSE** is only possible with a scheduled update strategy, and indicates that this update was an incremental update with no changes.

The **FU**=*last\_failed\_update* component contains the update last recently failed. The

failing update information is displayed whenever the DSA failed to send a shadowing PDU to the consumer DSA. The supplier DSA performs an error recovery procedure and re-tries to send the same request after the time interval specified in the Retry-Interval component of the SOB initiator policy or after 4 minutes by default. The **FU** component is cleaned up after a successful recovery. *last\_failed\_update* is displayed in the following format:

{**SID**=segment\_id,**UT**=update\_time,**AC**=area\_changes,**FR**=error\_code}

The **SID**=segment\_id component is an integer value identifying a time segment. It indicates which changes were shadowed within this update. For instance, **SID=3** means that all changes done until the end of the time segment 3 were transmitted within this update.

The **UT**=update\_time component is a time string in Generalized-Time format. It is the time at which the update was performed.

The **AC**=area\_changes component is a boolean value. A value of **TRUE** indicates that the update contained area changes. The value **FALSE** is only possible with a scheduled update strategy, and indicates that this update was an incremental update with no changes.

The **FR**=error\_code component is an integer value indicating the reason for the unsuccessful attempt to send this update.

The **NUT**=next\_update\_time component is a time string in Generalized-Time format. It is the time at which the next update is performed. This update is the next scheduled update or if the **FU** component is displayed the next attempt to send this update last recently failed.

The **AU**=actual\_update component contains information about the ongoing shadowing update the supplier DSA is sending to the consumer. *actual\_update* is displayed in the following format:

{**WS**=waiting\_state,**SID**=segment\_id,**UT**=update\_time,**AC**=area\_changes,**OS**=offset,**MC**=modification\_count,**LUE**=last\_updated\_entry}

The **WS**=waiting\_state component contains information about the state of the shadowing protocol handler with respect to the ongoing shadowing request. The value is one of the following keywords:

- **INVOKED** - indicates that the supplier has started to send a bind request or an update request to the consumer DSA.
- **WAIT\_BIND\_RESPONSE** - indicates that the supplier DSA is waiting for a response of the consumer DSA to the bind request. After a successful bind the supplier DSA starts to send update requests.
- **WAIT\_COORD\_RESPONSE** - indicates that the supplier DSA is waiting for a response of the consumer DSA that it is ready to receive the next update request.
- **SENT\_UPDATE** - indicates that the supplier DSA has sent one of several update messages and is collecting the next update message from its local database.

- **WAIT\_UPDATE\_RESPONSE** - indicates that the supplier DSA is waiting for the response of the consumer DSA to the last recently sent update request. The next update request to send is ready at that time. After a successful update the supplier DSA sends the next update request.

If the current update is an incremental one the following components are present:

The **SID=segment\_id** component is an integer value identifying a time segment. It indicates which changes were shadowed within this update. For instance, **SID=3** means that all changes done until the end of the time segment 3 were transmitted within this update.

The **UT=update\_time** component is a time string in Generalized-Time format. It is the time at which the update was performed.

The **AC=\*\*area\_changes** component is a Boolean value. A value of **\*TRUE** indicates that the update contained area changes. The value **FALSE** is only possible with a scheduled update strategy, and indicates that this update was an incremental update with no changes.

The **OS=offset** component is an integer value providing the offset in the journal file to the update request the supplier DSA is sending.

The **MC=modification\_count** component is an integer value providing the number of update requests that the supplier DSA has sent so far for the ongoing shadowing update.

The **LUE=last\_updated\_entry** component is the distinguished name of the last recently updated entry.

### **CUS=consumer\_update\_status**

Specifies the status of consumer updates. *consumer\_update\_status* in the following format:

{**LU=last\_update**,**CUS=cons\_upd\_stat**}

The *last\_update* parameter is a time string in Generalized Time format that specifies the time of the last update to the DIB. (See the **DirX Directory String Representation** chapter for a description of **Generalized Time** syntax.)

The *cons\_upd\_stat* parameter is one of the following keywords:

- **TOTAL-UPDATE** - indicates that the databases are not synchronized (e.g. replication was not yet started or errors occurred during the last replication process, for example the DSA was stopped).
- **INCREMENTAL-UPDATE** - indicates that the databases are synchronized (replication was started and the updates were transferred successfully).
- **TOTAL-RUNNING** - indicates that a total update is in progress.
- **INCREMENTAL-RUNNING** - indicates that an incremental update is in progress.

### 1.7.18.3. Description

The Update-Status is maintained by the DSA. It displays the current and planned updates of operational bindings between the present DSA and other DSAs or of LDIF agreements. DirX Directory supports only shadowing operational bindings (SOB). The Update-Status attribute syntax is only used for display purposes of the **dirxadm lob** and **sob show** command; it cannot be used for input.

### 1.7.18.4. Examples

```
US={D=FALSE,US={SUS={AC=CHANGED}}}
```

## 1.7.19. User-Policies

An attribute syntax used solely for the User Policy (USP), the attribute of the root DSE that specifies policies for DAP users, discriminating between them by name.

### 1.7.19.1. Synopsis

```
UserPolicies = {USN={DN_string}
  [,OPT=user_policy_options]
  [,AP={USER_authentication_policy}]
  [,SL=size_limit]
  [,PRSL=paged_result_size_limit]
  [,SPL=single_page_limit]
  [,ALS=attribute_limit:size] }
```

### 1.7.19.2. Attribute Type

#### UserPolicies

The abbreviation or OID that corresponds to a structured attribute with the User-Policies (USR\_POLICIES) syntax. For example, the USP (User Policy) is such an attribute. The USP attribute is a DSA-specific operational attribute of the root DSE, and is accessible with **dirxadm** only.

### 1.7.19.3. Components

#### USN=DN\_string

The distinguished name of a user, or the name of a group or a subtree name. The group can either be a groupOfNames (with member attributes) object including nested group support (auxiliary object class dirxImportGroup), or a groupOfUniqueNames (with uniqueMember attributes) object, or a dynamic group group object (object class dirxGroupOfUrls, attribute dirxMemberUrl).

Concerning the precedence of User Policy attributes, it is the specificity of the username that controls which value is relevant. During the bind operation, the DSA checks the User Policy (USP) attributes in the following order:

- If the user's distinguished name is found in the USN component of a User Policy, this

user policy is the relevant one.

- If there is a User Policy for a group of which the user is a member, this user policy is the relevant one. If the user is a member of more than one group occurring in USPs, the relevant user policy is constructed from the most restrictive policy settings of all user policies for groups of which the user is a member.
- If there is a User Policy for a subtree under which the user's entry is located, the lowest name defines the applicable user policy. Specify *DN\_String* as a forward slash (/) to apply the user policies to all otherwise unspecified users.

### **OPT=user\_policy\_options**

Policy options that specify whether the user can access the services of a DSA that can chain the user's operations, if necessary, and perform modify operations or search operations resulting in a sequential scan of the database.

Specify *user\_policy\_options* as one or more keywords or the value **0** (zero). Separate multiple keywords with the + (plus sign). The keywords are:

- **CHAINING** - Permits the DSA to generate chained operations on behalf of the user
- **MODIFY** - Permits the DSA to carry out local modify operations on behalf of the user
- **SEQSEARCH** - Permits the DSA to carry out subtree and one-level search operations that result in a sequential scan of the database on behalf of the user. The DSA performs a sequential scan if there is no index present for the matching rule specified and the attribute type used. For NOT filter items the ANY and the INITIAL/FINAL index are required to avoid a sequential scan. Regardless of the value of this option the sequential search is permitted if there are less than 128 entries located below the search base.

A contains filter searching for a single character contained in an attribute value, for example **cn=\*a\***, is always - independent from the presence of a CONTAINS index - performed as a sequential scan. It is rejected if the user policy denies sequential searches.

The default behavior of the DSA is to permit all options.

Supplying the value of **0** for *user\_policy\_options* means that the DSA will not perform chaining, modify operations, and sequential search operations on behalf of the user.

### **AP=user\_authentication\_policy**

Authentication policy for the user. Specify *user\_authentication\_policy* in the following format:

```
[AMB={auth_methods}]  
[,OPWD=any_octet_string]  
[,CV=credentials_validity]
```

where:

#### **AMB={auth\_methods}**

Specifies the authentication methods allowed for the user. Currently only the



following authentication methods are allowed. Specify *auth\_methods* in the following format:

#### **AM=authentication\_methods**

The *authentication\_methods* parameter is a sequence of the following keywords separated by semicolons:

- **NONE** - No authentication is performed. (Corresponds to an authentication level of **NONE** for access control.)
- **SIMPLE** - Simple authentication is performed, but no password is supplied. (Corresponds to an authentication level of **NONE** for access control.)
- **SIMPLE-WITH-PWD** - Simple authentication with a password is performed. The user's DN and password are passed in the clear. (Corresponds to an authentication level of **SIMPLE** for access control.)
- **SIMPLE-PROTECTED** - Simple authentication is performed with a secure token. A DN, password, a random number, and a timestamp are passed after being protected with hashing encoding function. (Corresponds to an authentication level of **SIMPLE** for access control.) The LDAP protocol does not support this method.
- **STRONG** - Strong authentication is based on public key cryptography. This is the case for ldap sasl bind operations with mechanism **EXTERNAL**.

The default is **NONE;SIMPLE;SIMPLE-WITH-PWD;SIMPLE-PROTECTED;STRONG**.

#### **OPWD=any\_octet\_string**

Specifies the password returned to the user by the DSA (if necessary) to validate the DSA. (Used only for **SIMPLE-PROTECTED** authentication.)

#### **CV=credentials\_validity**

Specifies the time period in seconds which is added to the expiration time of an incoming credential in order to check the credential's validity. (Used only with **SIMPLE-PROTECTED** authentication.)

The default is 30.

#### **SL=size\_limit**

Specifies the maximum number of entries of a search result for a search operation without paging. A smaller value of *size\_limit* overrules the size limit specified in the search request of the client. If the number of entries to be returned exceeds the *size\_limit* value the DSA returns *size\_limit* entries and a **SIZE LIMIT EXCEEDED** return code. The LDAP client, for example dirxcp, receives *size\_limit* SearchResultEntry messages and a SearchResultDone message containing sizeLimitExceeded resultCode. (See also **Managing Huge Search Results** in chapter **Managing Miscellaneous Scenarios** in the *Administration Guide*.)

A value of **-1** is unlimited. The default value for *size\_limit* is **2048**.

#### **PRSL=paged\_result\_size\_limit**

Specifies the maximum number of entries of a search result for a paged search

operation. The size limit **PRSL** is applied to the entire search result regardless of how many pages the result consists. A smaller value of *paged\_result\_size\_limit* overrides the size limit specified in the search request of the client. If the number of entries to be returned exceeds the *paged\_size\_limit* value the DSA returns *paged\_result\_size\_limit* entries, no search query reference (cookie for paging) and a **SIZE LIMIT EXCEEDED** return code. The LDAP client receives *paged\_result\_size\_limit* SearchResultEntry messages, a SearchResultDone message containing a sizeLimitExceeded resultCode and a pagedResultsControl with a cookie set to an empty value. The empty cookie indicates that there are no more entries returned in the paged search sequence. (See also **Managing Huge Search Results** in chapter **Managing Miscellaneous Scenarios** in the *Administration Guide*.)

The default value for *paged\_result\_size\_limit* is **16384**.

The maximum single page size limit is specified in the **SPL** component.

#### **SPL=**single\_page\_limit

Specifies the maximum number of entries per page of a search result for a search operation with paging. The size limit **SPL** is applied to each single result page. A smaller value of *single\_page\_limit* overrides the size limit specified in the search request of the client. If the number of entries to be returned exceeds the *single\_page\_limit* value the DSA returns *single\_page\_limit* entries per page, a search query reference (cookie for paging) and a **SIZE LIMIT EXCEEDED** return code. The LDAP client receives *single\_page\_limit* SearchResultEntry messages, a SearchResultDone message containing a sizeLimitExceeded resultCode and a paged resultsControl with a cookie set to a non-empty value. The non-empty cookie indicates that there are more entries that the server returns for subsequent requests of the paged search sequence. (See also **Managing Huge Search Results** in chapter **Managing Miscellaneous Scenarios** in the *Administration Guide*.) The client can continue to retrieve pages of searches by presenting the returned query reference.

The default value for *single\_page\_limit* is **2048**.

The maximum number of entries returned on all result pages is specified in the **PRSL** component.

#### **ALS=**attribute\_limit\_size

Specifies the maximum size of an attribute in an entry of a retrieval result. A value of **-1** is unlimited.

If an attribute exceeds the limit all of its values are omitted from the returned entry and "incompleteEntry" is set in the returned entry information. The limit is imprecise.

#### **1.7.19.4. Description**

The User-Policies syntax is for attributes that control a DSA's methods of interaction with users. The syntax is used for the multivalued USP attribute. The USP attribute specifies the policies for several individual users or group of users. These policies control:

- The authentication method required from the users

- Whether the users can access the chaining and modify services of the DSA
- The maximum number of entries of a retrieval result
- The maximum length of attribute values returned in a retrieval result.

#### 1.7.19.5. Example

```

USP={USN={/ },
      OPT=0,
      AP={AMB={AM=NONE}}
    };

USP={USN={/O=my-company/OU=asw},
      OPT=CHAINING+MODIFY,
      AP={AMB={AM=SIMPLE;SIMPLE-WITH-PWD},
          OPWD=my-pwd,
          CV=300}};
{USN={/O=my-company/OU=com},
  OPT=CHAINING,
  AP={AMB={AM=SIMPLE},
      OPWD=pwd}}
}
USP={{USN={/O=my-company/ou=asw },
      SL=1000, PRSL=5000, SPL=100};
      {USN={/O=my-company2/ou=mds },
      SL=2000, PRSL=20000, SPL=1000}}

```

#### 1.7.20. Continuation-Reference

A component syntax used for information returned by DSAs when the referral method of interaction is used for an operation. This syntax is described here for output purposes only; it cannot be used for input. This syntax is used by the **dirxcp** command to define the format and contents of displayed information.

##### 1.7.20.1. Synopsis

```

DSAR={TO=target-object
      [,ARDNS=aliased-rdns]
      ,OP=operation-progress
      [,RRDNS=resolved-rdns]
      ,RT=reference-type
      ,API=access-point_info;access-point_info;...
      ,EO=entry-only
      ,RTDUA=return-to-DUA
      ,NROM=name-resolve-on-master }

```

## 1.7.20.2. Components

### **TO=target-object**

A string that identifies the distinguished name of the object that should be used in continuing the operation on the referred-to DSA (for example, **/O=my-company**).

### **ARDNS=aliased-rdns**

An integer that identifies the number of aliases dereferenced to obtain the distinguished name of the target.

### **OP=operation-progress**

The state of the referral operation. *operation-progress* is in the format:

```
{NRP=name-resolution-phase  
[,NRDN=next-rdn-to-be-resolved]}
```

where:

#### **NRP=name-resolution-phase**

One of the following keywords:

- **NOT-STARTED**
- **PROCEEDING**
- **COMPLETED**

#### **NRDN=next-rdn-to-be-resolved**

An integer that indicates to the DSA which of the RDNs in the target object is the next to be resolved.

### **RRDNS=resolved-rdns**

An integer that identifies how many RDNs have currently been resolved using internal references only.

### **RT=reference-type**

The type referral. *reference-type* is one of the following keywords:

- **SUPERIOR**
- **SUBORDINATE**
- **CROSS**
- **NON-SPECIFIC-SUBORDINATE**
- **SUPPLIER**
- **MASTER**
- **IMMEDIATE-SUPERIOR**
- **SELF**

### **API=access-point\_info**

The access points for the master DSA and/or the shadow DSA in order to achieve the

continuation. *access-point\_info* is in the following format:

{**AP**=*MasterAndShadowAccPoints* [,**AAP**=*MasterAndShadowAccPoints*]}

See **Master-And-Shadow-Access-Points** for a description of *MasterAndShadowAccPoints* syntax. The **AAP** component specifies additional access points.

### **EO=entry-only**

A Boolean value that specifies whether the DSA should resolve only the named entry if alias names are encountered. (See **Simple Attribute Syntax** for a description of BOOLEAN syntax representation.) A value of **TRUE** indicates that name resolution should be performed only on the named entry.

### **RTDUA=return-to-DUA**

A Boolean value that specifies whether the requested information is available from the DUA. This component is used when for security reasons the DSA cannot return information from an intermediate DSA. (See **Simple Attribute Syntax** for a description of BOOLEAN syntax representation.) A value of **TRUE** indicates that the information must be obtained from the DUA.

### **NROM=name-resolve-on-master**

A Boolean value that specifies whether name resolution must be performed by the master DSA. (See **Simple Attribute Syntax** for a description of BOOLEAN syntax representation.) A value of **TRUE** indicates that name resolution must be performed by the master.

## **1.7.20.3. Description**

Continuation references passed during the referral method of interaction are represented with Continuation-Reference component syntax. The component contains the continuation reference for the referral

## **1.7.20.4. Examples**

```
DSAR={TO={/O=my-company/OU=asw/CN=dsa/CN=dsa20},
      OP={NRP=PROCEEDING,NRDN=5},
      RRDNS=4,
      RT=SUBORDINATE,
      API={MOS={MSAP={AE={/O=my-company/OU=asw/CN=dsa/CN=dsa20},
                    PSAP={TS=Server20,
                          NA=TCP/IP!internet=139.23.81.32+port=102}},
          CAT=MASTER}},
      EO=FALSE,
      RTDUA=FALSE,
      NROM=FALSE
}
```

## 1.7.21. Facsimile-Telephone-Number

An attribute syntax for attributes that specify Facsimile (FAX) numbers.

### 1.7.21.1. Synopsis

```
FacsimileTelephoneNumber = {PN=phone_number  
[,PAR=G3_FAX_NonBasicParameter]}
```

### 1.7.21.2. Attribute Type

#### FacsimileTelephoneNumber

The abbreviation or OID that corresponds to a structured attribute with the Facsimile-Telephone-Number (FAX\_TELEPHONE\_NO) syntax. For example, FTN (Fax-Telephone-Number) is such an attribute.

### 1.7.21.3. Components

#### **PN**=phone\_number

The fax number in Printable string format (up to 32 characters in length).

#### **PAR**=G3\_FAX\_NonBasicParameter

The settings for the G3 Fax parameters. Note that setting G3 parameters is hardly ever required. Specify *G3\_FAX\_NonBasicParameter* in the following format:

```
{[A3W=TRUE | FALSE]  
[,B4L= TRUE | FALSE]  
[,B4W= TRUE | FALSE]  
[,FR= TRUE | FALSE]  
[,TD= TRUE | FALSE]  
[,UC= TRUE | FALSE]  
[,UL= TRUE | FALSE]}
```

Specify **TRUE** to turn the parameter on or **FALSE** to turn it off.

### 1.7.21.4. Description

FAX numbers are represented with the Facsimile-Telephone-Number attribute syntax. The number consists of the country code and number in Printable string format. You can also set G3 non-basic parameters.

### 1.7.21.5. Example

```
FTN={PN=12345}
```

## 1.7.22. Name-And-Optional-UID

An attribute syntax for attributes that identify objects by a distinguished name and an optional identifier that can remove ambiguity from names that have been re-used.

### 1.7.22.1. Synopsis

*NameAndOptionalUID* = {**DN**=*{distinguished name}*  
[,**UID**=*BIT\_string*]}

### 1.7.22.2. Attribute Type

#### **NameAndOptionalUID**

The abbreviation or OID that corresponds to a structured attribute with the Name-And-Optional-UID syntax. For example, UM (Unique-Member) is such an attribute. The UM attribute can be used in objects which represent lists of names.

### 1.7.22.3. Components

#### **DN=distinguished name**

The distinguished name of the object. (See section **Distinguished Name** above in this chapter for details.)

#### **UID=BIT\_string**

A bit string that uniquely identifies the object. The bit string can be used when an object is removed from the Directory, and another object is subsequently given the same Directory name. The bit string must be chosen for uniqueness, such as a time-stamp (for example). (See section **Bit String Syntax** above in this chapter for details.)

### 1.7.22.4. Description

The identity of objects can be represented with the *Name-And-Optional-UID* attribute syntax. The attribute consists of the object distinguished name and an optional bit string that distinguishes between objects with the same distinguished name. The association between a user of the Directory and a name when the UID is present can be established by strong authentication using Version 2 (or later) certificates.

### 1.7.22.5. Examples

```
UM={DN={/O=my-company/OU=engineering/CN=Hughes H},
    UID=0010100101001011110};
{DN={/O=my-company/OU=engineering/CN=Smith J}
}
```

## 1.7.23. Partial-Outcome-Qualifier

A component syntax for information returned when DSA operations (search, list) cannot be completed. This syntax is only used for output purposes; it cannot be used for input. This syntax is used by the **dirxcp** command to define the format and contents of displayed information.

### 1.7.23.1. Synopsis

**POQF**={**LP**=*limit\_problem*  
[,**UDSA**=*continuation\_reference*[; ...],  
**UCE**=*unavailable-critical-extension*}

### 1.7.23.2. Components

#### **LP**=*limit\_problem*

One of the following keywords that indicate that a limit has been exceeded:

- **TIME LIMIT EXCEEDED**
- **SIZE LIMIT EXCEEDED**
- **ADMINISTRATIVE LIMIT EXCEEDED**

#### **UDSA**=*continuation-reference*

The DSA continuation reference for unexplored DSAs. See **Continuation-Reference** for the complete syntax of *continuation-reference*.

#### **UCE**=*unavailable-critical-extension*

A Boolean value that specifies whether or not mandatory protocol extensions are supported by the DSA involved the execution of this operation. (See **Simple Attribute Syntax** for a description of BOOLEAN syntax representation.)

### 1.7.23.3. Description

Information returned for partially completed operations is represented with the Partial-Outcome-Qualifier component syntax. The information includes a description of the limit that was exceeded, the continuation reference, and an indication of whether the DSA involved in the operation supports mandatory protocol extensions.

### 1.7.23.4. Examples

```
POQF={LP=SIZE LIMIT EXCEEDED,  
      UDSA={TO={/O=my-company/OU=asw/CN=dsa},  
            ARDNS=0,  
            OP={NRP=COMPLETED},  
            RT=SUBORDINATE,  
            API={MOS={MSAP={AE={/O=my-company/OU=asw/CN=dsa  
                                /CN=dsa20},  
                          PSAP={TS=Server20,  
                                NA='TCP/IP!internet=139.23.81.32+  
                                port=102'}},  
                      CAT=MASTER}}},  
      EO=FALSE,  
      RTDUA=FALSE,
```



```

        NROM=FALSE};
    {TO={/O=my-company/OU=asw/CN=dsa},
      ARDNS=0,
      OP={NRP=COMPLETED},
      RT=SUBORDINATE,
      API={MOS={MSAP={AE={/O=my-company/OU=asw/CN=dsa
                          /CN=dsa18},
                    PSAP={TS=Server18,
                          NA='TCP/IP!internet=139.23.81.32+
                          port=102'}}},
        CAT=MASTER}}},
    EO=FALSE,
    RTDUA=FALSE,
    NROM=FALSE}}
}

```

## 1.7.24. Postal-Address

A structured attribute syntax for postal addresses.

### 1.7.24.1. Synopsis

*PostalAddress* = {**PA1**=postal-address-string  
 [,**PA2**=postal-address-string]  
 [,**PA3**=postal-address-string]  
 [,**PA4**=postal-address-string]  
 [,**PA5**=postal-address-string]  
 [,**PA6**=postal-address-string]}

### 1.7.24.2. Attribute Type

#### PostalAddress

The abbreviation that corresponds to a structured attribute with the Postal-Address (POSTAL\_ADDR) syntax. For example, PA (Postal-Address) is such an attribute.

### 1.7.24.3. Components

#### *PAn*=postal-address-string

A string (of up to 30 characters long) in DirectoryString format that provides a line of a postal address.

### 1.7.24.4. Description

Postal addresses are represented with the Postal-Address attribute syntax. The attribute consists of up to 6 address lines in directory string format. Each address line is limited to 30 characters in (in accordance with ITU-T recommendation F.401). One address line is

required; the remaining 5 are optional.

#### 1.7.24.5. Example



In the following example backslashes are used to escape the comma, which is otherwise interpreted as a component-separation character.

```
PA={PA1=Tom Miller,  
    PA2=525\, Hacienda Ave.\,  
    PA3=Reading\, Massachusetts\, 01867  
    }
```

### 1.7.25. Presentation-Address

An attribute syntax for attributes that specify presentation addresses.

#### 1.7.25.1. Synopsis

```
PSA=[PS=P_Selector,  
    [SS=S_Selector,  
    TS=T_Selector,  
    NA=network_address[_network_address; ...],  
    [DNS=DNS_String]]
```

#### 1.7.25.2. Attribute Type

##### PSA

The abbreviation or OID that corresponds to a structured attribute with the Presentation-Address (PRESENTATION\_ADDR) syntax. For example, PSA (Presentation-Address) is such an attribute. The PSA attribute is used in entries that represent application entities or subclasses of application entities (such as DSAs).

#### 1.7.25.3. Components

##### **PS**=*P\_Selector*

The presentation selector address in Octet string format.

##### **SS**=*S\_Selector*

The session selector address in Octet string format.

##### **TS**=*T\_Selector*

The transport selector address in Octet string format of maximum 10 characters.

##### **NA**=*network\_address*

The network address in Octet string format. The use of NSAP macros as shown in the example is strongly recommended for entering network addresses. The following NSAP macros are available:

- IDM stack:  
**TCP/IP\_IDM!internet=\****IP\_address***+port=\****port\_number*
- OSI stack:  
**TCP/IP!internet=\****IP\_address***+port=\****port\_number*

where *IP\_address* is the IP address in dotted notation and *port\_number* is the port number.



**NA** is a mandatory component and thus must be specified even if the *DNS\_String* component is used for addressing.

## **DNS=DNS\_String**

A Printable String with the network address and additional information that supports the following features related to addressing:

- Network Address Translation (NAT)
- Secure X.500 protocols over the IDM stack
- IP version used by the IDM stack

This component is only supported for the IDM stack. If present, the *DNS\_String* is used for addressing and the network address specified in the **NA** component is ignored.

The *DNS\_String* is a printable string with the following internal format:

```
DNS=
'(HOST=name,
PLAINPORT=number
[,SSLPORT=number,]
[MODE=mode,]
[IPVERSION= ip_version])'
```

## **DNS\_String\_ Subcomponents**

### **HOST=name**

The name of the application entity's host. DirX Directory uses the DNS Service to resolve the IP addresses of the application entity. DirX Directory always uses the first IP address that the name service returns.

Using a name service is recommended to support network address translation (NAT) as provided, for example, in some firewall products. Using a name service is also recommended when communicating with DSAs over IPv6 protocol.

### **PLAINPORT=number**

The port number on which the application entity is listening for plain X.500 protocols. The resolved IP address and the plain port number are used to perform the DAP, DSP or DISP bind over plain IDM.

### **SSLPORT=number**

The port number the application on which the entity is listening for secure X.500 protocol.

In the context of an own address; that is, as the value of the DIRX\_OWN\_PSAP environment or as the value of the SELF entry in the **dirxcl.cfg** or **dirxldap.cfg** files, the SSLPORT subcomponent indicates whether the IDMS stack is to be initialized: If the address contains an SSLPORT with a value greater than zero, the IDMS stack is initialized. In the DSA process, IDMS starts a communication listener on the port specified by SSLPORT.

For outgoing connections, the resolved IP address and the SSL port number of the communication peer's PSAP are used to perform the DAP, DSP or DISP bind over IDM protected by SSL/TLS (IDMS). Suitable key material must be installed in order to be able to perform the X.500 protocols over SSL/TLS. See the chapter **Files** in the *DirX Directory Administration Reference* for details on IDMS configuration and key material.

### **MODE=mode**

The security mode for IDM. Specify one of the following values for *mode*:

- **PLAIN** - X.500 protocols are performed over plain IDM (default).
- **SSL** - X.500 protocols are performed over IDM protected by SSL/TLS (IDMS).

The *mode* value determines whether an outgoing connection to the application entity is performed to the plain port via plain IDM or to the SSL port via secure IDMS.

### **IPVERSION=ip\_version**

The IP stack version. Specify one of the following values for *ip\_version*:

- **4** - IPv4 address (default).
- **6** - IPv6 address.

The *ip\_version* specifies whether the IPv6 or IPv4 stack should be used for outgoing communication to the corresponding DSA. Keep in mind that you must administer the name service correctly before you can use this component.



older versions of DirX Directory support the DNS String format **DNS** = *host\*\*port[\*:ipversion]*. The latest version of DirX Directory also supports this older format, but using it prevents any benefit from the new IDMS feature.

#### **1.7.25.4. Description**

Presentation address is represented with the Presentation-Address attribute syntax. Presentation addresses are always associated with application entity entries. The PSAP macro utility handles presentation address attributes.

With respect to the IP version, DirX Directory is able to handle networks with DSAs of different versions running different protocol stacks. It is prohibited to run a mixed OSI- /

IDM-stack network. The OSI stack always uses IPv4 protocol. However running an IDM stack you can mix communication over IPv4 and IPv6 protocol. The network can consist of

- DSAs not understanding the **DNS** component:

These DSAs always communicate over IPv4 protocol. They use the IPv4 address specified in the **NA** component.

- DSAs understanding the **DNS** component but not the *\*:ip\_version* subcomponent:

These DSAs always communicate over IPv4 protocol. They use the *DNS\_name* to resolve the IP address from a name service. If the name service does not return a correct IPv4 address the DSA uses the IPv4 address specified in the **NA** component.

- DSAs understanding the **DNS** component and the *\*:ip\_version* subcomponent:

These DSAs can communicate over IPv4 and IPv6 protocol simultaneously if they are running on a dual-stack machine. They use the *DNS\_name* and the *ip\_version* to resolve the IP address from a name service. If the name service does not return the correct IPv4 address the DSA uses the IPv4 address specified in the **NA** component. If the name service does not return a correct IPv6 address establishing the connection fails.

With respect to secure X.500 protocols, DirX Directory is able to handle heterogeneous connections; that is, to use plain IDM to connect to set of communication peers while using IDMS to another set of communication peers.

#### 1.7.25.5. Examples

1. The following example shows a presentation address with:

- T-Selector (the **TS** component), an octet-string with octet values, set to the string "server," using ISO8824 encoding:
- Network address (the **NA** component) set to correspond to an IPv4 address of 123.45.67.89, with a port number of 31111:

```
PSA={TS=server,NA='TCP/IP_IDM!internet=123.45.67.89+port=31111'}
```

2. The following example shows a presentation address with a **DNS** component:

```
PSA={TS=DSA3,NA='TCP/IP_IDM!internet=139.25.13.11+port=23333',DNS='(HOST=DirXDSA3.my-company.de,PLAINPORT=23333)'} }
```

Let us assume that a DSA finds this PSAP address in the subordinate reference when chaining an update request to DSA3. It first reads the IPv4 address of **DirX-DSA3.my-company.de** from the name service and then performs the DSP bind to DSA3 with this IP address and the port number 23333.

3. The following example shows a presentation address with a **DNS** component for a DSA

that communicates over IPv6 protocol:

```
{TS=DSA4,NA='TCP/IP_IDM!internet=111.22.33.55+port=24444',DNS='(HOST=BOSDSA4,PLAINPORT=24444,IPVERSION=6)'}4
```

Keep in mind that for IPv6:

- You must use a machine that is able to handle IPv6 protocol. To be able to communicate also with DSAs communicating over IPv4 protocol, you must use a dual-stack machine.
  - You must administer the name service correctly.
  - For incoming communication, you must specify the environment variable **DIRX\_IP\_STACK**.
4. The following example shows the presentation address of a consumer DSA with a **DNS** component that specifies that communication to this DSA must be performed via secure IDM:

```
{TS=DSA3,NA='TCP/IP_IDM!internet=111.22.33.44+port=23333',DNS='(HOST=DURDSA3,PLAINPORT=21100,SSLPORT=21201,MODE=SSL)'}3
```

Let us assume that the supplier DSA finds this PSAP address in the Consumer Knowledge of the shadow consumer DSA when replicating an update via DISP to DSA3. It first resolves the IPv4 address of **DURDSA3** from the name service. If its own PSAP contains an SSL PORT greater than 0, it performs the DISP bind to DSA3 to the port number 21201 over secure IDM.

## 1.7.26. Protocol-Information

An attribute syntax for attributes that specify protocol information in the form of profiles (identified by object identifier) that are associated with particular network addresses.

### 1.7.26.1. Synopsis

```
ProtocolInformation = {NA=NSAP-Address,  
                  OID=object_identifier[_object_identifier..._]}
```

### 1.7.26.2. Attribute Type

#### ProtocolInformation

The abbreviation or OID that corresponds to a structured attribute with the Protocol-Information syntax. For example, PI (Protocol-Information) is such an attribute. The PI attribute is restricted to entries (e.g., those of object class application-entity or subclasses of application-entity), which themselves contain presentation-address attributes.

### 1.7.26.3. Components

#### **NA=NSAP-Address**

The NSAP network address of the communications object to be described. The NSAP address must be one of the NSAP addresses in the presentation address of the object being described.

#### **OID=object\_identifier**

The OID or an abbreviation for the OID that corresponds to a profile (e.g., as defined by OIW or EWOS) that is supported by the communications object. Note that these OIDs represent low-level communications characteristics (e.g., those associated with the network layer or below).

### 1.7.26.4. Description

Protocol-Information attribute syntax represents lower-layer communications characteristics of communications objects. This information consists of the NSAP address to which the profile applies and one or more object IDs that represent the profiles to which the object conforms.

### 1.7.26.5. Examples

```
PI={NA='TCP/IP!internet=123.45.67.89+port=102',  
    OID=1.3.14.65.237;1.3.14.65.239}
```

## 1.7.27. Subtree-Specification

An attribute syntax for attributes or for attribute components that specify subtrees. This attribute is used in the subtree-specification attribute of subentries to represent the target of policies held by the subentries. Note that this syntax is also used as part of the **Shadowing-Agreement-Info** attribute syntax and the **ACI-Item** syntax.

### 1.7.27.1. Synopsis

```
SubtreeSpecification = {[DEF=boolean_value]  
    [,BAS=base_object_of_subtree]  
    [,EXCL=specific_exclusions]  
    [,MIN=minimum_base_distance]  
    [,MAX=maximum_base_distance]  
    [,SF=specification_filter]}
```

### 1.7.27.2. Attribute Type

#### **SubtreeSpecification**

The abbreviation or OID that corresponds to a structured attribute with the SubtreeSpecification (SUBTREE\_SPEC) syntax. For example SS (Subtree-Specification) is such an attribute. SS is a mandatory attribute of a subentry.

### 1.7.27.3. Components

#### **DEF=boolean\_value**

A Boolean value that specifies whether the subtree consists of the entire administrative area. If **DEF=TRUE** (the subtree consists of entire administrative area), you should not specify the BAS and SF components. If the BAS and / or SF components are specified a **DEF=TRUE** is ignored.

#### **BAS=base\_object\_of\_subtree**

The base of the subtree, specified in a context-dependent way:

- In subentries - Relative to the administrative point
- In **Shadowing-Agreement-Info** - Relative to the context prefix
- In **ACI-Item (UC component)** - Absolute (relative to the root of the DIT)

In all cases, however, the Distinguished Name format is used (starts with a /). If this component is omitted, but the SF component is supplied, the SF component (below) is used as a filter for the entire administrative area. To specify the whole subtree (inclusive the base object of the subtree), use the notation **BAS={/}**.

#### **EXCL=specific\_exclusions**

This component defines a list of exclusions. It has the following format:

```
{CHB={chop_before}} | {CHA={chop_after}} [; {CHB={chop_before}} | {CHA={chop_after}} ...  
]
```

The **CHB=chop\_before** form defines a limit point that is to be excluded along with its subordinates from the subtree or subtree refinement. The limit point is an entry identified by a distinguished name relative to the base object of subtree.

The **CHA=chop\_after** form defines a limit point whose subordinates are to be excluded from the subtree or subtree refinement. The limit point is an entry identified by a distinguished name relative to the base object of subtree.

#### **MIN=minimum\_base\_distance**

An integer value that excludes all entries superior to entries that are *minimum\_base\_distance* RDN arcs below the base. The default value of zero corresponds to the base object of subtree.

This component is ignored when specified in a **Shadowing-Agreement-Info**.

#### **\_MAX=\_maximum\_base\_distance**

An integer value that excludes all entries subordinate to entries that are *maximum\_base\_distance* RDN arcs below the base. An absent of the component indicates that no lower limit should be imposed on the subtree or subtree refinement.

#### **SF=specification\_filter**

The *specification\_filter* is used to select entries in the subtree subordinate to the base defined by the BAS component or the entire administrative area. Specify *specification\_filter* in the following format:



```
{ITEM=object_class} |
{AND=specification_filter[_specification_filter_ ... ]} |
{OR=specification_filter[_specification_filter_ ... ]} |
{NOT=specification_filter} |
{FILTER=ldap_search_filter}
```

*object\_class* is either the OID of an object class or an object class abbreviation. The SF component selects entries that match the *specification\_filter* within their object class attributes.

*ldap\_search\_filter* is the string representation of a LDAP search filter as specified in RFC 4515. If **FILTER** is specified the **SF** component selects entries that match the LDAP search filter as specified. The **FILTER** component may be used exclusively within the *specification\_filter*. That is **FILTER** must not occur in any combination with the other components of *specification\_filter*. The **FILTER** component is not supported in subtree specifications of shadow agreements respectively LDIF agreements. Creating such an agreement results in an erroneous operation. Some restrictions may apply to the used attribute types and filter expressions. Please refer to the actual DirX Directory Release Notes of your deployed DirX Directoy server.

#### 1.7.27.4. Description

Specifications of subtrees used by subentries or in ACL items to define target entries for policies, are represented with the Subtree-Specification (SUBTREE\_SPEC) syntax.

#### 1.7.27.5. Examples

```
SS={DEF=TRUE}

SS={BAS={/OU=Research/OU=Metallurgy Department}}

SS={BAS={/OU=Research/OU=Metallurgy Department },SF={ITEM=ORP}}

SS={BAS={/O=sni},
    EXCL={CHB={/OU=Research/OU=Metallurgy}};
    MIN=0,MAX=10
    SF={ITEM=PER}}

SS={BAS={/O=sni},
    EXCL={CHB={/OU=Research/OU=Metallurgy}};{CHA={/OU=Sales}},
    MIN=0,MAX=10}

SS={SF={ITEM=2.5.6.7}}

SS={BAS={/OU=Sales},
    SF={AND={ITEM=ORP};{ITEM=IPER}}}
```

Selects all organizational persons of the organizational unit Sales that are also inet organizational persons. That are all entries under the base object /OU=Sales with the object class values ORP and IPER.

```
SS={BAS={/OU=Sales},
```

```
SF={AND={ITEM=ORP};{NOT={ITEM=IPER}}}}
```

Selects all organizational persons of the organizational unit Sales that are not inet organizational persons. That are all entries under the base object /OU=Sales with the object class value ORP and not IPER.

```
SS={BAS={/OU=Sales},
```

```
SF={OR={ITEM=ORP};{ITEM=OU}}}
```

Selects all organizational persons and all organizational units of the organizational unit Sales. That are all entries under the base object /OU=Sales with the object class values ORP or OU.

```
SS={BAS={/},
```

```
SF={Filter=(&(title=Manager*)(!(employeetype=contract)))}}
```

Selects all entries where attribute title begins with Manager and attribute employeetype is not contract.

## 1.7.28. Teletex-Terminal-Identifier

An attribute syntax for attributes that specify a Teletex terminal.

### 1.7.28.1. Synopsis

```
TeletexTerminalIdentifier = {TT=teletex-terminal  
[,PAR=teletex-non-basic-parameters] }
```

### 1.7.28.2. Attribute Type

#### TeletexTerminalIdentifier

The abbreviation or OID that corresponds to a structured attribute with the Teletex-Terminal-Identifier syntax. For example, TTI (TTX-Terminal-Identifier) is such an attribute.

### 1.7.28.3. Components

#### **TT**=teletex-terminal

A string (up to 1024 characters long) that identifies the terminal. Specify the *teletex-terminal* parameter using the Printable String syntax. See **Simple Attribute Syntax** for a description of Printable String Syntax representation.

### **PAR=teletex-non-basic-parameters**

A component that sets non-basic parameters for various teletex options. Specify *teletex-non-basic-parameters* in the following format:

```
{[CCS=control_character_sets]
  [,GCS=graphic_character_sets]
  [,MC=miscellaneous_capabilities]
  [,PF=page_formats]
  [,PU=private_use]}
```

where:

#### **CCS=control\_character\_sets**

A T61 string that defines the control character sets to use.

#### **GCS=graphic\_character\_sets**

A T61 string that defines the graphic character sets to use.

#### **MC=miscellaneous\_capabilities**

A T61 string that defines miscellaneous capabilities to use.

#### **PF=page\_formats**

An OCTET string that defines the page formats to use.

#### **PU=private\_use**

An OCTET string that defines user-defined Teletex parameters.

### **1.7.28.4. Description**

Teletex terminals and their parameters are represented by the Teletex-Terminal-Identifier attribute syntax. The information consists of a printable string that identifies the terminal and optional non-basic, advanced parameters that control terminal parameters such as character sets, page formats and other capabilities.

### **1.7.28.5. Examples**

```
{TTI={TT= My Company Teletex center}}
```

## **1.7.29. Telex-Number**

An attribute syntax for attributes that specify Telex numbers.

### **1.7.29.1. Synopsis**

```
TelexNumber = {TN=telex_number
  [,CC=country_code]
  [,AB=answer_back]}
```

### 1.7.29.2. Attribute Type

#### TelexNumber

The abbreviation or OID that corresponds to a structured attribute with the Telex-Number (TELEX\_NO) syntax. For example, TXN (Telex-Number) is such an attribute.

### 1.7.29.3. Components

#### TN=telex\_number

The telex number in Printable string format (of up to 14 characters long).

#### CC=country\_code

The country code in Printable string format (of up to 4 characters long).

#### AB=answer\_back

The short textual string (of up to 8 characters long), in Printable string format, with which the telex station responds when required to indicate its identity. (For example, when the telex station is connected to.)

### 1.7.29.4. Description

Telex numbers are represented with the Telex-Number attribute syntax. The number consists of the country code, Telex number, and answer-back code.

### 1.7.29.5. Example

```
{TXN={TN=24344,  
      CC=046,  
      AB=GAMEX B }}}
```

## 1.7.30. DirX Directory-Member-Url

An attribute syntax for attribute DRXMURL (in LDAP dirxMemberUrl).

### 1.7.30.1. Synopsis

*DRXMURL* = ldap:///BASE-DN??SEARCH-SCOPE?(SEARCH-FILTER)

### 1.7.30.2. Attribute Type

#### DRXMURL

The abbreviation or OID that corresponds to an attribute that requires a DirX Directory-Member-Url syntax. Per the schema definition, the syntax of the attribute is DirectoryString but DirX Directory DSA expects the values in the format described here. DRXMURL (DirX Directory-Member-Url) is an example of this type of attribute.

### 1.7.30.3. Components

#### **BASE-DN**

Specifies the distinguished name (DN) in LDAP syntax of the search base or the point from which the search is performed. This component is required.

Example: ou=Accounting,o=My-Company

#### **SEARCH-SCOPE**

Specifies the scope of the search. The following keywords are permitted:

- **base** retrieves information only about the DN specified in BASE-DN.
- **one** retrieves information about entries one level below the DN specified in BASE-DN.
- **sub** retrieves information about entries at all levels below the DN specified in BASE-DN.

This component is required.

Example: sub

#### **SEARCH-FILTER**

Specifies a search filter in LDAP syntax as described in RFC4515 that applies to entries within the specified scope of the search. This component is required and must be specified within parenthesis.

Example: (&(l=munich)(title=senior\*))

### 1.7.30.4. Description

This attribute syntax specifies a simplified LDAP URL syntax based on RFC2255. This syntax is identical for DAP and LDAPbinds. It is used to describe an LDAP search expression for managing dynamic groups. Some restrictions may apply to the search filter. Please refer to the *DirX Directory Release Notes* to see which restrictions apply in your deployed DirX Directory Service.

### 1.7.30.5. Example

```
{DRXMURL=ldap:///o=My-Company??sub?(&(l=munich)(title=senior*))}
```

## 2. DirX Directory String Representation for LDAP Binds

This chapter describes LDAP-style string representations of simple and structured DirX Directory attributes, search filters and distinguished names. This string representation is used with the administration program **dirxcp** for LDAP binds to enter and display directory information. (See section **Bind Types and Bind IDs** of chapter **DirX Directory Commands** in the *DirX Directory Administration Reference* for details.)

This chapter provides:

- A table of reserved characters for attributes
- An overview of the elements and format of simple and structured attributes
- An overview of elements and format of distinguished names
- An overview of elements and format of search filters
- A description of attribute syntax

Refer to the series of LDAP RFCs (RFC 4511 ff) for additional information about LDAP.

### 2.1. Reserved Characters

The following table describes reserved characters used for simple and structured attributes and distinguished names.

| Character | Purpose                                                                                                                                                                                                                                           |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| { }       | <b>For attributes and distinguished names:</b> Encloses the entire attribute (type and value) or distinguished name to indicate that white space is part of an attribute value. For example, <b>{cn=Henry Mueller}</b> or <b>{o=SNI AG, c=de}</b> |
| ;         | <b>For attributes:</b> Separates multiple values.                                                                                                                                                                                                 |
| \         | Escapes a reserved character.                                                                                                                                                                                                                     |



The section titled **String Representations for Structured Attribute Syntaxes** and RFC 2252 specifies additional reserved characters as separators, for example `""`, `"$"` or `"#"`, that also must be escaped if it should occur in that attribute values.

### 2.2. Simple and Structured Attributes

Simple attributes are in the form:

*type=simple\_value[;\_simple\_value\_ ...]*

Structured attributes are in the form:

`type=structured_value[:structured_value_ ... ]`

Each of the elements in simple and structured attributes is described in the following sections.

### 2.2.1. Attribute Types

For both simple and structured attributes, the *type* parameter identifies the attribute.

Internally attribute types are identified by OID (object identifiers), a unique series of integers separated by the period (.) character. For example, **2.5.4.3** is the OID for the Common-Name attribute type. You can identify attributes by OID, but to make it easier to specify attributes, the **dirxcp** commands allow you to identify them by LDAP names in command lines. There are no abbreviated and verbose forms as for attribute types for DAP binds.

The LDAP name(s) of an attribute are specified in the system schema. For example, the LDAP name **c** or **countryName** represents the Country attribute type, the LDAP name **cn** or **commonName** represents the Common-Name attribute type. When specifying LDAP names in **dirxcp** commands, LDAP names are treated case-insensitive; for example, **cn**, **Cn**, and **CN** are all valid ways to specify the LDAP name for the Common-Name attribute type independent of the exact value for the LDAP name specified in the system schema.

When you display attributes, **dirxcp** uses

- the exact LDAP name value specified in the system schema, when the LDAP name was specified in the request.
- the first exact LDAP name value specified in the system, when specifying options like **-allattr** in requests.
- the object identifier (OID), when the object identifier was specified in the request.

See sections **Attribute Table** and **Object Classes Table** of chapter **DirX Directory Default DSA Schema** in the *DirX Directory Administration Reference* for the LDAP names of attributes and object classes of the default schema.

### 2.2.2. Simple Attribute Values

For simple attributes, the *simple\_value* parameter is the value assigned to the attribute. *simple\_value* can be only a simple value, not another attribute. To enter more than one *simple\_value*, separate each with a semicolon as shown in the following examples:

```
{telephoneNumber=+1 964 123 456}  
{2.5.4.20=+1 964 123 456}  
{telephoneNumber=+1 964 123 456;+1 965 234 543}
```

Simple attribute values are always treated as UTF-8 strings for LDAP binds.

### 2.2.3. Structured Attribute Values

In contrast to the DAP protocol, the LDAP protocol usually handles all attribute values as UTF-8 strings. There is no common rule how to specify structured attribute values. Refer to section **String Representations for Structured Attribute Syntaxes** of this chapter and RFC 2252 (Attribute Syntax Definitions) for structured attribute syntaxes that are supported by the LDAP server and how they are specified. When displayed structured attribute values are not broken into subcomponents in pretty mode.

### 2.2.4. Attribute Lists for Simple and Structured Attributes

Most **dirxcp** operations allow you to specify more than one simple or structured attribute on the command line. To specify more than one attribute, separate each attribute type and value string with white space. The following **-attribute** option specifies an OCL, DSC and TN simple attributes, all separated by a space. Note that the DSC and TN attributes are enclosed in braces (**{ }**) because they contain white space for readability.

```
-attribute objectClass=organizationalUnit
           {description=Engineering Department}
           {telephoneNumber=+1 964 123 4567}
```

### 2.2.5. Binary Attribute Values

The LDAP v3 protocol supports the specification of attribute values in binary format, that is the ASN.1 encoding of the attribute value. To display and specify the binary attribute values on the user interface of **dirxcp** the Base-64 encoded representation of the attribute value is used. To specify or read attribute values in binary format the syntax is as follows:

|                                                 |                                       |
|-------------------------------------------------|---------------------------------------|
| <code>{attribute;binary}</code>                 | to read the attribute value binary    |
| <code>{attribute;binary=attribute_value}</code> | to specify the attribute value binary |

where *attribute* specifies the attribute type and *attribute\_value* specifies the Base-64 encoding of the binary attribute value when creating or modifying this value.

When the syntax of the attribute is not OCTET STRING null bytes in the value are not allowed.

Specifying the attribute value in binary format is possible for the following **dirxcp** operations:

- **obj create** (**-attribute** option)
- **obj modify** (**-addattr**, **-changeattr**, **-removeattr** options)
- **obj compare** (**-attribute** option)
- **obj search** (**-attribute** option)
- **obj show** (**-attribute** option)



### Examples:

The following example displays the Street Address attribute (**street**) of the person **cn=Digger, ou=Development, o=My-Company** Base-64 encoded:

```
show cn=Digger,ou=Development,o=my-company -attr {street;binary} -p
```

The output of the sample command as follows:

```
1) cn=Digger,ou=Development,o=my-company  
   street;binary : MjQgRG91Z2FuIFN0cmVldA==
```

## 2.2.6. Attribute Values in a File

For LDAPv3 binds, attribute values can also be specified in a file or written to a file. The syntax is as follows:

```
attribute[;binary]_FILE=filename1[;_filename2 ...]
```

where *attribute* specifies the attribute type and *filename* the name of the file containing the attribute value. When specifying multiple attribute values each value is saved in a separate file. *filename1\*;filename2 ...* then specify the value files. When reading multiple attribute values ***\*number is appended to filename where number is \*1 to n*** (total number of values). The first value is written to *filename*.

If the keyword **;binary** is specified, the file contains the valid ASN.1 encoded value of the attribute.

For the X.509 attributes Authority Revocation List, CA-Certificate, Certificate Revocation List, Cross-Certificate-Pair, Delta Revocation List, and User Certificate the file always contains the valid ASN.1 encoded value of the attribute regardless of specifying the keyword **;binary**.

Specifying the attribute value in a file is possible for the following **dirxcp** operations:

- **obj create** (-attribute option)
- **obj modify** (-addattr, -changeattr, -removeattr, -replaceattr options)
- **obj compare** (-attribute option)
- **obj show** (-attribute option)
- **obj search** (-attribute option)

### Examples:

1. The following example adds two pictures to the jpegPhoto attribute. The files `"/tmp/john1.jpg"` and `"/tmp/john2.jpg"` contain the binary representation of the pictures.

```
modify cn=Huber,ou=sales,o=my-company \
```

```
-addattr {jpegPhoto_FILE=/tmp/john1.jpg;/tmp/john2.jpg}
```

2. Compare the jpegPhoto attribute. The file "pict.jpg" contains the value of the jpegPhoto attribute in binary representation.

```
compare cn=zapf,ou=asw,o=my-company \  
-attr jpegPhoto_FILE=pict.jpg
```

3. Modify the value of the jpegPhoto attribute. The file "pict\_old.jpg" contains the old value and the file "pict\_new.jpg" contains the new value.

```
modify cn=Mayer,ou=sales,o=my-company \  
-changeattr jpegPhoto_FILE=pict_old.jpg \  
jpegPhoto_FILE=pict_new.jpg
```

4. Create an entry. The file "cert.cer" contains the ASN.1 encoded value of the user certificate attribute.

```
create cn=Huber,ou=sales,o=my-company -attr \  
{objectClass=person;organizationalPerson;strongAuthenticationUser}  
\  
sn=Huber {userCertificate;binary_FILE=cert.cer}
```

5. Show an entry. The ASN.1 encoded value of the user certificate attribute is written to the file "huber.cer".

```
show cn=Huber,ou=sales,o=my-company -attr \  
{userCertificate;binary_FILE=huber.cer}
```

6. Search several objects. The result contains eight objects two of them containing multiple attribute values.

```
search o=my-company -subtree -attr jpegPhoto_FILE=photo.jpg
```

The output is as follows:

```
{ou=Sales,o=my-company}  
{{cn=Smith John,ou=Sales,o=my-company}}  
{cn=Mayer,ou=Sales,o=my-company}
```

```
{cn=Hohner,ou=Sales,o=my-company  
  {jpegPhoto_FILE=photo.jpg;photo.jpg.1;photo.jpg.2}}  
{cn=Richter,ou=Sales,o=my-company}  
{cn=Abele,ou=Sales,o=my-company  
  {jpegPhoto_FILE=photo.jpg.3;photo.jpg.4}}  
{cn=Reichel,ou=Sales,o=my-company}  
{cn=hohner2,ou=Sales,o=my-company}
```

## 2.3. Distinguished Names

A distinguished name consists of a list of one or more relative distinguished names (RDNs), separated by a comma (,). The list of relative distinguished names starts with the last namepart and ends with the first namepart under the root entry. For example:

**cn=schmid+ou=ap11,ou=dap11,o=my-company**

Each RDN consists of one or more naming attributes in the following format:

*type=value[+type=value]...*

where *type* is an LDAP name or an OID that corresponds to a naming attribute type and *value* is the string representation that corresponds to the attribute syntax assigned to the attribute type. The plus sign (+) is used to separate multiple AVAs within one RDN. For example:

**c=de**

or

**2.5.6.2=de**

or

**ou=dap11+l=munich**

The maximum length of an ASN.1 encoded RDN is 470 bytes. The value length of the naming attribute is affected by

- the length of the object identifier
- the number of 2-byte UTF8 characters contained in the attribute value
- the number of attribute value assertions (AVAs) the RDN consists of.

When the name of the root entry is specified the slash (/) must be used.

The following special characters must be escaped in distinguished names:

- a SPACE or # character at the beginning of a string

- a SPACE character at the end of a string
- the characters comma (,), plus sign (+), quotation mark ("), backslash (\), lower than (<), greater than (>), or semicolon (;).

To escape these characters prefix the character by a backslash, for example **0=Sue\, Grabbit and Runn.** (See RFC 2253 for details.)

## 2.4. Search Filters

Use search filter expressions to specify a filter in a **dirxcp search** operation. A search filter is composed of one or more simple attributes, structured attributes, or distinguished name strings, and search filter operators. The following virtual attributes must not be used in search filters:

- numSubordinates
- numAllSubordinates
- structuralObjectClass

If the listed virtual attributes are used, an **Unwilling to Perform** error will be returned.

Recall from section **Tcl Command Language** in chapter **DirX Directory Commands** that you must enclose the filter expression in curly brackets ({} ) or quotation marks ("" ) if attribute values contain a space character. Specify a search filter in the following format:

*([logical\_operator](type matching\_operator value) [(type matching\_operator value) ...])*

where

### **logical\_operator**

is one of the following operators:

| Operator | Meaning                                     |
|----------|---------------------------------------------|
| &        | To “logically AND” two specified conditions |
|          | To “logically OR” two specified conditions  |
| !        | To “logically NEGATE” a specified condition |

### **type**

specifies an LDAP name or an object identifier.

### **matching\_operator**

is one of the following operators:

| Operator | Meaning                      |
|----------|------------------------------|
| =        | To specify equality          |
| ~=       | To specify phonetic matching |

| Operator | Meaning                                                             |
|----------|---------------------------------------------------------------------|
| >=       | To match values that are greater than or equal to a specified value |
| <=       | To match values that are less than or equal to a specified value    |

### **value**

specifies the attribute value in LDAP syntax. \* is used to specify substrings or to check for the presence of an attribute.

No SPACE character is permitted between *type* and *matching\_operator* and *matching\_operator* and *value*.

## **2.4.1. Search Filter Expression Example**

The following sample search filter string

```
(&(cn~=schmid) \
|(objectClass=organizationalPerson) \
(objectClass=residentialPerson)) \
(!(sn=ronnie)))
```

directs **dirxc** to search for names that meet all the following criteria:

- Have an object class attribute value of **Organizational-Person** or **Residential-Person**
- Have a **Common-Name** attribute value that approximately matches **schmid**
- Do not have a Surname attribute value of **ronnie**.

The following search filter string test for the presence of the **Common-Name** attribute type:

```
(&(c=de)(cn=*))
```

## **2.4.2. Certificates in Search Filters**

In order to support searching for attributes with the certificate syntax, for example userCertificates and caCertificates, DirX Directory supports the following two matching rules:

- The certificateExactMatching rule (OID **2.5.13.34**)
- The certificateMatching rule (OID **2.5.13.35**)

For both of these matching rules, LDAP clients can use the extensible filter feature to express such a search filter. (See RFC 4511 the LDAP protocol for details about the extensible filter feature.)

The components of a certificate that can be used in search filters are defined in X.509 in the

following CertificateAssertion ASN.1 definition:

```
CertificateAssertion :: SEQUENCE {
    serialNumber          [0] CertificateSerialNumber OPTIONAL,
    issuer                [1] Name                      OPTIONAL,
    subjectKeyIdentifier  [2] SubjectKeyIdentifier      OPTIONAL,
    authorityKeyIdentifier [3] AuthorityKeyIdentifier    OPTIONAL,
    certificateValid       [4] Time                     OPTIONAL,
    privateKeyValid       [5] GeneralizedTime          OPTIONAL,
    subjectPublicKeyAlgID [6] OBJECT IDENTIFIER         OPTIONAL,
    keyUsage               [7] KeyUsage                 OPTIONAL,
    subjectAltName         [8] AltNameType              OPTIONAL,
    policy                 [9] CertPolicySet            OPTIONAL,
    pathToName             [10] Name                    OPTIONAL,
    certificateInvalid     [20] Time                    OPTIONAL,
    certificateExpired     [21] Time                    OPTIONAL,
    certificateNotYetValid [22] Time                    OPTIONAL }
```

The keywords of the components, for example **serialNumber**, must be specified case-sensitive.

See section **Attribute Table** in chapter **DirX Directory Default DSA Schema** in the *DirX Directory Administration Reference* for attributes with Certificate syntax.

#### 2.4.2.1. The certificateExactMatching Rule

This rule specifies that two certificates are equal, if their **serialNumber** and **issuer** components are equal. These two components are mandatory in the matching rule assertion.

The syntax of an LDAP search filter for this rule is as follows:

1. `(certificateAttribute:2.5.13.34:={serialNumber serial_number,issuer rdnSequence:"distinguished_name"})`

or

2. `(certificateAttribute:2.5.13.34:=serial_number$distinguished_name)`

The *certificateAttribute* must be the LDAP name of an attribute with the syntax Certificate. The *serial\_number* must be specified as a decimal integer.

In the **dirxcp** command, the **\$** character used as separator must be escaped with a \ (backslash) character.

In order to use the attribute index for the exact certificate match there must be an INITIAL index for the certificate attribute.

#### 2.4.2.1.1. Example

Format 1:

```
dirxcp> search o=my-company -sub
        -filt {(usercertificate:2.5.13.34:={serialNumber 30,
            issuer rdnSequence:"cn=admin,o=pqgrupmann02"}}}
```

Format 2:

```
dirxcp> search o=my-company -sub
        -filt
(UserCertificate:2.5.13.34:=30\${cn=admin,o=pqgrupmann02})
```

#### 2.4.2.2. The certificateMatching Rule

This rule specifies a set of optional certificate components in the extended LDAP search filter. DirX Directory supports the components **keyUsage**, **issuer**, **certificateValid**, **certificateExpired** and **certificateNotYetValid**. According to this matching rule, a certificate matches the presented filter if the certificate contains all requested values.

In order to use the attribute index for the certificate matching rule there must be a CONTAINS index for the attribute.

The following sections provide the filter syntax for each component and how they can be combined.

##### 2.4.2.2.1. Searching for Certificates with a specific keyUsage:

The syntax of an LDAP search filter for this rule is as follows:

```
(certificateAttribute:2.5.13.35:={keyUsage {keyUsage_values}})
```

The *certificateAttribute* must be the LDAP name of an attribute with the syntax Certificate. A SPACE character must separate multiple *keyUsage\_value*. All specified values must be present in the key usage of a stored certificate to match the filter condition.

Valid *keyUsage\_values* (strings defined by X.509) are:

**digitalSignature**, **nonRepudiation**, **keyEncipherment**, **dataEncipherment**, **keyAgreement**, **keyCertSign**, **cRLSign**, **encipherOnly**, **decipherOnly**

##### 2.4.2.2.2. Example

In the following example the search operation will return all entries below o=my-company that userCertificate attribute must have either no keyUsage extension or must have a keyUsage extension with the values **keyEncipherment** and **digitalSignature**. (Additional values may be set in the keyUsage extension).

```
dirxcp> search o=my-company -sub
        -filt {(usercertificate:2.5.13.35:=
                {keyUsage {keyEncipherment digitalSignature}})}
```

#### 2.4.2.2.3. Searching for Certificates issued by a specific CA (issuer):

The syntax of an LDAP search filter for this rule is as follows:

*(certificateAttribute:2.5.13.35:={issuer {dn}})*

The *certificateAttribute* must be the LDAP name of an attribute with the syntax Certificate. The *dn* is the distinguished name of the issuer in LDAP format. (See **Distinguished Names** above.)

#### 2.4.2.2.4. Example

Usually the issuer component of a user certificate contains the distinguished name of the issuing CA. In the following example the search operation will return all entries below *o=my-company* that userCertificate was issued by the CA with the distinguished name *cn=trustedCA,o=CA,c=de*:

```
dirxcp> search o=my-company -sub
        -filt {(usercertificate:2.5.13.35:=
                {issuer {cn=trustedCA,o=CA,c=de}})}
```

#### 2.4.2.2.5. Searching for Certificates that are valid (certificateValid):

The syntax of an LDAP search filter for certificates that are valid for a specific time or time-period is as follows:

*(certificateAttribute:2.5.13.35:={certificateValid {time}})*

The *certificateAttribute* must be the LDAP name of an attribute with the syntax Certificate. Specify *time* in one of the following formats:

- An absolute Zulu timestamp in GeneralizedTime format, for example **20130703000000Z**. Specifying milliseconds is prohibited.
- An offset to the current time in the format

**now**{+|-}*integer*{**d**|**m**|**y**}

where **now** represents the current time, **+** represents the time-period after the current time, **-** represents the time-period before the current time, the *integer* the number of days, months or years, and **d** represents days, **m** months and **y** years; for example **now+3m** means a time-period of three month after the current time.



#### 2.4.2.2.6. Examples

In the following example the search operation will return all entries below o=my-company that userCertificate is valid on 3 July 2013:

```
dirxcp> search o=my-company -sub
        -filt {(usercertificate:2.5.13.35:=
                { certificateValid {20130703000000Z}})}
```

In the following example the search operation will return all entries below o=my-company that userCertificate is valid for three months from now on:

```
dirxcp> search o=my-company -sub
        -filt {(usercertificate:2.5.13.35:=
                {certificateValid {now+3m}})}
```

#### 2.4.2.2.7. Searching for Certificates that expires at a specific time (certificateExpired):

The syntax of an LDAP search filter for certificates that expires at a specific time is as follows:

*(certificateAttribute:2.5.13.35:={certificateExpired {time}})*

The *certificateAttribute* must be the LDAP name of an attribute with the syntax Certificate. See **Searching for Certificates that are valid (certificateValid)**: above about specifying *time*.

A NOT filter for **certificateValid** and a filter for **certificateExpired** may return different search results, for example

```
{ (! (usercertificate:2.5.13.35:={ certificateValid {
now }}))}
```



and

```
{ (usercertificate:2.5.13.35:={ certificateExpired { now
}})}
```

return different results for an entry that contains one expired and one valid certificate. The first filter will not return a result list containing this entry but the result list for the second filter will contain the entry.

#### 2.4.2.2.8. Example

In the following example the search operation will return all entries below o=my-company that userCertificate will expire in one month from now on:

```
dirxcp> search o=my-company -sub
        -filt {(usercertificate:2.5.13.35:=
                {certificateExpired {now+1m}})}
```

#### 2.4.2.2.9. Searching for Certificates that are not yet valid (certificateNotYetValid):

The syntax of an LDAP search filter for certificates that are not yet valid for a specific time is as follows:

*(certificateAttribute:2.5.13.35:={certificateNotYetValid {time}})*

The *certificateAttribute* must be the LDAP name of an attribute with the syntax Certificate. See **Searching for Certificates that are valid (certificateValid):** above about specifying *time*.

#### 2.4.2.2.10. Example

In the following example the search operation will return all entries below o=my-company that userCertificate is not valid before 1 January 2015:

```
dirxcp> search o=my-company -sub
        -filt {(usercertificate:2.5.13.35:=
                {certificateNotYetValid {20150101000000Z}})}
```

#### 2.4.2.2.11. Searching for Certificates with a combined extended LDAP filter:

When combining components in a single extended LDAP filter a comma (,) separates multiple components. All components are combined with the logically AND. The syntax is as follows:

*(certificateAttribute:2.5.13.35:={component\_keyword\_1 {component\_value\_1},  
component\_keyword\_2 {component\_value\_2},  
...,  
component\_keyword\_n {component\_value\_n}})*

The *certificateAttribute* must be the LDAP name of an attribute with the syntax Certificate. *component\_keyword* is one of the supported components keywords **keyUsage**, **issuer**, **certificateValid**, **certificateExpired** or **certificateNotYetValid** and *component\_value* the value of this component.

#### 2.4.2.2.12. Example

In the following example the search operation will return all entries below o=my-company

that `userCertificate` is expired and has either no `keyUsage` extension or that `keyUsage` extension has the values **keyCertSign** and **cRLSign**:

```
dirxcp> search o=my-company -sub
        -filt {(usercertificate:2.5.13.35:=
                {certificateExpired {now},
                keyUsage {keyCertSign cRLSign}})}
```

### 2.4.3. LDAP Matched Value Only Filter Control

Use the LDAP Matched Value Only Filter Control in search operation to further narrow the values returned in the search result. If an attribute holds many values - for example, the `userCertificate` attribute or an attribute containing hundreds of options - it may be desirable for the user to be able to selectively retrieve a subset of the values, specifically, those attribute values that match some user-defined selection criteria. Without using this control, a client must read all of the attribute's values and filter out the unwanted values, requiring the client to implement the matching rules. (The use of this control impacts performance, since the DSA normalizes and decodes each value of the selected multivalued attributes to be able to perform the value filtering.)

The filter string within this control is only used to narrow the attribute values to be returned by a search operation in a search result. The original search filter is first executed, producing an intermediate search result to which the filter in the control value is then applied. The control affects only those attribute types in the final result of the search, which appear in both the intermediate result from the original search filter *and* the filter string in the control value.

A search operation that carries the LDAP Matched Value Only Filter Control will do the following:

1. The Search Filter is first executed to determine which entries satisfy the search criteria. This step generates the filter result, which contains only the requested attributes with all attribute values.
2. For each attribute value of each entry contained in the filter result (prepared in step 1), the filter in the control value is evaluated:
  - If the attribute type of the filter result is not contained in the filter in the control value, the control has no effect: the attribute with all its values will be present in the final search result entry.
  - If the attribute type is contained in the filter in the control value, each value of this attribute type is checked against the filter in the control value. If the filter in the control value evaluates to FALSE, the given value will be removed from the final search result entry.

The value of this control is a filter string as described in the section **Search Filters** above with some further restrictions:

- If the values for more than one attribute type should be filtered, these different attribute

types must be put in an outermost OR filter.

- Otherwise, complex filter (AND, OR, NOT) can only contain the same attribute type.
- Extensible filter value matching is only supported for `certificateExactMatching` and `certificateMatching` rules with the same syntax and restrictions, as described in the section **Certificates in Search Filters**.

Therefore, the following filter strings are invalid examples in the control value:

```
(&(mail=*hotmail.com)(telephoneNumber=1234*))  
(!(|(mail=*hotmail.com)(telephoneNumber=1234*)))
```

and will be rejected by the server.

The following valid sample filter string in the control value

```
(|(mail=*hotmail.com)(mail=*gmail.com)(telephoneNumber=5*))
```

directs the server to filter further the values in the search result:

- All attribute values of attribute types other than `mail` or `telephoneNumber` will remain present on the output (same number of values, as without this control.)
- From the `mail` attribute, only the values ending with **hotmail.com** or **gmail.com** will be present on the output. (If no such matching value is found, also the attribute type will not be present on the output.)
- From the `telephoneNumber` attribute, only the values starting with **5** will be present on the output. (If no such matching value is found, also the attribute type will not be present on the output.)

The following sample filter string in the control value

```
(usercertificate:2.5.13.35:={issuer {cn=trustedCA,o=CA,c=de}})
```

directs the server to filter the `userCertificate` values, and return only those certificates that were issued by **{cn=trustedCA,o=CA,c=de}**.

Some attribute types are not supported in the filter in the control value; for example, attributes that are calculated at runtime, like `numSubordinates`.

The attributes occurring in the filter must have one of the following syntaxes:

- `DirectoryString` (including T61, Printable, BMP, Universal and UTF-8 String)
- `IA5 String`
- `Numeric String`

- Integer
- Boolean
- GeneralizedTime
- Certificate with matching rule certificateExactMatch
- Certificate with matching rule certificateMatch

## 2.5. Attribute Syntax

The attribute syntax of all attribute types is treated as UTF-8 strings for LDAP binds.

### 2.5.1. Undefined Types

To specify an attribute type that has not been assigned an LDAP name in the system schema, use the attribute type OID, for example, **1.2.325.67890.4.2**.

In the output, the **dirxcp** program returns a string in the form:

*oid=attribute\_value*

For example:

**1.2.325.67890.4.2=xyz**

which indicates that the attribute with the object identifier 1.2.325.67890.4.2 has a value of xyz. The value of the attribute type must be specified according to its attribute syntax.

## 2.6. String Representations for Simple Attribute Syntaxes

The following section describes the string representations for simple attribute syntaxes supported. All other simple attribute syntaxes not described in this section are treated as UTF-8 string.

### 2.6.1. Attribute Type Syntax

Specify an attribute type as an LDAP name (defined in the system schema) or the attribute type's object identifier (OID) in dotted notation (for example, 1.2.5.6).

### 2.6.2. Bit String Syntax

Specify a bit string as a sequence of 1's and 0's enclosed by the character ' and the character **B** appended (for example, '11110100100001001101101'**B**).

### 2.6.3. Boolean Syntax

Specify a boolean as the string **TRUE** or **FALSE**.

## 2.6.4. Object ID Syntax

Specify an OID as an LDAP name (defined in the system schema) or a dotted notation. For example, Organizational-Person could be specified as organizationalPerson or 2.5.6.7.

## 2.6.5. Generalized Time Syntax

The syntax is the same as in DAP style. (See section **Generalized Time Syntax** of chapter **DirX Directory String Representation for DAP Binds** for details.)

## 2.6.6. IA5 String Syntax

The syntax is the same as in DAP style. (See section **IA5 String Syntax** of chapter **DirX Directory String Representation for DAP Binds** for details.)

## 2.6.7. Integer String Syntax

The syntax is the same as in DAP style. (See section **Integer String Syntax** of chapter **DirX Directory String Representation for DAP Binds** for details.)

## 2.6.8. Numeric String Syntax

The syntax is the same as in DAP style. (See section **Numeric String Syntax** of chapter **DirX Directory String Representation for DAP Binds** for details.)

## 2.6.9. Preferred Delivery Method Syntax

Preferred-delivery-method syntax is a syntax for single-valued attributes that document the order of preference for message delivery methods. Specify this syntax in the following format:

*preferredDeliveryMethod=option [\$ option...]*

*option* is one or more of the following keywords that describe the delivery methods and the order of preference:

- **any**—Any method of delivery
- **mhs**—Message handling system delivery
- **physical**—Physical delivery
- **telex**—Telex delivery
- **teletex**—Teletex delivery
- **g3fax**—G3 FAX delivery
- **g4fax**—G4 FAX delivery
- **ia5**—IA5 terminal delivery
- **videotex**—Videotex delivery
- **telephone**—Telephone delivery

The keywords are specified diminishing order of preference, with the most preferred method first in the list. Separate multiple keywords with the string "\$". For example:

```
preferredDeliveryMethod=mhs $ teletex $ telephone
```

## 2.6.10. Printable String Syntax

The syntax is the same as in DAP style. (See section **Printable String Syntax** of chapter **DirX Directory String Representation for DAP Binds** for details.)

## 2.6.11. UTC Time Syntax

The syntax is the same as in DAP style. (See section **UTC Time Syntax** of chapter **DirX Directory String Representation for DAP Binds** for details.)

# 2.7. String Representations for Structured Attribute Syntaxes

This section describes the following structured attribute syntaxes supported by DirX Directory for LDAP binds.

- Syntaxes for schema attribute types
  - Attribute-Type-Description
  - Object-Class-Description
- Syntaxes for miscellaneous attribute types and subcomponents
  - Facsimile-Telephone-Number
  - Name-And-Optional-UID
  - Postal-Address
  - Teletex-Terminal-Identifier
  - Telex-Number



All attributes with a structure attribute syntax not described in this section must be specified in binary format for LDAP binds. (See section **Binary Attribute Values** above in this chapter for details.)

## 2.7.1. Attribute-Type-Description

An attribute syntax for schema attributes that specify attribute types. The attributeTypes (LAT) attribute is an example of such attributes in the default DSA schema.

### 2.7.1.1. Synopsis

*attributeTypes*=( *attribute\_identifier*

**NAME** [( ]'*attribute\_type\_name*' [... ]]  
**[DESC** '*attribute\_type\_description*' ]  
**[OBSOLETE** ]  
**[SUP** *derivation*]  
**[EQUALITY** *equality\_matching\_rule* ]  
**[ORDERING** *ordering\_matching\_rule* ]  
**[SUBSTR** *substrings\_matching\_rule* ]  
**[SYNTAX** *attribute\_syntax* [{*length*}]]  
**[SINGLE-VALUE** ]  
**[COLLECTIVE** ]  
**[NO-USER-MODIFICATION** ]  
**[USAGE** *usage* ])

### 2.7.1.2. Attribute Type

#### *attributeTypes*

The LDAP name or OID that corresponds to a structured attribute with the Attribute-Type-Description (ATTR\_TYPE\_DESCR) syntax. The LDAP Attribute-Types (LAT, *attributeTypes*) operational attribute, which specifies the attribute types used within the schema, has the ATTR\_TYPE\_DESCR attribute syntax. The LAT attribute is multivalued; each value describes one attribute type. The information held in this attribute should be complete and in accordance with the registered definition of each attribute type. The LAT attribute also provides the LDAP names of the attribute types.

### 2.7.1.3. Components

#### *attribute\_idenfier*

An object identifier (OID) that corresponds to an attribute type.

#### **NAME** '*attribute\_type\_name*'

A string (of up to 1024 characters long) that provides the LDAP name(s) for the attribute type. The values are enclosed by single quotation marks (') and separated by a whitespace character. A list of LDAP names is enclosed by brackets ([ ... ]). For LDAP names only the following characters are permitted: A to Z, a to z (case ignore), 0 to 9, and - (hyphen). LDAP names must start with a letter. A maximum of five LDAP names can be specified.

#### **DESC** '*attribute\_type\_description*'

A UTF-8 string (of up to 1024 characters long) that describes the attribute type. The value is enclosed by single quotation marks (').

#### **OBSOLETE**

The keyword **OBSOLETE** indicates that the attribute type is no longer supported (but its characteristics are maintained). If an attribute type is deleted it is set to **OBSOLETE**, that is this attribute cannot be added to an entry or modified. (The error **Unwilling to Perform** is returned.) The values of obsolete attributes are returned by search and read operations. The default is that the specified attribute type is supported and the keyword **OBSOLETE** is omitted.



### ***SUP derivation***

Specifies the attribute type LDAP name or OID that corresponds to the attribute type of which this attribute is a subtype. This component is used only for attributes defined with a supertype.

### ***EQUALITY equality\_matching\_rule***

Specifies the LDAP name or OID of an equality matching rule. This is an optional component, but it should be specified for attributes defined with equality matching rules. If additionally an ordering matching rule and / or a substrings matching rule is specified then all specified matching rules must have the same case sensitivity.

### ***ORDERING ordering\_matching\_rule***

Specifies the LDAP name or OID of an ordering matching rule. This is an optional component, but it should be specified for attributes defined with ordering matching rules.

### ***SUBSTR substrings\_matching\_rule***

Specifies the LDAP name or OID of a substrings matching rule. This is an optional component, but it should be specified for attributes defined with substrings matching rules.

### ***SYNTAX attribute\_syntax [{ length }]***

Specifies the OID of the attribute type syntax for use with LDAP, and an optional indication of the maximum length *length* of a value of this attribute.

### ***SINGLE-VALUE***

The keyword **SINGLE-VALUE** indicates that the attribute type is single-valued. The default is that the specified attribute type is multi-valued and the keyword **SINGLE-VALUE** is omitted.

### ***COLLECTIVE***

The keyword **COLLECTIVE** indicates that the attribute type is a collective attribute. The default is that the specified attribute type is not collective and the keyword **COLLECTIVE** is omitted.

### ***NO-USER-MODIFICATION***

The keyword **NO-USER-MODIFICATION** indicates that the attribute type is not modifiable by users. The default is that the specified attribute type is user modifiable and the keyword **NO-USER-MODIFICATION** is omitted.

### ***USAGE usage***

Specifies how the attribute is to be used. This value is one of the following keywords:

- **userApplications** - For normal user attributes
- **directoryOperation** - For operational attributes used by the DSA as part of non-distributed operations (for example, timestamp attributes, access control attributes)
- **distributedOperation** - For operational attributes used by several DSAs as part of distributed operations (e.g., knowledge-reference attributes)

- **dSAOperation** - For operational attributes that are used purely locally to this DSA

The default value is **userApplications**.

#### 2.7.1.4. Description

The LDAP Attribute-Types attribute is provided to permit an LDAP server to publish the static details of the attributes that it supports within its schema. It is also used to manage the schema. Attribute type descriptions are represented with the ATTR\_TYPE\_DESCR attribute syntax.

The components of this syntax are separated by a whitespace character.

For each attribute type either the **SUP** or the **SYNTAX** component must be specified.

The sequence of the components must be provided as specified in the synopsis section above.

Only creation and restricted modifications are allowed for attribute types. Deleting attribute types is prohibited. To modify the values of an existing attribute type the attribute type must be deleted and then added with modified values in a single modify operation. When specifying an attribute type for the **-removeattr** option only the *attribute\_identifier* component must be specified. (See the **dirxcp obj modify** operation in the *DirX Directory Administration Reference* for details.)

Only the values of the following components can be modified:

- **COLLECTIVE**
- **DESCRIPTION**
- **NAME**
- **NO-USER-MODIFICATION**
- **ORDERING** - If an equality matching rule (component **EQUALITY**) already exists in the old attribute type description an ordering matching rule can be specified.
- **SINGLE-VALUE** - This component can be omitted in the new specification. However it cannot be specified for an existing multi value attribute type.

All other already existing component values must be specified unchanged.

When creating a new attribute type that should also be used in the **-filter** option of search requests or as naming attribute an equality matching rule (component **EQUALITY**) must be specified.

#### 2.7.1.5. Examples

```
attributeTypes=
....
( 2.5.4.20 NAME 'MYtelephoneNumber' EQUALITY telephoneNumberMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.44{32} )
```

....

## 2.7.2. Object-Class-Description

An attribute syntax for schema attributes that specify object classes. The `objectClasses` (LOC) attribute is an example of such attributes in the default DSA schema.

### 2.7.2.1. Synopsis

```
objectClasses=( object_class_identifier
  NAME 'object_class_name'
  [DESC 'object_class_description' ]
  [OBSOLETE ]
  [SUP [( superior_object_class[$ ... ])]
  [kind ]
  [MUST [( attribute_type [$ ... ])]
  [MAY [( attribute_type [$ ... ])] )
```

### 2.7.2.2. Attribute Type

#### *objectClasses*

The LDAP name or OID that corresponds to a structured attribute with the Object-Class-Description (OBJECT\_CLASS\_DESCR) syntax. The LDAP Object-Classes (LOC, `objectClasses`) operational attribute is a multi-valued attribute of the schema used to describe which object classes are supported by the LDAP server. Each value describes one object class. The LOC attribute provides the LDAP names of the object classes.

The information specified for this attribute should be complete and in accordance with the registered definition of each object class.

### 2.7.2.3. Components

#### *object\_class\_identifier*

An OID that corresponds to an object class.

#### **NAME** '*object\_class\_name*'

A string (of up to 1024 characters long) that corresponds to LDAP name given to the object class. Only one LDAP name is provided. It is enclosed by single quotation marks ('). For LDAP names only the following characters are permitted: A to Z, a to z (case ignore), 0 to 9, and - (hyphen). The LDAP name must start with a letter.

#### **DESC** '*object\_class\_description*'

A UTF-8 string (of up to 1024 characters long) that describes the object class. The value is enclosed by single quotation marks (').

#### **OBSOLETE**

The keyword **OBSOLETE** indicates that the object class is no longer supported (but its characteristics are maintained). If an object class is deleted it is set to **OBSOLETE**, that is

this entries of this object class cannot be added or modified. (The error **Unwilling to Perform** is returned.) Entries of obsolete object classes are returned by search and read operations. The default is that the specified object class is supported and the keyword **OBSOLETE** is omitted.

#### ***SUP derivation***

Specifies the LDAP names or OIDs that corresponds to the object classes (if any) that are the superclass for this object lass. A list of object classes is enclosed by brackets and the object classes are separated by a **\$** character.

#### ***kind***

Specifies the kind of object-class that the object class registration specifies. The value is one of the following keywords:

- **STRUCTURAL** - Represents a real-world object (for example, device, organization) that is concrete enough to have a place in the DIT
- **AUXILIARY** - Descriptive of real-world objects (usually being applicable to more than one)
- **ABSTRACT** - Represents an abstraction of real-world objects which does not exist in it own right

#### ***MUST attribute\_types***

Specifies one or more LDAP names or OIDs that correspond to attributes that are registered as mandatory for entries of this object class. A list of attributes is enclosed by brackets and the attribute types are separated by a **\$** character.

#### ***MAY attribute\_types***

Specifies one or more LDAP names or OIDs that correspond to attributes that are registered as optional for entries of this object class. A list of attributes is enclosed by brackets and the attribute types are separated by a **\$** character.

### **2.7.2.4. Description**

The Object-Class-Description permits a LDAP server to publish the static details of the object-classes it supports. It is also used to manage the schema. Object class descriptions are represented with the OBJECT\_CLASS\_DESCR schema attribute syntax.

The components of this syntax are separated by a whitespace character.

The sequence of the components must be provided as specified in the synopsis section above.

Only creation and restricted modifications are allowed for object classes. Deleting an object class is prohibited. To modify the values of an existing object class the object class must be deleted and then added with modified values in a single operation. When specifying an object class for the **-removeattr** option only the *object\_class\_identifier* component must be specified. (See the **dirxcp obj modify** operation in the *DirX Directory Administration Reference* for details.)

Only the values of the following components can be modified:

- **DESCRIPTION**

- **MAY** - Only additional attribute types can be added. The already existing attribute types must also be specified. All these attribute type specifications must exist.
- **MUST** - Attribute types can be moved to the optional attribute types (component **MAY**). This modification is prohibited for the object class **top**.

- **NAME**

All other already existing component values must be specified unchanged.

#### 2.7.2.5. Examples

```
objectClasses=
....
( 2.5.6.16 NAME 'MYcertificationAuthority' SUP top AUXILIARY MUST (
  cACertificate $ certificateRevocationList $ authorityRevocationList )
  MAY crossCertificatePair )
....
```

### 2.7.3. Facsimile-Telephone-Number

An attribute syntax for attributes that specify Facsimile (FAX) numbers.

#### 2.7.3.1. Synopsis

*FacsimileTelephoneNumber=phone\_number*  
[*\$fax\_parameters*]

#### 2.7.3.2. Attribute Type

##### ***FacsimileTelephoneNumber***

The LDAP name or OID that corresponds to a structured attribute with the Facsimile-Telephone-Number (FAX\_TELEPHONE\_NO) syntax. For example, *facsimileTelephoneNumber* is such an attribute.

#### 2.7.3.3. Components

##### ***phone\_number***

The fax number in Printable string format (of up to 32 characters long).

##### ***fax\_parameters***

The settings for the G3 Fax parameters. Note that setting G3 parameters is hardly ever required. Specify *fax\_parameters* in the following format:

*fax\_parameter*[\**\$fax\_parameters*]

where *fax\_parameter* is one of the following identifiers:

- **twoDimensional**
- **fineResolution**
- **unlimitedLength**
- **b4Length**
- **a3Width**
- **uncompressed**

#### 2.7.3.4. Description

FAX numbers are represented with the Facsimile-Telephone-Number attribute syntax. The number consists the country code and number in Printable string format. You can also set G3 non-basic parameters.

The components of this attribute syntax are separated by a **\$** character.

#### 2.7.3.5. Example

```
facsimileTelephoneNumber={plus}49 89 12345
```

### 2.7.4. Name-And-Optional-UID

An attribute syntax for attributes that identify objects by a distinguished name and an optional identifier that can remove ambiguity from names that have been re-used.

#### 2.7.4.1. Synopsis

*NameAndOptionalUID=distinguished\_name*  
 [#BIT\_string]

#### 2.7.4.2. Attribute Type

##### ***NameAndOptionalUID***

The LDAP name or OID that corresponds to a structured attribute with the Name-And-Optional-UID syntax. For example, uniqueMember is such an attribute. The uniqueMember attribute can be used in objects which represent lists of names.

#### 2.7.4.3. Components

##### ***distinguished\_name***

The distinguished name of the object. (See section **Distinguished Name** above in this chapter for details.)

Although the **#** character is used as separator for the components of this syntax and it may occur in a string representation of a *distinguished\_name*, no additional special quoting is done.

### ***BIT\_string***

A bit string that uniquely identifies the object. The bit string can be used when an object is removed from the Directory, and another object is subsequently given the same Directory name. The bit string must be chosen for uniqueness, such as a time-stamp (for example). (See section **Bit String Syntax** above in this chapter for details.)

#### **2.7.4.4. Description**

The identity of objects can be represented with the *Name-And-Optional-UID* attribute syntax. The attribute consists of the object distinguished name and an optional bit string that distinguishes between objects with the same distinguished name. The association between a user of the Directory and a name when the UID is present can be established by strong authentication using Version 2 (or later) certificates.

The components of this attribute syntax are separated by a # character.

#### **2.7.4.5. Examples**

```
uniqueMember=o=sni,c=DE#'100'B
```

### **2.7.5. Postal-Address**

A structured attribute syntax for postal addresses.

#### **2.7.5.1. Synopsis**

```
PostalAddress=postal-address-string1  
[$postal-address-string2]  
[$postal-address-string3]  
[$postal-address-string4]  
[$postal-address-string5]  
[$postal-address-string6]
```

#### **2.7.5.2. Attribute Type**

##### ***PostalAddress***

The abbreviation that corresponds to a structured attribute with the Postal-Address (POSTAL\_ADDR) syntax. For example, postalAddress is such an attribute.

#### **2.7.5.3. Components**

##### ***postal-address-stringn***

A UTF-8 string (of up to 30 characters long) that provides a line of a postal address.

Backslashes (\) and \$ characters, if they occur in a component, are escaped by using an additional backslash, for example the string A\\$ represents the value A\$ in a component.

#### 2.7.5.4. Description

Postal addresses are represented with the Postal-Address attribute syntax. The attribute consists of up to 6 address lines in UTF-8 string format. Each address line is limited to 30 characters in (in accordance with ITU-T recommendation F.401). One address line is required; the remaining 5 are optional.

The components of this attribute syntax are separated by a \$ character.

#### 2.7.5.5. Example

```
postalAddress=My-Company AG$Sales Dpt$Einstein-Ring 4$D-81789  
Munich$Germany
```

### 2.7.6. Teletex-Terminal-Identifier

An attribute syntax for attributes that specify a Teletex terminal.

#### 2.7.6.1. Synopsis

```
TeletexTerminalIdentifier=teletex-terminal  
[$teletex-non-basic-parameters]
```

#### 2.7.6.2. Attribute Type

##### ***TeletexTerminalIdentifier***

The LDAP name or OID that corresponds to a structured attribute with the Teletex-Terminal-Identifier syntax. For example, `teletexTerminalIdentifier` is such an attribute.

#### 2.7.6.3. Components

##### ***teletex-terminal***

A string (of up to 1024 characters long) that identifies the terminal. Specify the *teletex-terminal* parameter using the Printable String syntax. See **Simple Attribute Syntax** for a description of Printable String Syntax representation.

##### ***teletex-non-basic-parameters***

A component that sets non-basic parameters for various teletex options. Specify *teletex-non-basic-parameters* in the following format:

```
[$control:control_character_sets]  
[$graphic:graphic_character_sets]  
[$misc:miscellaneous_capabilities]  
[$page:page_formats]  
[$private:private_use]
```

where:



**control:control\_character\_sets**

A string that defines the control character sets to use.

**graphic:graphic\_character\_sets**

A string that defines the graphic character sets to use.

**misc:miscellaneous\_capabilities**

A string that defines miscellaneous capabilities to use.

**page:page\_formats**

A string that defines the page formats to use.

**private:private\_use**

A string that defines user-defined Teletex parameters.

#### 2.7.6.4. Description

Teletex terminals and their parameters are represented by the Teletex-Terminal-Identifier attribute syntax. The information consists of a printable string that identifies the terminal and optional non-basic, advanced parameters that control terminal parameters such as character sets, page formats and other capabilities.

The components of this attribute syntax are separated by a \$ character.

#### 2.7.6.5. Examples

```
{teletexTerminalIdentifier=My Company AG Teletex center$page:letter}
```

### 2.7.7. Telex-Number

An attribute syntax for attributes that specify Telex numbers.

#### 2.7.7.1. Synopsis

```
TelexNumber=telex_number  
[$country_code]  
[$answer_back]}
```

#### 2.7.7.2. Attribute Type

**TelexNumber**

The LDAP name or OID that corresponds to a structured attribute with the Telex-Number (TELEX\_NO) syntax. For example, telexNumber is such an attribute.

#### 2.7.7.3. Components

**telex\_number**

The telex number in Printable string format (of up to 14 characters long).

**country\_code**

The country code in Printable string format (of up to 4 characters long).

**answer\_back**

The short textual string (of up to 8 characters long), in Printable string format, with which the telex station responds when required to indicate its identity. (For example, when the telex station is connected to.)

**2.7.7.4. Description**

Telex numbers are represented with the Telex-Number attribute syntax. The number consists of the country code, Telex number, and answer-back code.

The components of this attribute syntax are separated by a **\$** character.

**2.7.7.5. Example**

```
{telexNumber=24344$046$GAMEX B}
```

## 3. DirX Directory Attributes

This chapter describes the DirX Directory attributes defined in the DirX Directory default DSA schema. (See the chapter titled **DirX Directory Default DSA Schema** in the *DirX Directory Administration Reference* for details.) For each attribute it provides:

- A short description.
- The abbreviation that can be used in **dirxadm** and in **dirxcp** for DAP binds instead of object identifiers.
- The LDAP name(s) that can be used in **dirxcp** for LDAP binds instead of object identifiers.
- The syntax for DAP protocol.
- The syntax for LDAPv3 protocol.  
Note: When this subsection is omitted, the LDAPv3 syntax is UTF-8 string.
- An example for DAP binds.
- An example for LDAPv3 binds.

The LDAP name(s), the syntax, and an example for LDAPv3 binds are only provided for attributes that can be administered with the LDAP protocol. Only X.500 user attributes can be administered via LDAP.

For a detailed description of the attribute syntaxes, see the chapters titled **DirX Directory String Representation for DAP Binds** and **DirX Directory String Representation for LDAP Binds**.

### 3.1. X.500 User Application Attributes

User application attributes are attributes representing user information.

#### 3.1.1. Naming Attributes

The following attributes are used in DirX Directory to identify the entries. Note that naming attributes may also be used as normal attributes for an object.

##### 3.1.1.1. Common Name

The Common Name attribute specifies a name by which the object is commonly known and conforms to the naming conventions of the country or culture with which it is associated.

An attribute value for Common Name is chosen either by the person or organization it describes or by the organization responsible for the object it describes for devices and application entities.

##### 3.1.1.1.1. Abbreviation

CN

#### 3.1.1.1.2. LDAP Name(s)

cn, commonName

#### 3.1.1.1.3. Syntax (DAP)

Directory-String {64}

#### 3.1.1.1.4. Example (DAP)

```
{CN=John Smith}  
{CN=High Speed Modem}  
CN=Mayer
```

#### 3.1.1.1.5. Example (LDAP)

```
{cn=John Smith}  
{cn=High Speed Modem}  
commonName=Mayer
```

### 3.1.1.2. Country Name

The Country Name attribute specifies a country. It identifies the country in which the named object is located or with which it is associated.

It is not checked whether the specified value is contained in the list of valid country codes.

#### 3.1.1.2.1. Abbreviation

C

#### 3.1.1.2.2. LDAP Name(s)

c, countryName

#### 3.1.1.2.3. Syntax (DAP)

Country-Name (Only ISO/IEC IS3166 codes are valid values, see Appendix **Country Codes**.)

#### 3.1.1.2.4. Syntax (LDAP)

Printable String (Size (2)) (LDAP style)

#### 3.1.1.2.5. Example (DAP)

```
C=DE
```

## Example (LDAP)

```
c=DE  
countryName=UK
```

### 3.1.1.3. Locality Name

The Locality Name attribute specifies a locality (for example, a city). It identifies a geographical area or locality in which the named object is located or with which it is associated.

#### 3.1.1.3.1. Abbreviation

L

#### 3.1.1.3.2. LDAP Name(s)

l (character: lower l), localityName

#### 3.1.1.3.3. Syntax (DAP)

Directory-String {128}

#### 3.1.1.3.4. Example (DAP)

```
L=Munich
```

## Example (LDAP)

```
l=Munich  
localityName=Berlin
```

### 3.1.1.4. Organization Name

The Organization Name attribute specifies an organization.

An attribute value for Organization Name is chosen by the organization.

#### 3.1.1.4.1. Abbreviation

O

#### 3.1.1.4.2. LDAP Name(s)

o, organizationName

#### 3.1.1.4.3. Syntax (DAP)

Directory-String {64}

#### 3.1.1.4.4. Example (DAP)

```
{O=German Telecom}  
O=My-Company
```

#### 3.1.1.4.5. Example (LDAP)

```
{o=German Telecom}  
organizationName=My-Company
```

### 3.1.1.5. Organizational Unit Name

The Organizational Unit Name attribute specifies an organizational unit.

The designated organizational unit is a part of an organization designated by an Organization Name attribute. An Organizational Unit Name attribute must be associated with an Organization Name attribute.

An attribute value for Organizational Unit Name is chosen by the organization of which it is a part.

#### 3.1.1.5.1. Abbreviation

OU

#### 3.1.1.5.2. LDAP Name(s)

ou, organizationalUnitName

#### 3.1.1.5.3. Syntax (DAP)

Directory-String {64}

#### 3.1.1.5.4. Example (DAP)

```
{OU=Marketing Division}
```

#### 3.1.1.5.5. Example (LDAP)

```
{ou=Marketing Division}  
organizationalUnitName=Development
```

### 3.1.1.6. State or Province Name

The State or Province Name attribute specifies a state or province. It identifies the geographical subdivision in which the named object is located or with which it is associated.

#### 3.1.1.6.1. Abbreviation

SPN

#### 3.1.1.6.2. LDAP Name(s)

st, stateOrProvinceName

#### 3.1.1.6.3. Syntax (DAP)

Directory-String {128}

#### 3.1.1.6.4. Example (DAP)

```
SPN=Florida
```

#### Example (LDAP)

```
st=Florida  
{stateOrProvinceName=Freistaat Bayern}
```

### 3.1.1.7. Street Address

The Street Address attribute specifies a site for the local distribution and physical delivery in a postal address (that is, the street name, place, avenue, and the house number).

#### 3.1.1.7.1. Abbreviation

STA

#### 3.1.1.7.2. LDAP Name(s)

street, streetAddress

#### 3.1.1.7.3. Syntax (DAP)

Directory-String {128}

#### 3.1.1.7.4. Example (DAP)

```
{STA=Otto-Hahn-Ring 6}
```

#### 3.1.1.7.5. Example (LDAP)

```
{street=Otto-Hahn-Ring 6}
```

### 3.1.2. Names in General

#### 3.1.2.1. DMD Name

The DMD Name attribute specifies a Directory Management Domain (DMD). When used as a component of a directory name, it identifies a DMD which manages the named object.

##### 3.1.2.1.1. Abbreviation

DMD

##### 3.1.2.1.2. LDAP Name(s)

dmdName

##### 3.1.2.1.3. Syntax (DAP)

Directory-String {32768}

##### 3.1.2.1.4. Example (DAP)

```
DMD=dmd-1
```

##### 3.1.2.1.5. Example (LDAP)

```
dmdName=dmd-1
```

#### 3.1.2.2. DN Qualifier

The DN Qualifier attribute specifies disambiguating information to add to the relative distinguished name of an entry. It is used for entries held in multiple DSAs. Its value is the same in a given DSA and all the entries to which this information has been added.

##### 3.1.2.2.1. Abbreviation

DNQ

##### 3.1.2.2.2. LDAP Name(s)

dnQualifier



#### 3.1.2.2.3. Syntax (DAP)

Printable-String

#### 3.1.2.2.4. Syntax (LDAP)

Printable-String (LDAP style)

#### 3.1.2.2.5. Example (DAP)

```
{DNQ=the older}  
{DNQ=of My-Company-DSA}
```

#### 3.1.2.2.6. Example (LDAP)

```
{dnQualifier=the older}
```

### 3.1.2.3. Generation Qualifier

The Generation Qualifier attribute contains a string, which is used to provide generation information.

#### 3.1.2.3.1. Abbreviation

GQ

#### 3.1.2.3.2. LDAP Name(s)

generationQualifier

#### 3.1.2.3.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.2.3.4. Example (DAP)

```
GQ=sen .  
GQ=II
```

#### Example (LDAP)

```
generationQualifier=sen .  
generationQualifier=II
```

#### 3.1.2.4. Given Name

The Given Name attribute specifies the first name of an individual.

##### 3.1.2.4.1. Abbreviation

GN

##### 3.1.2.4.2. LDAP Name(s)

gn, givenName

##### 3.1.2.4.3. Syntax (DAP)

Directory-String {32768}

##### 3.1.2.4.4. Example (DAP)

```
GN=John
```

##### 3.1.2.4.5. Example (LDAP)

```
givenName=John
```

#### 3.1.2.5. Initials

The Initials attribute specifies the initials of some or all of an individual's name, but not the surname(s).

##### 3.1.2.5.1. Abbreviation

I

##### 3.1.2.5.2. LDAP Name(s)

initials

##### 3.1.2.5.3. Syntax (DAP)

Directory-String {32768}

##### 3.1.2.5.4. Example (DAP)

```
I=F . P .
```

##### 3.1.2.5.5. Example (LDAP)

```
initials=F.P.
```

### 3.1.2.6. Name

The Name attribute is the attribute supertype from which string attribute types typically used for naming are formed.

#### 3.1.2.6.1. Abbreviation

N

#### 3.1.2.6.2. LDAP Name(s)

name

#### 3.1.2.6.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.2.6.4. Example (DAP)

```
N=Miller
```

#### 3.1.2.6.5. Example (LDAP)

```
name=Miller
```

### 3.1.2.7. Organization Identifier

The Organization Identifier attribute specifies an identification of an organization different from the organization name.

#### 3.1.2.7.1. Abbreviation

ORGID

#### 3.1.2.7.2. LDAP Name(s)

organizationIdentifier

#### 3.1.2.7.3. Syntax (DAP)

Directory-String

#### 3.1.2.7.4. Example (DAP)

```
ORGID=org-123
```

#### 3.1.2.7.5. Example (LDAP)

```
organizationIdentifier=org-123
```

### 3.1.2.8. Pseudonym

The Pseudonym attribute a pseudonym for an object. It is used to name an object and make it clear that its name is a pseudonym.

#### 3.1.2.8.1. Abbreviation

PSNYM

#### 3.1.2.8.2. LDAP Name(s)

pseudonym

#### 3.1.2.8.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.2.8.4. Example (DAP)

```
PSNYM=xyz
```

#### 3.1.2.8.5. Example (LDAP)

```
pseudonym=xyz
```

### 3.1.2.9. Serial Number

The Serial Number attribute specifies an identifier, the serial number of a device.

#### 3.1.2.9.1. Abbreviation

SER

#### 3.1.2.9.2. LDAP Name(s)

serialNumber, ser

#### 3.1.2.9.3. Syntax (DAP)

Printable-String (Size(1..64))

#### 3.1.2.9.4. Syntax (LDAP)

Printable-String (Size(1..64)) (LDAP style)

#### 3.1.2.9.5. Example (DAP)

```
SER=A12345
```

#### 3.1.2.9.6. Example (LDAP)

```
serialNumber=A12345
```

### 3.1.2.10. Surname

The Surname attribute specifies the surname of an individual.

#### 3.1.2.10.1. Abbreviation

SN

#### 3.1.2.10.2. LDAP Name(s)

sn, surname

#### 3.1.2.10.3. Syntax (DAP)

Directory-String {64}

#### 3.1.2.10.4. Example (DAP)

```
SN=Smith
```

#### 3.1.2.10.5. Example (LDAP)

```
sn=Smith  
{surname=Meier-Luedenscheidt}
```

### 3.1.2.11. Unique Identifier

The Unique Identifier attribute specifies an identifier used to distinguish between object references when a distinguished name is reused. The attribute value is represented by a bit string.

#### 3.1.2.11.1. Abbreviation

UID

#### 3.1.2.11.2. LDAP Name(s)

x500UniqueIdentifier, uniqueIdentifier

#### 3.1.2.11.3. Syntax (DAP)

Unique-Identifier

#### 3.1.2.11.4. Syntax (LDAP)

Bit String

#### 3.1.2.11.5. Example (DAP)

```
UID=0011010100
```

#### 3.1.2.11.6. Example (LDAP)

```
uniqueIdentifier='0011010100'B
```

### 3.1.3. Attributes for Authentication

#### 3.1.3.1. Authority Revocation List

The Authority Revocation List attribute identifies Certification Authority (CA) certificates that have been revoked, typically because they have been compromised, or are otherwise no longer valid. The attribute is used solely by Certification Authorities. (See **Certificate-List** syntax for details.)

##### 3.1.3.1.1. Abbreviation

ARL

##### 3.1.3.1.2. LDAP Name(s)

authorityRevocationList

##### 3.1.3.1.3. Syntax (DAP)

Certificate-List

##### 3.1.3.1.4. Syntax (LDAP)

Specify the attribute value in binary format. (See section **Binary Attribute Values** of chapter **DirX Directory String Representations for LDAP Binds** for details.)

#### 3.1.3.1.5. Example (DAP)

```
ARL={CLU={VER=V2,  
    SIG={AL=1.3.14.3.2.3},  
    IS={/O=My-Company/OU=Corporate CA },  
    TU=9612011200Z,  
    NU=9612011300Z,  
    RC={SN=25,  
        RD=9603011500Z};  
    {SN=75,  
        RD=9608231005Z}}},  
ALG={AL=1.3.14.3.2.3},  
EV=1011010101101101011000}
```

#### 3.1.3.1.6. Example (LDAP)

```
{authorityRevocationList;binary=MIIBaTCB0wIBATANBgqhkiG9w0BAQUFADAtM  
QswCQYDVQQGEwJERTEMMAoGA1UEChMDUFFSMRAwDgYDVQQLEwdFbnRydXN0Fw050DA5MD  
IxMjIzMTThaFw050DA5MDMxMzEzMzJaoHIwCDBBBgNVHRwBAf8ENZaA1oDOgMaQvMC0xCzA  
JBgNVBAYTAkRFRMQwwCgYDVQQKEwNQUVIxEDA0BgNVBAsTB0VudHJ1c3QwCgYDVVR0UBAMC  
AQ4wHwYDVVR0jBBgwFoAUEy7QRSaZ8D9VFsYlK1V/XQsXJZ0wDQYJKoZIhvcNAQEFBQADg  
YEAki/dBwJ+v9qU1+rJA6Pkvi5BQgAfpGPCgoq5jWdkLw/I/GbPhxZAMEYDvvDPkUTh4M  
wkfL2HFpwwKKZVNJ25kfW0zdSEMtUuFEym4ofRi8B24r9PoivzGHvoZFRhtvG/jBEnDDm  
5f0BsoDvLk7Z+D0i/kNotrW4cZdDAaYg0irE=}
```

### 3.1.3.2. CA-Certificate

The CA-Certificate attribute specifies the public key of the user's certification authority (CA). (See **Certificate** syntax for details.)

#### 3.1.3.2.1. Abbreviation

CAC

#### 3.1.3.2.2. LDAP Name(s)

cACertificate

#### 3.1.3.2.3. Syntax (DAP)

Certificate

#### 3.1.3.2.4. Syntax (LDAP)

Specify the attribute value in binary format. (See section **Binary Attribute Values** of chapter **DirX Directory String Representations for LDAP Binds** for details.)

#### 3.1.3.2.5. Example (DAP)

```
CAC={CU={VER=V3,
        SN=10,
        SIG={AL=2.5.8.1.1,
            ALP=1024
        },
        IS={/O=my-company/OU=asw},
        VAL={VNA=9412011200Z,
            VNB=9412011500Z
        },
        SUB={/O=my-company/OU=asw/CN=wythenshawe},
        SUBK={AL={AL=2.5.8.1.1,
            ALP=1024
        },
        PK=0011000100110010001100111100100110000001100010000001
        },
        IUID=00101010110100101001,
        SUID=00111010110100111001
        },
        ALG={AL=2.5.8.1.1,
            ALP=1024
        },
        EV=1011010101101101011000}
```

#### 3.1.3.3. Certificate Revocation List

The Certificate Revocation List attribute identifies user certificates that have been revoked, typically because they have been compromised or are otherwise no longer valid. The attribute is used solely by Certification Authorities. (See **Certificate-List** syntax for details.)

##### 3.1.3.3.1. Abbreviation

CRL

##### 3.1.3.3.2. LDAP Name(s)

certificateRevocationList





#### 3.1.3.4.2. LDAP Name(s)

crossCertificatePair

#### 3.1.3.4.3. Syntax (DAP)

Certificate-Pair

#### 3.1.3.4.4. Syntax (LDAP)

Specify the attribute value in binary format. (See section **Binary Attribute Values** of chapter **DirX Directory String Representations for LDAP Binds** for details.)

#### 3.1.3.4.5. Example (DAP)

```
CCP={F={CU={VER=V3,
            SN=10,
            SIG={AL=2.5.8.1.1,
                ALP=1024
            },
            IS={/O=my-company/OU=asw},
            VAL={VNA=9412011200Z,
                VNB=9412011500Z
            },
            SUB={/C=ie/O=sse},
            SUBK={AL={AL=2.5.8.1.1,
                ALP=512
            },
                PK=0011000100110010001100111100100110000011000100
            },
            IUID=00101010110100101001,
            SUID=00111010110100111001
        },
        ALG={AL=2.5.8.1.1,
            ALP=1024
        },
        EV=1011010101101101011000
    },
    R={CU={VER=V3,
        SN=10,
        SIG={AL=2.5.8.1.1,
            ALP=1024
        },
        IS={/C=ie/O=sse},
```

```

        VAL={VNA=9412011200Z,
            VNB=9412011500Z
            },
        SUB={/O=my-company/OU=asw},
        SUBK={AL={AL=2.5.8.1.1,
            ALP=512
            },
            PK=00110001001100100011001111001001100000110001000
            },
        IUID=10101010110100101001,
        SUID=00111010110000101001
        },
        ALG={AL=2.5.8.1.1,
            ALP=1024
            },
        EV=1011010101101101011000
    }
}

```

### 3.1.3.5. Delta Revocation List

The Delta Revocation List attribute identifies certificates that have been recently revoked, typically because they have been compromised, or are otherwise no longer valid. The attribute is used solely by Certification Authorities. (See **Certificate-List** syntax for details.)

#### 3.1.3.5.1. Abbreviation

DRL

#### 3.1.3.5.2. LDAP Name(s)

deltaRevocationList

#### 3.1.3.5.3. Syntax (DAP)

Certificate-List

#### 3.1.3.5.4. Syntax (LDAP)

Specify the attribute value in binary format. (See section **Binary Attribute Values** of chapter **DirX Directory String Representations for LDAP Binds** for details.)

#### 3.1.3.5.5. Example (DAP)

```
DRL={CLU={VER=V2,
```

```

    SIG={AL=1.3.14.3.2.3},
        IS={/O=My-Company/OU=Corporate CA },
        TU=9612011200Z,
    NU=9612011300Z,
    RC={SN=25,
        RD=9603011500Z};
        {SN=75,
        RD=9608231005Z}}
    ALG={AL=1.3.14.3.2.3},
    EV=1011010101101101011000}

```

#### 3.1.3.5.6. Example (LDAP)

```

{deltaRevocationList;binary=MIIBaTCB0wIBATANBgkqhkiG9w0BAQUFADA+MQswC
QYDVQQGEwJERTEMMAoGA1UEChMDUFFSMRAwDgYDVQQLEwdFbnRydXN0Fw05ODA5MDIxMj
IzMThaFw05ODA5MDMxMzEzMzJaoHIwcDBBBgNVHRwBAf8ENzA1oDOgMaQvMC0xCzAJBgN
VBAYTAKRFMQwwCgYDVQQKEwNQUVIXEDAOBgNVBAsTB0VudHJ1c3QwCgYDVVR0UBAMCAQ4w
HwYDVROjBBGwFoAUEy7QRSaZ8D9VF5YlK1V/XQsXJZ0wDQYJKoZIhvcNAQEFBQADgYEAk
i/dBwJ+v9qU1+rJA6Pkvi5BQgAfpGPCgoq5jWdkLw/I/GbPhxZAMEYDvvDPkUTH4MwkfL
2HFpwwKKZVNJ25kfW0zdSEMtUuFEym4ofRi8B24r9PoivzGHvoZFRhtvG/jBEnDDm5f0B
soDvLk7Z+DOi/kNotrW4cZdDAaYg0irE=}

```

#### 3.1.3.6. NTS Security Identifier

The NTS Security Identifier attribute specifies the NT account name of the user. If an authenticated directory bind based on NT security should be performed, the user's entry must contain this attribute. (See **dse bind** and **obj bind** in the *DirX Directory Administration Reference* for details.)

##### 3.1.3.6.1. Abbreviation

NTUSER

##### 3.1.3.6.2. LDAP Name(s)

ntSecurityIdentifier

##### 3.1.3.6.3. Syntax (DAP)

Octet String

##### 3.1.3.6.4. Example (DAP)

```
NTUSER=nt-volp
```

#### 3.1.3.6.5. Example (LDAP)

```
ntSecurityIdentifier=nt-volp
```

#### 3.1.3.7. User Certificate

The User Certificate attribute specifies the public key of a user. (See **Certificate** syntax for details.) A user may obtain one or more certificates from one or more Certification Authorities. The object class must have the value **Strong-Authenticated-User (SAU)**.

##### 3.1.3.7.1. Abbreviation

UC

##### 3.1.3.7.2. LDAP Name(s)

userCertificate, uc

##### 3.1.3.7.3. Syntax (DAP)

Certificate

##### 3.1.3.7.4. Syntax (LDAP)

Specify the attribute value in binary format. (See section **Binary Attribute Values** of chapter **DirX Directory String Representations for LDAP Binds** for details.)

##### 3.1.3.7.5. Example (DAP)

```
UC={CU={VER=V3,
        SN=10,
        SIG={AL=2.5.8.1.1,
            ALP=1024
        },
        IS={/O=my-company/OU=asw},
        VAL={VNA=9412011200Z,
            VNB=9412011500Z
        },
        SUB={/O=my-company/OU=asw/CN=wythenshawe},
        SUBK={AL={AL=2.5.8.1.1,
            ALP=1024
        },
```

```

PK=0011100010011001000110011110010011000001100010000001
    },
    IUID=0000110101001011010100,
    SUID=1011010101101101011000
    },
    ALG={AL=2.5.8.1.1,
        ALP=1024
    },
    EV=1111010101101101011000}

```

### 3.1.3.8. User Password

The User Password attribute specifies the password of an object. It is a single-valued attribute. If the password is saved encrypted in the database it is not possible to perform a simple protected bind with **dirxadm** and **dirxc** (DAP-bind).

#### 3.1.3.8.1. Password Encryption

If the DirX Directory DSA password storage scheme is configured as "plain", the DSA stores the user password in the DBAM database as a symmetrically encoded value. In this mode, if an LDIF content file is written or a DAP search result is constructed, the password value is passed to the output stream as the decoded clear text password.

The DirX Directory DSA is configurable to support the password storage schemes "SHA" or "Salted SHA". Passwords that are formed according to one of these schemes have been one-way-encrypted or hashed with a secure hash algorithm and - to create a readable string - base64 encoded. If the salted scheme is in force, the hashed user passwords are concatenated with an arbitrary salt-value before the base64 encoding occurs. The created value string can be stored in the Octet String attribute value of the User Password attribute exactly as the clear text string.

The variant of the secure hash algorithm to be applied can be configured. The default algorithm for hashing user passwords is SHA-1, furthermore the DSA supports the algorithms SHA-224, SHA-256, SHA-384 and SHA-512.

Multiple levels of support of the SHA password storage schemes can be configured. If any of the SHA support levels is turned on, the DirX Directory DSA is able to operate with passwords that have already been hashed outside the server. Additionally the support level defines to what extent the DirX Directory DSA actively performs the clear text-to-hash value conversion of the User Password attribute value.

- **Support level 1 (Basic Level)** - This minimal level of support describes the fact that the DSA is able to operate with user passwords regardless of the format of the stored values. That is, if simple bind credentials are verified by the DSA, it considers the format of the entry's user password and conditionally applies the hash before the actual password matching is performed. When user password values are exported (for example, an LDIF content file is written or a DAP search result is constructed) the value is passed to the output stream as it is stored, regardless of whether it is clear text or hashed format.

- **Support level 2 (Read Level)** - This level introduces additionally to level 1 the feature that user password values are never exported in clear text format. Whenever the DSA retrieves a password from its database, it checks whether it is a clear text string or a hashed format. If the password is clear text, the hash algorithm is applied before the value is passed to the outgoing protocol. Therefore configuring support Level 2 makes passwords appear in LDIF files only in the hashed format. This also applies to the generation of LDIF change files, even if in this case the source of the information is an incoming request rather than the database.
- **Support level 3 (Read Write Level)** - This level introduces additionally to level 2 the feature that user password values are checked during the process of import to the DSA. If the passwords come in as clear text strings, they are automatically converted to the hashed format before they are stored in the DSA's database.

The generation of hashed user passwords follows the specification given in the internet-draft "Lightweight Directory Access Protocol (LDAP): Hashed Attribute values for 'userPassword' draft-stroeder-hashed-userpassword-values-01".

Note that even level 3 does not prevent userPassword values from continuing to exist as clear text strings in the DSA's database, if they were loaded before the password encryption was enabled.

Note also that even if level 2 or higher is activated, a DAP operation can return a non-hashed password if the entry information has been collected from other DSAs via DSP and these other DSAs are governed by another password policy.

Note that the use of hash algorithms distinct from SHA-1 is not recommended unless all DSAs participating in a floating master system are version DirX Directory Directory 8.3 or newer.

The DSA cannot compare two userPassword values that are both provided in a salted hash format; that is, all operations that compare two userPassword values, for example, a **dirxcp obj compare** operation, return an **Unwilling to Perform** error, if not one is a clear text user password.

The format of the password also affects the password syntax checking specified in the Password Policy subentry. If hashed passwords are used in add or modify operations, the server is unable to perform password quality checks. (See **Password Check Syntax** of section titled **Attributes of the Password Policy Subentry** for details.)

The DirX Directory DSA process generates an event log during the initialization process that informs the administrator of the current password storage scheme support level in force.

In principle, hashed passwords are not encrypted: The SHA algorithms are one-way functions resulting in a message digest. Once a user password value is converted into a hashed format, its clear text value cannot be restored.

If user password values are shadowed to directory servers that do not support the same password storage scheme or hash algorithm, these shadow directory servers are not able to use the password values in the context of the directory bind operation. Such a situation may result in the inability of all directory users to log into the directory system. This is particularly the case, if a DirX Directory Directory 8.3 supplier DSA replicates passwords that

were hashed with one of the SHA2 family of algorithms to an older consumer DSA.

Additionally, if the DSA stores a password in the hashed (SHA or SSHA) format, it cannot authenticate the user if the user provides the simple-protected choice of simple credentials.

User password values that are SHA-1 hashed start with the tag {SHA} followed by the base64 encoding of the SHA-1 cipher of the user password value. For example (in LDIF) userPassword: dirx is converted to  
userPassword: {SHA}qxghlHOkekweiJieHndsahdsjuew4n432==

User password values that are SSHA-1 hashed start with the tag {SSHA} followed by the base64 encoding of the SHA-1 cipher of the user password value. For example (in LDIF) userPassword: dirx is converted to  
userPassword: {SSHA}VJAdQg6I79lrPKgxAi/6vHkNUVc2OTkxM2ExYg==

User password values that are hashed with one of the SHA2 family of algorithms start with the tag {SHA224}, {SHA256}, {SHA384} or {SHA512}. For the salted hashed scheme the tags are {SSHA224}, {SSHA256}, {SSHA384} or {SSHA512}.

To enable password encryption, you must specify appropriate values for the Password Storage Scheme, Password Storage Scheme Algorithm and Password Storage Scheme Level attributes of the Password Policy subentry. (See section **Attributes of the Password Policy Subentry** for details.) Note that the Password Storage Scheme, Algorithm and Level are controlled by the respective attributes of the global Password Policy subentry **/cn=GlobalPasswordPolicy**. If subtree-specific Password Policy subentries contain Password Storage Scheme, Algorithm or Level attributes the DSA ignores these attributes; that is the global password storage scheme controls the storage of all user passwords stored in a DSA.

The following table summarizes the effect of the different settings for Password Storage Scheme and Password Storage Scheme Level (=Support level) on the storage and presented output of the User Password attribute.

| Storage Scheme | Storage Scheme Level | Value added as ... (e.g.: from LDIF file or ADD operation) | Value stored in DB as ... | Output as ... (LDIF file or search result) |
|----------------|----------------------|------------------------------------------------------------|---------------------------|--------------------------------------------|
| Plain          | Ignored              | Clear text                                                 | Symmetrical encoded       | Clear text                                 |
| Plain          | Ignored              | SHA                                                        | SHA                       | SHA                                        |
| Plain          | Ignored              | SSHA                                                       | SSHA                      | SSHA                                       |
| SHA            | 1                    | Clear text                                                 | Symmetrical encoded       | Clear text                                 |
| SHA            | 1                    | SHA                                                        | SHA                       | SHA                                        |
| SHA            | 1                    | SSHA                                                       | SSHA                      | SSHA                                       |
| SHA            | 2                    | Clear text                                                 | Symmetrical encoded       | SHA                                        |
| SHA            | 2                    | SHA                                                        | SHA                       | SHA                                        |
| SHA            | 2                    | SSHA                                                       | SSHA                      | SSHA                                       |



| Storage Scheme | Storage Scheme Level | Value added as ... (e.g.: from LDIF file or ADD operation) | Value stored in DB as ... | Output as ... (LDIF file or search result) |
|----------------|----------------------|------------------------------------------------------------|---------------------------|--------------------------------------------|
| SHA            | 3                    | Clear text                                                 | SHA                       | SHA                                        |
| SHA            | 3                    | SHA                                                        | SHA                       | SHA                                        |
| SHA            | 3                    | SSHA                                                       | SSHA                      | SSHA                                       |
| SSHA           | 1                    | Clear text                                                 | Symmetrical encoded       | Clear text                                 |
| SSHA           | 1                    | SHA                                                        | SHA                       | SHA                                        |
| SSHA           | 1                    | SSHA                                                       | SSHA                      | SSHA                                       |
| SSHA           | 2                    | Clear text                                                 | Symmetrical encoded       | SSHA (Note1)                               |
| SSHA           | 2                    | SHA                                                        | SHA                       | SHA                                        |
| SSHA           | 2                    | SSHA                                                       | SSHA                      | SSHA                                       |
| SSHA           | 3                    | Clear text                                                 | SSHA                      | SSHA                                       |
| SSHA           | 3                    | SHA                                                        | SHA                       | SHA                                        |
| SSHA           | 3                    | SSHA                                                       | SSHA                      | SSHA                                       |

Note 1: The output value will differ on each subsequent read in this combination.

#### 3.1.3.8.2. Abbreviation

UP

#### 3.1.3.8.3. LDAP Name(s)

userPassword

#### 3.1.3.8.4. Syntax (DAP)

Octet String (Size(1..128))

#### 3.1.3.8.5. Example (DAP)

UP=pw4712

#### 3.1.3.8.6. Example (LDAP)

userPassword=pw4712

#### 3.1.3.9. DirX TOTP Secret

The DirX TOTP Secret attribute specifies the 32-byte secret key associated with a time-based one-time password (TOTP) used in a DirX Directory TOTP two-factor (2FA) object

bind operation. It is a single valued attribute.

The DirX DSA generates the TOTP secret when a DirX Directory administrator adds the object class **dirxTOTPUser** to the object. Its value is chosen at random by a cryptographically-secure pseudorandom number generator. The DSA stores the secret in the database encrypted with the AES-256 algorithm. The encrypted value is decrypted into the following otpauth URI format when returned in a DAP or LDAP read operation or specified in an LDIF file:

**dirxTOTPSecret=otpauth://totp/\*User?issuer=issuer\_name&secret=\*secret**

where:

- **otpauth** indicates the scheme used by authenticator apps to generate one-time passwords (OTPs) see RFC [draft-linuxgemini-otpauth-uri-01](#).
- **totp** indicates the type of OTP being generated; in this case, time-based one-time passwords (TOTPs).
- The *User* parameter identifies the object associated with the TOTP secret and is the naming attribute of the object with the default value **2FA User**.
- The *issuer\_name* parameter identifies the provider or service associated with the TOTP secret and is either the value held by the DirX DSA environment variable **DIRX\_DSA\_TOTP\_ISSUER** or the value **DirX** if the environment variable is unset.
- The *secret* parameter is the actual secret in Base32 encoding.

This format is recognized by most TOTP authenticator apps and can be copied/pasted into these apps.

#### 3.1.3.9.1. Abbreviation

TOTPS

#### 3.1.3.9.2. LDAP Name(s)

dirxTOTPSecret

#### 3.1.3.9.3. Syntax (DAP)

Octet String

#### 3.1.3.9.4. Example (DAP)

```
TOTPS=++\++x6f++\++x74++\++x70++\++x61++\++x75++\++x74++\++x68++\++x3
a++\++x2f++\++x2f++\++x74++\++x6f++\++x74++\++x70++\++x2f++\++x55++\+
+x73++\++x65++\++x72++\++x3f++\++x69++\++x73++\++x73++\++x75++\++x65+
+\++x72++\++x3d++\++x44++\++x69++\++x72++\++x58++\++x26++\++x73++\++x
65++\++x63++\++x72++\++x65++\++x74++\++x3d++\++x4d++\++x47++\++x48++\
++x42++\++x46++\++x54++\++x54++\++x4d++\++x4f++\++x57++\++x54++\++x53
++\++x36++\++x46++\++x45++\++x4b++\++x4a++\++x35++\++x50++\++x44++\++
```

```
x58++\++x52++\++x58++\++x4d++\++x41++\++x4d++\++x52++\++x4c++\++x32++  
\++x57++\++x4a++\++x58++\++x41++\++x33++\++x53++\++x42++\++x32++\++x4  
3++\++x4a++\++x4f++\++x50++\++x47++\++x51++\++x41++\++x53++\++x44++\+  
+x36++\++x47++\++x4f++\++x34++\++x52++\++x51++\++x3d++\++x3d++\++x3d+  
+\++x3d
```

#### 3.1.3.9.5. Example (LDAP)

```
dirxTOTPSecret=otpauth://totp/User?issuer=DirX&secret=MGHBFTTMOWTS6FE  
KJ5PDXRXMAMRL2WJXA3SB2CJOPGQASD6G04RQ=====
```

### 3.1.4. Explanatory Attributes

#### 3.1.4.1. Business Category

The Business Category attribute specifies information concerning the occupation of some common objects; for example, people.

##### 3.1.4.1.1. Abbreviation

BC

##### 3.1.4.1.2. LDAP Name(s)

businessCategory

##### 3.1.4.1.3. Syntax (DAP)

Directory-String {128}

##### 3.1.4.1.4. Example (DAP)

```
{BC=Software Engineering}
```

##### 3.1.4.1.5. Example (LDAP)

```
{businessCategory=Software Engineering}
```

#### 3.1.4.2. Description

The Description attribute specifies information describing the associated object.

#### 3.1.4.2.1. Abbreviation

DSC

#### 3.1.4.2.2. LDAP Name(s)

description, dsc

#### 3.1.4.2.3. Syntax (DAP)

Directory-String {1024}

#### 3.1.4.2.4. Example (DAP)

```
{DSC=Feel free to phone he is very helpful}
```

#### 3.1.4.2.5. Example (LDAP)

```
{description=Feel free to phone he is very helpful}
```

### 3.1.5. Postal Addressing

#### 3.1.5.1. Physical Delivery Office Name

The Physical Delivery Office Name attribute specifies the name of the city, village, etc. where a physical delivery office is situated.

##### 3.1.5.1.1. Abbreviation

PDO

##### 3.1.5.1.2. LDAP Name(s)

physicalDeliveryOfficeName

##### 3.1.5.1.3. Syntax (DAP)

Directory-String {128}

##### 3.1.5.1.4. Example (DAP)

```
PDO=Munich-Trudering
```

##### 3.1.5.1.5. Example (LDAP)

```
physicalDeliveryOfficeName=Munich-Trudering
```

### 3.1.5.2. Postal Address

The Postal Address attribute specifies the address information required for physical delivery of postal messages by the postal authority to the named object. (See **Postal-Address** syntax for details.)

#### 3.1.5.2.1. Abbreviation

PA

#### 3.1.5.2.2. LDAP Name(s)

postalAddress, pa

#### 3.1.5.2.3. Syntax (DAP)

Postal-Address

#### 3.1.5.2.4. Syntax (LDAP)

Postal-Address (LDAP style)

#### 3.1.5.2.5. Example (DAP)

```
{PA={PA1=Frank Mayer,PA2=Otto-Hahn-Ring 6,PA3=D-81730 Munich}}
```

#### 3.1.5.2.6. Example (LDAP)

```
{postalAddress=Frank Mayer$Otto-Hahn-Ring 6$D-81730 Munich$Germany}
```

### 3.1.5.3. Postal Code

The Postal Code attribute specifies the postal code for a named object. It is part of the object's postal address.

#### 3.1.5.3.1. Abbreviation

PC

#### 3.1.5.3.2. LDAP Name(s)

postalCode

#### 3.1.5.3.3. Syntax (DAP)

Directory-String {40}

#### 3.1.5.3.4. Example (DAP)

```
PC=D-81730
```

#### 3.1.5.3.5. Example (LDAP)

```
postalCode=D-81730
```

### 3.1.5.4. Post Office Box

The Post Office Box attribute specifies the post office box at which the object will receive physical post delivery. It is part of the object's postal address.

#### 3.1.5.4.1. Abbreviation

POB

#### 3.1.5.4.2. LDAP Name(s)

postOfficeBox

#### 3.1.5.4.3. Syntax (DAP)

Directory-String {40}

#### 3.1.5.4.4. Example (DAP)

```
POB=1281730
```

#### 3.1.5.4.5. Example (LDAP)

```
postOfficeBox=1281730
```

## 3.1.6. Telecommunication Addresses

### 3.1.6.1. Destination Indicator

The Destination Indicator attribute specifies according to CCITT Rec. F.1 and Rec. F.31 the country and city associated with the object needed to provide the public telegram service.

#### 3.1.6.1.1. Abbreviation

DI

#### 3.1.6.1.2. LDAP Name(s)

destinationIndicator

#### 3.1.6.1.3. Syntax (DAP)

Destination-Indicator

#### 3.1.6.1.4. Syntax (LDAP)

Printable String {128} (LDAP style) only alphabetical characters

#### 3.1.6.1.5. Example (DAP)

```
{DI=F Paris}
```

#### 3.1.6.1.6. Example (LDAP)

```
{destinationIndicator=F Paris}
```

### 3.1.6.2. Fax Telephone Number

The Fax Telephone Number attribute specifies a telephone number for a facsimile (FAX) terminal. (See **Facsimile-Telephone-Number** for details).

#### 3.1.6.2.1. Abbreviation

FTN

#### 3.1.6.2.2. LDAP Name(s)

facsimileTelephoneNumber, ftn

#### 3.1.6.2.3. Syntax (DAP)

Facsimile-Telephone-Number

#### 3.1.6.2.4. Syntax (LDAP)

Facsimile-Telephone-Number (LDAP-style)

#### 3.1.6.2.5. Example (DAP)

```
FTN={PN=3456}
```

#### 3.1.6.2.6. Example (LDAP)

```
facsimileTelephoneNumber=3456
```

### 3.1.6.3. International ISDN Number

The International ISDN Number attribute specifies an International ISDN number associated with an object. The attribute value is a numeric string that complies with the internationally agreed format for ISDN addresses given in CCITT Rec. E. 164. (See **International-ISDN-Number** for details.)

#### 3.1.6.3.1. Abbreviation

IIN

#### 3.1.6.3.2. LDAP Name(s)

internationalISDNNumber

#### 3.1.6.3.3. Syntax (DAP)

International-ISDN-Number (see CCITT Rec. E.164 for valid values)

#### 3.1.6.3.4. Syntax (LDAP)

Numeric-String {16}

#### 3.1.6.3.5. Example (DAP)

```
IIN=27456
```

#### 3.1.6.3.6. Example (LDAP)

```
internationalISDNNumber=27456
```

### 3.1.6.4. Registered Address

The Registered Address attribute specifies a mnemonic for an address associated with an object at a particular city location. It is registered in the country in which the city is located and is used by the public telegram service. (See **Postal-Address** syntax for details.)

#### 3.1.6.4.1. Abbreviation

RA

#### 3.1.6.4.2. LDAP Name(s)

registeredAddress



#### 3.1.6.4.3. Syntax (DAP)

Postal-Address

#### 3.1.6.4.4. Syntax (LDAP)

Postal-Address (LDAP style)

#### 3.1.6.4.5. Example (DAP)

```
{RA={PA1=Frank Mayer,PA2=Otto-Hahn-Ring 6,PA3=D-81730 Munich}}
```

#### 3.1.6.4.6. Example (LDAP)

```
{postalAddress=Frank Mayer$Otto-Hahn-Ring 6$D-81730 Munich$Germany}
```

### 3.1.6.5. Telephone Number

The Telephone Number attribute specifies a telephone number associated with an object.

#### 3.1.6.5.1. Abbreviation

TN

#### 3.1.6.5.2. LDAP Name(s)

telephoneNumber, tn

#### 3.1.6.5.3. Syntax (DAP)

Printable-String (Size(1..32))

#### 3.1.6.5.4. Syntax (LDAP)

Printable-String (Size(1..32)) (LDAP style)

#### 3.1.6.5.5. Example (DAP)

```
{TN=+49 89 6745678;+48 89 636 567890}
```

#### 3.1.6.5.6. Example (LDAP)

```
{telephoneNumber=+49 89 6745678;+48 89 636 567890}
```

### 3.1.6.6. Teletex Terminal Identifier

The Teletex Terminal Identifier attribute specifies a teletex terminal identifier for a teletex terminal associated with an object. (See **Teletex-Terminal-Identifier** syntax for details.)

#### 3.1.6.6.1. Abbreviation

TTI

#### 3.1.6.6.2. LDAP Name(s)

teletexTerminalIdentifier

#### 3.1.6.6.3. Syntax (DAP)

Teletex-Terminal-Identifier

#### 3.1.6.6.4. Syntax (LDAP)

Teletex-Terminal-Identifier (LDAP style)

#### 3.1.6.6.5. Example (DAP)

```
{TTI={TT= My-Company AG Teletex center}}
```

#### 3.1.6.6.6. Example (LDAP)

```
{teletexTerminalIdentifier=My-Company AG Teletex center$page:letter}
```

### 3.1.6.7. Telex Number

The Telex Number attribute specifies a telex number, country code, and answerback code for a telex terminal associated with an object. (See **Telex-Number** syntax for details.)

#### 3.1.6.7.1. Abbreviation

TXN

#### 3.1.6.7.2. LDAP Name(s)

telexNumber

#### 3.1.6.7.3. Syntax (DAP)

Telex-Number

#### 3.1.6.7.4. Syntax (LDAP)

Telex-Number (LDAP style)

#### 3.1.6.7.5. Example (DAP)

```
{TXN={TN=24344,  
      CC=046,  
      AB=GAMEX B  }}
```

#### 3.1.6.7.6. Example (LDAP)

```
{telexNumber=24344$046$GAMEX B}
```

### 3.1.6.8. X.121 Address

The X.121 Address attribute specifies an address as defined by CCITT Rec. X.121 associated with an object. (See **X121-Address Syntax** for details.)

#### 3.1.6.8.1. Abbreviation

X1A

#### 3.1.6.8.2. LDAP Name(s)

x121Address

#### 3.1.6.8.3. Syntax (DAP)

X121-Address

#### 3.1.6.8.4. Syntax (LDAP)

Numeric-String {15}

#### 3.1.6.8.5. Example (DAP)

```
X1A=1234
```

#### 3.1.6.8.6. Example (LDAP)

```
x121Address=1234
```

## 3.1.7. Preference Attributes

### 3.1.7.1. Preferred Delivery Method

The Preferred Delivery Method attribute specifies the order of preference for message delivery methods. The attribute is single-valued. (See **Preferred Delivery Method Syntax** for

details.)

Valid values are:

| DAP bind  | LDAPv3 bind | Meaning                          |
|-----------|-------------|----------------------------------|
| ANY       | any         | Any method of delivery           |
| MHS       | mhs         | Message handling system delivery |
| PHYSICAL  | physical    | Physical delivery                |
| TELEX     | telex       | Telex delivery                   |
| TELETEX   | teletex     | Teletex delivery                 |
| G3_FAX    | g3fax       | G3 FAX delivery                  |
| G4_FAX    | g4fax       | G4 FAX delivery                  |
| IA5       | ia5         | IA5 terminal delivery            |
| VIDEOTEX  | videotex    | Videotex delivery                |
| TELEPHONE | telephone   | Telephone delivery               |

For DAP binds, multiple values are separated by a plus sign (+), with the most preferred method first in the list; for LDAP binds, multiple values are separated by a dollar sign (\$), with the most preferred method first in the list.

#### 3.1.7.1.1. Abbreviation

PDM

#### 3.1.7.1.2. LDAP Name(s)

preferredDeliveryMethod

#### 3.1.7.1.3. Syntax (DAP)

Preferred-Delivery-Method

#### 3.1.7.1.4. Syntax (LDAP)

Preferred-Delivery-Method (LDAP style)

#### 3.1.7.1.5. Example (DAP)

PDM=PHYSICAL+ANY

#### 3.1.7.1.6. Example (LDAP)

preferredDeliveryMethod=physical\$any

## 3.1.8. OSI Applications

### 3.1.8.1. Application Entity

The Application Entity attribute specifies the object identifier that is assigned to an application entity and allows it to be identified.

#### 3.1.8.1.1. Abbreviation

AE

#### 3.1.8.1.2. LDAP Name(s)

applicationEntity

#### 3.1.8.1.3. Syntax (DAP)

Object Identifier

#### 3.1.8.1.4. Syntax (LDAP)

Object Identifier (LDAP style) in dotted notation.

#### 3.1.8.1.5. Example (DAP)

```
AE=1.3.12.2.1107.0
```

#### 3.1.8.1.6. Example (LDAP)

```
applicationEntity=1.3.12.2.1107.0
```

### 3.1.8.2. Presentation Address

The Presentation Address attribute specifies a presentation address associated with an object representing an OSI application entity. (See **Presentation-Address** syntax for details.)

This attribute is not supported for LDAP protocol.

#### 3.1.8.2.1. Abbreviation

PSA

#### 3.1.8.2.2. Syntax (DAP)

Presentation-Address

#### 3.1.8.2.3. Example (DAP)

```
PSA={TS=server,NA='TCP/IP!internet=123.45.67.89+port=31111' }
```

### 3.1.8.3. Protocol Information

The Protocol Information attribute associates protocol information with each network address in the PSAP attribute. For each NSAP, address it identifies the protocol or profile for the network and transport layers. (See **Protocol Information** syntax for details.)

This attribute is not supported for LDAP protocol.

#### 3.1.8.3.1. Abbreviation

PI

#### 3.1.8.3.2. Syntax (DAP)

Protocol-Information

#### 3.1.8.3.3. Example (DAP)

```
PI={NA='TCP/IP!internet=123.45.67.89+port=102',  
    OID=1.3.14.65.237;1.3.14.65.239}
```

### 3.1.8.4. Supported Application Context

The Supported Application Context attribute specifies the object identifiers of supported application contexts. The attribute is contained in the root-DSE and is automatically generated when the database is created. It can be accessed using **dirxadm**.

Valid values are:

- **DAP** - Directory-Access-Application-Context (DUA or DSA functionality using DAP protocol)
- **DOPM** - Directory-OPERational-Binding-Management-Application-Context (DSA functionality using DOP protocol to coordinate HOBs (hierarchical operational bindings) or shadowing)
- **DSP** - Directory-System-Application-Context (DSA functionality using DSP protocol (chaining))
- **RSCI** - Reliable-Shadow-Consumer-Initiated-Application-Context (DSA functionality using DISP protocol (shadowing) over the RTSE protocol when the consumer initiates the shadowing)
- **RSSI** - Reliable-Shadow-Supplier-Initiated-Application-Context context (DSA functionality using DISP protocol (shadowing) over the RTSE protocol when the supplier initiates the shadowing)
- **SCI** - Shadow-Consumer-Initiated-Application-Context (DSA functionality using DISP

protocol when the consumer initiates the shadowing, and operations are synchronous; that is, no invoke can be initiated until a response to the last invoke has been received)

- **SSI** - Shadow-Supplier-Initiated-Application-Context (DSA functionality using DISP when the supplier initiates the shadowing, and operations are synchronous)
- **SSIA** - Shadow-Supplier-Initiated-Asynchronous-Application-Context (DSA functionality using DISP when the supplier initiates the shadowing, and operations can be asynchronous)
- **SCIA** - Shadow-Consumer-Initiated- Asynchronous-Application-Context (DSA functionality using DISP when the consumer initiates the shadowing, and operations can be synchronous; that is, another invoke can be initiated before a response to the last invoke has been received)

The DirX Directory DSA supports the application contexts **DAP**, **DSP**, **SCI**, and **SSI**.

Because this attribute is only accessible by using **dirxadm** no access using LDAP or DAP protocol is possible.

#### 3.1.8.4.1. Abbreviation

SAC

#### 3.1.8.4.2. Syntax

Object Identifier

#### 3.1.8.4.3. Example (dirxadm)

```
SAC=DAP ; DSP ; SCI ; SSI
```

### 3.1.8.5. Transfer Syntaxes Supported

The Transfer Syntaxes Supported attribute specifies the object identifiers of the transfer syntaxes and / or encoding rules supported. The attribute is contained in the root-DSE. It can be accessed using **dirxadm**.

The implementation supports the ASN.1 Basic and Distinguished Encoding Rules.

Because this attribute is only accessible by using **dirxadm**, no access using LDAP or DAP protocol is possible.

#### 3.1.8.5.1. Abbreviation

TSS

#### 3.1.8.5.2. Syntax

Object Identifier

#### 3.1.8.5.3. Example (dirxadm)

```
TSS=2.1.1;2.1.2.1 ASN.1 Basic and Distinguished Encoding Rules
```

#### 3.1.8.6. Upper Layer Profile Info

The Upper Layer Profile Info attribute specifies the object identifiers of the upper layer profile that are implemented in an application entity.

The implementation supports the EWOS profiles FOI11 and FOI5.

##### 3.1.8.6.1. Abbreviation

ULPI

##### 3.1.8.6.2. LDAP Name(s)

upperLayerProfileInfo

##### 3.1.8.6.3. Syntax (DAP)

Object Identifier

##### 3.1.8.6.4. Syntax (LDAP)

Object Identifier (LDAP style) in dotted notation.

##### 3.1.8.6.5. Example (DAP)

```
ULPI=1.0.10616;1.0.12072 EWOS profiles FOI11 and FOI5
```

##### 3.1.8.6.6. Example (LDAP)

```
upperLayerProfileInfo=1.0.10616;1.0.12072
```

### 3.1.9. Relational Attributes

Relational attributes are concerned with information regarding objects that are related to a particular object in certain ways.

#### 3.1.9.1. Aliased Object Name

The Aliased Object Name attribute specifies an alternative name for an object or object entry. It is contained in an alias entry.



#### 3.1.9.1.1. Abbreviation

AON

#### 3.1.9.1.2. LDAP Name(s)

aliasedObjectName, aliasedEntryName

#### 3.1.9.1.3. Syntax (DAP)

Distinguished Name

#### 3.1.9.1.4. Syntax (LDAP)

Distinguished Name (LDAP style)

#### 3.1.9.1.5. Example (DAP)

```
AON={/O=My-Company/OU=Sales/CN=Hohner}
```

#### 3.1.9.1.6. Example (LDAP)

```
aliasedObjectName=cn=Hohner,ou=Sales,o=My-Company
```

### 3.1.9.2. DirX Directory ImportFrom

The DirX Directory ImportFrom attribute specifies the DN of a groupOfNames entry from which member attributes are to be imported. It is used to build a nested group.

#### 3.1.9.2.1. Abbreviation

IMPFROM

#### 3.1.9.2.2. LDAP Name(s)

dirxImportFrom

#### 3.1.9.2.3. Syntax (DAP)

Distinguished Name

#### 3.1.9.2.4. Syntax (LDAP)

Distinguished Name (LDAP style)

#### 3.1.9.2.5. Example (DAP)

```
IMPFROM={/O=My-Company/CN=UserGroups}
```

#### 3.1.9.2.6. Example (LDAP)

```
dirxImportFrom= cn=UserGroups,o=My-Company
```

#### 3.1.9.3. DirX Directory MemberUrl

The DirX Directory MemberUrl attribute specifies an LDAP search expression. It is used to define a dynamic group.

##### 3.1.9.3.1. Abbreviation

DRXMURL

##### 3.1.9.3.2. LDAP Name(s)

dirxMemberUrl

##### 3.1.9.3.3. Syntax (DAP)

Directory-String in the format of DirX Directory-Member-Url attribute syntax (see the section **DirX Directory-Member-Url** in the chapter **DirX Directory String Representation for DAP Binds** for details)

##### 3.1.9.3.4. Syntax (LDAP)

Directory-String in the format of DirX Directory-Member-Url attribute syntax (see the section **DirX Directory-Member-Url** in the chapter **DirX Directory String Representation for DAP Binds** for details)

##### 3.1.9.3.5. Example (DAP)

```
DRXMURL=ldap:///o=My-Company??sub?(&(l=sunnyvale)(title=senior*))
```

##### 3.1.9.3.6. Example (LDAP)

```
dirxMemberUrl=ldap:///o=My-  
Company??sub?(&(l=sunnyvale)(title=senior*))
```

#### 3.1.9.4. DN

The Distinguished Name attribute specifies the name of an object.

##### 3.1.9.4.1. Abbreviation

DN

#### 3.1.9.4.2. LDAP Name(s)

distinguishedName

#### 3.1.9.4.3. Syntax (DAP)

Distinguished-Name

#### 3.1.9.4.4. Syntax (LDAP)

Distinguished-Name (LDAP style)

#### 3.1.9.4.5. Example (DAP)

```
DN={/O=My-Company/OU=Sales/CN=Miller}
```

#### 3.1.9.4.6. Example (LDAP)

```
distinguishedName=cn=Miller,ou=Sales,o=My-Company
```

### 3.1.9.5. Member

The Member attribute specifies a group of names associated with an object.

#### 3.1.9.5.1. Abbreviation

MEM

#### 3.1.9.5.2. LDAP Name(s)

member

#### 3.1.9.5.3. Syntax (DAP)

Distinguished-Name

#### 3.1.9.5.4. Syntax (LDAP)

Distinguished-Name (LDAP style)

#### 3.1.9.5.5. Example (DAP)

```
{MEM=/O=My-Company/OU=Sales/CN=distribution list}
```

#### 3.1.9.5.6. Example (LDAP)

```
{member=cn=distribution list,ou=Sales,o=My-Company}
```

### 3.1.9.6. Owner

The Owner attribute specifies the name of the object that is responsible for an associated object.

#### 3.1.9.6.1. Abbreviation

OWN

#### 3.1.9.6.2. LDAP Name(s)

owner

#### 3.1.9.6.3. Syntax (DAP)

Distinguished-Name

#### 3.1.9.6.4. Syntax (LDAP)

Distinguished-Name (LDAP style)

#### 3.1.9.6.5. Example (DAP)

```
OWN={/O=My-Company/OU=Sales/CN=Mayer}
```

#### 3.1.9.6.6. Example (LDAP)

```
owner=cn=Mayer,ou=Sales,o=My-Company
```

### 3.1.9.7. Role Occupant

The Role Occupant attribute specifies the name of an object that fulfills an organizational role.

#### 3.1.9.7.1. Abbreviation

RO

#### 3.1.9.7.2. LDAP Name(s)

roleOccupant

#### 3.1.9.7.3. Syntax (DAP)

Distinguished-Name

#### 3.1.9.7.4. Syntax (LDAP)

Distinguished-Name (LDAP style)

#### 3.1.9.7.5. Example (DAP)

```
RO={/O=My-Company/OU=Sales/CN=Mayer}
```

#### 3.1.9.7.6. Example (LDAP)

```
roleOccupant=cn=Mayer,ou=Sales,o=My-Company
```

### 3.1.9.8. See Also

The See Also attribute specifies the names of directory objects that can be other aspects of the same real world object.

#### 3.1.9.8.1. Abbreviation

SEA

#### 3.1.9.8.2. LDAP Name(s)

seeAlso

#### 3.1.9.8.3. Syntax (DAP)

Distinguished-Name

#### 3.1.9.8.4. Syntax (LDAP)

Distinguished-Name (LDAP style)

#### 3.1.9.8.5. Example (DAP)

```
SEA={/O=My-Company/OU=Sales/CN=Mayer}
```

#### 3.1.9.8.6. Example (LDAP)

```
seeAlso=cn=Mayer,ou=Sales,o=My-Company
```

### 3.1.9.9. Unique Member

The Unique Member attribute identifies objects by a distinguished name and an optional identifier. This attribute can remove ambiguity from names that have been reused. (See **Name-And-Optional-UID** syntax for details.)

#### 3.1.9.9.1. Abbreviation

UM

#### 3.1.9.9.2. LDAP Name(s)

uniqueMember

#### 3.1.9.9.3. Syntax (DAP)

Name-And-Optional-UID

#### 3.1.9.9.4. Syntax (LDAP)

Name-And-Optional-UID (LDAP style)

#### 3.1.9.9.5. Example (DAP)

```
UM={DN={/O=my-company/OU=engineering/CN=Hughes H},  
    UID=0010100101001011110};  
{DN={/O=my-company/OU=engineering/CN=Smith J}  
}
```

#### 3.1.9.9.6. Example (LDAP)

```
uniqueMember=cn=Hughes H,ou=engineering,o=my-company  
            #'0010100101001011110'B;  
            cn=Smith J,ou=engineering,o=my-company
```

### 3.1.10. Miscellaneous Attributes

#### 3.1.10.1. House Identifier

The House Identifier attribute specifies information that is used to identify a particular building; for example, a house number or house name relative to a street or town.

##### 3.1.10.1.1. Abbreviation

HID

##### 3.1.10.1.2. LDAP Name(s)

houseIdentifier

##### 3.1.10.1.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.10.1.4. Example (DAP)

```
HID=32
```

#### 3.1.10.1.5. Example (LDAP)

```
houseIdentifier=32
```

### 3.1.10.2. Knowledge Information

The Knowledge Information attribute specifies a human readable accumulated description of knowledge mastered by a specific DSA.

#### 3.1.10.2.1. Abbreviation

KNI

#### 3.1.10.2.2. LDAP Name(s)

knowledgeInformation

#### 3.1.10.2.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.10.2.4. Example (DAP)

```
{KNI=The entry is also shadowed by the My-Company-DSA}
```

#### 3.1.10.2.5. Example (LDAP)

```
{knowledgeInformation=The entry is also shadowed by the My-Company-  
DSA}
```

### 3.1.10.3. Labeled URI

The Labeled URI attribute as specified in RFC 2079 defines one of several Universal Resource Identifiers. The values must comply with RFC1738.

#### 3.1.10.3.1. Abbreviation

LURI

#### 3.1.10.3.2. LDAP Name(s)

labeledURI

#### 3.1.10.3.3. Syntax (DAP)

Directory-String

#### 3.1.10.3.4. Example (DAP)

```
LURI=http://www.sni.de
```

#### 3.1.10.3.5. Example (LDAP)

```
labeledURI=http://www.sni.de
```

### 3.1.10.4. Object Class

The ObjectClass attribute specifies the object identifiers of the object classes which indicates the nature of a particular entry. Each entry must contain this attribute.

There are three types of object classes:

- Abstract object classes - Used to derive other object classes and provide the common characteristics of such object classes. An entry cannot belong only to abstract object classes. The only abstract object class in DirX Directory is **TOP**.
- Structural object classes - Defines specific types of real-world objects, for example, object class organizational person (**ORP**). An entry is characterized by one structural object class superclass chain which has a single structural object class as the most subordinate object class. The structural object class of an entry cannot be changed.
- Auxiliary object classes - Defines characteristics that can be shared by more than one kind of real-world object. For example, message handling systems make use of the auxiliary object class MHS user (**MUS**) to specify a package of mandatory and optional MHS attributes for entries whose structural object class is variable (for example, **ORP** for organizational person or **REP** for residential person). An object may be member of one or more auxiliary object classes. The auxiliary object class of an entry may change over time.

Since all entries are member of the object class **TOP**, this value need not be specified.

The **Object Class Table** in the chapter titled **DirX Directory Default Schema** describes the LDAP name(s) (which is used as value for LDAP binds), the abbreviation (which is used as value for DAP binds), the type, the superclasses, and the mandatory and optional attributes.

#### 3.1.10.4.1. Abbreviation

OCL



#### 3.1.10.4.2. LDAP Name(s)

objectClass, ocl

#### 3.1.10.4.3. Syntax (DAP)

Object Identifier

#### 3.1.10.4.4. Syntax (LDAP)

Object Identifier (LDAP style) in dotted notation.

#### 3.1.10.4.5. Example (DAP)

```
{OCL=PER;ORP}
```

#### 3.1.10.4.6. Example (LDAP)

```
{objectClass=person;organizationalPerson}
```

### 3.1.10.5. RFC822 Mailbox

The RFC822 Mailbox attribute specifies a SMTP mail address (internet mail address).

#### 3.1.10.5.1. Abbreviation

RMB

#### 3.1.10.5.2. LDAP Name(s)

rfc822Mailbox, mail

#### 3.1.10.5.3. Syntax (DAP)

IA5-String

#### 3.1.10.5.4. Example (DAP)

```
RMB=John.Smithsales.my-company.de
```

#### 3.1.10.5.5. Example (LDAP)

```
mail=John.Smith@sales.my-company.de
```

### 3.1.10.6. Title

The Title attribute specifies the position or function of the object within an organization.

#### 3.1.10.6.1. Abbreviation

TIT

#### 3.1.10.6.2. LDAP Name(s)

title

#### 3.1.10.6.3. Syntax (DAP)

Directory-String {64}

#### 3.1.10.6.4. Example (DAP)

```
TIT=Manager
```

#### 3.1.10.6.5. Example (LDAP)

```
title=Manager
```

## 3.1.11. Collective Attributes

Collective attributes are user attributes whose values are the same for a collection of entries.

Although collective attributes appear to users of the Directory query operations as entry attributes, they are treated differently from entry attributes. This difference is that collective attributes cannot be administered (that is, modified) via the entries in which they appear but must be administered via their associated subentries. (See the chapter titled **Creating Collective Attributes in a Subtree** in the *Administration Guide* for details.)

The DSA does not support the use of collective attributes in filters.

Collective attributes can be excluded from particular entries by the use of the **Collective Exclusions** directory operational attribute.

### 3.1.11.1. Collective Fax Telephone Number

The Collective Fax Telephone Number attribute specifies a fax telephone number for a collection of entries

#### 3.1.11.1.1. Abbreviation

FTNC

#### 3.1.11.1.2. LDAP Name(s)

collectiveFacsimileTelephoneNumber, ftnc

#### 3.1.11.1.3. Syntax (DAP)

Facsimile-Telephone-Number

#### 3.1.11.1.4. Syntax (LDAP)

Facsimile-Telephone-Number (LDAP style)

#### 3.1.11.1.5. Example (DAP)

```
FTNC={PN=563212}
```

#### 3.1.11.1.6. Example (LDAP)

```
collectiveFacsimileTelephoneNumber=563212
```

### 3.1.11.2. Collective International ISDN Number

The Collective International ISDN Number attribute specifies a international ISDN number for a collection of entries

#### 3.1.11.2.1. Abbreviation

IINC

#### 3.1.11.2.2. LDAP Name(s)

collectiveInternationalISDNNumber

#### 3.1.11.2.3. Syntax (DAP)

International-ISDN-Number (see CCITT Rec. E.164 for valid values)

#### 3.1.11.2.4. Syntax (LDAP)

Numeric-String {16}

#### 3.1.11.2.5. Example (DAP)

```
IINC=1234
```

#### 3.1.11.2.6. Example (LDAP)

```
collectiveInternationalISDNNumber=1234
```

### 3.1.11.3. Collective Locality Name

The Collective Locality Name attribute specifies a locality name for a collection of entries

#### 3.1.11.3.1. Abbreviation

LC

#### 3.1.11.3.2. LDAP Name(s)

collectiveLocalityName

#### 3.1.11.3.3. Syntax (DAP)

Directory-String {128}

#### 3.1.11.3.4. Example (DAP)

```
LC=Munich
```

#### 3.1.11.3.5. Example (LDAP)

```
collectiveLocalityName=Munich
```

### 3.1.11.4. Collective Organization Name

The Collective Organization Name attribute specifies an organization name for a collection of entries.

#### 3.1.11.4.1. Abbreviation

ONC

#### 3.1.11.4.2. LDAP Name(s)

collectiveOrganizationName

#### 3.1.11.4.3. Syntax (DAP)

Directory-String {64}

#### 3.1.11.4.4. Example (DAP)

```
{ONC=German Telecom}
```

```
ONC=My-Company
```

#### 3.1.11.4.5. Example (LDAP)

```
{collectiveOrganizationName=German Telecom}  
collectiveOrganizationName=My-Company
```

### 3.1.11.5. Collective Organizational Unit Name

The Collective Organizational Unit Name attribute specifies an organizational unit for a collection of entries.

#### 3.1.11.5.1. Abbreviation

OUC

#### 3.1.11.5.2. LDAP Name(s)

collectiveOrganizationalUnitName

#### 3.1.11.5.3. Syntax (DAP)

Directory-String {64}

#### 3.1.11.5.4. Example (DAP)

```
{OUC=Marketing Division}
```

#### 3.1.11.5.5. Example (LDAP)

```
{collectiveOrganizationalUnitName=Marketing Division}
```

### 3.1.11.6. Collective Postal Address

The Collective Postal Address attribute specifies a postal address for a collection of entries.

#### 3.1.11.6.1. Abbreviation

PAC

#### 3.1.11.6.2. LDAP Name(s)

collectivePostalAddress

#### 3.1.11.6.3. Syntax (DAP)

Postal-Address

#### 3.1.11.6.4. Syntax (LDAP)

Postal-Address (LDAP style)

#### 3.1.11.6.5. Example (DAP)

```
{PAC={PA1=My-Company, PA2=Otto-Hahn-Ring 6, PA3=D-81234 Munich}}
```

#### 3.1.11.6.6. Example (LDAP)

```
{postalAddress=My-Company$Otto-Hahn-Ring 6$D-81730 Munich$Germany}
```

### 3.1.11.7. Collective Postal Code

The Collective Postal Code attribute specifies a postal code for a collection of entries.

#### 3.1.11.7.1. Abbreviation

PCC

#### 3.1.11.7.2. LDAP Name(s)

collectivePostalCode

#### 3.1.11.7.3. Syntax (DAP)

Directory-String {40}

#### 3.1.11.7.4. Example (DAP)

```
PCC=D-81234
```

#### 3.1.11.7.5. Example (LDAP)

```
collectivePostalCode=D-81234
```

### 3.1.11.8. Collective Physical Delivery Office Name

The Collective Physical Delivery Office Name attribute specifies a physical delivery office name for a collection of entries.

#### 3.1.11.8.1. Abbreviation

PDOC

#### 3.1.11.8.2. LDAP Name(s)

collectivePhysicalDeliveryOfficeName

#### 3.1.11.8.3. Syntax (DAP)

Directory-String {128}

#### 3.1.11.8.4. Example (DAP)

```
PDOC=Munich
```

#### 3.1.11.8.5. Example (LDAP)

```
collectivePhysicalDeliveryOfficeName=Munich
```

### 3.1.11.9. Collective Post Office Box

The Collective Post Office Box attribute specifies a post office box name for a collection of entries.

#### 3.1.11.9.1. Abbreviation

POBC

#### 3.1.11.9.2. LDAP Name(s)

collectivePostOfficeBox

#### 3.1.11.9.3. Syntax (DAP)

Directory-String {40}

#### 3.1.11.9.4. Example (DAP)

```
{POBC=Post Office Box 56789}
```

#### 3.1.11.9.5. Example (LDAP)

```
{collectivePostOfficeBox=Post Office Box 56789}
```

### 3.1.11.10. Collective State or Province Name

The Collective State or Province Name attribute specifies a state or province for a collection of entries.

#### 3.1.11.10.1. Abbreviation

SPNC

#### 3.1.11.10.2. LDAP Name(s)

collectiveStateOrProvinceName

#### 3.1.11.10.3. Syntax (DAP)

Directory-String {128}

#### 3.1.11.10.4. Example (DAP)

```
SPNC=Florida
```

#### 3.1.11.10.5. Example (LDAP)

```
collectiveStateOrProvinceName=Florida
```

### 3.1.11.11. Collective Street Address

The Collective Street Address attribute specifies street address for a collection of entries.

#### 3.1.11.11.1. Abbreviation

STAC

#### 3.1.11.11.2. LDAP Name(s)

collectiveStreetAddress

#### 3.1.11.11.3. Syntax (DAP)

Directory-String {128}

#### 3.1.11.11.4. Example (DAP)

```
{STAC=Otto-Hahn-Ring 6}
```

#### 3.1.11.11.5. Example (LDAP)



```
{collectiveStreetAddress=Otto-Hahn-Ring 6}
```

### 3.1.11.12. Collective Telephone Number

The Collective Telephone Number attribute specifies a telephone number for a collection of entries.

#### 3.1.11.12.1. Abbreviation

TNC

#### 3.1.11.12.2. LDAP Name(s)

collectiveTelephoneNumber, tnc

#### 3.1.11.12.3. Syntax (DAP)

Printable-String (SIZE(1..32))

#### 3.1.11.12.4. Syntax (LDAP)

Printable-String (SIZE(1..32)) (LDAP style)

#### 3.1.11.12.5. Example (DAP)

```
{TNC=+49 89 374526}
```

#### 3.1.11.12.6. Example (LDAP)

```
{collectiveTelephoneNumber=+49 89 374526}
```

### 3.1.11.13. Collective TTX Terminal Identifier

The Collective TTX Terminal Identifier attribute specifies a ttx terminal identifier for a collection of entries.

#### 3.1.11.13.1. Abbreviation

TTIC

#### 3.1.11.13.2. LDAP Name(s)

collectiveTeletexTerminalIdentifier

#### 3.1.11.13.3. Syntax (DAP)

Teletex-Terminal-Identifier

#### 3.1.11.13.4. Syntax (LDAP)

Teletex-Terminal-Identifier (LDAP style)

#### 3.1.11.13.5. Example (DAP)

```
{TTIC={TT=My-Company AG Teletex Center}}
```

#### 3.1.11.13.6. Example (LDAP)

```
{collectiveTeletexTerminalIdentifier=My-Company AG Teletex Center}
```

### 3.1.11.14. Collective Telex Number

The Collective Telex Number attribute specifies a telex number for a collection of entries.

#### 3.1.11.14.1. Abbreviation

TXNC

#### 3.1.11.14.2. LDAP Name(s)

collectiveTelexNumber

#### 3.1.11.14.3. Syntax (DAP)

Telex-Number

#### 3.1.11.14.4. Syntax (LDAP)

Telex-Number (LDAP style)

#### 3.1.11.14.5. Example (DAP)

```
{TXNC={TN=375679,CC=049,AB=GAMEX B}}
```

#### 3.1.11.14.6. Example (LDAP)

```
collectiveTelexNumber=375679$049$GAMEX B
```

## 3.1.12. Attributes of the LDAP Root Subentry

The LDAP protocol defines the so-called LDAP root as a collection of attributes that describe the LDAP server's capabilities and supported features. LDAP clients usually read these attributes before they start working with the LDAP server in order to retrieve information about useful features the server provides.

In DirX Directory, the LDAP root subentry **/CN=ldapRoot** is used to store these attributes. The DSA automatically creates an LDAP root subentry that specifies the default values for the DirX Directory LDAP server. The information within the LDAP root subentry should not be changed once it has been established and the LDAP server has been initialized.

Note that the LDAP root subentry contains local information and modifications to this entry are not propagated by shadowing.

The LDAP root subentry can be retrieved via LDAP by performing the following search request:

**obj search "" -allattr -baseobject -vtype SUBENTRY**

Like all other subentries it cannot be created via LDAP. However, it can be modified via LDAP.

### 3.1.12.1. LDAP Naming Contexts

The LDAP Naming Contexts attribute contains the distinguished names of the naming contexts that the contact DSA masters or shadows. This attribute is a mandatory attribute and must contain at a minimum a string with length 0. The DSA automatically synchronizes the LDAP naming context stored in this attribute with the X.500 context prefix. Therefore it is strongly recommended not to modify this attribute.

#### 3.1.12.1.1. Abbreviation

LNCO

#### 3.1.12.1.2. LDAP Name(s)

namingContexts

#### 3.1.12.1.3. Syntax (DAP)

Distinguished Name

#### 3.1.12.1.4. Syntax (LDAP)

Distinguished Name (LDAP style)

#### 3.1.12.1.5. Example (DAP)

```
LNCO={/C=de/O=my-company/OU=mds}
```

#### 3.1.12.1.6. Example (LDAP)

```
namingContexts=o=my-company
```

3.1.12.2. Alternate Servers

The Alternate Servers attribute specifies the URLs of other LDAP servers that can handle the client's request. This attribute is an optional attribute.

3.1.12.2.1. Abbreviation

ASER

3.1.12.2.2. LDAP Name(s)

altServer

3.1.12.2.3. Syntax (DAP)

Directory-String {32768}

3.1.12.2.4. Example (DAP)

```
ASER=ldap://ldapserver.sni.com
```

3.1.12.2.5. Example (LDAP)

```
altServer=ldap://ldapserver.sni.com
```

3.1.12.3. Supported Extensions

The Supported Extensions attribute specifies the object identifiers of the extensions that the LDAP server supports. This attribute is an optional attribute.

The DirX Directory LDAP server supports the following standardized extensions:

|                         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.3.6.1.4.1.1466.20037  | The Start-TLS extended operation.                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 1.3.6.1.4.1.4203.1.11.1 | <p>The LDAP Password Modify Extended Operation. The DirX Directory LDAP server supports this extension with the following restrictions:</p> <ul style="list-style-type: none"><li>• The operation changes the userPassword attribute (OID 2.5.4.35) of an entry stored in the directory</li><li>• The userIdentity field of the PasswdModifyRequestValue must contain a DistinguishedName</li><li>• The LDAP connection must be established using SSL/TLS protection (LDAPS or sasl EXTERNAL)</li></ul> |

#### 3.1.12.3.1. Abbreviation

SEXT

#### 3.1.12.3.2. LDAP Name(s)

supportedExtension

#### 3.1.12.3.3. Syntax (DAP)

Object Identifier

#### 3.1.12.3.4. Syntax (LDAP)

Object Identifier (LDAP style) in dotted notation.

#### 3.1.12.3.5. Example (DAP)

SEXT=1.3.6.1.4.1.1466.20037      LDAP server supports the Start-TLS extended operation

#### 3.1.12.3.6. Example (LDAP)

supportedExtension=1.3.6.1.4.1.1466.20037

### 3.1.12.4. Supported Control

The Supported Control attribute specifies the object identifiers of the controls that the LDAP server supports. This attribute is an optional attribute.

LDAP clients use the values of this attribute to evaluate whether they can successfully include the control values to their operations. The controls themselves are intended to optimize operations; for example, search operations with a large search result.

See **DirX Directory Commands** -> **dirxcp** -> **obj (dirxcp)** -> **-control Option** in the *DirX Directory Administration Reference* for details on how to specify controls for **dirxcp obj** operations.

The default value of the DirX Directory LDAP server is:

|                               |                                                                                                     |
|-------------------------------|-----------------------------------------------------------------------------------------------------|
| <b>1.2.840.113556.1.4.319</b> | <b>Simple Paged Results for request and response. The abbreviation for this control is SPAGRES.</b> |
| <b>1.2.840.113556.1.4.473</b> | Server Side Sorting Request.<br>The abbreviation for this control is <b>SSSREQ</b> .                |
| <b>1.2.840.113556.1.4.474</b> | Server Side Sorting Response.<br>The abbreviation for this control is <b>SSSRESP</b> .              |

|                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.2.840.113556.1.4.319     | Simple Paged Results for request and response. The abbreviation for this control is <b>SPAGRES</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 1.3.6.1.4.1.21008.108.63.1 | LDAP Session Tracking Control<br><br>The abbreviation for this control is <b>LSTRACK</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 1.3.6.1.1.22               | Don't Use Copy Control. This control is intended to direct the server to return master information in LDAP search and LDAP compare operations. It should not be attached to other operations. It overrides the settings for the respective operations service controls (ldapSearchServiceControls and ldapCompareServiceControls) configured in the configuration subentry of the LDAP server.<br><br>The abbreviation for this control is <b>DONTUSECOPY</b> .                                                                                                                     |
| 1.3.6.1.4.1.7628.5.101.1   | LDAP Subentries Control. The server returns only subentries in a search response. In DirX Directory for example, the <b>dirxcp obj search</b> command attaches the Subentry Control to the search request if the option <b>-vtype SUBENTRY</b> is specified.<br><br>The abbreviation for this control is <b>LSUBE</b> .                                                                                                                                                                                                                                                             |
| 1.3.6.1.4.1.42.2.27.8.5.1  | Password Policy Request and Password Policy Response according to the IETF draft RFC "Password Policy for LDAP Directories" (document: draft-behera-ldap-password-policy-07.txt).<br><br>The abbreviation for this control is <b>PPOREQRESP</b> .                                                                                                                                                                                                                                                                                                                                   |
| 2.16.840.1.113730.3.4.18   | LDAP Proxied Authorization Control according to the IETF RFC 4370.<br><br>The abbreviation for this control is <b>LPROXYAUTH</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 1.3.12.2.1107.1.3.2.12.5   | LDAP Relaxed Update Control. This is a proprietary control. It supports attaching a control to modify entry operations over LDAP that results in: <ul style="list-style-type: none"> <li>• Returning SUCCESS instead of a "no such attribute or value" error if the client request attempts to delete a non-existing attribute value.</li> <li>• Returning SUCCESS instead of an "attribute or attribute value already exists" error if the client request attempts to add an already existing attribute value.</li> </ul> The abbreviation for this control is <b>RELAXEDUPD</b> . |

|                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.2.840.113556.1.4.319    | Simple Paged Results for request and response. The abbreviation for this control is <b>SPAGRES</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 1.3.12.2.1107.1.3.2.12.6  | <p>LDAP Max Paging Timeout Control. This is a proprietary control. It directs the LDAP server to use a 3600 second (1 hour) timeout in paged search requests instead of the configured value in the PAGP (Paging-Policy) attribute.</p> <p>The abbreviation for this control is <b>MAXPAGTO</b>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 1.3.12.2.1107.1.3.2.12.7  | <p>LDAP Use MT, MN, CRT, CRN from DAP Request Control. This is a proprietary control. The server uses the modifyTimestamp (MT), the modifiersName (MN), the createTime (CRT), and the creatorsName (CRN) from the DAP request.</p> <p>The abbreviation for this control is <b>USEREQOPATTR</b>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 1.3.12.2.1107.1.3.2.12.9  | <p>LDAP No Modification Time Update Control. This is a proprietary control. The server does not update the modifyTimestamp (MT).</p> <p>The abbreviation for this control is <b>NOMODTIMEUPD</b>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 1.3.12.2.1107.1.3.2.12.10 | <p>LDAP Conditioned Operation Control. This is a proprietary control. A modify or delete operation is performed only if the condition specified in the control's value applies. The control's value must be specified.</p> <p>The abbreviation for this control is <b>CONDOP</b>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 1.3.12.2.1107.1.3.2.12.11 | <p>LDAP Search Result Info Control. This is a proprietary control. A search operation returns information about the search result like the number of result entries instead of the result.</p> <p>The abbreviation for this control is <b>SEARCHRESINFO</b>.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 1.3.12.2.1107.1.3.2.12.12 | <p>LDAP Incremental Replace Control. This is a proprietary control. The server increments (decrements negative values) attributes with syntax integer when performing a replace in a modify entry operation. The attribute must not be multi-valued. The control has no value. The default for <i>criticality</i> is <b>1</b> and cannot be changed. If none of the updates is a replace or none of the replaced attributes has the syntax integer, the control is ignored.</p> <p>For example,<br/> <b>dirxcp&gt; modify cn=MueUer,ou=Sales,o=My-Company -control INCRREPL -replaceattr -intAttr=*number increments the value of the attribute *intAttr of the entry cn=MueUer,ou=Sales,o=My-Company by number.</b></p> <p>The abbreviation for this control is <b>INCRREPL</b>.</p> |

|                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.2.840.113556.1.4.319    | Simple Paged Results for request and response. The abbreviation for this control is SPAGRES.                                                                                                                                                                                                                                                                                                                                                                                 |
| 1.3.12.2.1107.1.3.2.12.13 | <p>LDAP Matched Value Only Filter Control. This is a proprietary control. A search operation returns only those requested attribute values that match to the filter in the control's value. (See <b>DirX Directory String Representation for LDAP Binds -&gt; Search Filters -&gt; LDAP Matched Value Only Filter Control</b> for details.) For example, it can be used to retrieve one particular certificate.</p> <p>The abbreviation for this control is <b>MVOF</b>.</p> |

The DirX Directory DSA supports the controls for search request and responses when the suitable indices are created in the DSA's database.

If an LDAP search operation requests server-side sorting, the search operation must be performed with "dereference aliases always" or the DSA returns an "**Unwilling to Perform**" error. This behavior is independent from whether or not the database actually contains alias entries.

Be aware that using some of the listed LDAP controls may break the server behavior described in the LDAP/X.500 standards.

#### 3.1.12.4.1. Abbreviation

SCON

#### 3.1.12.4.2. LDAP Name(s)

supportedControl

#### 3.1.12.4.3. Syntax (DAP)

Object Identifier

#### 3.1.12.4.4. Syntax (LDAP)

Object Identifier (LDAP style) in dotted notation.

#### 3.1.12.4.5. Example (DAP)

```
{SCON=2.16.840.1.113730.3.4.9;1.2.840.113556.1.4.473;1.2.840.113556.1.4.474;1.2.840.113556.1.4.319;2.16.840.1.113730.3.4.10}
```

#### 3.1.12.4.6. Example (LDAP)

```
{supportedControl=2.16.840.1.113730.3.4.9;1.2.840.113556.1.4.473;1.2.
```



```
840.113556.1.4.474;1.2.840.113556.1.4.319;2.16.840.1.113730.3.4.10}
```

### 3.1.12.5. Supported SASL Mechanism

The Supported SASL Mechanism attribute specifies the names of the SASL authentication mechanisms that the LDAP server supports. This attribute is an optional attribute. (See *RFC 2222* for details on SASL.)

The LDAP server supports the EXTERNAL SASL (attribute value **EXTERNAL**) mechanism. This mechanism type means that the LDAP server supports certificate-based client authentication based on SSL/TLS.

#### 3.1.12.5.1. Abbreviation

SSME

#### 3.1.12.5.2. LDAP Name(s)

supportedSASLMechanisms

#### 3.1.12.5.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.12.5.4. Example (DAP)

```
SSME=EXTERNAL
```

#### 3.1.12.5.5. Example (LDAP)

```
supportedSASLMechanisms=EXTERNAL
```

### 3.1.12.6. Supported LDAP Versions

The Supported LDAP Versions attribute specifies the LDAP protocol versions that the LDAP server supports. This attribute is a mandatory attribute. Valid values are **2**, and **3**.

#### 3.1.12.6.1. Abbreviation

SLVE

#### 3.1.12.6.2. LDAP Name(s)

SupportedLDAPVersion

#### 3.1.12.6.3. Syntax (DAP)

Integer

#### 3.1.12.6.4. Syntax (LDAP)

Integer String (LDAP style)

#### 3.1.12.6.5. Example (DAP)

```
SLVE=2;3    LDAP server supports LDAP protocol versions 2 and 3
```

#### 3.1.12.6.6. Example (LDAP)

```
SupportedLDAPVersion=2;3
```

### 3.1.12.7. Og Supported Profile

The Og Supported Profile attribute specifies the object identifiers of the profiles that the LDAP server supports. This attribute is a mandatory attribute.

#### 3.1.12.7.1. Abbreviation

OSPR

#### 3.1.12.7.2. LDAP Name(s)

ogSupportedProfile

#### 3.1.12.7.3. Syntax (DAP)

Object Identifier

#### 3.1.12.7.4. Syntax (LDAP)

Object Identifier (LDAP style) in dotted notation.

#### 3.1.12.7.5. Example (DAP)

```
OSPR=1.2.826.0.1050.11.1.1; 1.2.826.0.1050.11.2.1  
LDAP server supports the Open Group Profiles Read-Only LDAP server  
and Read-Write LDAP server
```

#### 3.1.12.7.6. Example (LDAP)

```
ogSupportedProfile=1.2.826.0.1050.11.1.1; 1.2.826.0.1050.11.2.1
```

### 3.1.12.8. Vendor Name

The Vendor Name attribute specifies the name of the LDAP server implementer. This attribute is a single-valued, optional attribute.

#### 3.1.12.8.1. Abbreviation

RFCVN

#### 3.1.12.8.2. LDAP Name(s)

vendorName

#### 3.1.12.8.3. Syntax (DAP)

Directory-String

#### 3.1.12.8.4. Syntax (LDAP)

Directory-String (LDAP style)

#### 3.1.12.8.5. Example (DAP)

```
RFCVN=Eviden
```

#### 3.1.12.8.6. Example (LDAP)

```
vendorName=Eviden
```

### 3.1.12.9. Vendor Version

The Vendor Version attribute specifies the version of the LDAP server. This attribute is a single-valued, optional attribute.

#### 3.1.12.9.1. Abbreviation

RFCVV

#### 3.1.12.9.2. LDAP Name(s)

vendorVersion

#### 3.1.12.9.3. Syntax (DAP)

Directory-String

#### 3.1.12.9.4. Syntax (LDAP)

Directory-String (LDAP style)

#### 3.1.12.9.5. Example (DAP)

```
RFCVV=DirX Directory V8.2 A 8.6.146 2011:05:24 20:10 32-Bit
```

#### 3.1.12.9.6. Example (LDAP)

```
vendorVersion=DirX Directory V8.2 A 8.6.146 2011:05:24 20:10 32-Bit
```

### 3.1.12.10. LDAP Subschema Subentry

The LDAP Subschema Subentry attribute specifies the distinguished names of the LDAP subschema subentries. This attribute is a mandatory attribute.

#### 3.1.12.10.1. Abbreviation

LSSU

#### 3.1.12.10.2. LDAP Name(s)

subSchemaSubentry

#### 3.1.12.10.3. Syntax (DAP)

Distinguished Name

#### 3.1.12.10.4. Syntax (LDAP)

Distinguished Name (LDAP Style)

#### 3.1.12.10.5. Example (DAP)

```
LSSU={/CN=Schema}
```

#### 3.1.12.10.6. Example (LDAP)

```
subSchemaSubentry=cn=Schema
```

### 3.1.13. Attributes of the Schema Subentry

The schema information (attribute types and object classes) is saved in the attributes of the schema subentry. The schema subentry is created automatically by the DirX Directory DSA. LDAP clients usually read this subentry to get LDAP server's schema information. The distinguished name (DN) of the schema subentry is saved in the LDAP Subschema Subentry attribute of the LDAP root subentry and must consist of only one relative distinguished name (RDN). By default the name of the schema subentry is **cn=Schema**.

It is not recommended to modify the default name of the schema. However there may be reasons that you must rename the schema. In this case, you modify the LDAP Subschema Subentry attribute value of the LDAP root subentry. After modifying this value, you must restart the DirX Directory Service so that the modification becomes effective. The DSA will rename the schema subentry automatically.

Note that the LDAP root subentry contains local information and modifications to this entry are not propagated by shadowing. You must perform the modification to the LDAP Subschema Subentry attribute value on all servers participating in your network. Schema modifications, however, are propagated by shadowing.

The schema must be legal and consistent for all servers to work correctly. All attribute types and all object classes of the schema must have an LDAP name. The LDAP server does not start if it detects an attribute type or object class in the schema without an LDAP name assigned. If an LDAP client uses attributes not present in the schema, they will be ignored in the request. (See section titled **Schema Check** in chapter titled **DirX Directory Default DSA Schema** in the *DirX Directory Administration Reference* how to perform a schema check.)

Schema modifications become effective immediately in the DSA and in the LDAP server where the schema has been changed. Additional LDAP servers must be restarted. If the schema is changed not via the LDAP server - for example, new attributes have been added by **dirxcp** after a DAP bind - all LDAP servers must be restarted. (See the **dirxadm ldap restart** operation in the chapter **DirX Directory Commands** in the *DirX Directory Administration Reference* for details.)

The LDAP schema subentry can be retrieved by **dirxcp** via LDAP by performing the following request:

**obj show cn=Schema -allattr**

Like all other subentries, it cannot be created via LDAP. However, it can be modified via LDAP by

- performing **dirxcp obj** operations after performing an LDAP bind
- importing an LDIF content file containing schema information
- DirX Directory Manager.

### 3.1.13.1. LDAP AT

The LDAP AT attribute is a multi-valued attribute specifying the attribute types used within the schema. The component **NAME *\*attribute\_type\_name\**** provides the LDAP names of the attribute type. (See the section titled **Attribute-Type-Description** in the chapter **DirX Directory String Representation for LDAP Binds** for details.)

#### 3.1.13.1.1. Abbreviation

AT

#### 3.1.13.1.2. LDAP Name(s)

attributeTypes, at

#### 3.1.13.1.3. Syntax (LDAP)

Attribute-Type-Description (LDAP style)

#### 3.1.13.1.4. Example (LDAP)

```
attributeTypes=
....
( 2.5.4.20 NAME 'MYtelephoneNumber' EQUALITY telephoneNumberMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.44 {32} )
....
```

### 3.1.13.2. LDAP OC

The LDAP OC attribute is a multi-valued attribute specifying the object classes used within the schema. The component **NAME *"object\_class\_name"*** provides the LDAP name of the object class. You can specify only one LDAP name for an object class. (See the section titled **Object-Class-Description** in the chapter **DirX Directory String Representation for LDAP Binds** for details.)

#### 3.1.13.2.1. Abbreviation

OC

#### 3.1.13.2.2. LDAP Name(s)

objectClasses, oc

#### 3.1.13.2.3. Syntax (LDAP)

Object-Class-Description (LDAP style)

#### 3.1.13.2.4. Example (LDAP)

```
objectClasses=
....
( 2.5.6.16 NAME 'MYcertificationAuthority' SUP top AUXILIARY MUST
(cACertificate $ certificateRevocationList $ authorityRevocationList)
MAY crossCertificatePair )
....
```

### 3.1.14. Attributes for LDAP Server Configuration

Attributes for LDAP server configuration are attributes of the LDAP configuration subentry with the default name **/CN=ldapConfiguration**. Several LDAP configuration subentries may be present within the same database with unique common names. Specify the Common Name of the LDAP configuration subentry of the first / default LDAP server to be started in the environment variable `DIRX_DEFAULT_LDAP_SERVER`; if not, the default LDAP configuration subentry with the name **ldapConfiguration** should be used.

Run the script **ldap\_admin.cp** (using **dirxcp**) to create a default LDAP configuration subentry that specifies default values for the DirX Directory LDAP server. Use the **dirxcp obj** operations to modify the attributes of the default LDAP configuration subentry or to create new LDAP configuration subentries. Like all other subentries, it cannot be created via LDAP. However, it can be modified via LDAP.

Modifications of an LDAP configuration subentry become effective after a restart of the LDAP server. (See the **dirxadm ldap restart** operation in the *DirX Directory Administration Reference* for details.)

The attributes of the LDAP configuration subentry

- specify the service controls that the LDAP server uses when sending an operation to the DSA (LDAP Search Service Controls, LDAP Compare Service Controls, LDAP Add Service Controls, LDAP Remove Service Controls, LDAP Modify Service Controls, LDAP Modify DN Service Controls).
- control the options of an LDAP search request:
  - the maximum number of requested attributes (LDAP Maximum Requested Attributes)
  - the maximum number of filter items (LDAP Maximum Filter Items)
- control connection handling:
  - the TCP port number (LDAP Port Number)
  - the secure port number (LDAP Secure Port Number)
  - the name of the SSL configuration subentry (LDAP SSL configuration subentry CN)
  - whether or not clients can switch from a normal (plain) LDAP bind to a secure LDAP bind (LDAP Start TLS Enabled)
  - the maximum number of simultaneous connections (LDAP Anonymous Connections)
  - the number of simultaneous anonymous connections (LDAP Anonym DAP Pool Size)
  - the number of seconds that an idle client connection remains open (LDAP Connection Idle Time)
  - the number of seconds that an authenticated connection remains open after the last performed unbind (LDAP Unbind Delay Time)
  - whether clients can share the same connection (LDAP Backend Sharing, LDAP Maximum Share Count)

- whether the LDAP server grants or denies access to clients with a specific IP address or distinguished name, or denies anonymous or authenticated access (LDAP Allow IP List, LDAP Deny IP List, LDAP Allowed Users List, LDAP Allowed Groups, LDAP Denied Users List, LDAP Denied Groups, LDAP Deny Anonymous Access, LDAP Only Allow Anonymous Access)
- policies for users (LDAP User Policies) or groups (LDAP Group Policies)
- whether clients can bind to the LDAP server only over a specific IP address (LDAP Listen IP List)
- control the encoding and conversion of attributes (LDAP ASN1 Header, LDAP Charset Conv Request, LDAP Charset Conv Result).
- control the operations that the LDAP server is willing to perform (LDAP Read Only Server).
- control the thread pool size (LDAP Thread Pool Size).
- control the LDAP flow to avoid capacity overload over the socket layer:
  - the TCP keep alive (LDAP Socket Keep Alive)
  - the use of non-blocking sockets (LDAP Asynchronous Sockets)
  - the maximum timeout for send and receive operation calls (LDAP Max Send Timeout, LDAP Max Recv Timeout)
  - the maximum number of incomplete operations per connection (LDAP Max Incomplete Ops per Connection)
  - the operation stack size limit (LDAP Op Stack Limit)
  - the number of overflow threads (LDAP Overflow Threads)
- control the LDAP cache:
  - enable or disable the LDAP cache (LDAP Cache)
  - control the storage of search results (LDAP Cache Max Cached Elements, LDAP Cache Maximum Memory Size, LDAP Cache Min Cache Time, LDAP Cache Max Cache Time, LDAP Cache Min Cache Entries, LDAP Cache Max Cache Entries, LDAP Cache Min Attributes, LDAP Cache Max Attributes, LDAP Cache Min Values, LDAP Cache Max Values, LDAP Cache Table Size, LDAP Cache Update Strategy)
- specify the name of the LDAP audit configuration subentry (LDAP Audit Configuration Subentry CN).
- control the accessibility of LDAP extended operations:
  - specify the users and groups that can perform LDAP extended operations (LDAP Extended Operation Admins, LDAP Extended Operation Read Users, LDAP Extended Operations Execute Users, LDAP Extended Operations Monitoring Users, LDAP Extended Operation Admin Groups, LDAP Extended Operation Read Groups, LDAP Extended Operations Execute Groups, LDAP Extended Operations Monitoring Groups)
  - specify the LDAP extended read, execute and monitoring operations (LDAP Extended Read Operations, LDAP Extended Execute Operations, LDAP Extended Monitoring Operations)



The following sections provide information about these attributes.

#### 3.1.14.1. LDAP Search Service Controls

The LDAP Search Service Controls attribute specifies the service controls that apply to the search operation. This attribute is an optional, single-valued attribute. Specify the value in the following format:

**PreferChaining=Flag:ChainingProhibited=Flag:LocalScope=Flag:DontUseCopy=Flag:DontDereferenceAlias=Flag:Subentries=Flag:CopyShallDo=Flag:Priority=Integer:\_TimeLimit=\_Integer:SizeLimit=Integer:ScopeOfReferral=Integer:AttributeSizeLimit=Integer**

where:

##### **PreferChaining=Flag**

indicates whether chaining is preferred, rather than referrals. Specify **T** (true - indicates to prefer chaining) or **F** (false - indicates to prefer referrals). The default value is **T** (prefer chaining).

##### **ChainingProhibited=Flag**

indicates whether chaining and other methods of distributing the request around the Directory are prohibited. Specify **T** (true - indicates to prohibit chaining) or **F** (false - indicates to permit chaining). The default value is **F** (permit chaining).

##### **LocalScope=Flag**

indicates whether the operation is to be limited to a local scope as specified in the **Local Scope Policy**. (See **Local Scope Policy** attribute and **Local Scope Policy Syntax** for details.) Specify **T** (true - indicates limitation to a local scope) or **F** (false - indicates no limitation to a local scope). The default value is **F** (no limitation to a local scope).

##### **DontUseCopy=Flag**

indicates whether copied (shadowed) information shall not be used to provide the service. Specify **T** (true - indicates not to use copies) or **F** (false - indicates to use copies). The default value is **T** (not to use copies) and **F** (to use copies) for LDAP Search and Compare Service Controls attributes.

##### **DontDereferenceAlias=Flag**

indicates whether an alias entry used to identify the entry affected by an operation is not to be dereferenced. This is necessary to refer to an alias entry itself rather than the aliased entry. In modify operations, alias dereferencing is not supported. Specify **F** (false - indicates to dereference aliases) or **T** (true - indicates not to dereference aliases). The default value is **F** (dereference aliases).

##### **Subentries=Flag**

indicates whether a search or list operation returns only subentries and normal entries become inaccessible. Specify **T** (true - indicates to return only subentries) or **F** (false - indicates to return only normal entries). The default value is **F** (return only normal entries).

### ***CopyShallDo=Flag***

indicates, when an operation is able to partly but not fully satisfy a query at a copied (shadowed) information of an entry, whether it need not chain the query. This service control is meaningful only if **DontUseCopy=F**. Specify **T** (true - indicates to permit operations that do not fully satisfy a query with shadowed information) or **F** (false - indicates to prohibit operations that do not completely satisfy a query with shadowed information and to chain the query).

The default value is **F** (partial results using copied information are prohibited) except for LDAP Search and Compare Service Controls attributes. There the default value is **T** (partial results using copied information are permitted).

### ***Priority=Integer***

specifies the priority at which the service is to be provided. Specify one of the following values:

- 0** - indicates low priority.
- 1** - indicates medium priority.
- 2** - indicates high priority.

The default value is **1** (medium).

### ***TimeLimit=Integer***

specifies the maximum elapsed time in seconds for completion of an operation. Specify a non-negative integer. A value of **0** (zero) indicates no time limit. The default value is **0**.

### ***SizeLimit=Integer***

specifies the maximum number of objects returned for a list or search operation. Specify a non-negative integer. A value of **0** (zero) indicates no size limit. The default value is **0**.

### ***ScopeOfReferral=Integer***

specifies the scope to which a referral returned by a DSA should be relevant. Specify one of the following values:

- 0** - indicates **DMD** (Directory Management Domain).
- 1** - indicates **Country**.
- 2** - indicates **Unlimited**.

The default value is **2** (no limitation).

### ***AttributeSizeLimit=Integer***

specifies the maximum size (in octets) of any attribute (the type and all its values) to be returned. If an attribute exceeds this limit, all of its values are omitted. Specify a non-negative integer. A value of **0** (zero) indicates no size limit. The default value is **0**.

#### **3.1.14.1.1. Abbreviation**

LSSES

#### 3.1.14.1.2. LDAP Name(s)

ldapSearchSvcCtl

#### 3.1.14.1.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.1.4. Example (DAP)

```
LSSES={SizeLimit=11:ScopeOfReferral=1:PreferChaining=F}
```

#### 3.1.14.1.5. Example (LDAP)

```
ldapSearchSvcCtl=SizeLimit=11:ScopeOfReferral=1:PreferChaining=F
```

### 3.1.14.2. LDAP Compare Service Controls

The LDAP Compare Service Controls attribute specifies the service controls that apply to the compare operation. This attribute is an optional, single-valued attribute. Specify the value in the same format as described for the **LDAP Search Service Controls** (see this section for details).

#### 3.1.14.2.1. Abbreviation

LCOS

#### 3.1.14.2.2. LDAP Name(s)

ldapCompareSvcCtl

#### 3.1.14.2.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.2.4. Example (DAP)

```
LCOS={SizeLimit=11:ScopeOfReferral=1:PreferChaining=F}
```

#### 3.1.14.2.5. Example (LDAP)

```
ldapCompareSvcCtl=SizeLimit=11:ScopeOfReferral=1:PreferChaining=F
```

### 3.1.14.3. LDAP Add Service Controls

The LDAP Add Service Controls attribute specifies the service controls that apply to the create operation. This attribute is an optional, single-valued attribute. Specify the value in the same format as described for the **LDAP Search Service Controls** (see there for details).

#### 3.1.14.3.1. Abbreviation

LADS

#### 3.1.14.3.2. LDAP Name(s)

ldapAddSvcCtl

#### 3.1.14.3.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.3.4. Example (DAP)

```
LADS={SizeLimit=11:ScopeOfReferral=1:PreferChaining=F}
```

#### 3.1.14.3.5. Example (LDAP)

```
ldapAddSvcCtl=SizeLimit=11:ScopeOfReferral=1:PreferChaining=F
```

### 3.1.14.4. LDAP Remove Service Controls

The LDAP Remove Service Controls attribute specifies the service controls that apply to the remove/delete operation. This attribute is an optional, single-valued attribute. Specify the value in the same format as described for the **LDAP Search Service Controls** (see this section for details).

#### 3.1.14.4.1. Abbreviation

LRES

#### 3.1.14.4.2. LDAP Name(s)

ldapRemoveSvcCtl

#### 3.1.14.4.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.4.4. Example (DAP)

```
LRES={SizeLimit=11:ScopeOfReferral=1:PreferChaining=F}
```

#### 3.1.14.4.5. Example (LDAP)

```
ldapRemoveSvcCtl=SizeLimit=11:ScopeOfReferral=1:PreferChaining=F
```

### 3.1.14.5. LDAP Modify Service Controls

The LDAP Modify Service Controls attribute specifies the service controls that apply to the modify operation. This attribute is an optional, single-valued attribute. Specify the value in the same format as described for the **LDAP Search Service Controls** (see there for details).

#### 3.1.14.5.1. Abbreviation

LMOS

#### 3.1.14.5.2. LDAP Name(s)

ldapModifySvcCtl

#### 3.1.14.5.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.5.4. Example (DAP)

```
LMOS={SizeLimit=11:ScopeOfReferral=1:PreferChaining=F}
```

#### 3.1.14.5.5. Example (LDAP)

```
ldapModifySvcCtl=SizeLimit=11:ScopeOfReferral=1:PreferChaining=F
```

### 3.1.14.6. LDAP ModifyDN Service Controls

The LDAP ModifyDN Service Controls attribute specifies the service controls that apply to the modify DN operation. This attribute is an optional, single-valued attribute. Specify the value in the same format as described for the **LDAP Search Service Controls** (see there for details).

#### 3.1.14.6.1. Abbreviation

LMDS

#### 3.1.14.6.2. LDAP Name(s)

ldapModifyDNSvcCtl

#### 3.1.14.6.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.6.4. Example (DAP)

```
LMDS={SizeLimit=11:ScopeOfReferral=1:PreferChaining=F}
```

#### 3.1.14.6.5. Example (LDAP)

```
ldapModifyDNSvcCtl=SizeLimit=11:ScopeOfReferral=1:PreferChaining=F
```

### 3.1.14.7. LDAP Maximum Requested Attributes

The LDAP Maximum Requested Attributes attribute specifies the maximum number of requested attributes that a client is allowed to specify in an LDAP search request. This attribute is an optional, single-valued attribute.

Huge lists of requested attributes can significantly increase the request size, the audit record size and the processing time of an LDAP search operation. You can use this attribute to avoid these situations and to block malicious client applications. If a search request contains more than the specified maximum, the search is rejected with an **administrative limit exceeded** error code and a corresponding LDAP message.

If this attribute is not specified or has a value of **0**, the default is to allow an unlimited number of requested attributes.

When specified in an LDAP search operation, the LDAP-defined abbreviations **\*** (all user attributes) and **+** (all operational attributes) are treated as one requested attribute each.

#### 3.1.14.7.1. Abbreviation

LMAXRA

#### 3.1.14.7.2. LDAP Name(s)

ldapMaxReqAttrs

#### 3.1.14.7.3. Syntax (DAP)

Integer

#### 3.1.14.7.4. Example (DAP)

```
LMAXRA=100
```

#### 3.1.14.7.5. Example (LDAP)

```
ldapMaxReqAttrs=100
```

#### 3.1.14.8. LDAP Maximum Filter Items

The LDAP Maximum Filter Items attribute specifies the maximum number of filter Items that a client is allowed to specify in the filter of an LDAP search request. This attribute is an optional, single-valued attribute.

A huge number of filter items can significantly increase the request size, the audit record size and the processing time of an LDAP search operation. You can use this attribute to avoid these situations and to block malicious client applications. If a search request contains a filter expression with more than the specified number of filter items, the search is rejected with an **administrative limit exceeded** error code and a corresponding LDAP message.

If this attribute is not specified or has a value of **0**, the default is to allow an unlimited number of filter items is.

##### 3.1.14.8.1. Abbreviation

LMAXFI

##### 3.1.14.8.2. LDAP Name(s)

ldapMaxFilterItems

##### 3.1.14.8.3. Syntax (DAP)

Integer

##### 3.1.14.8.4. Example (DAP)

```
LMAXFI=128
```

##### 3.1.14.8.5. Example (LDAP)

```
ldapMaxFilterItems=128
```

#### 3.1.14.9. LDAP Port Number

The LDAP Port Number attribute specifies the TCP port number on which the LDAP server is to listen. This attribute is an optional, single-valued attribute. The default value is **389**.

#### 3.1.14.9.1. Abbreviation

LPNU

#### 3.1.14.9.2. LDAP Name(s)

ldapPortNumber

#### 3.1.14.9.3. Syntax (DAP)

Integer

#### 3.1.14.9.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.14.9.5. Example (DAP)

```
LPNU=6666
```

#### 3.1.14.9.6. Example (LDAP)

```
ldapPortNumber=6666
```

### 3.1.14.10. LDAP Secure Port Number

The LDAP Secure Port Number attribute specifies the port number on which the LDAP server is to listen for Secure Socket Layer (SSL) secured LDAP protocol. A value of **0** indicates that SSL is disabled. To enable SSL, it is recommended to specify the value **636** because this is the standard port for SSL. This attribute is an optional, single-valued attribute. The default value is **0** (SSL is turned off).

When SSL is enabled, the LDAP server tries to read also the LDAP SSL configuration subentry. The distinguished name of the LDAP SSL configuration subentry is specified in the LDAP SSL Configuration Subentry attribute of the LDAP configuration subentry. (See also the sections **LDAP SSL Configuration Subentry** and **Attributes for LDAP Server SSL Configuration**.)

#### 3.1.14.10.1. Abbreviation

LSPN

#### 3.1.14.10.2. LDAP Name(s)

ldapSecurePortNumber

#### 3.1.14.10.3. Syntax (DAP)

Integer



#### 3.1.14.10.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.14.10.5. Example (DAP)

```
LSPN=636
```

#### 3.1.14.10.6. Example (LDAP)

```
ldapSecurePortNumber=636
```

### 3.1.14.11. LDAP SSL Configuration Subentry CN

The LDAP SSL Configuration Subentry CN attribute specifies the common name attribute of the LDAP SSL configuration subentry. This attribute is an optional single-valued attribute. When this attribute is omitted the LDAP server reads the LDAP SSL configuration subentry with the default common name value **LdapSSLConfiguration**.

#### 3.1.14.11.1. Abbreviation

LSCCN

#### 3.1.14.11.2. LDAP Name(s)

ldapSSLCfgSubentryCN

#### 3.1.14.11.3. Syntax (DAP)

Printable String

#### 3.1.14.11.4. Syntax (LDAP)

Printable String (LDAP style)

#### 3.1.14.11.5. Example (DAP)

```
LSCCN={LDAPSSLConfigurationSubentry}
```



Do not specify the 'CN' attribute type in the value!

#### 3.1.14.11.6. Example (LDAP)

```
ldapSSLCfgSubentryCN=LDAPSSLConfigurationSubentry
```



Do not specify the 'cn' attribute type in the value!

### 3.1.14.12. LDAP Start TLS Enabled

The LDAP Start TLS Enabled attribute specifies whether or not the LDAP extended operation startTLS is allowed. This operation allows a client to turn on the SSL/TLS security layer for an LDAP connection that was created as a normal (plain) LDAPv3 connection. See also the section **DirX Directory Commands obj (dirxcp)** in the *DirX Directory Administration Reference*.

A value of 1 means that the DirX Directory LDAP server accepts a startTLS extended operation. A value of 0 means that startTLS is disabled in the configuration.

Note that you are allowed to enable the startTLS operation even though the LDAP server does not provide secure access via the Secure LDAP Port (LDAPS); that is, the LDAP Start TLS Enabled attribute can be set to 1, while the LDAP Secure Port Number can be set to 0.

Note also that a client does not need to have the permission to run LDAP extended operations (via the LDAP Extended Operation Admins attribute) in order to perform the startTLS operation. If LDAP Start TLS Enabled is set to 1, all clients are allowed to perform the startTLS operation.

If the LDAP Start TLS Enabled attribute is set to 1, the LDAP server tries to read the LDAP SSL configuration subentry at startup. As a result, the appropriate SSL key material must be provided.

#### 3.1.14.12.1. Abbreviation

LSTEN

#### 3.1.14.12.2. LDAP Name(s)

ldapStartTLSEnabled

#### 3.1.14.12.3. Syntax (DAP)

Integer

#### 3.1.14.12.4. Syntax (LDAP)

Integer (LDAP Style)

#### 3.1.14.12.5. Example (DAP)

```
LSTEN=1
```

#### 3.1.14.12.6. Example (LDAP)

```
ldapStartTLSEnabled=0
```

### 3.1.14.13. LDAP Max Connections

The LDAP Max Connections attribute specifies the maximum number of simultaneous LDAP connections that the LDAP server supports. This attribute is an optional, single-valued attribute. The default value is **4000**.

System resources, the value of this attribute, and the behavior concerning LDAP backend sharing determine the maximum number of simultaneous connections that can be established. See the section titled **LDAP Backend Sharing** for details.) The value of this attribute also includes the number of connections established for anonymous binds. (See the section titled **LDAP Anonym DAP Pool Size** for details.)

The system file descriptor limit must be set accordingly (**ulimit**) to support the configured value. On Windows systems, a value of **4000** is appropriate if the IDM stack is used. (See the environment variable **DIRX\_OWN\_PSAP** in the *DirX Directory Administration Reference* for information on how to enable the IDM stack.) On Linux systems, a value of **4000** is appropriate if the IDM stack is used and the file system descriptor limit is set to **8192** (**ulimit -n 8192**).

#### 3.1.14.13.1. Abbreviation

LMCO

#### 3.1.14.13.2. LDAP Name(s)

ldapMaxConnections

#### 3.1.14.13.3. Syntax (DAP)

Integer

#### 3.1.14.13.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.14.13.5. Example (DAP)

```
LMCO=124
```

#### 3.1.14.13.6. Example (LDAP)

```
ldapMaxConnections=124
```

#### 3.1.14.14. LDAP Anonym DAP Pool Size

The LDAP Anonym DAP Pool Size attribute is provided to configure the number of simultaneous DAP connections that the LDAP server establishes to the DSA for anonymous binds. Additionally the LDAP server establishes one DAP connection for internal operations. This attribute is an optional, single-valued attribute. The default value is **5**.

Increasing the value of this attribute may result in improving the performance for anonymous binds. However the maximum number of connections established for authenticated binds decreases. The maximum number of simultaneous connections that can be established is determined by system resources, by the maximum number of connections specified in the LDAP Max Connections attribute, and by the behavior concerning LDAP Backend Sharing.

Modifications to this attribute become effective only after a restart of the LDAP server.

##### 3.1.14.14.1. Abbreviation

LADPS

##### 3.1.14.14.2. LDAP Name(s)

ldapAnonymDapPoolSize

##### 3.1.14.14.3. Syntax (DAP)

Integer

##### 3.1.14.14.4. Syntax (LDAP)

Integer-String (LDAP style)

##### 3.1.14.14.5. Example (DAP)

```
LADPS=3
```

##### 3.1.14.14.6. Example (LDAP)

```
ldapAnonymDapPoolSize=3
```

#### 3.1.14.15. LDAP Connection Idle Time

The LDAP Connection Idle Time attribute specifies the number of seconds of client inactivity after which the LDAP server is to close an LDAP connection to an LDAP client. This attribute is an optional, single-valued attribute. The default value is 300 seconds.

##### 3.1.14.15.1. Abbreviation

LCIT

#### 3.1.14.15.2. LDAP Name(s)

ldapConnectionIdleTime

#### 3.1.14.15.3. Syntax (DAP)

Integer

#### 3.1.14.15.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.14.15.5. Example (DAP)

```
LCIT=600
```

#### 3.1.14.15.6. Example (LDAP)

```
ldapConnectionIdleTime=600
```

### 3.1.14.16. LDAP Unbind Delay Time

By default, the LDAP server tries to share DAP backend connections among the same authenticated LDAP client connections; for example, two LDAP clients bound with the same credentials use the same DAP backend connection. The LDAP Unbind Delay Time attribute specifies whether authenticated DAP binds shall be kept for a certain time or shall be unbound immediately by the LDAP server if no references to a specific bind are present. This attribute is an optional, single-valued attribute. Specify one of the following values:

**0** - close authenticated DAP connections immediately after the last LDAP client using that connection has unbound.

$n > 0$  - keep authenticated DAP connections for at least  $n$  seconds after the last LDAP client using that connection has unbound.

The default value is **0**.

The value of this attribute does not affect anonymous binds.

It is recommended to close all authenticated binds immediately at unbind time if password lockout policy (Password Lockout attribute of the password policy subentry) or password aging policy (Password Maximum Age attribute of the password policy subentry) is enabled, because existing binds are not closed automatically if accounts are locked out or passwords expire due to the password policy enabled. New performed binds using locked out accounts or expired passwords are accepted if an authenticated DAP connection using these credentials is already open. (See the section titled **Attributes of the Password Policy Subentry** for details.)

#### 3.1.14.16.1. Abbreviation

LUDT

#### 3.1.14.16.2. LDAP Name(s)

ldapUnbindDelayTime

#### 3.1.14.16.3. Syntax (DAP)

Integer

#### 3.1.14.16.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.14.16.5. Example (DAP)

```
LUDT=60
```

#### 3.1.14.16.6. Example (LDAP)

```
ldapUnbindDelayTime=60
```

### 3.1.14.17. LDAP Backend Sharing

The LDAP Backend Sharing attribute type specifies whether

- Clients with the same credentials share the same DAP connection and
- Newly performed binds with existing credentials are accepted as long as at least one bind with these credentials exists. These binds are accepted even though the password has been modified and the client uses the old password.

or

- Clients with the same credentials do not share the same DAP connection and
- Newly performed binds are rejected if the client uses a password that has been changed while other clients are still bound with this password.

Note that binds exist until an unbind operation is performed or when there has been no operation performed for a specific time. (See the **LDAP Connection Idle Time** and the **LDAP Unbind Delay Time** attribute sections in this chapter for details.) Existing binds are not aborted when the password is modified.

Specify one of the following values:

**TRUE** - Newly performed binds using the old password are accepted as long as at least one bind with the old password exists. Clients with identical credentials share the same

connection.

**FALSE** - Newly performed binds using the old password are rejected. Each client uses its own connection. Note that this increases the use of system resources.

The default value is **TRUE**.

The value of this attribute does not affect anonymous binds.

It is recommended to disable backend sharing if the password lockout policy (Password Lockout attribute of the password policy subentry) or the password aging policy (Password Maximum Age attribute of the password policy subentry) are enabled because there is no automatic disabling of backend sharing if accounts are locked out or passwords expire due to the enabled password policy. New performed binds using locked-out accounts or expired passwords are accepted if an authenticated DAP connection using these credentials is already open. (See the section titled **Attributes of the Password Policy Subentry** for details.)

#### 3.1.14.17.1. Abbreviation

LBS

#### 3.1.14.17.2. LDAP Name(s)

ldapBackendSharing

#### 3.1.14.17.3. Syntax (DAP)

Boolean

#### 3.1.14.17.4. Syntax (LDAP)

Boolean (LDAP Style)

#### 3.1.14.17.5. Example (DAP)

```
LBS=FALSE
```

#### 3.1.14.17.6. Example (LDAP)

```
ldapBackendSharing=FALSE
```

### 3.1.14.18. LDAP Maximum DAP Share Count

The single-valued LDAP Maximum DAP Share Count attribute type specifies how many LDAP users can share the same DAP connection when they authenticate with the same credentials if LDAP backend sharing is enabled. (See the **LDAP Backend Sharing** attribute section of this chapter for details.)

The default value is **5**. Using more DAP connections can increase the throughput but also increases the resource usage for file descriptors.

The value of this attribute does not affect anonymous binds.

The values of this attribute are also stored in the LDAP MIB static table. (See the **ldap mib** operation described in the **DirX Directory Commands** chapter and the **LDAP MIB Static Table** section in the **LDAP MIB Tables** appendix in the *DirX Directory Administration Reference* for details.)

#### 3.1.14.18.1. Abbreviation

LMDSC

#### 3.1.14.18.2. LDAP Name(s)

ldapMaxDapShareCount

#### 3.1.14.18.3. Syntax (DAP)

Integer

#### 3.1.14.18.4. Syntax (LDAP)

Integer (LDAP Style)

#### 3.1.14.18.5. Example (DAP)

```
LMDSC=10
```

#### 3.1.14.18.6. Example (LDAP)

```
ldapMaxDapShareCount=8
```

### 3.1.14.19. LDAP Allow IP List

The LDAP server grants access to clients by evaluating the LDAP Allow IP List attribute. If the IP address of the client matches a value in the LDAP Allow IP List attribute the LDAP server grants access to the client. If the client's IP address also matches a value in LDAP Deny List attribute, the LDAP server denies access to the client. (A client's IP address must match the allow list and must not match the deny list to get access.)

Specify the IP4 address values in the shortest dotted notation (no blanks or initial zeros allowed), for example 192.23.81.23 (and not 192.023.081.023 or 192. 23. 81. 23). Use the wildcard (\*) character to specify subnets, for example 192.23.\*.81 (all IP addresses from 192.23.0.81 through 192.23.255.81).

Use the value **\*.\*.\*** or the keyword **all** to specify all possible IP addresses.



Use the value **0.0.0.0** or the keyword **none** to specify no IP address.

The default value is **all**.

The values of this attribute are also stored in the LDAP MIB static table. (See the **ldap mib** operation described in the **DirX Directory Commands** chapter and the **LDAP MIB Static Table** section in the **LDAP MIB Tables** appendix in the *DirX Directory Administration Reference* for details.)

See also section **LDAP Deny IP List** of this chapter.

Note that access is only granted to clients if both allow and deny list evaluation lead to a positive result for a given IP address. That is an IP address must be allowed and must not be denied to match the acceptance criteria.

#### 3.1.14.19.1. Abbreviation

LAIP

#### 3.1.14.19.2. LDAP Name(s)

ldapAllowIPList

#### 3.1.14.19.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.19.4. Example (DAP)

```
{LAIP=123.23.81.23;130.*.*.*}  
LAIP=all           All IP addresses are allowed.  
LAIP=none          No IP address is allowed.
```

#### 3.1.14.19.5. Example (LDAP)

```
{ldapAllowIPList=123.23.81.23;130.*.*.*}  
ldapAllowIPList=all       All IP addresses are allowed.  
ldapAllowIPList=none      No IP address is allowed.
```

### 3.1.14.20. LDAP Deny IP List

The LDAP server denies access to clients by evaluating the LDAP Deny IP List attribute. If the IP address of the client matches a value in the LDAP Deny IP List attribute the LDAP server denies access to the client.

Specify the IP4 address values in the shortest dotted notation, for example 192.23.81.23 (and not 192.023.081.023). Use the wildcard (\*) character to specify subnets, for example 192.23.\*.81 (that are all IP addresses from 192.23.0.81 through 192.23.255.81).

Use the value **\*.\*.\*** or the keyword **all** to specify all possible IP addresses.

Use the value **0.0.0.0** or the keyword **none** to specify no IP address.

The default value is **none**.

The values of this attribute are also stored in the LDAP MIB static table. (See the **ldap mib** operation of the **DirX Directory Commands** chapter and the **LDAP MIB Static Table** section of the **LDAP MIB Tables** appendix in the *DirX Directory Administration Reference* for details.)

See also the section **LDAP Allow IP List** in this chapter.

Note that access is only granted to clients if both allow and deny list evaluation lead to a positive result for a given IP address; that is, an IP address must be allowed and must not be denied to match the acceptance criteria.

#### 3.1.14.20.1. Abbreviation

LDIP

#### 3.1.14.20.2. LDAP Name(s)

ldapDenyIPList

#### 3.1.14.20.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.20.4. Example (DAP)

```
{LDIP=124.23.81.23;140.*.*.23}
LDIP=.*.*.*      All IP addresses are denied.
LDIP=none        No IP address is denied.
```

#### 3.1.14.20.5. Example (LDAP)

```
{ldapDenyIPList=124.23.81.23;140.*.*.23}
ldapDenyIPList=.*.*.*      All IP addresses are denied.
ldapDenyIPList=none        No IP address is denied.
```

### 3.1.14.21. LDAP Allowed Users List

The LDAP server grants or denies access to clients by evaluating the multi-valued attributes LDAP Allowed Users List, LDAP Allowed Groups, LDAP Denied Users List and LDAP Denied Groups. If the DN of the client matches one of the values in the LDAP Allowed Users List attribute, the LDAP server grants access to the client. If the client's DN also matches a value in the LDAP Denied Users List attribute, the LDAP server denies access to the client; that is,

a user name must match the allow list **and** must not match the deny list to get access.

Specify the DN attribute values in LDAP format; for example, cn=Abele,ou=sales,o=my-company. (See the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds** for details.)

If a client binds via SASL with the EXTERNAL mechanism, the DN is taken from the user's certificate subject field independent from the mapping method defined by the attribute **LDAP SASL Authz Id Mapping**. Specify the user's subject DN in LDAP format; for example, uid=12345,givenName=Max Mustermann,o=my-company.

Use the keyword **all** to specify that all distinguished names get access.

Use the keyword **none** to specify that no distinguished name gets access.

The default value is **all**.

A DN matches if one of the values of the Allowed Users List attribute is **contained** in the given DN; that is, if the string from the attribute is part of the given user name. Specify the complete distinguished name of the root node of a subtree if you want to grant access to all users below this subtree node; for example, if you want to grant access to all employees of the sales department of your company, specify the value ou=sales,o=my-company.

The value(s) of this attribute are treated and compared as LDAP DN's by eliminating unspecific blanks and normalizing to lowercase; for example:

CN = Abele, O=My-Company is normalized to cn=abele,o=my-company

The value(s) of this attribute are also stored in the LDAP MIB static table. (See the **ldap mib** operation described in the **DirX Directory Commands** chapter and the **LDAP MIB Static Table** section in the **LDAP MIB Tables** appendix in the *DirX Directory Administration Reference* for details.)

See also the sections **LDAP Allowed Groups**, **LDAP Denied Users List** and **LDAP Denied Groups** in this chapter.

Note that access is only granted to clients if both allow and deny list and groups evaluation lead to a positive result for a given DN; that is, a DN must be allowed and must not be denied to match the acceptance criteria.

#### 3.1.14.21.1. Abbreviation

LAUSL

#### 3.1.14.21.2. LDAP Name(s)

ldapAllowedUsers

#### 3.1.14.21.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.21.4. Example (DAP)

```
{LAUSL=cn=admin,o=my-company;ou=sales,o=my-company}
```

The LDAP server grants access to the Administrator of My-Company (cn=admin,o=my-company) and to all employees of the Sales department (ou=sales,o=my-company).

```
LAUSL=all      Access is granted to all users (default).
LAUSL=none     No user can access the LDAP server. Only anonymous
               access is possible.
```

#### 3.1.14.21.5. Example (LDAP)

```
{ldapAllowedUsers=cn=admin,o=my-company;ou=sales,o=my-company}
ldapAllowedUsers=all
ldapAllowedUsers=none
```

#### 3.1.14.22. LDAP Allowed Groups

The LDAP server grants or denies access to clients by evaluating the multi-valued attributes LDAP Allowed Users List, LDAP Allowed Groups, LDAP Denied Users List and LDAP Denied Groups. The LDAP Allowed Groups attribute specifies the DN's of groups with the object class **groupOfNames**. The group members are treated in the same way as the DN's listed in the attribute LDAP Allowed Users List.

Specify the attribute values for the group names in LDAP format; for example, cn=validUserGroup,o=my-company. (See the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds** for details.)

If this attribute is not specified, no access restriction is applied to any user unless other access attributes are in effect.

The value(s) of this attribute are treated and compared as LDAP DN's by eliminating unspecific blanks and normalizing to lowercase; for example:

CN = Group1 , O=My-Company is normalized to cn=group1,o=my-company

The value(s) of this attribute are also stored in the LDAP MIB static table. (See the **ldap mib** operation described in the **DirX Directory Commands** chapter and the **LDAP MIB Static Table** section of the **LDAP MIB Tables** appendix in the *DirX Directory Administration Reference* for details.)

See also the sections **LDAP Allowed Groups**, **LDAP Denied Users List** and **LDAP Denied Groups** in this chapter.

Note that access is only granted to clients if both allow and deny list and groups evaluation

lead to a positive result for a given DN; that is, a DN must be allowed and must not be denied to match the acceptance criteria.

#### 3.1.14.22.1. Abbreviation

LAGRP

#### 3.1.14.22.2. LDAP Name(s)

ldapAllowedGroups

#### 3.1.14.22.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.22.4. Example (DAP)

```
{LAGRP=cn\=validUserGroup\,o\=my-company}
```

The LDAP server grants access to all members of the group (cn=validUserGroup,o=my-company).

#### 3.1.14.22.5. Example (LDAP)

```
{ldapAllowedGroups=cn=validUserGroup,o=my-company}
```

### 3.1.14.23. LDAP Denied Users List

The LDAP server grants or denies access to clients by evaluating the multi-valued attributes LDAP Allowed Users List, LDAP Allowed Groups, LDAP Denied Users List and LDAP Denied Groups. If the DN of the client matches one of the values in the LDAP Denied Users List attribute, the LDAP server denies access to the client.

Specify the attribute values in LDAP format; for example, cn=Abele,ou=sales,o=my-company. (See the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds** for details.)

If a client binds via SASL with the EXTERNAL mechanism, the DN is taken from the user's certificate subject field independent of the mapping method defined by the attribute **LDAP SASL Authz Id Mapping**. Specify the user's subject DN in LDAP format; for example, uid=12345,givenName=Max Mustermann,o=my-company.

Use the keyword **all** to specify that the LDAP server denies access to all distinguished names.

Use the keyword **none** to specify that the LDAP server does not deny access to any distinguished name.

The default value is **none**.

A DN matches if one of the values of the Denied Users List attribute is **contained** in the given DN; that is, if the string from the attribute is part of the given user name. Specify the complete distinguished name of the root node of a subtree if you want to deny access to all users below this subtree node; for example, if you want to deny access to all employees of the development department of your company, specify the value `ou=development,o=my-company`.

The value(s) of this attribute are treated and compared as LDAP DN's by eliminating unspecific blanks and normalizing to lowercase, for example:

CN = Abele, O=My-Company is normalized to `cn=abele,o=my-company`

The value(s) of this attribute are also stored in the LDAP MIB static table. (See the **ldap mib** operation described in the **DirX Directory Commands** chapter and the **LDAP MIB Static Table** section in the **LDAP MIB Tables** appendix in the *DirX Directory Administration Reference* for details.)

See also the sections **LDAP Allowed Users List**, **LDAP Allowed Groups** and **LDAP Denied Groups** in this chapter.

Note that access is only granted to clients if both allow and deny list and groups evaluation lead to a positive result for a given DN; that is, a DN must be allowed and must not be denied to match the acceptance criteria.

#### 3.1.14.23.1. Abbreviation

LDUSL

#### 3.1.14.23.2. LDAP Name(s)

ldapDeniedUsers

#### 3.1.14.23.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.23.4. Example (DAP)

```
{LDUSL=cn\=richter\,ou\=sales\,o\=my-company}
```

The LDAP server denies access to Mr. Richter an employee of the Sales department (`cn=Richter,ou=sales,o=my-company`).

|            |                                                                        |
|------------|------------------------------------------------------------------------|
| LDUSL=all  | No user can access the LDAP server. Only anonymous access is possible. |
| LDUSL=none | Access is granted to all users (default).                              |

#### 3.1.14.23.5. Example (LDAP)

```
{ldapDeniedUsers=cn=richter,ou=sales,o=my-company}  
ldapDeniedUsers=all  
ldapDeniedUsers=none
```

#### 3.1.14.24. LDAP Denied Groups

The LDAP server grants or denies access to clients by evaluating the multi-valued attributes LDAP Allowed Users List, LDAP Allowed Groups, LDAP Denied Users List and LDAP Denied Groups. The LDAP Denied Groups attribute specifies the DNs of groups with the object class **groupOfNames**. The group members are treated in the same way as the DNs listed in the attribute LDAP Denied Users List.

Specify the attribute values for the group names in LDAP format; for example, cn=invalidUserGroup,o=my-company. (See the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds** for details.)

If this attribute is not specified, no access restriction is applied to any user unless other access attributes are in effect.

The value(s) of this attribute are treated and compared as LDAP DNs by eliminating unspecific blanks and normalizing to lowercase; for example:

CN = Group1 , O=My-Company is normalized to cn=group1,o=my-company

The value(s) of this attribute are also stored in the LDAP MIB static table. (See the **ldap mib** operation described in the **DirX Directory Commands** chapter and the **LDAP MIB Static Table** section in the **LDAP MIB Tables** appendix in the *DirX Directory Administration Reference* for details.)

See also the sections **LDAP Allowed Groups**, **LDAP Denied Users List** and **LDAP Denied Groups** in this chapter.

Note that access is only granted to clients if both allow and deny list and groups evaluation lead to a positive result for a given DN; that is, a DN must be allowed and must not be denied to match the acceptance criteria.

##### 3.1.14.24.1. Abbreviation

LDGRP

##### 3.1.14.24.2. LDAP Name(s)

ldapDeniedGroups

##### 3.1.14.24.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.24.4. Example (DAP)

```
{LDGRP=cn\=GroupInvalidUsers\,o\=my-company}
```

The LDAP server denies access to all members of the group (cn=GroupInvalidUsers,o=my-company).

#### 3.1.14.24.5. Example (LDAP)

```
{ldapDeniedGroups=cn=GroupInvalidUsers,o=my-company}
```

### 3.1.14.25. LDAP Deny Anonymous Access

The LDAP Deny Anonymous Access attribute specifies whether or not anonymous LDAP client binds are allowed. Especially in security environments, it might be useful to allow only authorized user access to the database.

This feature should not be used if you work with standard LDAP clients. It should only be used when you know that your clients are able to handle it correctly.

Specify one of the following values:

**TRUE** - The LDAP server rejects anonymous LDAP client binds. LDAP clients can only perform simple authenticated or SASL-authenticated binds. Note that this is not LDAPv3 conformant. (See the **dirxcp obj bind** operation of the of the **DirX Directory Commands** chapter in the *DirX Directory Administration Reference* for details.)

**FALSE** - The LDAP server accepts anonymous LDAP client binds.

The default value is **FALSE** (anonymous users are allowed).

Note that this attribute controls only the LDAP server and not the DSA and is independent of any user policy settings in the database.

Keep in mind that if the LDAP Deny Anonymous Access attribute is set to **TRUE** and the LDAP Only Allow Anonymous Access attribute is set to **1**, no LDAP client is able to bind to the LDAP server.

#### 3.1.14.25.1. Abbreviation

LDAA

#### 3.1.14.25.2. LDAP Name(s)

ldapDenyAnonymousAccess

#### 3.1.14.25.3. Syntax (DAP)

Boolean



#### 3.1.14.25.4. Syntax (LDAP)

Boolean (LDAP Style)

#### 3.1.14.25.5. Example (DAP)

```
LDAA=TRUE
```

#### 3.1.14.25.6. Example (LDAP)

```
ldapDenyAnonymousAccess=TRUE
```

### 3.1.14.26. LDAP Only Allow Anonymous Access

The LDAP Only Allow Anonymous Access attribute specifies whether or not only anonymous LDAP client binds are allowed.

Specify one of the following values:

**1** - The LDAP server accepts only anonymous LDAP client binds. It rejects all authenticated LDAP client binds with the error code LDAP\_INAPPROPRIATE\_AUTH (48). It does not establish an anonymous bind automatically as RFC 2251 (and newer) requires. (See the **dirxcp obj bind** operation of the of the **DirX Directory Commands** chapter in the *DirX Directory Administration Reference* for details.)

**0** - The LDAP server accepts authenticated LDAP client binds.

The default value is **0**.

Note that this attribute controls only the LDAP server and not the DSA and is independent of any user policy settings in the database.

Keep in mind that if the LDAP Only Allow Anonymous Access attribute is set to **1** and the LDAP Deny Anonymous Access attribute is set to **TRUE**, no LDAP client is able to bind to the LDAP server.

#### 3.1.14.26.1. Abbreviation

LOAAL

#### 3.1.14.26.2. LDAP Name(s)

ldapOnlyAnonymousAllowed

#### 3.1.14.26.3. Syntax (DAP)

Integer

#### 3.1.14.26.4. Syntax (LDAP)

Integer (LDAP Style)

#### 3.1.14.26.5. Example (DAP)

```
LOAAL=1
```

#### 3.1.14.26.6. Example (LDAP)

```
ldapDenyAnonymousAccess=1
```

#### 3.1.14.27. LDAP User Policies

The LDAP User Policies attribute specifies policies that apply to a single user or a set of users. This attribute is an optional, multi-valued attribute. Specify the values in the following format:

```
Class=dn:CONLIMIT=Integer:SIZELIMIT=Integer:TIMELIMIT=Integer:  
FORB_SB=dn:FORB_SB_SUB=dn:FORB_TGT=dn:FORB_TGT_SUB=dn:  
REQUIRED_SB=dn:REQUIRED_SB_SUB=dn:  
REQUIRED_TGT=dn:REQUIRED_TGT_SUB=dn:  
MIN_SB_LEVEL=Integer:TLS_REQUIRED=Flag:  
PRIO=Integer:INHERIT=Flag:DISCLOSE_VIOLATION=Flag:  
QUOTA_MAX_OPS=IntervalUnitLimit:  
QUOTQ_MAX_RES_BYTES= IntervalUnitLimit:  
MUST_CONTACT_DSA=DSAName:PREF_CONTACT_DSA=DSAName
```

where:

**Class=dn**

specifies the user(s) for which the policies apply. *Class* is mandatory and must be the first assignment of the policies attribute. Specify one of the following keywords:

**USER** - specifies that the policies apply to a single user. *dn* specifies the full qualified user name in LDAP format; for example, **USER=cn=admin,o=my-company**. For details about distinguished names in LDAP format see the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds**.

For binds with SASL EXTERNAL authentication, specify *dn* in LDAP format; for example, **USER=uid=12345,givenName=Max Mustermann,o=my-company**. For details about distinguished names in DAP format, see the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds**. If a client binds via SASL with the EXTERNAL mechanism, the DN is taken from the user's certificate subject field independent of the mapping method defined by the attribute **LDAP SASL Authz Id Mapping**.

Instead of specifying a full qualified distinguished name, specify one of the following

keywords:

**anonymous** - specifies that the policies apply to all users bound anonymously to the LDAP server; that is, to all users that performed an anonymous bind, to all LDAP clients that did not perform a bind, or to users that performed a bind with an **invalid credentials** error code.

**all** - specifies that the policies apply to all users. If there are multiple policies, the user policies for **all** have the lowest priority. Anonymous users belong to all if there are no policies for **anonymous**.

The class **USER** cannot be overruled by policies of any other class. It is only allowed for the LDAP User Policies attribute.

**SUBUSER** - specifies that the policies apply to all users below a node in the DIT. *dn* specifies the full qualified name in LDAP format of the node in the DIT in LDAP format; for example, **SUBUSER=ou=sales,o=my-company**.

Policies of the class **SUBUSER** are not applied to users bound with SASL EXTERNAL authentication.

The class **SUBUSER** is only allowed for the LDAP User Policies attribute.

**WCUSER** - specifies that *dn* represents a regular expression syntax string according to Linux/Perl standards. A colon (:) is not allowed in *dn*; for example, **WCUSER=^cn=Digger.\***.

If multiple **WCUSER** class policies match the same user, the policies with the highest priority apply. (See **PRIO** for details.) If multiple policies match the same user with the same priority, the LDAP server randomly selects the policies.

The class **WCUSER** is only allowed for the LDAP User Policies attribute.

**GROUP** - specifies that the policies apply to a group of users. *dn* specifies the full qualified name in LDAP format of an entry with object class **groupOfNames (GON)**; for example, **GROUP=cn=my-group,o=my-company**. If a user is member of several groups, the policies with the highest priority apply. (See **PRIO** for details.) If the priority is also the same, the LDAP server randomly selects the policies.

The class **GROUP** is only allowed for the LDAP Group Policies attribute.

#### **CONLIMIT=Integer**

specifies the maximum number of concurrent LDAP connections a user can open. Specify a positive integer; for example, **CONLIMIT=5**. Only one **CONLIMIT** value can be specified.

#### **SIZELIMIT=Integer**

specifies the maximum number of objects returned for a list or search operation. Specify a positive integer; for example, **SIZELIMIT=100**. Only one **SIZELIMIT** value can be specified.

This value overrides the value of the LDAP Search Service Controls attribute. However, it

cannot increase the size limit configured for the DSA.

#### ***TIMELIMIT=Integer***

specifies the maximum elapsed time in seconds for completion of an operation. Specify a positive integer; for example, **SIZELIMIT=10**. Only one **TIMELIMIT** value can be specified.

This value overrides the value of the LDAP Search Service Controls attribute. However, it cannot increase the size limit configured for the DSA.

#### ***FORB\_SB=dn***

specifies a base object that is not allowed in search requests. *dn* is the full qualified distinguished name of a leaf or a node in the DIT in LDAP format; for example, **FORB\_SB=ou=finance,o=my-company**. For details about distinguished names in LDAP format, see the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds**. Multiple **FORB\_SB** assertions can be specified.

#### ***FORB\_SB\_SUB=dn***

specifies the distinguished name of the base of a subtree that is not allowed in search requests. *dn* is the full qualified distinguished name a node in the DIT in LDAP format; for example, **FORB\_SB\_SUB=o=my-company**. For details about distinguished names in LDAP format, see the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds**. Multiple **FORB\_SB\_SUB** assertions can be specified.

#### ***FORB\_TGT=dn***

specifies an entry that is not allowed in update operations. *dn* is the full qualified distinguished name of a leaf or a node in the DIT in LDAP format; for example, **FORB\_TGT=cn=smith,ou=finance,o=my-company**. For details about distinguished names in LDAP format, see the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds**. Multiple **FORB\_TGT** assertions can be specified.

#### ***FORB\_TGT\_SUB=dn***

specifies the distinguished name of the base of a subtree that is not allowed in update operations. *dn* is the full qualified distinguished name a node in the DIT in LDAP format; for example, **FORB\_TGT\_SUB=ou=finance,o=my-company**. For details about distinguished names in LDAP format, see the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds**. Multiple **FORB\_TGT\_SUB** assertions can be specified.

#### ***REQUIRED\_SB=dn***

specifies a base object that is allowed in search requests. Subordinate *dns* are not included. *dn* is the full qualified distinguished name of a leaf or a node in the DIT in LDAP format; for example, **REQUIRED\_SB=ou=finance,o=my-company**. For details about distinguished names in LDAP format, see the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds**. Multiple **REQUIRED\_SB** assertions can be specified.

#### ***REQUIRED\_SB\_SUB=dn***

specifies the distinguished name of the base of a subtree that is allowed in search

requests. *dn* is the full qualified distinguished name a node in the DIT in LDAP format; for example, **REQUIRED\_SB\_SUB=o=my-company**. For details about distinguished names in LDAP format, see the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds**. Multiple **REQUIRED\_SB\_SUB** assertions can be specified.

#### **REQUIRED\_TGT=dn**

specifies an entry that is allowed in update operations. *dn* is the full qualified distinguished name of a leaf or a node in the DIT in LDAP format; for example, **REQUIRED\_TGT=cn=smith,ou=finance,o=my-company**. For details about distinguished names in LDAP format, see the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds**. Multiple **REQUIRED\_TGT** assertions can be specified.

#### **REQUIRED\_TGT\_SUB=dn**

specifies the distinguished name of the base of a subtree that is allowed in update operations. *dn* is the full qualified distinguished name a node in the DIT in LDAP format; for example, **REQUIRED\_TGT\_SUB=ou=finance,o=my-company**. For details about distinguished names in LDAP format, see the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds**. Multiple **REQUIRED\_TGT\_SUB** assertions can be specified.

#### **MIN\_SB\_LEVEL=Integer**

specifies the minimum level starting at the root object at which a base object must exist for a search request. Specify a positive integer; for example, **MIN\_SB\_LEVEL=3**. Only one **MIN\_SB\_LEVEL** value can be specified.

#### **TLS\_REQUIRED=Flag**

indicates whether or not a bind over SSL/TLS is required. Specify **1** (true - indicates that a bind over SSL/TLS is required) or **0** (false - indicates that no bind over SSL/TLS is required). The default value is **TLS\_REQUIRED=0** (any bind is allowed). Only one **TLS\_REQUIRED** flag can be specified.

#### **PRIO=Integer**

specifies the priority of policies in the same class. Specify a positive integer. The highest priority is **0**. The default priority is **PRIO=1**. The lower the priority value, the higher is the priority. Only one **PRIO** value can be specified.

If there are multiple policies for the same user, the policies with the highest priority (the lowest priority value) apply.

If there are multiple policies for the same user with the same priority, the first policies found apply. It is not predictable which policies are the first ones found.

#### **INHERIT=Flag**

indicates whether or not policies are inherited. Specify **INHERIT=1** (true - indicates that policies are inherited) or **INHERIT=0** (false - indicates that policies are not inherited). The default value is **0** (policies are not inherited). This flag can only be specified for the **USER=all** class policies. Only one **INHERITED** flag can be specified.

### **DISCLOSE\_VIOLATION=Flag**

indicates whether or not detailed information about the specified limit is provided in the error message if the limit is exceeded. Specify **DISCLOSE\_VIOLATION=1** (true - indicates that detailed information is provided) or **DISCLOSE\_VIOLATION=0** (false - indicates that no detailed information is provided). The default value is **1** (detailed information is provided). This flag affects violations of **CONLIMIT**, **QUOTA\_MAXOPS** and **QUOTA\_MAX\_RES\_BYTES**. Only one **DISCLOSE\_VIOLATION** flag can be specified.

### **QUOTA\_MAX\_OPS=IntervalUnitLimit**

specifies the maximum number of operations that a user is allowed to perform in the specified time interval. Specify **QUOTA\_MAX\_OPS** in the format *IntervalUnitLimit*

where

#### **Interval**

is a positive integer.

#### **Unit**

is one of the following characters:

**s** for seconds

**m** for minutes

**h** for hours

**d** for days

#### **Limit**

is a positive integer specifying the maximum number of operations. The operations abandon and unbind do not increase the number of operations.

Only one **QUOTA\_MAX\_OPS** assertion can be specified.

Example: **QUOTA\_MAX\_OPS=24h1000**

### **QUOTA\_MAX\_RES\_BYTES=IntervalUnitLimit**

specifies the maximum number of bytes that a user is allowed to receive from search operations in the specified time interval. Specify **QUOTA\_MAX\_RES\_BYTES** in the format *IntervalUnitLimit*

where

#### **Interval**

is a positive integer.

#### **Unit**

is one of the following characters:

**s** for seconds

**m** for minutes

**h** for hours

**d** for days

### Limit

is a positive integer specifying the maximum number of bytes.

Only one **QUOTA\_MAX\_RES\_BYTES** assertion can be specified.

Example: **QUOTA\_MAX\_RES\_BYTES=3600s1000000**

### **MUST\_CONTACT\_DSA=DSAName**

specifies the DSA that must be contacted in a multiple contact DSA configuration. (See **Using Multiple Contact DSAs** in the *DirX Directory Administration Guide* for details.) *DSAName* is the name of the DSA as specified in the LDAP server configuration file; for example, **MUST\_CONTACT\_DSA=/CN=DSA1**. (See **LDAP Server Configuration File** in the *DirX Directory Administration Reference* for details.)

If the **MUST\_CONTACT\_DSA** is not available and no **PREF\_CONTACT\_DSA** is specified, the operation fails.

If the **MUST\_CONTACT\_DSA** is not available and a **PREF\_CONTACT\_DSA** is specified, the LDAP server continues with contacting the **PREF\_CONTACT\_DSA**.

If the **PREF\_CONTACT\_DSA** is also unavailable, the LDAP server starts a round-robin selection to contact a DSA.

Only one **MUST\_CONTACT\_DSA** can be specified.

### **PREF\_CONTACT\_DSA=DSAName**

specifies the DSA that is preferred to be contacted in a multiple contact DSA configuration. (See **Using Multiple Contact DSAs** in the *DirX Directory Administration Guide* for details.) *DSAName* is the name of the DSA as specified in the LDAP server configuration file; for example, **PREF\_CONTACT\_DSA=/CN=DSA3**. (See **LDAP Server Configuration File** in the *DirX Directory Administration Reference* for details.)

If the **PREF\_CONTACT\_DSA** is unavailable, the LDAP server starts a round-robin selection to contact a DSA.

Only one **PREF\_CONTACT\_DSA** can be specified.

The LDAP server reads the policies attributes at start-up. The policies attributes can be updated dynamically via LDAP extended operations (**ldap\_cfg\_upd**). (See **Changing the LDAP Configuration** in chapter **Extending the DirX Directory Server** in the *DirX Directory Administration Guide* for details.) When updating the policies attributes dynamically, the LDAP server keeps the old policies to avoid severe problems between old and new policies for a specific user. This operation can consume a lot of memory and performance. Consequently, we strongly recommended updating the policies only on demand and only if it is really necessary to avoid serious memory problems. If the LDAP server cannot update the policies attributes due to memory problems, the old policies remain in effect.

Use the **CLASS** property to specify the user or set of users to which the policy apply. **CLASS**

is the only mandatory property. It must be specified as the first property of a policies value. All other properties are optional. They specify the rules that apply for the user(s) specified in **CLASS**.

Policies for a specific user can appear in several classes. The LDAP server follows the following strategy when evaluating user and group policies for a specific user:

1. Search for a **USER** rule with the user's *dn*.
2. Search for a **GROUP** rule with the user's *dn* as member of the group.
3. Search for a **WCUSER** rule whose regular expression matches the user's *dn*.
4. Search for a **SUBUSER** rule that matches the parent of the user's *dn*.
5. Search for a **USER** rule with **all** as *dn*.
6. If the LDAP server does not find a rule in steps 1 through 5, no rules apply to the user.

The LDAP server stops evaluating the user policies for a specific user when it finds a rule while performing steps 1 through 5. You can control the priority of rules by specifying the **PRIO** property for a rule.

If a user or group policy is exceeded, the LDAP server returns an **unwilling to perform** error together with detailed information about the exceeded policy. You can hide the detailed information about the exceeded policy by specifying the **DISCLOSE\_VIOLATION** flag.

The LDAP audit log file provides an overview on the number of rules specified.

The LDAP server writes a log file in readable format every time it reads the policies attributes either at start-up or when updating the policies dynamically. The name of the log file is **LUP\_\*pid\_counter.txt\***, where *pid* is the PID of the LDAP server process and *counter* is an integer starting with 1. The *counter* is incremented by 1 each time the LDAP server reads the policies attributes and writes the log file. The log file is located at *install\_path\*/ldap/log\**. The format of the file is identical to the output of the LDAP extended operation **ldap\_show\_policy\_rules**.

#### 3.1.14.27.1. Abbreviation

LUSP

#### 3.1.14.27.2. LDAP Name(s)

ldapUserPolicies

#### 3.1.14.27.3. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.27.4. Example (DAP)

```
LUSP=SUBUSER\=ou\=Development\,o\=My-  
Company:CONNLIMIT\=10:PRIO\=1:SIZELIMIT\=1555:QUOTA_MAX_OPS\=2s10:TLS
```



```
_REQUIRED\=1:DISCLOSE_VIOLATION\=1:FORB_SB\=ou\=Development\,o\=My-
Company;USER\=anonymous:CONNLIMIT\=5:SIZELIMIT\=2000:QUOTA_MAX_OPS\=5
m100:TLS_REQUIRED\=0:DISCLOSE_VIOLATION\=0:FORB_TGT\=cn\=admin\o\=My-
Company;USER\=cn\=emil\,ou\=CertTest\,o\=My-
Company:PRIO\=1:SIZELIMIT\=10:TIMELIMIT\=1:TLS_REQUIRED\=1:DISCLOSE_V
IOLATION\=0
```

#### 3.1.14.27.5. Example (LDAP)

```
ldapUserPolicies=SUBUSER=ou=Development,o=My-
Company:CONNLIMIT=10:SIZELIMIT=100000:QUOTA_MAX_RES_BYTES=30s100000:T
LS_REQUIRED=0:DISCLOSE_VIOLATION=0:FORB_SB=ou=hb,o=My-
Company;USER=anonymous:CONNLIMIT=5:SIZELIMIT=2000:QUOTA_MAX_OPS=5m100
:TLS_REQUIRED=0:DISCLOSE_VIOLATION=0:FORB_TGT=cn=admin,o=My-
Company;USER=cn=emil,ou=CertTest,o=My-
Company:PRIO=1:SIZELIMIT=10:TIMELIMIT=1:TLS_REQUIRED=1:DISCLOSE_VIOLA
TION=0;WCUSER=*admin*:CONNLIMIT=100:PRIO=0:SIZELIMIT=1000000:TLS_REQU
IRED=0:DISCLOSE_VIOLATION=0
```

#### 3.1.14.28. LDAP Group Policies

The LDAP Group Policies attribute specifies policies that apply to a group of users. This attribute is an optional, multi-valued attribute. Specify the value in the same format as described for the **LDAP User Policies** (see that section for details).

The LDAP Group Policies attribute can only be specified for groups with a maximum of one million members. The LDAP server discards groups with more than one million members.

See **LDAP User Policies** for detailed information about processing the LDAP User Policies and LDAP Group policies attributes.

##### 3.1.14.28.1. Abbreviation

LGRP

##### 3.1.14.28.2. LDAP Name(s)

ldapGroupPolicies

##### 3.1.14.28.3. Syntax (DAP)

Directory-String {32768}

##### 3.1.14.28.4. Example (DAP)

```
LGRP={GROUP\=cn\=a11E\,o\=My-  
Company:CONNLIMIT\=10:PRI0\=1:SIZELIMIT\=200:  
MIN_SB_LEVEL\=2:TLS_REQUIRED\=0:DISCLOSE_VIOLATION\=0}
```

#### 3.1.14.28.5. Example (LDAP)

```
ldapGroupPolicies=GROUP=cn=a11E,o=My-Company:CONNLIMIT=10:PRI0=1:  
SIZELIMIT=200:MIN_S_LEVEL=2:TLS_REQUIRED=0:DISCLOSE_VIOLATION=0
```

#### 3.1.14.29. LDAP Listen IP List

The LDAP Listen IP List attribute specifies exactly one IP address over which clients can bind to the LDAP server.

By default, the LDAP server establishes its ports (plain and SSL) for any available IP address of the host; that is, clients can bind to the LDAP server over all available IP addresses. An administrator, however, may want to restrict remote binds to only one IP address, for example, due to security issues.

Specify the IP4 address value in the shortest dotted notation, for example 192.23.81.23 (and not 192.023.081.023).

Use the keyword **all** to specify all possible IP addresses. This is the default.

If an invalid IP address is specified, the LDAP server does not start.

If the local loop back address **127.0.0.1** is specified, only local clients can access the LDAP server. This may be useful for administrative tasks.

The value of the LDAP Listen IP List attribute has no effect on RPC or the OSI/IDM connections of the LDAP server to the DSA. These ports are bound to all available IP addresses.

##### 3.1.14.29.1. Abbreviation

LLIPL

##### 3.1.14.29.2. LDAP Name(s)

ldapListenIPList

##### 3.1.14.29.3. Syntax (DAP)

Directory String

##### 3.1.14.29.4. Syntax (LDAP)

Directory String (LDAP Style)

#### 3.1.14.29.5. Example (DAP)

```
LLIPL=192.10.11.12
```

#### 3.1.14.29.6. Example (LDAP)

```
ldapListenIPList=192.10.11.12
```

### 3.1.14.30. LDAP ASN1 Header

The LDAP ASN1 Header attribute specifies whether X.509 attributes are returned in binary ASN1 encoding (value: **FALSE**) or in “[ASN]<hexdump of asn1 encoding>” format (value: **TRUE**). This format applies only to LDAP v2 accesses. This attribute is an optional, single-valued attribute. The default value is **FALSE** (X.509 attributes are returned as binary in LDAP v2 results).

#### 3.1.14.30.1. Abbreviation

LAHE

#### 3.1.14.30.2. LDAP Name(s)

ldapASN1Header

#### 3.1.14.30.3. Syntax (DAP)

Boolean

#### 3.1.14.30.4. Syntax (LDAP)

Boolean (LDAP Style)

#### 3.1.14.30.5. Example (DAP)

```
LAHE=TRUE
```

#### 3.1.14.30.6. Example (LDAP)

```
ldapASN1Header=TRUE
```

### 3.1.14.31. LDAP Charset Conv Request

The LDAP Charset Conv Request attribute specifies the representation of string attributes that the LDAP server expects for string values in LDAP requests. This format applies only to LDAPv2 directory access operations. This attribute is an optional, single-valued attribute. In

LDAPv3, strings are always represented in UTF-8 format. Specify one of the following values:

**LATIN1** - all strings are represented as specified in ISO8859-1.

**UTF8** - all strings are represented in UTF-8 format.

**T61** - all strings are represented in T61 format.

The default value is **UTF8**.

#### 3.1.14.31.1. Abbreviation

LCCQ

#### 3.1.14.31.2. LDAP Name(s)

ldapCharsetConvRequest

#### 3.1.14.31.3. Syntax (DAP)

Directory String {32768}

#### 3.1.14.31.4. Example (DAP)

```
LCCQ=T61
```

#### 3.1.14.31.5. Example (LDAP)

```
ldapCharsetConvRequest=T61
```

### 3.1.14.32. LDAP Charset Conv Result

The LDAP Charset Conv Result attribute specifies the conversion of string attributes that the LDAP server performs for LDAP search results. This format applies only to LDAPv2 directory access operations. This attribute is an optional, single-valued attribute. In LDAPv3, strings are always represented in UTF-8 format. Specify one of the following values:

**LATIN1** - all strings are converted as specified in ISO8859-1.

**UTF8** - all strings are converted in UTF-8 format.

**T61** - all strings are converted in T61 format.

The default value is **UTF8**.

The following table shows the impact of this attribute value on the representation of string values in the LDAP v2 protocol.

| Attribute value syntax in DIR.X DSA | Value of LDAP Charset Conv Result | String representation in LDAPv2 |
|-------------------------------------|-----------------------------------|---------------------------------|
| Printable String                    | T61 String                        | Printable String                |
| T61 String                          | T61 String                        | T61 String                      |
| BMP String                          | T61 String                        | T61 String                      |
| Printable String                    | LATIN1                            | Printable String                |
| T61 String                          | LATIN1                            | ISO8859-1                       |
| BMP String                          | LATIN1                            | ISO8859-1                       |
| Printable String                    | UTF8                              | Printable String                |
| T61 String                          | UTF8                              | UTF8                            |
| BMP String                          | UTF8                              | UTF8                            |

#### 3.1.14.32.1. Abbreviation

LCCS

#### 3.1.14.32.2. LDAP Name(s)

ldapCharsetConvResult

#### 3.1.14.32.3. Syntax (DAP)

Directory String {32768}

#### 3.1.14.32.4. Example (DAP)

```
LCCS=LATIN1
```

#### 3.1.14.32.5. Example (LDAP)

```
ldapCharsetConvResult=LATIN1
```

### 3.1.14.33. LDAP Read Only Server

The LDAP Read Only Server attribute specifies whether the LDAP server allows update operations (add, delete, modify and moddn) (value: **FALSE**) or not (value: **TRUE**). When this attribute is set to the value **TRUE**, update operations are rejected with the error **Administrative Limit Exceeded**. The default value is **FALSE** (update operations permitted).

#### 3.1.14.33.1. Abbreviation

LROS

#### 3.1.14.33.2. LDAP Name(s)

ldapReadOnlyServer

#### 3.1.14.33.3. Syntax (DAP)

Boolean

#### 3.1.14.33.4. Syntax (LDAP)

Boolean (LDAP Style)

#### 3.1.14.33.5. Example (DAP)

```
LROS=TRUE
```

#### 3.1.14.33.6. Example (LDAP)

```
ldapReadOnlyServer=TRUE
```

### 3.1.14.34. LDAP Thread Pool Size

The LDAP Thread Pool Size attribute specifies the maximum number of operation threads available to operate LDAP client requests in parallel. This pool size is static and cannot be modified without restarting the server.

The value of this attribute does not limit the possible simultaneous connections or the possible simultaneous operations, but it can affect the LDAP server performance and should therefore be chosen with care. The value to be used depends on the available system resources and the expected search profiles (big results or small results, LDAP cache on/off). Especially if the LDAP cache is enabled and provides a good hit rate, it might be possible to decrease the value. If the LDAP cache is disabled or most operations stress the backend with long-duration operations, the value can be increased to improve performance. If all pool threads are busy, further operations will be queued for processing.

If the value is set too high for your system, permanent re-scheduling of threads can occur, thereby lowering LDAP performance.

The default value of this attribute is **32**.

#### 3.1.14.34.1. Abbreviation

LTPS

#### 3.1.14.34.2. LDAP Name(s)

ldapThreadPoolSize

#### 3.1.14.34.3. Syntax (DAP)

Integer

#### 3.1.14.34.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.14.34.5. Example (DAP)

```
LTPS=24
```

#### 3.1.14.34.6. Example (LDAP)

```
ldapThreadPoolSize=24
```

### 3.1.14.35. LDAP Socket Keep Alive

The LDAP Socket Keep Alive attribute specifies whether the TCP keep-alive feature is enabled for all LDAP server connections (value **1**) or not (value **0**). The default value of this attribute is that the feature is enabled (**1**).

If the TCP keep-alive feature is enabled, the LDAP server requests TCP to verify whether an idle connection is still intact by sending keep-alive packets. In the event that a connection is no longer intact (for example, the LAN cable is no longer connected) or a remote client has not properly closed the connection, TCP aborts this connection.

The system administrator must set the timings and intervals for this mechanism to reasonable values. (See the section **Enabling KEEPALIVE Time/Interval** in the **Environment Variables** chapter in the *DirX Directory Administration Reference* for details.)

#### 3.1.14.35.1. Abbreviation

LSOKA

#### 3.1.14.35.2. LDAP Name(s)

ldapSockKeepAlive

#### 3.1.14.35.3. Syntax (DAP)

Integer

#### 3.1.14.35.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.14.35.5. Example (DAP)

```
LSOKA=0
```

#### 3.1.14.35.6. Example (LDAP)

```
ldapSockKeepAlive=0
```

### 3.1.14.36. LDAP Asynchronous Sockets

The LDAP Asynchronous Sockets attribute specifies whether the LDAP server performs non-blocking socket interface operations (value **1**) or not (value **0**). The default value is **1** (perform non-blocking operations).

When an LDAP client establishes a new connection, the LDAP server performs a socket interface accept operation that creates a new socket representing the TCP connection to the client. When the client and the server exchange data, the LDAP server performs send and receive operations. Usually the sockets are created in blocking mode; that is, all operations exchanging data block the connection until all data is processed (all data is copied to the TCP transport buffers). If, for example, a client stops collecting or sending data before the end of data has been reached, the socket is blocked forever. To avoid such a deadlock, the LDAP server should use non-blocking sockets.

Using non-blocking sockets may be a bit slower than using blocking sockets because a single blocking send / receive operation may result in multiple send / receive operations.

#### 3.1.14.36.1. Abbreviation

LASYS

#### 3.1.14.36.2. LDAP Name(s)

ldapAsyncSockets

#### 3.1.14.36.3. Syntax (DAP)

Integer

#### 3.1.14.36.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.14.36.5. Example (DAP)

```
LASYS=0
```



#### 3.1.14.36.6. Example (LDAP)

```
ldapAsyncSockets=0
```

#### 3.1.14.37. LDAP Max Send Timeout

The Max Send Timeout attribute specifies the maximum timeout value in seconds after which the LDAP server closes a non-responding connection and stops sending data. This attribute affects only non-blocking sockets. (See the **LDAP Asynchronous Sockets** attribute above for details.) The default value is **30** seconds.

In the event that there is no idle socket to send data, the non-blocking socket send operation returns an error indicating that the data cannot be sent. The LDAP server sets the time limit specified in the LDAP Max Send Timeout attribute for completing the intended send operation. If the time has elapsed and the data has not been sent, the LDAP server closes the connection without any further I/O. It writes an exception log indicating the timeout.

Usually the LDAP server sends PDUs in a single operation. However, for search results, it may perform several send operations. The LDAP server evaluates the timeout value for each operation. In the event of a timeout, the LDAP server aborts the current operation and all possible subsequent traffic for the given connection. Then the LDAP server closes the connection.

##### 3.1.14.37.1. Abbreviation

LMSTI

##### 3.1.14.37.2. LDAP Name(s)

ldapMaxSendTimeout

##### 3.1.14.37.3. Syntax (DAP)

Integer

##### 3.1.14.37.4. Syntax (LDAP)

Integer String (LDAP Style)

##### 3.1.14.37.5. Example (DAP)

```
LMSTI=40
```

##### 3.1.14.37.6. Example (LDAP)

```
ldapMaxSendTimeout=40
```

### 3.1.14.38. LDAP Max Recv Timeout

The Max Recv Timeout attribute specifies the maximum timeout value in seconds after which the LDAP server closes a non-responding connection and stops receiving data. This attribute affects only non-blocking sockets. (See the **LDAP Asynchronous Sockets** attribute above for details.) The default value is **30** seconds.

In the event that there is no idle socket to receive data, the non-blocking socket receive operation returns an error indicating that the data cannot be received. The LDAP server sets the time limit specified in the LDAP Max Recv Timeout attribute for completing the intended receive operation. If the time has elapsed and the data has not been received, the LDAP server closes the connection without any further I/O. It writes an exception log indicating the timeout.

Usually the LDAP server receives PDUs in a single operation. However, for search results, it may perform several receive operations. The LDAP server evaluates the timeout value for each operation. In the event of a timeout, the LDAP server aborts the current operation and all possible subsequent traffic for the given connection. Then the LDAP server closes the connection.

#### 3.1.14.38.1. Abbreviation

LMRTI

#### 3.1.14.38.2. LDAP Name(s)

ldapMaxRecvTimeout

#### 3.1.14.38.3. Syntax (DAP)

Integer

#### 3.1.14.38.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.14.38.5. Example (DAP)

```
LMRTI=40
```

#### 3.1.14.38.6. Example (LDAP)

```
ldapMaxRecvTimeout=40
```

### 3.1.14.39. LDAP Max Incomplete Operations Per Connection

The LDAP Max Incomplete Operations Per Connection attribute specifies the maximum number of incomplete operations per connection. By default, a client can send several requests without collecting the results. Such operations remain incomplete and increase

the system load on the server side. Thus a single client may consume the entire system resources. To avoid such situations, it is possible to limit the number of incomplete operations per connection by specifying a value greater than 1 for this attribute.

If a client exceeds the maximum number of incomplete operations, the LDAP server rejects all new incoming requests from that client until the client finishes incomplete operations and the number of incomplete operations is lower than the specified limit.

A value greater than 0 limits the number of incomplete operations to this value.

A value of **1** enforces a synchronous client behavior; that is, the client must collect all results of an operation before it can send the next request.

A value of **0** disables this feature and specifies an unlimited number of incomplete operations; that is, the only limits are the system resources. This is the default value.

It is recommended to enable this feature only if there are problems with the default behavior.

#### **3.1.14.39.1. Abbreviation**

LMIOC

#### **3.1.14.39.2. LDAP Name(s)**

ldapMaxIncompleteOpsPerConn

#### **3.1.14.39.3. Syntax (DAP)**

Integer

#### **3.1.14.39.4. Syntax (LDAP)**

Integer String (LDAP Style)

#### **3.1.14.39.5. Example (DAP)**

```
LMIOC=8
```

#### **3.1.14.39.6. Example (LDAP)**

```
ldapMaxIncompleteOpsPerConn=8
```

### **3.1.14.40. LDAP OpStack Size Limit**

The LDAP OpStack Size Limit attribute specifies the maximum number of operations that are accepted by the LDAP server. Whenever the listener thread of the LDAP server detects an activity on an existing connection, it creates a new operation and puts it onto an internal operation stack (OpStack) for subsequent processing by one of the pool threads. Usually

there should be always a thread available to process the operation immediately. However, during system overload, all pool threads are busy, for example, due to long-lasting operations. In this case, the internal operation stack grows until the system resources are exhausted. To avoid exhausting system resources and long-lasting operations due to waiting time in the internal operation stack, it is recommended to limit the internal operation stack size.

The default value is **320**; this is the tenfold default thread pool size. (See the **LDAP Thread Pool Size** attribute above for details.) A value of **0** is an unlimited number.

If the limit is exceeded, specific overflow threads process all incoming operations immediately by returning a busy error code (51). (See the **LDAP Overflow Threads** attribute description for details.)

#### 3.1.14.40.1. Abbreviation

LOSLI

#### 3.1.14.40.2. LDAP Name(s)

ldapOpStackLimit

#### 3.1.14.40.3. Syntax (DAP)

Integer

#### 3.1.14.40.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.14.40.5. Example (DAP)

```
LOSLI=200
```

#### 3.1.14.40.6. Example (LDAP)

```
ldapOpStackLimit=200
```

### 3.1.14.41. LDAP Overflow Threads

The LDAP Overflow Threads attribute specifies the number of threads handling requests if the internal operation stack limit is exceeded. (See the **LDAP OpStack Size Limit** attribute above for details.) An overflow thread just returns the busy error code (51). This attribute has an effect only if the LDAP OpStack Size Limit attribute is greater than **0**. In this case, there must be at least one overflow thread.

The default value is **1**. The maximum value is the maximum number of pool threads. (See the **LDAP Thread Pool Size** attribute description for details.) The number of overflow

threads decreases the maximum number of operation threads (LDAP Thread Pool Size).

#### 3.1.14.41.1. Abbreviation

LOFTH

#### 3.1.14.41.2. LDAP Name(s)

ldapOverflowThreads

#### 3.1.14.41.3. Syntax (DAP)

Integer

#### 3.1.14.41.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.14.41.5. Example (DAP)

```
LOFTH=2
```

#### 3.1.14.41.6. Example (LDAP)

```
ldapOverflowThreads=2
```

### 3.1.14.42. LDAP Cache

The LDAP Cache attribute specifies whether the LDAP cache for LDAP search results is enabled. This mechanism can provide a dramatic performance increase if repeated LDAP search requests appear frequently, for the cost of extra memory.

If the value is **TRUE**, the LDAP cache is enabled; that is, an LDAP search result is retrieved in the LDAP cache first. An LDAP result is retrieved successfully in the LDAP cache when the very same user sends the identical search request several times.

LDAP search results are not stored in the LDAP cache when the following conditions apply:

- The result has already been stored in the cache.
- The request of a result contains controls (LDAP v3 only).
- The result was assembled via DSA chaining. You can enforce caching of results of chained searches by specifying the value **1** for the environment variable `DIRX_LDAP_IGNORE_CACHABLE_FLAG`.
- The request of a result contains as a requested attribute at least one of the attribute types that are configured in the environment variable `DIRX_LDAP_CACHE_BANNED_REQ_ATTR`.
- The request of a result has a base object whose DN matches one of the DNs configured

in the environment variable DIRX\_LDAP\_CACHE\_BANNED\_BASE\_OBJ.

- The request of a result was performed by a user whose DN is not configured in the environment variable DIRX\_LDAP\_CACHE\_ALLOWED\_USER.

If the value is **FALSE**, the LDAP cache is disabled; that is, all LDAP search results are retrieved in the DSA database.

The default value of this attribute is **FALSE** (LDAP cache disabled).

Modifications to this attribute become effective only after a restart of the LDAP server.

To enable or disable the LDAP cache dynamically, use the **dirxadm ldap cache –start** or **–stop** operations. (See the **ldap cache** section in the **dirxadm** command description in the *DirX Directory Administration Reference* for details on a set of operations for dynamic administration of the LDAP cache.)

#### 3.1.14.42.1. Abbreviation

LCA

#### 3.1.14.42.2. LDAP Name(s)

ldapCache

#### 3.1.14.42.3. Syntax (DAP)

Boolean

#### 3.1.14.42.4. Syntax (LDAP)

Boolean (LDAP Style)

#### 3.1.14.42.5. Example (DAP)

```
LCA=FALSE
```

#### 3.1.14.42.6. Example (LDAP)

```
ldapCache=FALSE
```

### 3.1.14.43. LDAP Cache Max Cached Elements

The LDAP Cache Max Cached Elements optional, single-valued attribute is provided to configure the maximum number of LDAP search results that are stored in the LDAP server's cache. An LDAP search result consists always of one or more LDAP result messages. Each result message describes one resulting entry. The last result message contains the result code of the search. If the search results in no entry, the LDAP search result consists only of the last result message containing the result code.

The value selected for this attribute depends on the value of the LDAP Cache Table Size attribute and the system memory available. Each cached result requires at least 1 KB. This size is for a result that contains just the return code and no entries. The cached size of typical results that contain not only the result code but also entries depends on the size of the retrieved attribute sizes. The default value is **10000** (that is if the LDAP cache is filled it consumes at least 10 MB of memory).

Modifications to this attribute become effective only after a restart of the LDAP server.

#### **3.1.14.43.1. Abbreviation**

LCMAXCEL

#### **3.1.14.43.2. LDAP Name(s)**

ldapCacheMaxCachedElements

#### **3.1.14.43.3. Syntax (DAP)**

Integer

#### **3.1.14.43.4. Syntax (LDAP)**

Integer-String (LDAP style)

#### **3.1.14.43.5. Example (DAP)**

```
LCMAXCEL=15000
```

#### **3.1.14.43.6. Example (LDAP)**

```
ldapCacheMaxCachedElements=15000
```

### **3.1.14.44. LDAP Cache Maximum Memory Size**

The LDAP Cache Maximum Memory Size optional, single-valued attribute is provided to configure the maximum memory size in MB used for the LDAP result cache.

The default value is **0** that is the memory size is unlimited. If the value is greater than **0**, the LDAP server stores a result in the cache if there is enough memory space left. If the memory size specified would be exceeded after storing, the result it is not cached.

Keep in mind that the LDAP Cache Max Cached Elements attribute also limits the storing of results. If either limit is exceeded, the LDAP server does not store further results.

Modifications to this attribute become effective only after a restart of the LDAP server.

#### 3.1.14.44.1. Abbreviation

LCMAXMES

#### 3.1.14.44.2. LDAP Name(s)

ldapCacheMaxMemSize

#### 3.1.14.44.3. Syntax (DAP)

Integer

#### 3.1.14.44.4. Syntax (LDAP)

Integer-String (LDAP style)

#### 3.1.14.44.5. Example (DAP)

```
LCMAXMES=256
```

The upper bound of the memory used by the LDAP cache is 256 MB.

#### 3.1.14.44.6. Example (LDAP)

```
ldapCacheMaxMemSize=256
```

### 3.1.14.45. LDAP Cache Min Cache Time

The LDAP Cache Min Cache Time optional, single-valued attribute is provided to configure the minimum number of seconds that an LDAP search result is stored in the LDAP cache. A cached result is not removed from the cache until this amount of time has elapsed. The default value is **0**.

Modifications to this attribute become effective only after a restart of the LDAP server.

#### 3.1.14.45.1. Abbreviation

LCMINCTI

#### 3.1.14.45.2. LDAP Name(s)

ldapCacheMinCacheTime

#### 3.1.14.45.3. Syntax (DAP)

Integer



#### 3.1.14.45.4. Syntax (LDAP)

Integer-String (LDAP style)

#### 3.1.14.45.5. Example (DAP)

```
LCMINCTI=60
```

#### 3.1.14.45.6. Example (LDAP)

```
ldapCacheMinCacheTime=60
```

### 3.1.14.46. LDAP Cache Max Cache Time

The LDAP Cache Max Cache Time attribute optional, single-valued is provided to configure the maximum number of seconds that an LDAP search result is stored in the LDAP cache. The default value is **43200** (12 hours).

Modifications to this attribute become effective only after a restart of the LDAP server.

#### 3.1.14.46.1. Abbreviation

LCMAXCTI

#### 3.1.14.46.2. LDAP Name(s)

ldapCacheMaxCacheTime

#### 3.1.14.46.3. Syntax (DAP)

Integer

#### 3.1.14.46.4. Syntax (LDAP)

Integer-String (LDAP style)

#### 3.1.14.46.5. Example (DAP)

```
LCMAXCTI=7200
```

#### 3.1.14.46.6. Example (LDAP)

```
ldapCacheMaxCacheTime=7200
```

#### 3.1.14.47. LDAP Cache Min Cache Entries

The LDAP Cache Min Cache Entries optional, single-valued attribute is provided to configure the minimum number of entries that an LDAP search result must have to be stored in the LDAP cache. The default value is **0**. (That is that even empty results – results that only contain the result code - are cached.)

Modifications to this attribute become effective only after a restart of the LDAP server.

##### 3.1.14.47.1. Abbreviation

LCMINCEN

##### 3.1.14.47.2. LDAP Name(s)

ldapCacheMinCacheEntries

##### 3.1.14.47.3. Syntax (DAP)

Integer

##### 3.1.14.47.4. Syntax (LDAP)

Integer-String (LDAP style)

##### 3.1.14.47.5. Example (DAP)

```
LCMINCEN=5
```

##### 3.1.14.47.6. Example (LDAP)

```
ldapCacheMinCacheEntries=5
```

#### 3.1.14.48. LDAP Cache Max Cache Entries

The LDAP Cache Max Cache Entries optional, single-valued attribute is provided to configure the maximum number of entries that an LDAP search result can have to be stored in the LDAP cache. The default value is **2000** (an LDAP search result is not stored in the cache if it consists of more than 2,000 entries). By default, the DirX Directory DSA is limited to returning only the first 2,000 entries of a search request.

Modifications to this attribute become effective only after a restart of the LDAP server.

##### 3.1.14.48.1. Abbreviation

LCMAXCEN

#### 3.1.14.48.2. LDAP Name(s)

ldapCacheMaxCacheEntries

#### 3.1.14.48.3. Syntax (DAP)

Integer

#### 3.1.14.48.4. Syntax (LDAP)

Integer-String (LDAP style)

#### 3.1.14.48.5. Example (DAP)

```
LCMAXCEN=2500
```

#### 3.1.14.48.6. Example (LDAP)

```
ldapCacheMaxCacheEntries=2500
```

### 3.1.14.49. LDAP Cache Min Attributes

The LDAP Cache Min Attributes optional, single-valued attribute is provided to configure the minimum number of attributes that an LDAP search result must have to be stored in the LDAP cache. Note that this attribute specifies the minimum number of attributes for the whole result and not for a single entry of the result. The default value is **0**.

Modifications to this attribute become effective only after a restart of the LDAP server.

#### 3.1.14.49.1. Abbreviation

LCMINATS

#### 3.1.14.49.2. LDAP Name(s)

ldapCacheMinAttrs

#### 3.1.14.49.3. Syntax (DAP)

Integer

#### 3.1.14.49.4. Syntax (LDAP)

Integer-String (LDAP style)

#### 3.1.14.49.5. Example (DAP)

```
LCMINATS=10 (A search result must contain at least 10 attributes to
```

be stored in the cache.)

#### 3.1.14.49.6. Example (LDAP)

```
ldapCacheMinAttrs=10
```

#### 3.1.14.50. LDAP Cache Max Attributes

The LDAP Cache Max Attributes optional, single-valued attribute is provided to configure the maximum number of attributes that an LDAP search result can have to be stored in the LDAP cache. Note that this attribute specifies the maximum number of attributes for the whole result and not for a single entry of the result. The default value is **2000**. (That is an LDAP search result is not stored in the cache if it contains more than 2,000 attributes.)

Modifications to this attribute become effective only after a restart of the LDAP server.

##### 3.1.14.50.1. Abbreviation

LCMAXATS

##### 3.1.14.50.2. LDAP Name(s)

ldapCacheMaxAttrs

##### 3.1.14.50.3. Syntax (DAP)

Integer

##### 3.1.14.50.4. Syntax (LDAP)

Integer-String (LDAP style)

##### 3.1.14.50.5. Example (DAP)

```
LCMAXATS=2500 (A search result can contain up to 2,500 attributes to  
be stored in the cache.)
```

##### 3.1.14.50.6. Example (LDAP)

```
ldapCacheMaxAttrs=2500
```

#### 3.1.14.51. LDAP Cache Min Values

The LDAP Cache Min Values optional, single-valued attribute is provided to configure the minimum number of attribute values that an LDAP search result must have to be stored in the LDAP cache. Note that this attribute specifies the minimum number of attribute values

for the whole result and not for a single entry of the result. The default value is **0**.

Modifications to this attribute become effective only after a restart of the LDAP server.

#### **3.1.14.51.1. Abbreviation**

LCMINAVS

#### **3.1.14.51.2. LDAP Name(s)**

ldapCacheMinValues

#### **3.1.14.51.3. Syntax (DAP)**

Integer

#### **3.1.14.51.4. Syntax (LDAP)**

Integer-String (LDAP style)

#### **3.1.14.51.5. Example (DAP)**

```
LCMINAVS=10 (A search result must contain at least 10 attribute
values to be stored in the cache.)
```

#### **3.1.14.51.6. Example (LDAP)**

```
ldapCacheMinValues=10
```

### **3.1.14.52. LDAP Cache Max Values**

The LDAP Cache Max Values optional, single-valued attribute is provided to configure the maximum number of attribute values that an LDAP search result can have to be stored in the LDAP cache. Note that this attribute specifies the maximum number of attribute values for the whole result and not for a single entry of the result. The default value is **2000** (an LDAP search result is not stored in the cache if it contains more than 2,000 attribute values).

Modifications to this attribute become effective only after a restart of the LDAP server.

#### **3.1.14.52.1. Abbreviation**

LCMAXAVS

#### **3.1.14.52.2. LDAP Name(s)**

ldapCacheMaxValues

#### 3.1.14.52.3. Syntax (DAP)

Integer

#### 3.1.14.52.4. Syntax (LDAP)

Integer-String (LDAP style)

#### 3.1.14.52.5. Example (DAP)

```
LCMAXAVS=2500 (A search result can contain up to 2,500 attribute values to be stored in the cache.)
```

#### 3.1.14.52.6. Example (LDAP)

```
ldapCacheMaxValues=2500
```

### 3.1.14.53. LDAP Cache Table Size

The LDAP Cache Table Size optional, single-valued attribute is provided to configure the size of the internal cash table. For best performance, the value of this component should be at least  $(max\_cached\_elements / 3)$ . It must be at least **128**. The default value is **4096**.

DirX Directory prohibits increasing the table size beyond 65535. The number of cached results is not limited (only by memory), but cache performance will decrease if too many results are stored in a small table.

Modifications to this attribute become effective only after a restart of the LDAP server.

#### 3.1.14.53.1. Abbreviation

LCTBSIZE

#### 3.1.14.53.2. LDAP Name(s)

ldapCacheTableSize

#### 3.1.14.53.3. Syntax (DAP)

Integer

#### 3.1.14.53.4. Syntax (LDAP)

Integer-String (LDAP style)

#### 3.1.14.53.5. Example (DAP)

```
LCTBSIZE=5000
```

#### 3.1.14.53.6. Example (LDAP)

```
ldapCacheTableSize=5000
```

#### 3.1.14.54. LDAP Cache Update Strategy

The LDAP Cache Update Strategy optional, single-valued attribute is provided to configure the update strategy of the LDAP cache. Specify one of the following values:

- **0** - (Full erase strategy). After the database has been modified by an update operation, all LDAP search results are removed from the LDAP cache. This strategy should only be applied if there are only a small number of modifications.
- **1** - (DN strategy). After the database has been modified by an update operation, all LDAP search results that may contain the distinguished name of the modified object are removed from the LDAP cache.
- **2** - (Attribute strategy). After the database has been modified by an update operation, all LDAP search results that may contain the modified attribute are removed from the LDAP cache.
- **3** - (DN + attribute strategy.) After the database has been modified by an update operation, all LDAP search results that may contain the distinguished name of the modified object or may contain the modified attribute are removed from the LDAP cache.

The default value is **3**.

Modifications to this attribute become effective only after a restart of the LDAP server.

##### 3.1.14.54.1. Abbreviation

LCUPDSTR

##### 3.1.14.54.2. LDAP Name(s)

ldapCacheUpdateStrategy

##### 3.1.14.54.3. Syntax (DAP)

Integer

##### 3.1.14.54.4. Syntax (LDAP)

Integer-String (LDAP style)

#### 3.1.14.54.5. Example (DAP)

```
LCUPDSTR=0
```

#### 3.1.14.54.6. Example (LDAP)

```
ldapCacheUpdateStrategy=0
```

### 3.1.14.55. LDAP Audit Configuration Subentry CN

The LDAP Audit Configuration Subentry CN attribute specifies the common name attribute of the LDAP Audit configuration subentry. This attribute is an optional single-valued attribute. When this attribute is omitted, the LDAP server reads the LDAP audit configuration subentry with the default common name value **ldapAudit**.

#### 3.1.14.55.1. Abbreviation

LACCN

#### 3.1.14.55.2. LDAP Name(s)

ldapAuditCfgSubentryCN

#### 3.1.14.55.3. Syntax (DAP)

Printable String

#### 3.1.14.55.4. Syntax (LDAP)

Printable String (LDAP style)

#### 3.1.14.55.5. Example (DAP)

```
LACCN=LDAPAAuditConfigurationSubentry
```

#### 3.1.14.55.6. Example (LDAP)

```
ldapAuditCfgSubentryCN=LDAPAAuditConfigurationSubentry
```

### 3.1.14.56. Attributes Controlling LDAP Extended Operations

DirX Directory provides a set of attributes for controlling LDAP extended operations. These attributes specify groups of users that are allowed to perform specific extended operations and groups of operations. The following attributes grant specific rights to perform a specific set of LDAP extended operations:



- LDAP Extended Operations Admins - allows users listed in this attribute to perform all LDAP extended operations.
- LDAP Extended Operations Admin Groups - allows users who are members of the group(s) listed in this attribute to perform all LDAP extended operations.
- LDAP Extended Operations Read Users - allows users listed in this attribute to perform all LDAP extended read operations.
- LDAP Extended Operations Read Groups - allows users who are members of the group(s) listed in this attribute to perform all LDAP extended read operations.
- LDAP Extended Operations Execute Users - allows users listed in this attribute to perform all LDAP extended execute operations.
- LDAP Extended Operations Execute Groups - allows users who are members of the group(s) listed in this attribute to perform all LDAP extended execute operations.
- LDAP Extended Operations Monitoring Users - allows users listed in this attribute to perform all LDAP extended monitoring operations.
- LDAP Extended Operations Monitoring Groups - allows users who are members of the group(s) listed in this attribute to perform all LDAP extended monitoring operations.

At startup, the LDAP server reads the groups attributes LDAP Extended Operations Admin Groups, LDAP Extended Operations Read Groups, LDAP Extended Operations Execute Groups and LDAP Extended Operations Monitoring Groups and evaluates their members. Changes made to these groups attributes while the LDAP server is running can be dynamically activated with the LDAP extended operation **ldap\_cfg\_upd**. See the **ldap\_cfg\_upd** reference description in *DirX Directory LDAP Extended Operations*.

The number of members in a group affects the LDAP server's startup time. Specifying very large groups - more than 1,000 members - may result in a long wait time before the LDAP server becomes responsive after startup.

The following attributes group the LDAP extended operations:

- LDAP Extended Read Operations - specifies all operations that retrieve data and require read access rights.
- LDAP Extended Execute Operations - specifies all operations that execute or change data and require execute access rights.
- LDAP Extended Monitoring Operations - specifies all operations that are used for special Nagios monitoring and require monitoring access rights.

There are default LDAP extended read, execute and monitoring operations. These default operations specify the default values for the attributes grouping the LDAP extended operations. If an LDAP extended operation is not contained in any attribute, the default grouping applies. An LDAP extended operation can only be contained in one grouping attribute. The following LDAP extended operations do not belong to any group and no specific access rights are required to perform them. It is recommended not to add them to a group attribute because they are used for internal operations:

- startTLS (**dirxcp startTLS**) - enables SSL/TLS security for a LDAP connection; that is, the LDAP server performs extended operations. Note that the LDAP Start TLS Enabled

(LSTEN, ldapStartTLSEnabled) attribute must be set to 1 to perform this operation.

- ldap\_extop\_info - displays a list of all supported extended operations.
- ldap\_cfg\_defaults - displays the default values for the LDAP server configuration.
- ldap\_ssl\_ciphers - displays the LDAP SSL cipher names.
- ldap\_show\_config\_dsas - displays the contact DSAs configured for an LDAP server.

See the **dirxextop** command reference page in the *DirX Directory Administration Reference* for a list of extended operations provided by DirX Directory and how to perform them.

The following sections describe the attributes that control the LDAP extended operations.

#### 3.1.14.56.1. LDAP Extended Operations Admins

The LDAP Extended Operations Admins attribute controls the accessibility of extended operations. This attribute is an optional, multi-valued attribute. The default is that performing extended operations is prohibited; that is, the LDAP server only performs the extended operations that require no access rights (see the list in the section **Attributes Controlling LDAP Extended Operations** above) or that users are allowed to perform due to the values specified in the LDAP Extended Operations Read, Execute, or Monitoring attributes.

The users listed in this attribute are allowed to execute all LDAP extended operations.

Specify the DN attribute values in LDAP format; for example, cn=admin,o=my-company. (See the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds** for details.) Specify complete distinguished names of users that can perform an LDAP bind.

If a client binds via SASL with the EXTERNAL mechanism, the DN is taken from the user's certificate subject field. independent of the mapping method defined by the attribute **LDAP SASL Authz Id Mapping**. Specify the user's subject DN in LDAP format; for example, uid=12345,givenName=Max Mustermann,o=my-company. (See the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds**.)

The LDAP server performs the extended operation if the client's DN in the bind operation or the user's certificate subject field matches one of the attribute values.

The value(s) of the attribute are treated and compared as LDAP DN's by eliminating unspecific blanks and normalizing to lowercase, for example:

CN = Admin, O=My-Company is normalized to cn=admin,o=my-company

The keyword **all** specifies that everyone can perform extended operations and the keyword **none** specifies that no one can perform extended operations.

#### 3.1.14.56.2. Abbreviation

LEXOA

#### 3.1.14.56.3. LDAP Name(s)

ldapExtOpAdmins

#### 3.1.14.56.4. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.56.5. Syntax (LDAP)

Directory-String {32768} (LDAP Style)

#### 3.1.14.56.6. Example (DAP)

```
{LEXOA=cn++\+=admin++\+,o++\+=my-  
company;cn++\+=metaadm++\+,o++\+=my-company}
```

The LDAP server performs extended operations only if one of the two My-Company administrators (cn=admin,o= my-company or cn=metaadm,o=my-company) is bound.

#### 3.1.14.56.7. Example (LDAP)

```
{ldapExtOpAdmins=cn=admin,o=my-company;cn=metaadm,o=my-company}
```

#### 3.1.14.56.8. LDAP Extended Operations Admin Groups

The LDAP Extended Operations Admin Groups attribute allows you to control the accessibility of extended operations by specifying groups. This attribute is an optional, multi-valued attribute. The groups must have the object class **groupOfNames**.

The members of the groups specified in this attribute are allowed to execute all LDAP extended operations.

The default is that performing extended operations is prohibited; that is, the LDAP server only performs the extended operations that require no access rights (see the list in the section **Attributes Controlling LDAP Extended Operations** above) or that users are allowed to perform due to the values specified in the LDAP Extended Operations Read, Execute or Monitoring attributes.

Specify the DN attribute values in LDAP format; for example, cn=admin,o=my-company. (See the section **Distinguished Names** in the chapter **DirX Directory String Representation for LDAP Binds** for details.) Specify complete distinguished names of groups that contain members that can perform an LDAP bind.

If a client binds via SASL with the EXTERNAL mechanism, the DN is taken from the user's certificate subject field independent of the mapping method defined by the attribute **LDAP SASL Authz Id Mapping**. There is no difference between simple- and SASL-bound

users when the privilege is granted due to the group membership.

The LDAP server performs the extended operation if the client's DN in the bind operation matches a member name in one of the specified group attribute values.

The value(s) of the attribute are treated and compared as LDAP DN's by eliminating unspecific blanks and normalizing to lowercase, for example:

CN = Admin, O=My-Company is normalized to cn=admin,o=my-company

#### **3.1.14.56.9. Abbreviation**

LEXOAG

#### **3.1.14.56.10. LDAP Name(s)**

ldapExtOpAdminGroups

#### **3.1.14.56.11. Syntax (DAP)**

Directory-String {32768}

#### **3.1.14.56.12. Syntax (LDAP)**

Directory-String {32768} (LDAP Style)

#### **3.1.14.56.13. Example (DAP)**

```
{LEXOAG=cn\=adminGroup\,o\=my-company;cn\=metaadmGroup\,o\=my-company}
```

The LDAP server performs extended operations only for those users that are members of one of the two listed My-Company administrator groups: cn=adminGroup,o=my-company or cn=metaadmGroup,o=my-company.

#### **3.1.14.56.14. Example (LDAP)**

```
{ldapExtOpAdminGroups=cn=adminGroup,o=my-company;cn=metaadmGroup,o=my-company}
```

#### **3.1.14.56.15. LDAP Extended Operations Read Users**

The LDAP Extended Operations Read Users attribute controls the accessibility of LDAP extended read operations. This attribute is an optional, multi-valued attribute. The LDAP Extended Read Operations attribute specifies all LDAP extended read operations.

The default is that no user is allowed to perform LDAP extended read operations.

Specify the DN attribute values as described in the section **LDAP Extended Operations Admins**.

#### 3.1.14.56.16. Abbreviation

LEXORU

#### 3.1.14.56.17. LDAP Name(s)

ldapExtOpReadUsers

#### 3.1.14.56.18. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.56.19. Syntax (LDAP)

Directory-String {32768} (LDAP Style)

#### 3.1.14.56.20. Example (DAP)

```
{LEXORU=cn\=admin\,o\=my-company;cn\=metaadm\,o\=my-company}
```

The LDAP server performs LDAP extended read operations only if one of the two My-Company administrators (cn=admin,o= my-company or cn=metaadm,o=my-company) is bound.

#### 3.1.14.56.21. Example (LDAP)

```
{ldapExtOpReadUsers=cn=admin,o=my-company;cn=metaadm,o=my-company}
```

#### 3.1.14.56.22. LDAP Extended Operations Read Groups

The LDAP Extended Operations Read Groups attribute controls the accessibility of LDAP extended read operations by specifying groups. The groups must have the object class **groupOfNames**.

Members of the groups specified in this attribute are allowed to execute the LDAP extended operations that require read privilege.

All members of the specified groups have the same permissions as the users listed explicitly with their entry DN's in the LDAP Extended Operations Read Users attribute.

This attribute is an optional, multi-valued attribute.

The LDAP Extended Operations Read Groups attribute specifies all LDAP extended read operations.

The default is that no group is allowed to perform LDAP extended read operations.

Specify the DN attribute values as described in the section **LDAP Extended Operations Admin Groups**.

#### 3.1.14.56.23. Abbreviation

LEXORG

#### 3.1.14.56.24. LDAP Name(s)

ldapExtOpReadGroups

#### 3.1.14.56.25. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.56.26. Syntax (LDAP)

Directory-String {32768} (LDAP Style)

#### 3.1.14.56.27. Example (DAP)

```
{LEXORG=cn\=readUserGroup\, o\=my-company; }
```

The LDAP server performs LDAP extended read operations only if the bound user is a member of the specified group cn=ReadUserGroup,o=my-company.

#### 3.1.14.56.28. Example (LDAP)

```
{ldapExtOpReadUsers=cn=ReadUserGroup, o=my-company}
```

#### 3.1.14.56.29. LDAP Extended Operations Execute Users

The LDAP Extended Operations Execute Users attribute controls the accessibility of LDAP extended execute operations. This attribute is an optional, multi-valued attribute. The LDAP Extended Execute Operations attribute specifies all LDAP extended execute operations.

The default is that no user is allowed to perform LDAP extended execute operations.

Specify the DN attribute values as described in section **LDAP Extended Operations Admins**.

#### 3.1.14.56.30. Abbreviation

LEXOEU

#### 3.1.14.56.31. LDAP Name(s)

ldapExtOpExecuteUsers

#### 3.1.14.56.32. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.56.33. Syntax (LDAP)

Directory-String {32768} (LDAP Style)

#### 3.1.14.56.34. Example (DAP)

```
{LEXOEU=cn\=admin\,o\=my-company;cn\=metaadm\,o\=my-company}
```

The LDAP server performs LDAP extended execute operations only if one of the two My-Company administrators (cn=admin,o= my-company or cn=metaadm,o=my-company) is bound.

#### 3.1.14.56.35. Example (LDAP)

```
{ldapExtOpExecuteUsers=cn=admin,o=my-company;cn=metaadm,o=my-company}
```

#### 3.1.14.56.36. LDAP Extended Operations Execute Groups

The LDAP Extended Operations Execute Groups attribute controls the accessibility of LDAP extended execute operations by specifying groups. The groups must have the object class **groupOfNames**.

Members of the groups specified in this attribute are allowed to execute the LDAP extended operations that require execute privilege.

All members of the specified groups have the same permissions as the users listed explicitly with their entry DN's in the LDAP Extended Operations Execute Users attribute.

This attribute is an optional, multi-valued attribute.

The LDAP Extended Operations Execute Groups attribute specifies all LDAP extended execute operations.

The default is that no group is allowed to perform LDAP extended execute operations.

Specify the DN attribute values as described in the section **LDAP Extended Operations Admin Groups**.

#### 3.1.14.56.37. Abbreviation

LEXOEG

#### 3.1.14.56.38. LDAP Name(s)

ldapExtOpExecuteGroups

#### 3.1.14.56.39. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.56.40. Syntax (LDAP)

Directory-String {32768} (LDAP Style)

#### 3.1.14.56.41. Example (DAP)

```
{LEXOEG=cn\=Group4711\, o\=my-company
```

The LDAP server performs LDAP extended read operations only if the bound user is a member of the specified group cn=Group4711,o= my-company.

#### 3.1.14.56.42. Example (LDAP)

```
{ldapExtOpExecuteGroups=cn=Group4711, o=my-company}
```

#### 3.1.14.56.43. LDAP Extended Operations Monitoring Users

The LDAP Extended Operations Execute Users attribute controls the accessibility of LDAP extended monitoring operations. This attribute is an optional, multi-valued attribute. The LDAP Extended Monitoring Operations attribute specifies all LDAP extended monitoring operations.

The default is that no user is allowed to perform LDAP extended monitoring operations.

Specify the DN attribute values as described in section **LDAP Extended Operations Admins**.

#### 3.1.14.56.44. Abbreviation

LEXOMU

#### 3.1.14.56.45. LDAP Name(s)

ldapExtOpMonitoringUsers

#### 3.1.14.56.46. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.56.47. Syntax (LDAP)

Directory-String {32768} (LDAP Style)



#### 3.1.14.56.48. Example (DAP)

```
{LEXOMU=cn\=admin\,o\=my-company;cn\=metaadm\,o\=my-company}
```

The LDAP server performs LDAP extended execute operations only if one of the two My-Company administrators (cn=admin,o= my-company or cn=metaadm,o=my-company) is bound.

#### 3.1.14.56.49. Example (LDAP)

```
{ldapExtOpMonitoringUsers=cn=admin,o=my-company;cn=metaadm,o=my-company}
```

#### 3.1.14.56.50. LDAP Extended Operations Monitoring Groups

The LDAP Extended Operations Monitoring Groups attribute controls the accessibility of LDAP extended monitoring operations by specifying groups. The groups must have the object class **groupOfNames**.

Members of the groups specified in this attribute are allowed to execute the LDAP extended operations that require monitoring privilege.

All members of the specified groups have the same permissions as the users listed explicitly with their entry DN's in the LDAP Extended Operations Monitoring Users attribute.

This attribute is an optional, multi-valued attribute.

The LDAP Extended Operations Monitoring Groups attribute specifies all LDAP extended monitoring operations.

The default is that no group is allowed to perform LDAP extended monitoring operations.

Specify the DN attribute values as described in the section **LDAP Extended Operations Admin Groups**.

#### 3.1.14.56.51. Abbreviation

LEXOMG

#### 3.1.14.56.52. LDAP Name(s)

ldapExtOpMonitoringGroups

#### 3.1.14.56.53. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.56.54. Syntax (LDAP)

Directory-String {32768} (LDAP Style)

#### 3.1.14.56.55. Example (DAP)

```
{LEXOMG=cn\=NagiosUserGroup\,o\=my-company}
```

The LDAP server performs LDAP extended read operations only if the bound user is a member of the specified group `cn=NagiosUserGroup,o= my-company`.

#### 3.1.14.56.56. Example (LDAP)

```
{ldapExtOpMonitoringGroups=cn=NagiosUserGroup,o=my-company}
```

#### 3.1.14.56.57. LDAP Extended Read Operations

The LDAP Extended Read Operations attribute specifies all LDAP extended read operations. This attribute is an optional, multi-valued attribute.

Specify the OID or the keyword of the LDAP extended operation. See the **dirxextop** reference page in the *DirX Directory Administration Reference* for a complete list of extended operations.

The default LDAP extended read operations are:

- ldap\_mib\_static
- ldap\_mib\_total
- ldap\_mib\_current
- ldap\_mib\_assoc
- ldap\_mib\_env
- ldap\_cache\_info
- ldap\_audit\_info
- ldap\_host\_ips
- ldap\_pstack
- ldap\_bt\_dump
- ldap\_idm\_dump
- ldap\_cache\_dump
- ldap\_rusage
- ldap\_pfiles
- ldap\_top
- ldap\_status

- ldap\_pmap
- ldap\_ctxinfo
- ldap\_netstat
- ldap\_exceptions
- ldap\_op\_history
- ldap\_show\_op\_priv
- ldap\_show\_priv\_users
- ldap\_show\_cfg\_audit
- ldap\_show\_cfg\_general
- ldap\_show\_cfg\_ssl
- ldap\_show\_cfg\_upd\_attr
- ldap\_show\_cfg\_upd\_attr\_history
- dsa\_nmi\_show
- dsa\_dbam\_mib\_show
- dsa\_entry\_info\_get
- dsa\_nmi\_show\_dap
- dsa\_nmi\_show\_dsp
- dsa\_nmi\_show\_disp
- dsa\_pstack
- dsa\_bt\_dump
- dsa\_idm\_dump
- dsa\_disp\_flow\_counter
- dsa\_rusage
- dsa\_pfiles
- dsa\_top
- dsa\_status
- dsa\_pmap
- dsa\_dbam\_devinfo
- dsa\_iostat
- dsa\_ctxinfo
- dsa\_audit\_info
- dsa\_dbam\_config
- dsa\_netstat
- dsa\_exceptions
- dsa\_history\_dap

- dsa\_history\_dsp
- dsa\_history\_disp
- dsa\_index\_info
- dsa\_disp\_monitor

Modify the attribute value by simply adding or deleting an LDAP extended operation.

To perform the LDAP extended read operations, the client's DN must match either a value contained in the LDAP Extended Operations Admins attribute or a value in the LDAP Extended Operations Read Users attribute.

#### 3.1.14.56.58. Abbreviation

LEXROP

#### 3.1.14.56.59. LDAP Name(s)

ldapExtReadOperations

#### 3.1.14.56.60. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.56.61. Syntax (LDAP)

Directory-String {32768} (LDAP Style)

#### 3.1.14.56.62. Example (DAP)

```
{LEXROP=ldap_dsa_getopmode;1.3.12.2.1107.1.3.2.13.33}
```

The LDAP extended read operations group contains all default LDAP extended read operations not contained in any other group attribute and the operations displaying the DSA operation mode (ldap\_dsa\_getopmode) and the status of a database backup (1.3.12.2.1107.1.3.2.13.33).

#### 3.1.14.56.63. Example (LDAP)

```
{ldapExtReadOperations=ldap_dsa_getopmode;1.3.12.2.1107.1.3.2.13.33}
```

#### 3.1.14.56.64. LDAP Extended Execute Operations

The LDAP Extended Execute Operations attribute specifies all LDAP extended execute operations. This attribute is an optional, multi-valued attribute.

Specify the OID or the keyword of the LDAP extended operation. See the **dirxextop** reference page in the *DirX Directory Administration Reference* for a complete list of extended operations.

The default LDAP extended execute operations are:

- ldap\_audit\_start
- ldap\_audit\_stop
- ldap\_cache\_start
- ldap\_cache\_stop
- ldap\_cache\_clear
- ldap\_audit\_evaluate
- ldap\_audit\_err
- ldap\_disable\_config\_dsa
- ldap\_enable\_config\_dsa
- ldap\_cfg\_upd
- dsa\_dbam\_mib\_enable
- dsa\_dbam\_mib\_disable
- dsa\_audit\_evaluate
- dsa\_audit\_enable
- dsa\_audit\_disable
- dsa\_dbam\_backup
- dsa\_dbam\_aac

Modify the attribute value by simply adding or deleting an LDAP extended operation.

To perform the LDAP extended execute operations, the client's DN must match either a value contained in the LDAP Extended Operations Admins attribute or a value in the LDAP Extended Operations Execute Users attribute.

#### **3.1.14.56.65. Abbreviation**

LEXEOP

#### **3.1.14.56.66. LDAP Name(s)**

ldapExtExecuteOperations

#### **3.1.14.56.67. Syntax (DAP)**

Directory-String {32768}

#### **3.1.14.56.68. Syntax (LDAP)**

Directory-String {32768} (LDAP Style)

#### 3.1.14.56.69. Example (DAP)

```
{LEXEOP=ldap_dsa_getopmode;1.3.12.2.1107.1.3.2.13.33}
```

The LDAP extended execute operations group contains all default LDAP extended execute operations not contained in any other group attribute and the operations displaying the DSA operation mode (ldap\_dsa\_getopmode) and the status of a database backup (1.3.12.2.1107.1.3.2.13.33).

#### 3.1.14.56.70. Example (LDAP)

```
{ldapExtExecuteOperations=ldap_dsa_getopmode;1.3.12.2.1107.1.3.2.13.33}
```

#### 3.1.14.56.71. LDAP Extended Monitoring Operations

The LDAP Extended Monitoring Operations attribute specifies all LDAP extended monitoring operations. This attribute is an optional, multi-valued attribute.

Specify the OID or the keyword of the LDAP extended operation. See the **dirxextop** reference page in the *DirX Directory Administration Reference* for a complete list of extended operations.

The default LDAP extended monitoring operations are:

- ldap\_mib\_total\_tevd
- ldap\_mib\_current\_tevd
- ldap\_no\_of\_fds
- dsa\_server\_files
- dsa\_dbam\_preload\_status
- dsa\_backup\_status
- dsa\_no\_of\_fds

The Nagios plugins use the operations listed above to retrieve monitoring information from the DirX Directory servers.

Modify the attribute value by simply adding or deleting an LDAP extended operation.

To perform the LDAP extended read operations, the client's DN must match either a value contained in the LDAP Extended Operations Admins attribute or a value in the LDAP Extended Operations Monitoring Users attribute.

#### 3.1.14.56.72. Abbreviation

LEXMOP

#### 3.1.14.56.73. LDAP Name(s)

ldapExtMonitoringOperations

#### 3.1.14.56.74. Syntax (DAP)

Directory-String {32768}

#### 3.1.14.56.75. Syntax (LDAP)

Directory-String {32768} (LDAP Style)

#### 3.1.14.56.76. Example (DAP)

```
{LEXMOP=ldap_mib_static;1.3.12.2.1107.1.3.2.11.5}
```

The LDAP extended monitoring operations group contains all default LDAP extended monitoring operations not contained in any other group attribute and the operations displaying the LDAP MIB static table (ldap+\_+\_mib+\_+\_static) and the LDAP MIB environment table (1.3.12.2.1107.1.3.2.11.5).

#### 3.1.14.56.77. Example (LDAP)

```
{ldapExtMonitoringOperations=ldap_mib_static;1.3.12.2.1107.1.3.2.11.5  
}
```

### 3.1.15. Attributes for LDAP Server SSL Configuration

Attributes for LDAP server SSL configuration are attributes of the LDAP server SSL configuration subentry with the default name **/CN=LdapSSLConfiguration**. (This name is referenced in the LDAP SSL Configuration Subentry CN (LSCCN) attribute of the LDAP Server Configuration subentry.) All attributes are optional. Run the script **ldap\_sslcfg.cp** (using **dirxcp**) to create a default LDAP SSL configuration subentry that specifies default values for the DirX Directory LDAP server. Use the **dirxcp obj** operations to modify the attributes of the default LDAP SSL configuration subentry or to create new LDAP SSL configuration subentries. Like all other subentries, it cannot be created via LDAP. However, it can be modified via LDAP. Modifications of an LDAP SSL configuration subentry become effective after a restart of the LDAP server. The LDAP server SSL configuration subentry is only read at start time when SSL is enabled; that is, when the value of the LDAP Secure Port Number attribute is greater than zero. (See section **LDAP Secure Port Number** for details.)

#### 3.1.15.1. LDAP Security Level A

The LDAP Security Level A single-valued attribute specifies the minimum required security level of the LDAP server's own key material for OpenSSL. (See **LDAP Own Keymaterial PEM** attribute and section **LDAP Server Key Material** in the *DirX Directory Administration*

Reference for details.)

The default value (as of OpenSSL V3.1.0) for this attribute is **1**.

Specify one of the following values:

#### **0 - Level 0**

Everything is permitted.

This setting retains compatibility with previous versions of OpenSSL.

#### **1 - Level 1 (OpenSSL default)**

The security level corresponds to a minimum of 80 bits of security.

Any parameters offering below 80 bits of security are excluded.

As a result, RSA, DSA, and DH keys shorter than 1024 bits and ECC keys shorter than 160 bits are prohibited.

All export cipher suites are prohibited since they all offer less than 80 bits of security.

SSL version 2 is prohibited.

Any cipher suite using MD5 for Mac OS X is also prohibited.

#### **2 - Level 2 (The default *DirX* Directory keymaterial fullfills the requirements for this level.)**

The security level is set to 112 bits of security.

As a result, RSA, DSA, and DH keys shorter than 2048 bits and ECC keys shorter than 224 bits are prohibited.

In addition to the Level 1 exclusions, any cipher suite using RC4 is also prohibited.

SSL version 3 is also not allowed.

Compression is disabled.

#### **3 - Level 3**

The security level is set to 128 bits of security.

As a result, RSA, DSA, and DH keys shorter than 3072 bits and ECC keys shorter than 256 bits are prohibited.

In addition to the Level 2 exclusions, cipher suites that do not offer forward secrecy are prohibited.

TLS versions below 1.1 are not permitted.

Session tickets are disabled.

#### **4 - Level 4**

The security level is set to 192 bits of security.

As a result, RSA, DSA, and DH keys shorter than 7680 bits and ECC keys shorter than 384 bits are prohibited.

Cipher suites using SHA1 for Mac OS X are prohibited.

TLS versions below 1.2 are not permitted.

#### **5 - Level 5**

Security level set to 256 bits of security.

As a result, RSA, DSA, and DH keys shorter than 15360 bits and ECC keys shorter than 512 bits are prohibited.

#### **Notes:**



1. If no value is specified (the attribute is missing in the subentry) a default takes place that is chosen by OpenSSL.
2. The values may differ in meaning for later OpenSSL versions. You may want to search for the function **SSL\_set\_security\_level** in the OpenSSL documentation for more details and levels.
3. Be aware that the higher the level, the more challenging it will be to provide a suitable PKI key material. The current default (OpenSSL V3.1.0) for both values is 1. The demo PKI key material delivered along with the DirX Directory installation is suitable for levels 1 and 2 only. Thus, if you set higher levels, ensure that your PKI material fits the needs.

#### 3.1.15.1.1. Abbreviation

LSLA

#### 3.1.15.1.2. LDAP Name(s)

ldapSecLevelA

#### 3.1.15.1.3. Syntax (DAP)

Integer

#### 3.1.15.1.4. Example (DAP)

```
LSLA=2
```

#### 3.1.15.1.5. Example (LDAP)

```
ldapSecLevelA=2
```

### 3.1.15.2. LDAP Security Level B

The LDAP Security Level B single-valued attribute specifies the minimum required security level for incoming LDAP client connections for OpenSSL.

The default value (as of OpenSSL V3.1.0) for this attribute is **1**.

See the **LDAP Security Level A** attribute for the allowed values and details.

#### 3.1.15.2.1. Abbreviation

LSLB

#### 3.1.15.2.2. LDAP Name(s)

ldapSecLevelB

#### 3.1.15.2.3. Syntax (DAP)

Integer

#### 3.1.15.2.4. Example (DAP)

```
LSLB=2
```

#### 3.1.15.2.5. Example (LDAP)

```
ldapSecLevelB=2
```

### 3.1.15.3. LDAP Supported Encryption Strength

The LDAP Supported Encryption Strength multi-valued attribute specifies the cipher suites that are accepted in the SSL handshake protocol for version TLSv1.0 up to TLSv1.2. Cipher suites are algorithms for signing, encryption, and hashing.

The values specified in this attribute comprise short names and explicit cipher suite names. Examples for valid short names values are **HIGH**, **MEDIUM**, **RSA** or **ALL**. Examples for explicit cipher suite names are **ECDH-RSA-DES-CBC3-SHA** or **EDH-RSA-DES-CBC-SHA**.

Use the **openssl** command "**openssl ciphers**" to retrieve the complete list of explicit cipher suite names supported by the DirX Directory LDAP server. The complete list of the supported short names are documented on <https://www.openssl.org/docs/man1.0.2/man1/ciphers.html>.

The default value of this attribute is **RSA**: all cipher suites that use the RSA algorithm are accepted.

See the description of the attribute LDAP Supported Encryption Strength Extended for details on how to set ciphers for the TLS protocol version 1.3.

#### 3.1.15.3.1. Abbreviation

LSSES

#### 3.1.15.3.2. LDAP Name(s)

supportedEncryptionStrength

#### 3.1.15.3.3. Syntax (DAP)

Directory String {40}

#### 3.1.15.3.4. Example (DAP)

```
LSSSES=ECDHE-RSA-AES256-SHA; ECDH-RSA-DES-CBC3-SHA; ECDH-RSA-AES128-SHA
```

These values explicitly set the specified three values as the only accepted cipher suites.

#### 3.1.15.3.5. Example (LDAP)

```
supportedEncryptionStrength=HIGH; RSA
```

These two values set the supported cipher suites to those that belong to the groups HIGH or RSA.

#### 3.1.15.4. LDAP Supported Encryption Strength Extended

The LDAP Supported Encryption Strength Extended multi-valued attribute specifies the cipher suites that are accepted in the SSL handshake protocol for version TLSv1.3. Cipher suites are algorithms for signing, encryption, and hashing.

The values specified in this attribute correspond to the TLS1.3 specification. The following list comprises all five valid cipher suites for the TLS 1.3 protocol:

TLS\_AES\_256\_GCM\_SHA384

TLS\_CHACHA20\_POLY1305\_SHA256

TLS\_AES\_128\_GCM\_SHA256

TLS\_AES\_128\_CCM\_8\_SHA256

TLS\_AES\_128\_CCM\_SHA256

Possible values are also the keyword **ALL** (all five cipher suites) or **DEFAULT** (the first three cipher suites of the previous list).

The default value of this attribute is **DEFAULT**.

See the description of the attribute LDAP Supported Encryption Strength for details on how to set accepted ciphers for the TLS protocol versions 1.0 up to 1.2.

##### 3.1.15.4.1. Abbreviation

LSSSEX

##### 3.1.15.4.2. LDAP Name(s)

supportedEncryptionStrengthExt

##### 3.1.15.4.3. Syntax (DAP)

Directory String {1024}

#### 3.1.15.4.4. Example (DAP)

```
LSSSESEX={TLS++_++AES++_++256++_++GCM++_++SHA384;TLS++_++AES++_++128++_++GCM++_++SHA256}
```

These values explicitly set the specified two values as the only accepted cipher suites.

#### 3.1.15.4.5. Example (LDAP)

```
supportedEncryptionStrengthExt=DEFAULT
```

These value set the default cipher suites for the TLSv1.3 protocol.

### 3.1.15.5. LDAP Supported Security Protocols

The LDAP Supported Security Protocols multi-valued attribute specifies the supported security protocols. The actual protocol version used for a particular SSL/TLS connection is negotiated during SSL connection establishment. Valid values are:

- **TLSv1.0** - The LDAP server accepts TLSv1.0 security protocol.
- **TLSv1.1** - The LDAP server accepts TLSv1.1 security protocol.
- **TLSv1.2** - The LDAP server accepts TLSv1.2 security protocol.
- **TLSv1.3** - The LDAP server accepts TLSv1.3 security protocol.

By default, the LDAP server accepts all versions of the security protocol listed above (so the default for this attribute is all values).

#### 3.1.15.5.1. Abbreviation

LSSSP

#### 3.1.15.5.2. LDAP Name(s)

supportedSecurityProtocols

#### 3.1.15.5.3. Syntax (DAP)

Directory String {128}

#### 3.1.15.5.4. Example (DAP)

```
LSSSP=TLSv1.0
```

#### 3.1.15.5.5. Example (LDAP)

```
supportedSecurityProtocols=TLSv1.2;TLSv1.2
```

### 3.1.15.6. LDAP SSL Trace File

The LDAP SSL Trace File attribute specifies the full pathname of the directory in which the SSL-specific log files are located. SSL-specific log file names are in the format:

**sslpid.log**

where *pid* is the process ID of the LDAP server process that created the log file.

The default value of this attribute is *install\_path/ldap/log*

#### 3.1.15.6.1. Abbreviation

LSTFI

#### 3.1.15.6.2. LDAP Name(s)

SSLTraceFile

#### 3.1.15.6.3. Syntax (DAP)

Directory String {1024}

#### 3.1.15.6.4. Example (DAP)

(on Linux systems)

```
LSTFI=/opt/dirx/ldap/log/ssl
```

(on Windows systems)

```
LSTFI={C:/Program Files/DirX/Directory/ldap/log/ssl}
```

#### 3.1.15.6.5. Example (LDAP)

(on Linux systems)

```
SSLTraceFile=/opt/dirx/ldap/log/ssl
```

(on Windows systems)

```
{SSLTraceFile=C:/Program Files/DirX/Directory/ldap/log/ssl}
```

Note that the forward slash (/) and the backslash (\) characters can be used as separators

for Linux and Windows pathnames because the value is read and interpreted only by the **dirxldapv3** program. Note also that the curly braces ({} ) in the Windows example are necessary because of the blank character contained in the example's pathname.

### 3.1.15.7. LDAP SSL Trace Level

The LDAP SSL Trace Level attribute specifies the level of SSL internal logging. Valid values are the integers **0** and **1**, where **0** indicates that SSL logging is turned off.

The default value of this attribute is **0** (SSL logging is turned off).

It is recommended to turn on SSL internal logging only if TLS handshake problems must be debugged. The logging can result in huge logging files that may contain confidential data.

#### 3.1.15.7.1. Abbreviation

LSTLE

#### 3.1.15.7.2. LDAP Name(s)

SSLTraceLevel

#### 3.1.15.7.3. Syntax (DAP)

Integer

#### 3.1.15.7.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.15.7.5. Example (DAP)

```
LSTLE=1
```

#### 3.1.15.7.6. Example (LDAP)

```
SSLTraceLevel=1
```

### 3.1.15.8. LDAP SSL Client Authentication Required

The LDAP SSL Client Authentication Required attribute specifies whether or not the LDAP server requires SSL Client Authentication for SSL protected connections. When a client performs SSL Client Authentication, he/she requires a user certificate and a private key.

If the value is **FALSE**, clients are not required to perform SSL Client Authentication when establishing the SSL/TLS connection.

If the value is **TRUE**, clients are required to perform SSL Client Authentication when establishing the SSL/TLS connection.

In the LDAPv3 protocol, SSL Client Authentication is performed with a SASL bind when the **-mechanism EXTERNAL** option is specified. (See the recommendations entitled *Lightweight Directory Access Protocol (LDAP): The Protocol (RFC 4511)* and *Lightweight Directory Access Protocol (LDAP): Authentication Methods and Security Mechanisms (RFC 4513)* for details.)

The **dirxcp** command performs SASL-authenticated binds when the **-sasl** option is specified in the **obj bind** operation. See the **obj bind** operation in the *DirX Directory Administration Reference* for details.

The default value of this attribute is **FALSE** (SSL Client Authentication is not required).

In the security layer, SSL/TLS client authentication takes place during connection establishment. The LDAP server's security layer verifies the authentication during the process that establishes the connection.

If the LDAP SSL Client Authentication Required attribute is **TRUE**, the LDAP server processing also depends on the settings of the following attributes:

- LDAP SASL Authz Id Mapping (ldapSaslAuthzIdMapping)
- LDAP Trusted CA Certs (ldapTrustedCACerts)
- LDAP SSL CRL Checking (ldapSslCrlChecking)

The LDAP server maps an LDAP SASL bind with the **EXTERNAL** mechanism onto a special DAP bind to the DSA. The DSA checks whether a trusted LDAP server has issued this DAP bind by evaluating the IP address of the LDAP server's host. By default, only local LDAP servers are trusted, but you can specify the range of accepted IP addresses in the environment variable DIRX\_SSL\_HOSTS. If the DSA does not trust the LDAP server due to the IP address, the following error message is written to the EXC files or Windows eventing: **Host ip-address not allowed as SSL bind initiator.**

#### 3.1.15.8.1. Abbreviation

LSCAR

#### 3.1.15.8.2. LDAP Name(s)

requireSSLClientAuth

#### 3.1.15.8.3. Syntax (DAP)

Boolean

#### 3.1.15.8.4. Syntax (LDAP)

Boolean (LDAP Style)

#### 3.1.15.8.5. Example (DAP)

```
LSCAR=FALSE
```

#### 3.1.15.8.6. Example (LDAP)

```
requireSSLClientAuth=FALSE
```

#### 3.1.15.9. LDAP SASL Authz Id Mapping

The LDAP SASL Authz Id Mapping attribute specifies how a client entity authenticated by an SASL-authenticated bind over SSL-protected LDAPv3 protocol (ldap sasl bind) with the **EXTERNAL** mechanism is mapped onto an Authorization ID.

When performing an LDAP SASL bind with the mechanism **EXTERNAL**, the SSL/TLS layer performs the client authentication. It is a mutual strong authentication performed between the LDAP client and the LDAP server. On successful authentication, the user's public key certificate is sent to the DSA. The LDAP SASL Authz Id Mapping attribute allows you to define, on a LDAP server process basis, the mapping from the user's certificate onto the distinguished name that will be used for successive access control decisions. The attribute LDAP SASL Authz Id Mapping attribute is only evaluated for SASL binds; that is, if the LDAP SSL Client Authentication is activated by setting the LDAP SSL Client Authentication Required attribute to **TRUE**. (See the LDAP SSL Client Authentication Required attribute for details.)

DirX Directory supports the following choices for the mapping onto the DN representing the Authorization ID. The string values are case sensitive:

- **Certificate.subjectDN:**

This value instructs the DSA to use the DN subjectDN contained in the userCertificate as the Authorization ID directly. No mapping onto an existing entry is performed. After passing the SSL/TLS authentication, an LDAP SASL bind that is configured with **Certificate.subjectDN** cannot fail with the error code **invalid Credentials**. An entry with the subjectDN does not need to exist in the DIT of the DSA. However, the DN subjectDN or groups containing that DN can be used in Access Control Items.

- **Certificate:**

This value instructs the DSA to map onto an existing entry using the userCertificate attribute value with the OID 2.5.4.36. During the bind procedure performed for SASL binds, the DSA performs an internal local search operation that uses the whole certificate as the search filter. The Certificated Exact matching rule applies the certificate matches if the serialNumber and the issuerDN are identical. The userCertificate attribute must be indexed (INITIAL and PRESENT), otherwise the search is rejected and the LDAP SASL bind fails. If the DSA finds exactly one entry in the DIT that owns a certificate with the same value as was used in the LDAP SASL bind, the bind is completed successfully and the Authorization ID is set to the found entry's DN. An LDAP SASL bind that is configured with **Certificate** may fail despite the successful completion of the SSL/TLS authentication. See the paragraph below.



- **Certificate.extensions.altName.email:**

This value instructs the DSA to map onto an existing entry using the value of the subjectAlternateName extension from the user's certificate. During the bind procedure performed for SASL binds, the DSA performs an internal local search operation that uses the value contained in the subjectAlternateName extension with the choice "rfc822 name" of the certificate as the search filter. The attribute type of the search filter is by default the email attribute value with the OID 1.2.840.113549.1.9.1.

The email attribute must be indexed (INITIAL and PRESENT), otherwise the search is rejected and the LDAP SASL bind fails. If the DSA finds exactly one entry in the DIT that owns an email with the same value as in the extension of the certificate that was used in the LDAP SASL bind, the bind is completed successfully and the Authorization ID is set to the found entry's DN.

It is possible to use another attribute of the entry to map onto the Authorization ID by setting the environment variable **DIRX\_MAP\_CERT\_ALTNAME\_ATTR**. The value of this environment variable must be the OID of the attribute to be used in dotted notation. The attribute must have an IA5Str Syntax, its matching rules may be caseIgnore or caseExact. With respect to indexing the same preconditions apply as for the email attribute as described above.

An LDAP SASL bind that is configured with **Certificate.extensions.altName.email** may fail despite the successful completion of the SSL/TLS authentication.

Values different from the three valid values above are accepted by the directory server when changing the LDAP SSL Configuration Subentry. However, LDAP SASL binds performed via the respective LDAP server will fail with **LDAP operation error**. The LDAP message that is returned with the bind error to the LDAP client is **sasl bind: invalid mapping configured: the-invalid-configuration-value**.

The default value of this attribute is **Certificate.subjectDN**.

The LDAP SASL bind may fail in different stages of the operation:

- SSL/TLS client authentication may fail because the cryptographic validation of the authentication token results in an error or because the Certificate issuer is not trusted. (The DirX Directory LDAP server uses the attribute LDAP Trusted CA Certs for the purpose of specifying trusted issuers.) Another SSL/TLS handshake error can occur, if CertificateRevocation List Checking is enabled (the LDAP SSL CRL Checking attribute has the value **1**) and the revocation check of the user's certificate fails. If an SSL/TLS handshake error occurs, the SASL bind fails with a remote abort receive error (Can't connect to the LDAP server).
- An **Invalid Credentials Error** may occur only if the LDAP SASL Authz Id Mapping attribute is set to **Certificate** or **Certificate.extensions.altName.email**:
- The DSA returns an **Invalid Credentials Error** if the mapping fails; that is, if the internal search operation returns **0** or more than one single entry.
- The DSA also returns an **Invalid Credentials Error** if the supplied userCertificate is unsuitable for the configured mapping method; for example, the certificate does not contain an email in the altName extension while the configuration is set to

### **Certificate.extensions.altName.email.**

- An **LDAP operation error** may occur due to various defects:
- The internal search operation is rejected due to the lack of the INITIAL and PRESENT index on the userCertificate or email attribute.
- A runtime error (lack of resources) occurred.
- A data error (for example, decoding of the certificate extensions) occurred.

#### **3.1.15.9.1. Abbreviation**

LSAIM

#### **3.1.15.9.2. LDAP Name(s)**

ldapSaslAuthzIdMapping

#### **3.1.15.9.3. Syntax (DAP)**

Directory String {1024}

#### **3.1.15.9.4. Syntax (LDAP)**

Directory String {1024}

#### **3.1.15.9.5. Example (DAP)**

```
LSAIM=Certificate.subjectDN
```

#### **3.1.15.9.6. Example (LDAP)**

```
ldapSaslAuthzIdMapping=Certificate.subjectDN
```

### **3.1.15.10. LDAP SSL CRL Checking**

The LDAP SSL CRL Checking attribute specifies whether or not the client certificate specified in the context of an LDAP SASL EXTERNAL bind is also checked against a certificate revocation list (CRL). To perform LDAP SASL binds with the mechanism EXTERNAL, the LDAP SSL Client Authentication Required attribute must be **TRUE**.

If the LDAP SSL CRL Checking attribute has the value **1**, CRL checking is performed. If its value is **0**, CRL checking is not performed. Once CRL checking is enabled, at least one CRL file must be specified. (See the attribute LDAP SSL CRL Filenames (ldapSslCrlFilenames)). If no proper CRL file is configured, the server does not start.

The DirX Directory LDAP server uses CRLs stored locally in Privacy Enhanced Mail (PEM)-formatted files that are read during LDAP server process startup or when explicitly required by an LDAP extended operation. (See the attribute LDAP SSL CRL Filenames (ldapSslCrlFilenames) for details.)

If CRL checking is enabled, the following attributes of the LDAP SSL Configuration subentry control the check procedure:

- LDAP SSL CRL Filenames (ldapSslCrIFilenames)
- LDAP SSL Tolerate Missing CRL (ldapSSLTolerateMissingCrl)
- LDAP SSL Allow NYV CRL (ldapSSLAllowNYVcrl)
- LDAP SSL Allow Expired CRL (ldapSSLAllowExpiredCrl)

The CRL check is the last check that the LDAP server performs during the authentication process. Trusting the issuer of the client certificate is the precondition. (See the LDAP Trusted CA Certs (ldapTrustedCACerts) attribute for details.)

The default value of this attribute is **0**; that is, CRL checking is disabled by default.

#### 3.1.15.10.1. Abbreviation

LSCRLC

#### 3.1.15.10.2. LDAP Name(s)

ldapSslCrlChecking

#### 3.1.15.10.3. Syntax (DAP)

Integer

#### 3.1.15.10.4. Syntax (LDAP)

Integer

#### 3.1.15.10.5. Example (DAP)

```
LSCRLC=1
```

#### 3.1.15.10.6. Example (LDAP)

```
ldapSslCrlChecking=1
```

### 3.1.15.11. LDAP SSL CRL Filenames

The LDAP SSL CRL Filenames attribute specifies the name(s) of the files containing the PEM-formatted certificate revocation lists (CRLs) that are to be used to check the certificate for revocation in the context of an LDAP SASL bind with the mechanism type **EXTERNAL**. DirX Directory does not support delta CRLs.

This attribute is multi-valued. Each value contains the name of a file that contains one CRL in PEM Privacy Enhanced Mail (PEM) format. A PEM formatted CRL consists of the base64

encoded binary CRL enclosed in the lines

```
-----BEGIN X509 CRL-----
```

```
-----END X509 CRL-----
```

To convert binary CRLs to PEM formatted CRLs you can use the following **openssl** command:

```
openssl crl -inform DER -in certlist.crl -outform PEM -out crl.pem.
```

Incorrect file content results in a fatal error at LDAP server startup.

Specify the filename values either as fully-qualified pathnames or as filenames without any path. If a filename without any path is specified, the LDAP server expects the file to reside in the directory *install\_path/ldap/conf*.

By default, the value of this attribute is absent.

As the attribute is multi-valued, more than one CRL can be loaded for certificate verification. Each CRL usually refers to a different certification authority (CA), but you can specify more than one CRL file for the same CA with different names. For example, FileA-CRL is an older version of a FileB-CRL. If both names are added to the list, the general rule is that the most recently loaded CRL overwrites any existing version. The sequence of the names cannot be predicted when they are read from the subentry.

The size of a specified CRL has a significant impact on the LDAP server start-time and the time to generate the internal SSL\_CTX into which the CRL is stored; for example, a CRL of 30 MB lasts up to 1 minute to be loaded.

If multiple LDAP servers refer to the same CRL file, the behavior during server start-up or CRL refresh is undefined. To avoid conflicts, use separate CRL names for each server; that is, there may be several copies of the same CRL, one copy for each LDAP server.

To update the CRLs for an LDAP server during runtime, use the LDAP extended operation **ldal\_ssl\_ctx\_update**. Use the **dirxextop** command or DirX Directory Manager (**Monitoring View -> LDAP -> SSL -> Create New Context (CRL refresh)** or **Configuration View -> LDAP Configuration Subentries -> LDAP SSL Configuration Subentry -> Client Authentication -> Refresh CRLs**) to perform this LDAP extended operation. (See the **dirxextop** reference page in the *DirX Directory Administration Reference* for details.) The LDAP server creates a new SSL context, re-reads the values of the LDAP SSL CRL Filenames attribute and re-loads the CRLs. While this dynamic CRL update is in progress, no LDAP bind operation is blocked or postponed. SSL SASL binds are checked against the old CRLs. Once the new CRLs are fully loaded, SSL SASL binds are checked against the new CRLs.

#### 3.1.15.11.1. Abbreviation

LSCRLF

#### 3.1.15.11.2. LDAP Name(s)

ldapSslCrlFileNames

#### 3.1.15.11.3. Syntax (DAP)

Directory String {1024}

#### 3.1.15.11.4. Syntax (LDAP)

Directory String {1024}

#### 3.1.15.11.5. Example (DAP)

*(on Linux systems)*

```
LSCRLF=/home/dirx/ldap/conf/ssl/mycrl.pem
```

*(on Windows systems)*

```
{LSPPF=C:/Program Files/DirX/Directory/ldap/conf/ssl/mycrl.pem}
```

#### 3.1.15.11.6. Example (LDAP)

```
ldapSslCrlFileNames=mycrl.pem
```

This file must be located under `install_path/ldap/conf`.

### 3.1.15.12. LDAP SSL Tolerate Missing CRL

The LDAP SSL Tolerate Missing CRL attribute controls whether or not a client certificate presented in the context of an LDAP SASL bind with the EXTERNAL mechanism is rejected if a suitable CRL cannot be found in the given CRL list.

When a client certificate arrives, the server tries to find a matching CRL; that is, the issuer of the CRL and the issuer of the certificate must match. If a suitable CRL is found, this attribute has no further meaning. If no suitable CRL is found, this attribute controls whether the server rejects or accepts a certificate during the SSL handshake.

If this attribute has the value **1**, certificates are accepted even if a matching CRL cannot be found. If the value is **0**, certificates are rejected if a matching CRL cannot be found.

The default value of this attribute is **0**.

#### 3.1.15.12.1. Abbreviation

LSTMCR

#### 3.1.15.12.2. LDAP Name(s)

ldapSslTolerateMissingCrl

#### 3.1.15.12.3. Syntax (DAP)

Integer

#### 3.1.15.12.4. Syntax (LDAP)

Integer

#### 3.1.15.12.5. Example (DAP)

```
LSTMCRL=1
```

#### 3.1.15.12.6. Example (LDAP)

```
ldapSslTolerateMissingCrl=1
```

### 3.1.15.13. LDAP SSL Allow NYV CRL

The LDAP SSL Allow NYV CRL attribute controls whether or not a client certificate presented in the context of an LDAP SASL bind with the mechanism EXTERNAL is rejected if the corresponding CRL is not yet valid.

A CRL usually contains the lastUpdateTime field which specifies the issuing date. This issuing date can be a date in the future. During certificate checking, OpenSSL can generate a Crl-Not-Yet-Valid error if the CRL's lastUpdateTime is not yet reached. The LDAP SSL Allow NYV CRL attribute specifies whether or not this error is ignored.

If this attribute has the value **1**, certificates are accepted even if the CRL is not yet valid. If the value is **0**, certificates are rejected if the CRL is not yet valid.

The default value of this attribute is **0**.

#### 3.1.15.13.1. Abbreviation

LSANCRL

#### 3.1.15.13.2. LDAP Name(s)

ldapSslAllowNYVCrl

#### 3.1.15.13.3. Syntax (DAP)

Integer

#### 3.1.15.13.4. Syntax (LDAP)

Integer

#### 3.1.15.13.5. Example (DAP)

```
LSANCRL=1
```

#### 3.1.15.13.6. Example (LDAP)

```
ldapSslAllowNYVCrl=1
```

### 3.1.15.14. LDAP SSL Allow Expired CRL

The LDAP SSL Allow Expired CRL attribute controls whether or not a client certificate presented in the context of an LDAP SASL bind with the mechanism EXTERNAL is rejected if the corresponding CRL has expired.

A CRL contains a field `nextUpdateTime` that specifies the date when the CA issues an updated version of the current CRL. If the `nextUpdateTime` is in the past, the OpenSSL provider generates a CRL-Expired error during Client-Certificate-Validation. The attribute `ldapSSLAllowExpiredCrl` controls whether or not this error is ignored.

If this attribute has the value **1**, certificates are accepted even if the CRL has expired. If the value is **0**, certificates are rejected if the CRL has expired. If the value is **0**, the CRL must be updated before its expiration date. Otherwise, all certificates issued by the corresponding CA are rejected after the CRL has expired.

The default value of this attribute is **0**.

#### 3.1.15.14.1. Abbreviation

LSAECRL

#### 3.1.15.14.2. LDAP Name(s)

`ldapSslAllowExpiredCrl`

#### 3.1.15.14.3. Syntax (DAP)

Integer

#### 3.1.15.14.4. Syntax (LDAP)

Integer

#### 3.1.15.14.5. Example (DAP)

```
LSAECRL=1
```

#### 3.1.15.14.6. Example (LDAP)

```
ldapSslAllowExpiredCrl=1
```

#### 3.1.15.15. Attributes of the LDAP SSL Configuration Subentry for the LDAP Server Personal Security Environment (PSE)

To establish an LDAP connection over the SSL/TLS layer successfully, the DirX Directory LDAP server must have a Personal Security Environment (PSE). This PSE comprises the following items:

- The LDAP server's own key material, which consists of the public key certificate and the private key. This key material is required for every kind of SSL connection. It is used in the context of the SSL/TLS handshake protocol for the purpose of server authentication. The key and the certificate must be enclosed in a password-protected PEM file. Usually, the LDAP client application must store the server public key certificate in a local certificate database because it might check the acceptability of the server authentication (for example for Java-based clients, the certificate must be imported into the key store of the application).

The LDAP server's own public key certificate and private key is stored in the LDAP Own Keymaterial PEM attribute.

- The password that is used to protect the private key in the PEM file.

The full pathname to the file that contains this password is stored in the LDAP PKCS12 Password File attribute.

- The list of Certification Authority (CA) certificates the LDAP server trusts to issue user certificates, if SSL/TLS client authentication is required (see the section **LDAP SSL Client Authentication Required** for details). The LDAP server uses the list of CA certificates to decide whether a trusted CA issued the certificate provided by the user during the SSL/TLS handshake protocol.

The list of CA certificates is stored in the multi-valued LDAP Trusted CA Certs attribute.

The following sections provide information about the attributes of the LDAP server PSE.

##### 3.1.15.15.1. LDAP Own Keymaterial PEM

The LDAP Own Keymaterial PEM attribute stores the LDAP server's PEM-formatted key material. The PEM input must contain:

1. The LDAP server's the encrypted private key
2. The LDAP server's public key certificate
3. Optionally the CA certificates of the issuing CAs



in the order specified above.

The decryption password is stored in a separate file. (See the **LDAP PKCS12 Password File** reference page for details.)

The DirX Directory installation copies an example key material file at the location *install\_path/ldap/conf/cert\_ldapserver.pem*.

Because certificates have a limited lifetime, they may expire and lose their validity. The DirX Directory LDAP server checks the remaining validity period of its public key certificate. If this period is less than 30 days, the LDAP server sends the dirxCertExpiry SNMP trap and issues a warning, and the LDAP server exception file contains the following entry:

SEC\_error: "WARNING: Your Server-SSL key material will expire soon!! Please refresh!

The LDAP server checks the expiration date during every SSL handshake (SSL-Bind). The environment variable **DIRX\_LDAP\_SSL\_EXPIRY\_WARN\_TIME** controls expiration alerting.

#### 3.1.15.15.2. Abbreviation

LSOKP

#### 3.1.15.15.3. LDAP Name(s)

ldapOwnKeymaterialPEM

#### 3.1.15.15.4. Syntax (DAP)

Octet String

#### 3.1.15.15.5. Example (DAP)

In the following example, the PEM-formatted content of the file **cert\_ldapserver.pem** is stored as the LSOKP attribute value of the LDAP SSL configuration subentry using the **obj modify** operation of the **dirxcp** command:

```
dirxcp> modify /o=my-company/cn=ldapsslConfiguration -add  
LSOKP_FILE=cert_ldapserver.pem
```

#### 3.1.15.15.6. Example (LDAP)

In the following example, the PEM-formatted content of the file **cert\_ldapserver.pem** is stored as the **ldapOwnKeymaterialPEM** attribute value of the LDAP SSL configuration subentry using the **obj modify** operation of the **dirxcp** command:

```
dirxcp> modify cn=ldapsslConfiguration,o=my-company -add  
ldapOwnKeymaterialPEM_FILE=cert_ldapserver.pem
```

#### 3.1.15.15.7. LDAP Trusted CA Certs

The multi-valued LDAP Trusted CA Certs attribute stores CA Certificates of those authorities that the LDAP server trusts to issue valid user certificates. The values of this attribute are plain binary X.509 CA Certificates; usually the content of a **.cer/.der** file. (For information how to store the file content as attribute values, see the section **Attribute Values in a File** in the chapter **DirX Directory String Representation for DAP Binds**.)

The LDAP Trusted CA Certs attribute is only used when the LDAP server supports client authentication. (See the section **LDAP SSL Client Authentication Required** for details.) There is no default value for this attribute.

For information about SSL/TLS client authentication and its relationship to LDAPv3, see the recommendations entitled *Authentication Methods and Security Mechanisms (RFC 4513)*.

#### 3.1.15.15.8. Abbreviation

LSTCC

#### 3.1.15.15.9. LDAP Name(s)

ldapTrustedCACerts

#### 3.1.15.15.10. Syntax (DAP)

Certificate

#### 3.1.15.15.11. Example (DAP)

In the following example, the content of the file **cacert1.der** is stored as LSTCC attribute values of the LDAP SSL configuration subentry using the **obj modify** operation of the **dirxcp** command:

```
dirxcp> modify /o=my-company/cn=ldapsslConfiguration -add  
LSTCC_ANY_FILE=cacert1.der
```

#### 3.1.15.15.12. Example (LDAP)

In the following example, the content of the file **cacert1.der** is stored as ldapTrustedCACerts attribute values of the LDAP SSL configuration subentry using the **obj modify** operation of the **dirxcp** command:

```
dirxcp> modify cn=ldapsslConfiguration,o=my-company -add  
ldapTrustedCACerts_ANY_FILE=cacert1.der
```

#### 3.1.15.15.13. LDAP PKCS12 Password File

The LDAP PKCS12 Password File attribute specifies the full pathname of the file that contains the password for access to the private key of the LDAP server. This key is stored in

the LDAP Own KeymaterialPEM attribute (in older DirX Directory versions, the key is stored in the LDAP Own Keymaterial attribute).The password is used during the process of creating the PEM file, for example, by using the **openssl** command line tool.

The format of the value specified in the password file is indicated by an optional tag that is also contained in this file.After the initial administration, it is the clear text password.Once the LDAP server has accessed the file one time, it changes the content to the encrypted format of the password.Note that the clear text password must not be longer than 160 characters and must not start with the string "{ENC}".

It is strongly recommended to protect the password file by means of the operating system so that access is only granted to the LDAP server process.

The default value of this attribute is *install\_path/ldap/conf/dirx\_pkcs12.pwd*. It is strongly recommended to use this default.

#### 3.1.15.15.14. Abbreviation

LSPPF

#### 3.1.15.15.15. LDAP Name(s)

Pkcs12PasswordFile

#### 3.1.15.15.16. Syntax (DAP)

Directory String {1024}

#### 3.1.15.15.17. Example (DAP)

*(on Linux systems)*

```
LSPPF=/opt/dirx/ldap/conf/ssl/dirx_pkcs12.pwd
```

*(on Windows systems)*

```
{LSPPF=C:/Program Files/DirX/Directory/ldap/conf/ssl/dirx_pkcs12.pwd}
```

#### 3.1.15.15.18. Example (LDAP)

*(on Linux systems)*

```
Pkcs12PasswordFile=/opt/dirx/ldap/conf/ssl/dirx_pkcs12.pwd
```

*(on Windows systems)*

```
{Pkcs12PasswordFile=C:/Program  
Files/DirX/Directory/ldap/conf/ssl/dirx_pkcs12.pwd} +
```

Note that the forward slash (/) and the backslash (\) characters can be used as separators for Linux and Windows pathnames because the values are interpreted only by the **dirxldapv3** program. Also note that curly braces must enclose a Windows pathname when it contains space characters, as shown in the example above.

### 3.1.15.16. Obsolete Attributes of the LDAP SSL Configuration Subentry for the LDAP Server PSE

This section provides information about the following attributes:

- LDAP Own Keymaterial
- LDAP Own Keymaterial File
- LDAP Valid Root CAs File

These attributes are obsolete and are supported only for compatibility reasons. In a shadowing environment, DirX Directory V8.2B servers may store configuration subentries used by directory server instances running older DirX Directory versions.

It is not allowed to use these attributes to store key material for DirX Directory V8.2B LDAP servers and newer. The LDAP Own KeymaterialPEM and LDAP Trusted CA Certs attributes must be used to store the own key material and the trusted CA certs. If the LDAP SSL configuration subentry contains the attributes LDAP Own KeymaterialPEM and LDAP Trusted CA Certs, the LDAP server ignores the obsolete attributes LDAP Own Keymaterial, LDAP Own Keymaterial File and LDAP Valid Root CAs File.

## 3.1.16. Attributes for LDAP Server Audit Configuration

Attributes for LDAP server audit configuration are attributes of the LDAP server audit configuration subentry with the default name **/CN=ldapAudit**. (This name is referenced in the LDAP Audit Configuration Subentry CN (LACCN) attribute of the LDAP Server Configuration subentry.) All attributes are optional. Run the script **ldap\_auditcfg.cp** (using **dirxcp**) to create a default LDAP audit configuration subentry that specifies default values for the DirX Directory LDAP server. Use the **dirxcp obj** operations to modify the attributes of the default LDAP audit configuration subentry or to create new LDAP audit configuration subentries. Like all other subentries, it cannot be created via LDAP. However, it can be modified via LDAP. Modifications of an LDAP audit configuration subentry become effective after a restart of the LDAP server or by performing the **dirxadm ldap audit -config** operation.

### 3.1.16.1. LDAP Audit On

The LDAP Audit On single-valued attribute specifies whether LDAP server auditing is enabled (value **TRUE**) at start-up time of the LDAP server or disabled (value **FALSE**). This attribute is also evaluated when you perform re-configuration via the **dirxadm ldap audit -config** operation. The default value is **FALSE**.

If LDAP auditing is enabled, you can use the LDAP Audit Start Time (LASTA) and the LDAP Audit Stop Time (LASTO) attributes to specify a time period when the LDAP server records audit information. If no start time (LASTA attribute) is specified, the LDAP server starts recording of audit information at LDAP server start-up or at LDAP server re-configuration

by performing the **dirxadm ldap audit –config** operation.

LDAP auditing can be enabled dynamically by performing the **dirxadm ldap audit –start** operation. LDAP auditing can be disabled dynamically by performing the **dirxadm ldap audit –stop** operation. The current LDAP audit status can be displayed by performing the **dirxadm ldap audit –info** operation.

#### 3.1.16.1.1. Abbreviation

LAON

#### 3.1.16.1.2. LDAP Name(s)

ldapAuditOn

#### 3.1.16.1.3. Syntax (DAP)

Boolean

#### 3.1.16.1.4. Syntax (LDAP)

Boolean (LDAP Style)

#### 3.1.16.1.5. Example (DAP)

```
LAON=TRUE LDAP auditing enabled at startup or after re-configuration
LAON=FALSE LDAP auditing disabled at startup or after re-configuration
```

#### 3.1.16.1.6. Example (LDAP)

```
ldapAuditOn=TRUE LDAP auditing enabled at startup or after re-configuration
ldapAuditOn=FALSE LDAP auditing disabled at startup or after re-configuration
```

### 3.1.16.2. LDAP Audit Destination

The LDAP Audit Destination single-valued attribute specifies the initial pathname of the directory in which the LDAP server's audit log file with the name **audit.log** is located. Each LDAP server opens its audit log file exclusively. Therefore, the name of the LDAP server's configuration subentry is appended as the subdirectory name. This directory must exist before LDAP server auditing is enabled for the first time. Otherwise the LDAP server exits.

The full pathname of the LDAP server's audit log file is in the format:

*LDAP\_Audit\_Destination*\*/**LDAP\_Configuration\_Subentry\_Name**/audit.log\*

where *LDAP\_Audit\_Destination* is the value of the LDAP Audit Destination attribute, *LDAP\_Configuration\_Subentry\_Name* is the name of the LDAP server's current configuration subentry, and **audit.log** is the name of the LDAP server's audit log file. (See the section **Setting Up Multiple LDAP Servers** of chapter **Setting Up an LDAP Server** in the *Administration Guide* for details on multiple LDAP servers.)

The default value of the LDAP Audit Destination attribute is *install\_path*\*/ldap/audit\*. The default LDAP server's configuration subentry name is **ldapConfiguration**. The default full pathname of the audit log file is *install\_path*/ldap/audit/ldapConfiguration/audit.log.

#### 3.1.16.2.1. Abbreviation

LADE

#### 3.1.16.2.2. LDAP Name(s)

ldapAuditDestination

#### 3.1.16.2.3. Syntax (DAP)

Directory String {32768}

#### 3.1.16.2.4. Example (DAP)

(on Linux systems)

```
LADE=\/opt\/dirx\/ldap\/log\/audit
```

Assuming that the default LDAP configuration subentry is used the full pathname of the audit log file is **/opt/dirx/ldap/log/audit/ldapConfiguration/audit.log**.

(on Windows systems)

```
LADE=C:\\Program Files\\DirX\\Directory\\ldap\\log\\audit
```

Assuming that the default LDAP configuration subentry is used, the full pathname of the audit log file is **C:\\Program Files\\DirX\\Directory\\ldap\\log\\audit\\ldapConfiguration\\audit.log**.



The slash (/) and the backslash (\\) character must be escaped by a backslash (\\) character.

#### 3.1.16.2.5. Example (LDAP)

(on Linux systems)

```
ldapAuditDestination=\/opt\/dirx\/ldap\/log\/audit
```

Assuming that the default LDAP configuration subentry is used, the full pathname of the audit log file is `/opt/dirx/ldap/log/audit/ldapConfiguration/audit.log`.

(on Windows systems)

```
ldapAuditDestination=C:\\Program  
Files\\DirX\\Directory\\ldap\\log\\audit
```

Assuming that the default LDAP configuration subentry is used, the full pathname of the audit log file is `C:\\Program Files\\DirX\\Directory\\ldap\\log\\audit\\ldapConfiguration\\audit.log`.



The slash (/) and the backslash (\\) character must be escaped by a backslash (\\) character.

### 3.1.16.3. LDAP Audit Size Limit

The LDAP Audit Size Limit single valued attribute specifies the maximum number of audit records of the LDAP server's audit log file. Every LDAP operation creates one audit record. The value of **0** allows files with an unlimited number of records. The default value is **50000**. The maximum number of audit records must be greater than 100. Otherwise the default value is used.

The LDAP operations logged can be specified in the attribute LDAP OP Selection. The size of one record in bytes depends on the data and the configuration settings specified in the attributes LDAP Audit Level and LDAP Audit Value Limit.

Independent of the setting of the LDAP Audit Size Limit attribute, the size of the LDAP audit log file is limited to 256 MB by default because the **dirxauddecode** command must be used to evaluate the binary audit log files. Due to the 32-bit I/O mechanism, **dirxauddecode** is not able to handle files greater than 2GB, neither as input nor as output. **dirxauddecode** may run into a deadlock. The administrator can specify another maximum file size than 256 MB in the LDAP Audit Max File Size attribute (See section below for details.)

If either the LDAP Audit Size Limit (maximum number of audit records) or the LDAP Audit Max File Size (maximum size of the audit file) is exceeded, the LDAP Audit Overflow attribute specifies the action to take. (See the section **LDAP Audit Overflow** in this chapter for details.)

#### 3.1.16.3.1. Abbreviation

LASL

#### 3.1.16.3.2. LDAP Name(s)

ldapAuditSizeLimit

#### 3.1.16.3.3. Syntax (DAP)

Integer

#### 3.1.16.3.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.16.3.5. Example (DAP)

```
LASL=500
```

#### 3.1.16.3.6. Example (LDAP)

```
ldapAuditSizeLimit=500
```

### 3.1.16.4. LDAP Audit Max File Size

The LDAP Audit Max File Size single-valued attribute specifies the maximum file size of the LDAP server's audit log file in MB. A value of **0** specifies an unlimited file size. Note that this behavior may lead to errors due to the 32-bit I/O mechanism when files exceed 2 GB. The default file size is **256** MB.

Additionally to the LDAP Audit Max File Size attribute, the maximum number of records specified in the LDAP Audit Size Limit attribute also limits the size of the LDAP server's audit log file. (See section above for details.)

If either the LDAP Audit Max File Size or the LDAP Audit Size Limit is exceeded, the LDAP Audit Overflow attribute specifies the action to take. (See section **LDAP Audit Overflow** of this chapter for details.)

It is recommended not to specify too small limits because wrapping or moving the audit log file is rather expensive compared to writing an audit record.

The **dirxauddecode** command evaluates the binary LDAP audit log file and generates an ASCII output file. Due to 32-bit code and internal data representation, the **dirxauddecode** command cannot handle files greater than 2GB. Unfortunately, it is not possible to determine the size of the generated ASCII output file from the binary audit log file. However, the generated ASCII output file is usually five to ten times greater than the binary LDAP audit log file. Therefore, it is necessary to limit the maximum file size.

#### 3.1.16.4.1. Abbreviation

LAMFS

#### 3.1.16.4.2. LDAP Name(s)

ldapAuditMaxFileSize



#### 3.1.16.4.3. Syntax (DAP)

Integer

#### 3.1.16.4.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.16.4.5. Example (DAP)

```
LAMFS=512
```

#### 3.1.16.4.6. Example (LDAP)

```
ldapAuditMaxFileSize=512
```

### 3.1.16.5. LDAP Audit Overflow

The LDAP Audit Overflow single-valued attribute specifies the action to take when the maximum record number specified in the LDAP Audit Size Limit attribute or the file size specified in the LDAP Audit Max File Size attribute is reached. Specify one of the following values:

- **stopService** - Shut down the LDAP server. An exception message is written. The LDAP server is not re-started automatically.
- **stopAudit** - Cease auditing. LDAP server continues without auditing. An exception message is written.
- **wrapAround** - Begin writing at the top of the log file, overwriting the old content.
- **moveFile** - Close the audit file currently in use and rename it by appending the current UTC time in ZULU format and a counter from 000 to 999, for example **audit.log.20010510104511Z\_001**. This renamed file can be evaluated by using the **dirxauddecode** command. A new audit log file is created. Auditing is continued using the current configuration settings.
- **multiWrap** - Begin writing at the top of the next log file, overwriting the old content. The LDAP Audit Number Wraparound Files specifies the number of additional audit log files in use. (See section below for details.)

The default value is **wrapAround**.

#### 3.1.16.5.1. Abbreviation

LAOF

#### 3.1.16.5.2. LDAP Name(s)

ldapAuditOverFlow

#### 3.1.16.5.3. Syntax (DAP)

Directory String {32768}

#### 3.1.16.5.4. Example (DAP)

```
LAOF=moveFile
```

Closes the currently used file when LDAP Audit Size Limit is exceeded, renames the closed file to **audit.log.timestamp** and continues auditing in the newly created file **audit.log**.

#### 3.1.16.5.5. Example (LDAP)

```
ldapAuditOverflow=moveFile
```

Closes the currently used file when LDAP Audit Size Limit is exceeded, renames the closed file to **audit.log.timestamp** and continues auditing in the newly created file **audit.log**.

#### 3.1.16.6. LDAP Audit Number Wraparound Files

The LDAP Audit Number Wraparound Files single-valued attribute specifies the number of additional audit log files in use when the LDAP Audit Overflow attribute specifies the strategy **multiWrap**. Specify a value greater or equal to **1**. Do not specify a value greater than 99. The default value is **1** (one additional audit log file is used; that is, there are two audit log files in use: **audit.log** for current logging and **audit.PID.01** for wrapping around).

If auditing is enabled, the LDAP server writes audit file records in the file **audit.log**. (See the section **LDAP Audit Destination** for details.) If the maximum number of audit file records (LDAP Audit Size Limit attribute) or the maximum audit file size (LDAP Audit Max File Size attribute) is exceeded, the LDAP server closes this file and renames it to **audit.log.\*PID.number**, where **PID** is the process identifier of the LDAP server and **number** is a sequential number running from **\*01** through the value specified in LDAP Audit Number Wraparound Files attribute. In the event that a file with this name already exists, the LDAP server first deletes this file and then creates a new audit log file **audit.log**. Auditing is continued using the current configuration settings. The result is that there are LDAP Audit Number Wraparound Files audit files and the file **audit.log** in use.

While the LDAP server is running and the set of audit log files is in use, you must ensure that the LDAP server is able to access all these files. For example, do not move, lock, rename, or delete any of these files.

If the LDAP server is restarted for any reason and gets a previously used process identifier, it overwrites already existing files from the previous process.

Auditing is disabled automatically if any of the file operations fails.

#### 3.1.16.6.1. Abbreviation

LANWF

#### 3.1.16.6.2. LDAP Name(s)

ldapAuditNumWrapFiles

#### 3.1.16.6.3. Syntax (DAP)

Integer

#### 3.1.16.6.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.16.6.5. Example (DAP)

```
LANWF=9
```

#### 3.1.16.6.6. Example (LDAP)

Currently the LDAP server uses multi-wrapping with 4 additional log files. That is, the values currently in use are:

```
ldapAuditOverflow=multiWrap  
ldapAuditNumWrapFiles=4 (create 4 additional audit log files)
```

Now the administrator decides to use ten audit log files because he wants to evaluate a longer time period. So he specifies the value 9 for the additional number of audit log files:

```
ldapAuditNumWrapFiles=9 (create 9 additional audit log files)
```

The administrator restarts the LDAP server to enable the new audit configuration and to get process-specific audit log files. The LDAP server gets the PID 1308. It finds the audit log file **audit.log** from the just stopped LDAP server process with the PID 1352. First, the LDAP server renames this file to **audit.log.20060605091435Z\_001** and then creates a new **audit.log** file. When **audit.log** exceeds one of the limits, the LDAP server renames the file to **audit.log.1308.01** and creates a new file **audit.log**. After generating **audit.log.1308.09**, the LDAP server starts wrapping around with **audit.log.1308.01**. It deletes this file before renaming **audit.log**. The administrator finds the following audit log files:

- From the current running LDAP server with PID 1308:

```
audit.log           (currently active audit log file)  
audit.log.1308.01   (multi wrapping file 1)
```

|                   |                         |
|-------------------|-------------------------|
| audit.log.1308.02 | (multi wrapping file 2) |
| audit.log.1308.03 | (multi wrapping file 3) |
| audit.log.1308.04 | (multi wrapping file 4) |
| audit.log.1308.05 | (multi wrapping file 5) |
| audit.log.1308.06 | (multi wrapping file 6) |
| audit.log.1308.07 | (multi wrapping file 7) |
| audit.log.1308.08 | (multi wrapping file 8) |
| audit.log.1308.09 | (multi wrapping file 9) |

Because the LDAP server with the PID 1308 is running, the administrator must ensure write access for the LDAP server. It is prohibited to delete, lock, or rename these files. However, the administrator can copy them to another location for evaluating the files.

- From the previously running LDAP server with PID 1352:

|                               |                                     |
|-------------------------------|-------------------------------------|
| audit.log.20060605091435Z_001 | (last recent active audit log file) |
| audit.log.1352.01             | (multi wrapping file 1)             |
| audit.log.1352.02             | (multi wrapping file 2)             |
| audit.log.1352.03             | (multi wrapping file 3)             |
| audit.log.1352.04             | (multi wrapping file 4)             |

Because the LDAP server with the PID 1352 is not running the administrator can delete or move these files.

The creation dates of the files provide a hint about the chronological sequence.

### 3.1.16.7. LDAP Audit Level

The LDAP Audit Level single-valued attribute specifies the level of detail of audit information logged for attribute values of the request parameters of the operations add, modify, compare, and of the LDAPv3 control parameters. This attribute is useful to limit the size of the audit information. Specify one of the following values:

- **max** - The object name, all attribute types, and the attribute values are logged: If the value length exceeds the length specified in the LDAP Audit Value Limit attribute, only the length of the value is logged. If the value contains unprintable characters, **dirxauddecode** delivers a hex dump format of the value.
- **med** - The object name, all attribute types, and the length of the attribute values are logged.
- **min** - Only the object name is logged. No attribute information is logged.

The default value is **med**.

#### 3.1.16.7.1. Abbreviation

LALE

#### 3.1.16.7.2. LDAP Name(s)

ldapAuditLevel

#### 3.1.16.7.3. Syntax (DAP)

Directory String {32'768}

#### 3.1.16.7.4. Example (DAP)

1. The following sample output was generated by the **dirxauddecode** command and

LALE=max

**dirxauddecode** sample output:

```
----- OPERATION 000002 -----
Start Time      :Thu May 10 10:47:21 2001
Duration        :0.040 sec
User           :
IP              :127.0.0.1
Op-Name         :LDAP_Con0_Op1
ConnID          :0
Operation       :ADD
Version         :3
MessageID       :2
Entry           :cn=dummy,o=my-company
Attributes      :2
  Attr Type     :description
  Attr Val      :thisisalongdescription
  Attr Type     :cn
  Attr Val      :dummy2
Result Code     :65 (objectClassViolation)

----- OPERATION 000002 -----
Start Time      :Mon May 14 13:29:16 2001
Duration        :0.040 sec
User           :
IP              :127.0.0.1
Op-Name         :LDAP_Con0_Op1
ConnID          :0
```

```

Operation      :ADD
Version        :2
MessageID      :2
Entry          :cn=dummy,o=my-company
Attributes      :2
  Attr Type     :description
  Attr Val(hex):(26)
  000000: 61 64 65 73 63 72 69 70 74 69 6f 6e 0a 0d 77 69
:adescription.. wi <- binary
  000010: 74 68 0a 0d 62 69 6e 61 72 79
:th..binary
  Attr Type     :cn
  Attr Val      :dummy2
Result Code     :65 (objectClassViolation)

```

2. The following sample output was generated by the **dirxauddecode** command and LALE=med

**dirxauddecode** sample output:

```

----- OPERATION 000002 -----
Start Time      :Thu May 10 11:18:55 2001
Duration        :0.040 sec
User           :
IP             :127.0.0.1
Op-Name        :LDAP_Con0_Op1
ConnID         :0
Operation      :ADD
Version        :2
MessageID      :2
Entry          :cn=dummy,o=my-company
Attributes      :2
  Attr Type     :description
  Attr Val(len):22      <-- only the length is evaluated
  Attr Type     :cn
  Attr Val(len):6       <-- only the length is evaluated
Result Code     :65 (objectClassViolation)

```

3. The following sample output was generated by the **dirxauddecode** command and LALE=min

**dirxauddecode** sample output:

```
----- OPERATION 000002 -----
Start Time      :Thu May 10 11:27:57 2001
Duration        :0.040 sec
User            :
IP              :127.0.0.1
Op-Name         :LDAP_Con0_Op1
ConnID          :0
Operation       :ADD
Version         :2
MessageID       :2
Entry           :cn=dummy,o=my-company    <- no attribute
information
Result Code     :65 (objectClassViolation)
```

#### 3.1.16.7.5. Example (LDAP)

```
ldapAuditLevel=min
```

#### 3.1.16.8. LDAP Audit Value Limit

The LDAP Audit Value Limit single-valued attribute specifies the maximum length of an attribute value that is logged. If the length of the attribute value exceeds this limit, only the value's length is logged. (Like **LDAP Audit Level med**; see the section **LDAP Audit Level** for details.) The value of **0** allows the logging of values with a length of an unlimited number of bytes. The default value is **256**.

Note that this attribute can greatly affect the audit log file size if clients perform modifications with large attribute values; for example, pictures.

This attribute is relevant only if the LDAP Audit Level attribute is set to **max**.

##### 3.1.16.8.1. Abbreviation

LAVL

##### 3.1.16.8.2. LDAP Name(s)

ldapAuditValueLimit

##### 3.1.16.8.3. Syntax (DAP)

Integer

#### 3.1.16.8.4. Syntax (LDAP)

Integer String (LDAP Style9)

#### 3.1.16.8.5. Example (DAP)

```
LAVL=50
```

If the length of the attribute value is lower than or equal to 50 bytes, the value is logged; else only the value length is logged.

#### 3.1.16.8.6. Example (LDAP)

```
ldapAuditValueLimit=50
```

### 3.1.16.9. LDAP Audit Op Selection

The LDAP Audit Op Selection multi-valued attribute specifies a list of operations that are logged. Specify the value in the following format:

*operation*[:*\_operation\_*...]

where *operation* is one of the following keywords:

- **all** - All operations
- **ldap** - All LDAP server operations except 'other' and 'unknown' operations.
- **abandon** - All abandon operations
- **add** - All add operations
- **bind** - All bind operations
- **compare** - All compare operations
- **extop** - All LDAPv3 extended operations; for example, unsolicited notification or startTLS
- **moddn** - All moddn operations
- **modify** - All modify operations
- **other** - All other unexpected client operations that cannot be identified
- **remove** - All delete operations
- **search** - All search operations
- **unbind** - All unbind operations
- **unknown** - All unexpected client operations that indicate client misbehavior, for example closing the socket layer without initiating an LDAP unbind operation or clients that send illegal PDUs etc.

The default value is **all**.



Note that this attribute provides an operation filter for auditing in the LDAP server at the time of writing audit records. At the time of evaluation, the **dirxauddecode** command also provides the facility to filter out operation types.

#### 3.1.16.9.1. Abbreviation

LAOS

#### 3.1.16.9.2. LDAP Name(s)

ldapAuditOpSelection

#### 3.1.16.9.3. Syntax (DAP)

Directory String {40}

#### 3.1.16.9.4. Example (DAP)

```
LAOS=ldap
```

Logs all LDAP operations except the other and unknown operations.

```
{LAOS=bind;add;search}
```

Logs all bind, add, and search operations.

#### 3.1.16.9.5. Example (LDAP)

```
ldapAuditOpSelection=ldap
```

Logs all LDAP operations except the other and unknown operations.

```
{ldapAuditOpSelection=bind;add;search}
```

Logs all bind, add, and search operations.

### 3.1.16.10. LDAP Audit Erroneous Operations

The LDAP Audit Erroneous Operations single-valued attribute specifies whether or not erroneous operations are recorded in the audit log file.

Specify one of the following values:

- 1** - The LDAP server records erroneous operations in the audit log file.
- 0** - The LDAP server does not record erroneous operations in the audit log file.

The default value is **1**.

If the administrator wants only erroneous operations to be recorded in the audit log file, he specifies the value **1** for the LDAP Audit Erroneous Operations attribute and the value **none** for the LDAP Audit Op Selection attribute.

#### 3.1.16.10.1. Abbreviation

LAERR

#### 3.1.16.10.2. LDAP Name(s)

ldapAuditErrors

#### 3.1.16.10.3. Syntax (DAP)

Integer

#### 3.1.16.10.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.16.10.5. Example (DAP)

```
LAERR=0
```

#### 3.1.16.10.6. Example (LDAP)

```
ldapAuditErrors=0
```

### 3.1.16.11. LDAP Audit Internal Buffer Size

The LDAP Audit Internal Buffer Size single-valued attribute specifies whether audit information is buffered internally before it is written to the audit log file. A value of **0** specifies that all audit information is written to the audit log file when an operation was completed. A value greater than 0 specifies the buffer size in number of bytes. The audit information is written to the audit log file when the buffer size is exceeded.

The default value is **0** (no buffering of audit information).

#### 3.1.16.11.1. Abbreviation

LAIBS

#### 3.1.16.11.2. LDAP Name(s)

ldapAuditIntBufSize

#### 3.1.16.11.3. Syntax (DAP)

Integer

#### 3.1.16.11.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.16.11.5. Example (DAP)

```
LAIBS=1024
```

All internal buffered audit records are written to the audit log file when the internal buffer size of 1024 bytes is exceeded.

#### 3.1.16.11.6. Example (LDAP)

```
ldapAuditIntBufSize=1024
```

All internal buffered audit records are written to the audit log file when the internal buffer size of 1024 bytes is exceeded.

### 3.1.16.12. LDAP Audit Encryption

The LDAP Audit Encryption single valued attribute specifies whether or not the audit log file shall be encrypted. The value of this attribute specifies the encryption mechanism. Specify one of the following values:

- **none** - The audit log file is not encrypted.
- **scramble** - The audit log file is encrypted using a scramble mechanism.

The default value is **none** (no encryption).

If scramble encryption mechanism is enabled, the file specified in the LDAP Audit Encryption Key File attribute must contain the correct key material. If the key material is incorrect or the specified file does not exist or cannot be read, the LDAP server terminates.

#### 3.1.16.12.1. Abbreviation

LAE

#### 3.1.16.12.2. LDAP Name(s)

ldapAuditEncryption

#### 3.1.16.12.3. Syntax (DAP)

Directory String {32768}

#### 3.1.16.12.4. Example (DAP)

LAE=scramble The audit log file is encrypted by using a scramble mechanism.

#### 3.1.16.12.5. Example (LDAP)

ldapAuditEncryption=scramble

The audit log file is encrypted by using a scramble mechanism.

### 3.1.16.13. LDAP Audit Encryption Key File

The LDAP Audit Encryption Key File single-valued attribute specifies the full pathname of the file that contains the key material to encrypt the audit information to be saved in the audit log file. This key material may be a password that is used as a cipher key, a pass phrase, or a PIN that is used by Personal Security Environment to access the current encryption key. If the key material is incorrect or the specified file does not exist or cannot be read, the LDAP server terminates.

This attribute is ignored if no encryption mechanism is specified in the LDAP Audit Encryption attribute (LAE=none).

If an encryption mechanism is specified in the LDAP Audit Encryption attribute (LAE≠none) the password stored in this file must be specified in the **-k** option of the **dirxauddecode** command in order to evaluate a scrambled audit file.

The default value is *install\_path\*/ldap/conf/dirx\_audit.pwd\**.

#### 3.1.16.13.1. Abbreviation

LAEKF

#### 3.1.16.13.2. LDAP Name(s)

ldapAuditEncrKeyFile

#### 3.1.16.13.3. Syntax (DAP)

Directory String {32768}

#### 3.1.16.13.4. Example (DAP)

(on Linux systems)

LAEKF=\\opt\\dirx\\ldap\\log\\audit\\key.pwd

(on Windows systems)

```
LAЕКF=C:\\Program Files\\DirX\\Directory\\ldap\\log\\audit\\key.pwd
```



The slash (/) and the backslash (\) character must be escaped by a backslash (\) character.

#### 3.1.16.13.5. Example (LDAP)

(on Linux systems)

```
ldapAuditEncrKeyFile=\/opt\/dirx\/ldap\/log\/audit\/key.pwd
```

(on Windows systems)

```
ldapAuditEncrKeyFile=C:\\Program  
Files\\DirX\\Directory\\ldap\\log\\audit\\key.pwd
```



The slash (/) and the backslash (\) character must be escaped by a backslash (\) character.

#### 3.1.16.14. LDAP Audit Start Time

The LDAP Audit Start Time single-valued attribute specifies the time when the LDAP server starts recording of audit information. A value of **none** specifies no start time and does not affect the audit behavior. Specify the time in the following syntax:

*DD.MM.YYYY hh:mm:ss*

where

*DD* specifies the day of month (**01** up to **31**)

*MM* specifies the month (**01** up to **12**)

*YYYY* specifies the year

*hh* specifies the hour (**00** up to **23**)

*mm* specifies the minute (**00** up to **59**)

*ss* specifies the seconds (**00** up to **59**)

If a start time is specified the LDAP Audit On (LAON) attribute must be set to **TRUE**.

The default value is **none** (no start time specified).

#### 3.1.16.14.1. Abbreviation

LASTA

#### 3.1.16.14.2. LDAP Name(s)

ldapAuditStartTime

#### 3.1.16.14.3. Syntax (DAP)

Directory String {32'68}

#### 3.1.16.14.4. Example (DAP)

```
{LASTA=03.05.2001 13:05:30}
```

LDAP server auditing is started on May 3<sup>rd</sup>, 2001 at 1:05:30 PM.

#### 3.1.16.14.5. Example (LDAP)

```
{ldapAuditStartTime=03.05.2001 13:05:30}
```

LDAP server auditing is started on May 3<sup>rd</sup>, 2001 at 1:05:30 PM.

### 3.1.16.15. LDAP Audit Stop Time

The LDAP Audit Stop Time single-valued attribute specifies the time when the LDAP server stops recording of audit information. A value of **none** specifies no stop time and does not affect the audit behavior. Specify the time in the following syntax:

*DD.MM.YYYY hh:mm:ss*

where

*DD* specifies the day of month (**01** up to **31**)

*MM* specifies the month (**01** up to **12**)

*YYYY* specifies the year

*hh* specifies the hour (**00** up to **23**)

*mm* specifies the minute (**00** up to **59**)

*ss* specifies the seconds (**00** up to **59**)

The default value is **none** (no stop time specified).

#### 3.1.16.15.1. Abbreviation

LASTO

#### 3.1.16.15.2. LDAP Name(s)

ldapAuditStopTime

#### 3.1.16.15.3. Syntax (DAP)

Directory String {32768}

#### 3.1.16.15.4. Example (DAP)

```
{LASTO=03.05.2001 20:05:30}
```

LDAP server auditing is finished on May 3<sup>rd</sup>, 2001 at 8:05:30 PM.

#### 3.1.16.15.5. Example (LDAP)

```
{ldapAuditStopTime=03.05.2001 20:05:30}
```

LDAP server auditing is finished on May 3<sup>rd</sup>, 2001 at 8:05:30 PM.

### 3.1.16.16. LDAP Audit Cron Job Start Time

The LDAP Audit Cron Job Start Time single-valued attribute specifies the time when the LDAP server starts a cron job that performs the action specified in the LDAP Audit Overflow attribute despite whether any of the specified limits (maximum number of records or maximum file size) has been exceeded. (See the section **LDAP Audit Overflow** in this chapter for details.) The LDAP Audit Cron Job Interval attribute specifies when the next **cron** job is started. (See the section below for details.) This attribute may be useful for example, to generate audit log files that cover a certain time frame.

The value **never** disables starting **cron** jobs. This is the default value.

The value **now** starts the **cron** job immediately.

Specify the time in the following syntax:

*DD.MM.YYYY hh:mm:ss*

where

*DD* specifies the day of month (**01** up to **31**)

*MM* specifies the month (**01** up to **12**)

*YYYY* specifies the year

*hh* specifies the hour (**00** up to **23**)

*mm* specifies the minute (**00** up to **59**)

*ss* specifies the seconds (**00** up to **59**)

If the LDAP server reads a start time in the past and a cron job interval is specified in the LDAP Audit Cron Job Interval attribute (greater than 60 seconds), the LDAP server calculates the next start time of the cron job automatically.

If a start time is specified, LDAP audit logging must be enabled; that is, the LDAP Audit On (LAON) attribute must be set to **TRUE**. Keep in mind that you may have disabled auditing by specifying the attributes LDAP Audit Start and / or LDAP Audit Stop Time.

The LDAP Audit Cron Job Start Time attribute has no effect on whether or not the LDAP server performs auditing.

#### 3.1.16.16.1. Abbreviation

LACRS

#### 3.1.16.16.2. LDAP Name(s)

ldapAuditCronStart

#### 3.1.16.16.3. Syntax (DAP)

Directory String {32768}

#### 3.1.16.16.4. Syntax (LDAP)

Directory String {32768} (LDAP Style)

#### 3.1.16.16.5. Example (DAP)

```
{LACRS=03.05.2001 20:05:30}
```

The LDAP audit cron job is started on May 3<sup>rd</sup>, 2001 at 8:05:30 PM.

#### 3.1.16.16.6. Example (LDAP)

```
{ldapAuditCronStart=03.05.2001 20:05:30}
```

The LDAP audit cron job is started on May 3<sup>rd</sup>, 2001 at 8:05:30 PM.

#### 3.1.16.17. LDAP Audit Cron Job Interval

The LDAP Audit Cron Job Interval single-valued attribute specifies the wait time in seconds after which the LDAP server starts the next **cron** job that performs the action specified in



the LDAP Audit Overflow attribute despite whether any of the specified limits (maximum number of records or maximum file size) has been exceeded. (See the section **LDAP Audit Overflow** in this chapter for details.) The LDAP Audit Cron Job Start Time attribute specifies the date when the **cron** job is started the first time. (See the section above for details.)

Specify a value greater than 60 seconds. The default value is **3600** seconds. If a value lower than 60 seconds is specified, the value 60 seconds is set automatically.

If the value **0** is specified, the LDAP server starts the **cron** job once at LDAP Audit Cron Job Start Time.

If a cron job start time and interval are specified, LDAP audit logging must be enabled; that is, the LDAP Audit On (LAON) attribute must be set to **TRUE**. Keep in mind that you may have disabled auditing by specifying the attributes LDAP Audit Start and / or LDAP Audit Stop Time.

The LDAP Audit Cron Job Interval has no effect on whether or not the LDAP server performs auditing.

The following example illustrates how to use **cron** jobs:

Suppose you want to generate audit log files, where each file contains the LDAP operations for one hour. The current time is February 28, 2006, 7:35 PM. The LDAP audit is turned on (LDAP Audit On (LAON) attribute is **TRUE**). Neither a LDAP Audit Start nor a LDAP Audit Stop is specified. The overflow action to take is moving the audit log file (ldapAuditOverFlow=moveFile). Specify the following values for managing the cron job:

```
{ldapAuditCronStart=28.02.2006 20:00:00}
```

```
ldapAuditCronInterval=3600
```

At 8:00 PM, the LDAP server starts the first cron job, moving the current audit log file to a file named **audit.log.20060228200000Z\_001** containing the audit records until 8:00 PM. One hour (3600 seconds) later, at 9:00 PM, the LDAP server again starts a new cron job, moving the current audit log file to a file named **audit.log.20060228210000Z\_002** containing the audit records from 8:00 PM through 9:00 PM. This action now takes place every hour, moving the current audit log file.

If, meanwhile, either the maximum number of records or the maximum file size is exceeded, the LDAP server takes the overflow action additionally.

If the LDAP server is re-started (for example, at 10:45 PM), it finds a cron job start time in the past. The LDAP server now calculates the next start time of the cron job for 11:00 PM to get audit log files containing the operations on the hour.

The results are several audit log files (preferably one per hour) that may be easier to evaluate; for example, to compare the number of operations for different time periods. Note that if the audit log file has exceeded a specified limit or the LDAP server has been re-started for any reason, there is more than one audit log file per hour.

To stop the **cron** job feature, change the LDAP Audit Cron Job Interval value to **0**.

#### 3.1.16.17.1. Abbreviation

LACRI

#### 3.1.16.17.2. LDAP Name(s)

ldapAuditCronInterval

#### 3.1.16.17.3. Syntax (DAP)

Integer

#### 3.1.16.17.4. Syntax (LDAP)

Integer (LDAP Style)

#### 3.1.16.17.5. Example (DAP)

```
LACRI=2400
```

#### 3.1.16.17.6. Example (LDAP)

```
ldapAuditCronInterval=2400
```

### 3.1.16.18. LDAP Audit Session Tracking Enabled

The LDAP Audit Session Tracking Enabled single-valued attribute specifies whether or not the LDAP server writes the values of the session tracking control (if provided) to the LDAP audit file.

Specify a value of **1** to enable session tracking and a value of **0** to disable session tracking.

An LDAP client can include the session tracking control with its LDAP request. The session tracking control provides the following information:

- The sessionSourceIP - The IP address of the originating LDAP client application with a maximum length of 128 characters.
- The sessionSourceName - A description of the originating system, with a maximum length of 65536 bytes. The value might be, for example, a hostname, a distinguished name, or a web service address.
- The formatOID - A description of the semantics of the sessionSourceName and the sessionTrackingIdentifier, with a maximum length of 1024 bytes.
- The sessionTrackingIdentifier - The tracking identifier; for example, a username that has already been authenticated by the application that is generating the session, with an unlimited length (but it should not be longer than 65536 characters).

Control the number of bytes written to the audit log file by specifying the LDAP Audit Session Tracking InfoLen attribute.

See the Internet-Draft “LDAP Session Tracking Control” (draft-wahl-ldap-session-03.txt) for details about session tracking.

#### 3.1.16.18.1. Abbreviation

LASTE

#### 3.1.16.18.2. LDAP Name(s)

ldapAuditSessionTrackingEnabled

#### 3.1.16.18.3. Syntax (DAP)

Integer

#### 3.1.16.18.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.16.18.5. Example (DAP)

```
LASTE=1
```

#### 3.1.16.18.6. Example (LDAP)

```
ldapAuditSessionTrackingEnabled=0
```

### 3.1.16.19. LDAP Audit Session Tracking InfoLen

The LDAP Audit Session Tracking InfoLen single-valued attribute specifies the maximum length of each field (the sessionSourceIP, the sessionSourceName, the formatOID, and the sessionTrackingIdentifier) of the session tracking control that the LDAP server writes to the audit log file. The default value is **64**; that is, the LDAP server writes the initial 64 bytes of the value to the audit log file.

Set the value of the LDAP Audit Session Tracking Enabled attribute to the value **1** to enable session tracking. (See the LDAP Audit Session Tracking Enabled attribute above for details.)

#### 3.1.16.19.1. Abbreviation

LASTI

#### 3.1.16.19.2. LDAP Name(s)

ldapAuditSessionTrackingInfoLen

#### 3.1.16.19.3. Syntax (DAP)

Integer

#### 3.1.16.19.4. Syntax (LDAP)

Integer String (LDAP Style)

#### 3.1.16.19.5. Example (DAP)

```
LASTI=125
```

#### 3.1.16.19.6. Example (LDAP)

```
ldapAuditSessionTrackingInfoLen=125
```

### 3.1.17. Attributes of the Password Policy Subentry

Attributes for Password Policy are attributes of Password Policy subentries. These subentries administer the password policy for the directory. DirX Directory supports:

- A global Password Policy subentry **CN=GlobalPasswordPolicy**:  
The attributes of this subentry specify the standard password policy for the directory. DirX Directory automatically creates this subentry with a default subtree specification so that it applies to all entries in the DIT (**{SS={DEF=TRUE}}**). The administrator may modify the attributes specifying the password policy using the DAP or LDAP protocol. The global Password Policy subentry applies to all entries that are not controlled by a more specific Password Policy subentry.

The attributes specifying the format of the stored user password attributes (Password Storage Scheme, Password Storage Scheme Algorithm and Password Storage Scheme Level) can be administered only in this global Password Policy subentry. They are ignored if specified in a specific Password Policy subentry.

- Specific Password Policy subentries:  
Specific Password Policy subentries can be created below an administrative point (ADM-Point) if the Administrative Role of the ADM-Point has the value **passwordPolicySpecificArea (PPSA)**. As for every subentry the Subtree Specification attribute controls the relevance of the subentry. The object class of specific Password Policy subentries must contain the values **subEntry** and **pwdPolicy**.

No **innerPasswordPolicySpecificArea** is supported. Furthermore the DSA rejects the coexistence of more than one specific Password Policy subentry in a particular passwordPolicySpecificArea. An object class violation error is returned.

There is exactly one Password Policy subentry for each entry in the DIT, either the global one or a specific one - if there is a specific Password Policy subentry up in the tree which is relevant for the entry in terms of the Subtree Specification.

The attributes of the Password Policy subentries control how passwords are used and administered. They provide a set of rules that ensures that

- Passwords meet construction requirements to prevent users from choosing easy-to-guess passwords (attributes: Password Check Syntax, Password Minimum Length, Password Maximum Length, Password Minimum Lower Case Characters, Password Minimum Upper Case Characters, Password Minimum Digits, Password Minimum Special Character, Password Exclude Name)
- The re-use of old passwords is restricted (attribute: Password In History)
- Users change their password periodically (attributes: Password Minimum Age, Password Maximum Age, Password Expire Warning, Password Grace Login Limit)
- Accounts are locked out after failed logins (attributes: Password Lockout, Password Lockout Duration, Password Failure Counter Interval, Password Maximum Failure, Password Must Change)
- Login information is stored at the user's entry, for example the number of failed logins (attribute: Password Store Login Info)
- Particular users can be excluded from the password policy rules by means of the Password Policy Exclusions attribute

They also provide information about level and algorithm of password hashing (attributes: Password Storage Scheme, Password Storage Scheme Algorithm, Password Storage Scheme Level). The DSA evaluates these attributes only if they are contained in the global Password Policy subentry. Adding them to specific Password Policy subentries has no effect on the storage of user passwords.

To support the rules for the restricted re-use of old passwords, the password expiration, and the account lockout each user entry provides a set of directory operational password policy attributes. See section titled **Password Policy Attributes** for details.

The DSA creates the global Password Policy subentry under the root DSE with the DN **/cn=globalPasswordPolicy** if this subentry does not exist at start up time of the DSA. This subentry contains the following attributes specifying the default password policy:

- Password Check Syntax (pwdCheckQuality, pwdCheckSyntax, PPCHSY, value: **0** – Syntax checking disabled)
- Password Minimum Length (pwdMinLength, PPMINL, value: **0**)
- Password Maximum Length (pwdMaxLength, PPMAXL, value: **0**)
- Password Minimum Lower Case Character (pwdMinLowerCase, PPMINLO, value: **0**)
- Password Minimum Upper Case Character (pwdMinUpperCase, PPMINUP, value: **0**)
- Password Minimum Digit (pwdMinDigit, PPMINDI, value: **0**)
- Password Minimum Special Character (pwdMinSpecialChar, PPMINS, value: **0**)
- Password Exclude Name (pwdExcludeName, PPEXCNA, value: **0**)
- Password In History (pwdInHistory, PPINHI, value: **0**)
- Password Minimum Age (pwdMinAge, PPMINA, value: **0**)
- Password Maximum Age (pwdMaxAge, PPMAXA, value: **0**)
- Password Expire Warning (pwdExpireWarning, PPEXWA, value: **0**)

- Password Grace Login Limit (pwdGraceLoginLimit, PPGRLL, value: **0**)
- Password Lockout (pwdLockout, PPLOCK, value: **FALSE**)
- Password Lockout Duration (pwdLockoutDuration, PPLODU, value: **0**)
- Password Maximum Failure (pwdMaxFailure, PPMAXF, value: **0**)
- Password Failure Counter Interval (pwdFailureCountInterval, PPFCIN, value: **0**)
- Password Must Change (pwdMustChange, PPMUCH, value: **FALSE**)
- Password Store Login Info (pwdStoreloginInfo, PPSTLI, value: **0**)
- Password Storage Scheme (pwdStorageScheme, PPSTSC, value: **ssha** – User password values are hashed according to salted SHA)
- Password Storage Scheme Algorithm (pwdStorageSchemeAlgo, PPSTSA, value: **SHA-256**)
- Password Storage Scheme Level (pwdStorageSchemeLevel, PPSTSL, value: **2**)
- Password Policy Exclusions (pwdExclusions, PPEXCL, no value)

Use the **dirxcp obj** operations to modify the attributes of the global Password Policy subentry or to create and modify subtree specific Password Policy subentries. Modifications of a Password Policy subentry become effective immediately.

The following sections provide information about the attributes of password policy subentries.

#### 3.1.17.1. Password Check Syntax

The Password Check Syntax attribute is a single-valued attribute specifying how the password syntax is checked. Specify one of the following values:

- **0** - Syntax checking is disabled.
- **1** - Syntax checking is enabled for password values in clear text format. If the server is unable to check the syntax (due to a hashed password) the password is accepted.
- **2** - Syntax checking is enabled for password values in clear text format. If the server is unable to check the syntax the password is rejected. (That is only passwords in clear text format are accepted if the syntax is checked successful.)

The default value is **0**.

The value of this attribute affects checking with respect to the following attributes:

- Password Minimum Length
- Password Maximum Length
- Password Minimum Lower Case Character
- Password Minimum Upper Case Character
- Password Minimum Digit
- Password Minimum Special Character

- Password Exclude Name
- Password In History

If another value than described above is specified the DSA ignores the value of this attribute and establishes the default behavior.

#### 3.1.17.1.1. Abbreviation

PPCHSY

#### 3.1.17.1.2. LDAP Name(s)

pwdCheckQuality, pwdCheckSyntax

#### 3.1.17.1.3. Syntax (DAP)

Integer

#### 3.1.17.1.4. Syntax (LDAP)

Integer

#### 3.1.17.1.5. Example (DAP)

```
PPCHSY=2
```

#### 3.1.17.1.6. Example (LDAP)

```
pwdCheckQuality=2
```

### 3.1.17.2. Password Minimum Length

The Password Minimum Length attribute is a single-valued attribute specifying the minimum number of octets that must be used in a password when syntax checking is enabled (that is the Password Check Syntax attribute value is **1** or **2**). (See section titled **Password Check Syntax** above for details.)

The default value is **0** (no minimum password length is enforced).

#### 3.1.17.2.1. Abbreviation

PPMINL

#### 3.1.17.2.2. LDAP Name(s)

pwdMinLength

#### 3.1.17.2.3. Syntax (DAP)

Integer

#### 3.1.17.2.4. Syntax (LDAP)

Integer

#### 3.1.17.2.5. Example (DAP)

```
PPMINL=6
```

#### 3.1.17.2.6. Example (LDAP)

```
pwdMinLength=6
```

### 3.1.17.3. Password Maximum Length

The Password Maximum Length attribute is a single-valued attribute specifying the maximum number of octets that can be used in a password when syntax checking is enabled (that is the Password Check Syntax attribute value is **1** or **2**). (See section titled **Password Check Syntax** above for details.)

The default value is **0**. The maximum password length is not restricted by the password policy. However it is restricted to 128 octets by the X.520 standard.

If a value lower than Password Minimum Length or the sum of Password Minimum Lower Case Character, Password Upper Case Character, Password Minimum Digit and Password Minimum Special Character is specified, the DSA ignores the value of this attribute and establishes the default behavior.

#### 3.1.17.3.1. Abbreviation

PPMAXL

#### 3.1.17.3.2. LDAP Name(s)

pwdMaxLength

#### 3.1.17.3.3. Syntax (DAP)

Integer

#### 3.1.17.3.4. Syntax (LDAP)

Integer



#### 3.1.17.3.5. Example (DAP)

```
PPMAXL=24
```

#### 3.1.17.3.6. Example (LDAP)

```
pwdMaxLength=26
```

### 3.1.17.4. Password Minimum Lower Case Character

The Password Minimum Lower Case Character attribute is a single-valued attribute specifying the minimum number of lowercase characters (a-z) that must be used in a password when syntax checking is enabled (that is, when the Password Check Syntax attribute value is **1** or **2**). (See the section **Password Check Syntax** for details.)

The default value is **0** (no minimum number of lowercase characters is enforced).

#### 3.1.17.4.1. Abbreviation

PPMINLO

#### 3.1.17.4.2. LDAP Name(s)

pwdMinLowerCase

#### 3.1.17.4.3. Syntax (DAP)

Integer

#### 3.1.17.4.4. Syntax (LDAP)

Integer

#### 3.1.17.4.5. Example (DAP)

```
PPMINLO=2
```

#### 3.1.17.4.6. Example (LDAP)

```
pwdMinLowerCase=2
```

### 3.1.17.5. Password Minimum Upper Case Character

The Password Minimum Upper Case Character attribute is a single-valued attribute specifying the minimum number of uppercase characters (A-Z) that must be used in a password when syntax checking is enabled (that is, when the Password Check Syntax

attribute value is **1** or **2**). (See the section **Password Check Syntax** for details.)

The default value is **0** (no minimum number of uppercase characters is enforced).

#### 3.1.17.5.1. Abbreviation

PPMINUP

#### 3.1.17.5.2. LDAP Name(s)

pwdMinUpperCase

#### 3.1.17.5.3. Syntax (DAP)

Integer

#### 3.1.17.5.4. Syntax (LDAP)

Integer

#### 3.1.17.5.5. Example (DAP)

```
PPMINUP=2
```

#### 3.1.17.5.6. Example (LDAP)

```
pwdMinUpperCase=2
```

### 3.1.17.6. Password Minimum Digit

The Password Minimum Digit attribute is a single-valued attribute specifying the minimum number of digits (0-9) that must be used in a password when syntax checking is enabled (that is, when the Password Check Syntax attribute value is **1** or **2**). (See the section **Password Check Syntax** for details.)

The default value is **0** (no minimum number of digits is enforced).

#### 3.1.17.6.1. Abbreviation

PPMINDI

#### 3.1.17.6.2. LDAP Name(s)

pwdMinDigit

#### 3.1.17.6.3. Syntax (DAP)

Integer

#### 3.1.17.6.4. Syntax (LDAP)

Integer

#### 3.1.17.6.5. Example (DAP)

```
PPMINDI=1
```

#### 3.1.17.6.6. Example (LDAP)

```
pwdMinDigit=1
```

### 3.1.17.7. Password Minimum Special Character

The Password Minimum Special Character attribute is a single-valued attribute specifying the minimum number of special characters (any octet outside the range of A-z and 0-9) that must be used in a password when syntax checking is enabled (that is, when the Password Check Syntax attribute value is **1** or **2**). (See the section **Password Check Syntax** for details.)

The default value is **0** (no minimum number of special characters is enforced).

#### 3.1.17.7.1. Abbreviation

PPMINS

#### 3.1.17.7.2. LDAP Name(s)

pwdMinSpecialChar

#### 3.1.17.7.3. Syntax (DAP)

Integer

#### 3.1.17.7.4. Syntax (LDAP)

Integer

#### 3.1.17.7.5. Example (DAP)

```
PPMINS=2
```

#### 3.1.17.7.6. Example (LDAP)

```
pwdMinSpecialChar=2
```

### 3.1.17.8. Password Exclude Name

The Password Exclude Name attribute is a single-valued attribute specifying whether the password is rejected if it has a substring that matches the user's name. The rule is checked when syntax checking is enabled (that is, when the Password Check Syntax attribute value is **1** or **2**). (See the section **Password Check Syntax** for details.)

Specify one of the following values:

- **0** - No restriction.
- **1** - Passwords are rejected if the value is a case-ignored substring of the user's distinguished name (DN) in LDAP format; for example, the user **cn=abele,ou=sales,o=my-company** cannot set his password to **Bele,Ou**.
- **2** - Passwords are rejected if the user's relative distinguished name (RDN) is a case-ignored substring of the password; for example, the user **cn=abele,ou=sales,o=my-company** cannot set his password to **Abele1**.
- **3** - Passwords are rejected if either the condition for value **1** or the condition for value **2** applies.

The default value is **0**.

#### 3.1.17.8.1. Abbreviation

PPEXCNA

#### 3.1.17.8.2. LDAP Name(s)

#### 3.1.17.8.3. pwdExcludeName

#### 3.1.17.8.4. Syntax (DAP)

Integer

#### 3.1.17.8.5. Syntax (LDAP)

Integer

#### 3.1.17.8.6. Example (DAP)

```
PPEXCNA=2
```

#### 3.1.17.8.7. Example (LDAP)

```
pwdExcludeName=3
```

### 3.1.17.9. Password In History

The Password In History attribute is a single-valued attribute specifying the number of most recently used passwords that are prohibited for reuse when syntax checking is enabled (that is, when the Password Check Syntax attribute value is **1** or **2**). (See the section **Password Check Syntax** for details.) The most recently used password values are stored in the directory operational Password History attribute.

The default value is **0**, which permits any previously used password to be reused.

The maximum supported number is **24**. If a number greater than 24 is specified, the DSA ignores the value of this attribute and establishes the default behavior.

#### 3.1.17.9.1. Abbreviation

PPINHI

#### 3.1.17.9.2. LDAP Name(s)

pwdInHistory

#### 3.1.17.9.3. Syntax (DAP)

Integer

#### 3.1.17.9.4. Syntax (LDAP)

Integer

#### 3.1.17.9.5. Example (DAP)

```
PPINHI=5
```

#### 3.1.17.9.6. Example (LDAP)

```
pwdInHistory=5
```

### 3.1.17.10. Password Minimum Age

The Password Minimum Age attribute is a single-valued attribute specifying the number of seconds that must elapse between modifications to the password. The default value is **0** (seconds).

#### 3.1.17.10.1. Abbreviation

PPMINA

#### 3.1.17.10.2. LDAP Name(s)

pwdMinAge

#### 3.1.17.10.3. Syntax (DAP)

Integer

#### 3.1.17.10.4. Syntax (LDAP)

Integer

#### 3.1.17.10.5. Example (DAP)

```
PPMINA=86400 (The password must remain unchanged for at least one
day.)
```

#### 3.1.17.10.6. Example (LDAP)

```
pwdMinAge=86400
```

### 3.1.17.11. Password Maximum Age

The Password Maximum Age attribute is a single-valued attribute specifying the number of seconds after which a password expires. The default value is **0** (the password never expires).

If a password maximum age is specified, we recommend disabling LDAP backend sharing and setting the LDAP Unbind delay time to 0 to avoid successful binds using an expired password. (See the sections **LDAP Unbind Delay Time** and **LDAP Backend Sharing** in this chapter for details.)

#### 3.1.17.11.1. Abbreviation

PPMAXA

#### 3.1.17.11.2. LDAP Name(s)

pwdMaxAge

#### 3.1.17.11.3. Syntax (DAP)

Integer

#### 3.1.17.11.4. Syntax (LDAP)

Integer

#### 3.1.17.11.5. Example (DAP)

```
PPMAXA=8640000 (The password expires after 100 days.)
```

#### 3.1.17.11.6. Example (LDAP)

```
pwdMaxAge=3600
```

### 3.1.17.12. Password Expire Warning

The Password Expire Warning attribute is a single-valued attribute specifying when the DSA starts to return password expiration warning messages to an authenticating user. A successful bind operation returns the warning message "**Warning: password expires in *n* seconds.**" where *n* specifies the number of seconds after which the user's password will expire. If a value *n* greater than 0 is specified, the password expires *n* seconds after sending the first expiration warning.

The default value **0** specifies that no warnings are returned.

#### 3.1.17.12.1. Abbreviation

PPEXWA

#### 3.1.17.12.2. LDAP Name(s)

pwdExpireWarning

#### 3.1.17.12.3. Syntax (DAP)

Integer

#### 3.1.17.12.4. Syntax (LDAP)

Integer

#### 3.1.17.12.5. Example (DAP)

```
PPEXWA=6000
```

#### 3.1.17.12.6. Example (LDAP)

```
pwdExpireWarning=6000
```

### 3.1.17.13. Password Grace Login Limit

The Password Grace Login Limit attribute is a single-valued attribute specifying the maximum number of times an expired password can be used to authenticate. A successful bind operation due to the Password Grace Login Limit attribute returns the warning message "**Warning:  $n$  grace logins left.**" where  $n$  specifies the number of remaining grace logins.

The default value is **0** (authentication fails when using an expired password).

#### 3.1.17.13.1. Abbreviation

PPGRLL

#### 3.1.17.13.2. LDAP Name(s)

pwdGraceLoginLimit

#### 3.1.17.13.3. Syntax (DAP)

Integer

#### 3.1.17.13.4. Syntax (LDAP)

Integer

#### 3.1.17.13.5. Example (DAP)

```
PPGRLL=3
```

#### 3.1.17.13.6. Example (LDAP)

```
pwdGraceLoginLimit=3
```

### 3.1.17.14. Password Lockout

The Password Lockout attribute is a single-valued attribute specifying whether the password can be used to authenticate after a specific number of consecutive failed bind attempts. The maximum number of consecutive failed bind attempts is specified in the Password Maximum Failure (PPMAXF, pwdMaxFailure) attribute. The time period within which the failed logins are counted is specified in the Password Failure Counter Interval (PPFCIN, pwdFailureCountInterval) attribute. The lockout duration is specified in the Password Lockout Duration (PPLODU, pwdLockoutDuration) attribute.

Specify one of the following values for the Password Lockout attribute:

- **TRUE** - The password cannot be used to authenticate after pwdMaxFailure number of consecutive failed bind attempts. (Password lockout policy is enabled.)



- **FALSE** - The password can be used to authenticate after pwdMaxFailure number of consecutive failed bind attempts. (Password lockout policy is disabled.)

The default value is **FALSE**.

If a password lockout policy is enabled, we recommend disabling LDAP backend sharing and setting the LDAP Unbind delay time to **0** to avoid successful binds using a locked-out account. (See the sections **LDAP Unbind Delay Time** and **LDAP Backend Sharing** in this chapter for details.)

#### 3.1.17.14.1. Abbreviation

PPLOCK

#### 3.1.17.14.2. LDAP Name(s)

pwdLockout

#### 3.1.17.14.3. Syntax (DAP)

Boolean

#### 3.1.17.14.4. Syntax (LDAP)

Boolean

#### 3.1.17.14.5. Example (DAP)

```
PPLOCK=TRUE
```

#### 3.1.17.14.6. Example (LDAP)

```
pwdLockout=TRUE
```

### 3.1.17.15. Password Lockout Duration

The Password Lockout Duration attribute is a single-valued attribute specifying the number of seconds during which a password cannot be used to authenticate after a specific number of consecutive failed bind attempts. This attribute is evaluated only if the password lockout policy is enabled. (The Password Lockout attribute value is **TRUE**.)

The default value is **0**. (The password cannot be used to authenticate until unlocked by an administrator.)

An administrator can unlock an account at any time. (See the Password Locked Time attribute for information on how to unlock a password.)

#### 3.1.17.15.1. Abbreviation

PPLODU

#### 3.1.17.15.2. LDAP Name(s)

pwdLockoutDuration

#### 3.1.17.15.3. Syntax (DAP)

Integer

#### 3.1.17.15.4. Syntax (LDAP)

Integer

#### 3.1.17.15.5. Example (DAP)

```
PPLODU=1200
```

#### 3.1.17.15.6. Example (LDAP)

```
pwdLockoutDuration=1200
```

### 3.1.17.16. Password Maximum Failure

The Password Maximum Failure attribute is a single-valued attribute specifying the number of consecutive failed bind attempts after which the password cannot be used to authenticate. This attribute is evaluated only if the password lockout policy is enabled. (The Password Lockout attribute value is **TRUE**.)

The default value is **0**. (Disables the password lockout policy.)

#### 3.1.17.16.1. Abbreviation

PPMAXF

#### 3.1.17.16.2. LDAP Name(s)

pwdMaxFailure

#### 3.1.17.16.3. Syntax (DAP)

Integer

#### 3.1.17.16.4. Syntax (LDAP)

Integer

#### 3.1.17.16.5. Example (DAP)

```
PPMAXF=3
```

#### 3.1.17.16.6. Example (LDAP)

```
pwdMaxFailure=3
```

### 3.1.17.17. Password Failure Counter Interval

The Password Failure Counter Interval attribute is a single-valued attribute specifying the number of seconds after which the password failures are purged from the failure counter (the DSA operational Password Failure Time attribute) even though no successful authentication was performed. This attribute is evaluated only if the password lockout policy is enabled. (The Password Lockout attribute value is **TRUE**.)

The default value **0** specifies that the failure counter is only reset by a successful authentication.

#### 3.1.17.17.1. Abbreviation

PPFCIN

#### 3.1.17.17.2. LDAP Name(s)

pwdFailureCountInterval

#### 3.1.17.17.3. Syntax (DAP)

Integer

#### 3.1.17.17.4. Syntax (LDAP)

Integer

#### 3.1.17.17.5. Example (DAP)

```
PPFCIN=600
```

#### 3.1.17.17.6. Example (LDAP)

```
pwdFailureCountInterval=600
```

### 3.1.17.18. Password Must Change

The Password Must Change attribute is a single-valued attribute specifying whether the user must change his password when he binds to the directory for the first time after an administrator has set or reset his password. Specify one of the following values:

- **TRUE** - The user must change his password after an administrator has set or reset the password.
- **FALSE** - The user is not required to change his password.

Whether an administrator has set or reset a password is saved in the directory operational Password Reset (pwdReset, PPORSET) attribute of each entry. (See the Password Reset (pwdReset, PPORSET) attribute for details.)

The default value is **FALSE**.

#### 3.1.17.18.1. Abbreviation

PPMUCH

#### 3.1.17.18.2. LDAP Name(s)

pwdMustChange

#### 3.1.17.18.3. Syntax (DAP)

Boolean

#### 3.1.17.18.4. Syntax (LDAP)

Boolean

#### 3.1.17.18.5. Example (DAP)

```
PPMUCH=TRUE
```

#### 3.1.17.18.6. Example (LDAP)

```
pwdMustChange=TRUE
```

### 3.1.17.19. Password Store Login Info

The Password Store Login Info attribute is a single-valued attribute specifying login-related information (bind operation-related information) stored at the user's entry. Specify one of the following values:

- **0** - No information about logins is stored.
- **1** - The number of failed logins is stored in the user's entry. If a bind operation fails due to

an **Invalid Credentials** error, the value of the user entry's Password Number of Failed Logins attribute is incremented. (See the Password Number of Failed Logins attribute (pwdNumberOfFailedLogins, PPONOFLL) for details.)

- **2** - The time of the most recent successfully-performed bind operation is stored in the user entry's Password Last Login Time attribute. (See the Password Last Login Time attribute (pwdLastLoginTime, PPOLLLOT) for details.)
- **3** - The number of failed logins and the time of the most recent successfully-performed bind operation are stored at the user's entry.

The default value is **0**.

If a value other than the values described here is specified, the DSA ignores the value of this attribute and establishes the default behavior.

#### 3.1.17.19.1. Abbreviation

PPSTLI

#### 3.1.17.19.2. LDAP Name

pwdStoreloginInfo

#### 3.1.17.19.3. Syntax (DAP)

Integer

#### 3.1.17.19.4. Syntax (LDAP)

Integer

#### 3.1.17.19.5. Example (DAP)

```
PPSTLI=2
```

#### 3.1.17.19.6. Example (LDAP)

```
pwdStoreloginInfo=2
```

### 3.1.17.20. Password Storage Scheme

The Password Storage Scheme attribute is a single-valued attribute specifying the algorithm used to hash user password values. (See also **Password Encryption** in **User Password**.) Specify one of the following values (case-insensitive):

- **sha** - A SHA algorithm is used to hash user password values.
- **ssha** - A Salted SHA algorithm is used to hash user password values.
- **plain** - User password values are not hashed.

The default value is **ssha**.

If a value other than **plain** is specified, specify the level of password hashing in the Password Storage Scheme Level attribute.

If the value **plain** is specified and the user password of an entry is stored in hashed format - for example, the entry has been created by an LDIF file import - this entry cannot bind using the clear text password. To perform the bind successfully, the Password Storage Scheme attribute must be set to **sha** or **ssha** and the Password Storage Scheme Level attribute must be set at least to **1** (Basic Level).

The SSHA format is the most secure format because the SHA algorithm is applied to a concatenation of the current user password value and a random string.

In order to be able to compare two user password values, one of the values must be a clear text value. It is theoretically impossible to compare two values that are salted hashed or hashed with different algorithms. In these cases, the following operations always fail and return an **Unwilling to Perform** error:

- **obj compare** (-attribute option)
- **obj modify** (-changeattr, -removeattr, -replaceattr options)
- **obj search** (-filter option)

These operations work if one of the user password values (usually the value in the request) is specified in clear text format, regardless of the hashing scheme and algorithm used for the stored user password value.

The DSA evaluates this attribute only if it is contained in the global Password Policy subentry. It ignores this attribute if it is contained in specific Password Policy subentries.

#### 3.1.17.20.1. Abbreviation

PPSTSC

#### 3.1.17.20.2. LDAP Name(s)

pwdStorageScheme

#### 3.1.17.20.3. Syntax (DAP)

Directory String {32768}

#### 3.1.17.20.4. Syntax (LDAP)

Directory String {32768}

#### 3.1.17.20.5. Example (DAP)

```
PPSTSC=sha
```

#### 3.1.17.20.6. Example (LDAP)

```
pwdStorageScheme=sha
```

#### 3.1.17.21. Password Storage Scheme Algorithm

The Password Storage Scheme Algorithm attribute is a single-valued attribute specifying the hash algorithm that the DSA uses to hash a user password if the value of the Password Storage Scheme attribute is not **plain**. Specify one of the following values:

- **1** - The DSA uses the algorithm SHA-1 for hashing the user password. This action results in user password values starting with the tag {SHA1}.
- **224** - The DSA uses the algorithm SHA-224 for hashing the user password. This results in user password values starting with the tag {SHA224}.
- **256** - The DSA uses the algorithm SHA-256 for hashing the user password. This action results in user password values starting with the tag {SHA256}.
- **384** - The DSA uses the algorithm SHA-384 for hashing the user password. This action results in user password values starting with the tag {SHA384}.
- **512** - The DSA uses the algorithm SHA-1 for hashing the user password. This action results in user password values starting with the tag {SHA384}.

The default value is **1**. (See the section **Password Encryption** in the **User Password** section for details.)

The DSA evaluates this attribute only if it is contained in the global Password Policy subentry. It ignores this attribute if it is contained in specific Password Policy subentries.

Note: Do not use a value other than **1** if the password is going to be used by a DSA that is older than version DirX Directory 8.3.

##### 3.1.17.21.1. Abbreviation

PPSTSA

##### 3.1.17.21.2. LDAP Name(s)

pwdStorageSchemeAlgo

##### 3.1.17.21.3. Syntax (DAP)

Integer

##### 3.1.17.21.4. Syntax (LDAP)

Integer

##### 3.1.17.21.5. Example (DAP)

```
PPSTSA=1
```

#### 3.1.17.21.6. Example (LDAP)

```
pwdStorageSchemeAlgo=1
```

### 3.1.17.22. Password Storage Scheme Level

The Password Storage Scheme Level attribute is a single-valued attribute specifying the level of password hashing if the value of the Password Storage Scheme attribute is not **plain**. Specify one of the following values:

- **1** (Basic Level) - The hashing of a stored password is taken into account when performing an operation; for example, when evaluating a filter in a search operation.
- **2** (Read Level) - When a password in clear text format is read, it is converted into the corresponding hash format before returning it to the requestor or writing it into an LDIF file.
- **3** (Read and Write Level) - When a password in clear text format is to be written into the database, it is converted into the corresponding hash format before saving it.

The default value is **2**. (See section titled **Password Encryption** in the **User Password** section for details.)

The DSA evaluates this attribute only if it is contained in the global Password Policy subentry. It ignores this attribute if it is contained in specific Password Policy subentries.

#### 3.1.17.22.1. Abbreviation

```
PPSTSL
```

#### 3.1.17.22.2. LDAP Name(s)

```
pwdStorageSchemeLevel
```

#### 3.1.17.22.3. Syntax (DAP)

```
Integer
```

#### 3.1.17.22.4. Syntax (LDAP)

```
Integer
```

#### 3.1.17.22.5. Example (DAP)

```
PPSTSL=3
```



#### 3.1.17.22.6. Example (LDAP)

```
pwdStorageSchemeLevel=3
```

#### 3.1.17.23. Password Attribute Type

The Password Attribute Type attribute is a single-valued attribute specifying the password attribute type to which the password policy is applied. The default value is the OID of the userPassword attribute type (**2.5.4.35**). The DSA ignores the value of this attribute. It always applies the specified policies to the userPassword attribute type.

##### 3.1.17.23.1. Abbreviation

PPATTR

##### 3.1.17.23.2. LDAP Name(s)

pwdAttribute

##### 3.1.17.23.3. Syntax (DAP)

Object Identifier

##### 3.1.17.23.4. Syntax (LDAP)

Object Identifier (LDAP style) in dotted notation.

##### 3.1.17.23.5. Example (DAP)

```
PPATTR=2.5.4.35
```

##### 3.1.17.23.6. Example (LDAP)

```
pwdAttribute=2.5.4.35
```

#### 3.1.17.24. Password Allow User Change

The Password Allow User Change attribute is a single-valued attribute specifying whether a user can change his own password. Specify one of the following values:

- **TRUE** - The user can change his own password.
- **FALSE** - The user cannot change his own password.

The default value is **TRUE**.

The DSA does not support this attribute. Access control can be used to allow or deny a user to change his password.

#### 3.1.17.24.1. Abbreviation

PPAUCH

#### 3.1.17.24.2. LDAP Name(s)

pwdAllowUserChange

#### 3.1.17.24.3. Syntax (DAP)

Boolean

#### 3.1.17.24.4. Syntax (LDAP)

Boolean

#### 3.1.17.24.5. Example (DAP)

```
PPAUCH=FALSE
```

#### 3.1.17.24.6. Example (LDAP)

```
pwdAllowUserChange=FALSE
```

### 3.1.17.25. Password Safe Modify

The Password Safe Modify attribute is a single-valued attribute specifying whether or not the old password value must be specified when changing the password. Specify one of the following values:

- **TRUE** - The old password value must be specified.
- **FALSE** - The old password value does not need to be specified.

The default value is **FALSE**.

The DSA does not support this attribute.

#### 3.1.17.25.1. Abbreviation

PPSAMO

#### 3.1.17.25.2. LDAP Name(s)

pwdSafeModify

#### 3.1.17.25.3. Syntax (DAP)

Boolean

#### 3.1.17.25.4. Syntax (LDAP)

Boolean

#### 3.1.17.25.5. Example (DAP)

```
PPSAMO=TRUE
```

#### 3.1.17.25.6. Example (LDAP)

```
pwdSafeModify=TRUE
```

### 3.1.17.26. Password Policy Exclusions

The Password Policy Exclusions attribute is a multi-valued attribute specifying distinguished names of entries. The distinguished names can point to leaves or nodes. Password Policy checks concerning Password Aging and Account Locking are not performed during an operation if the target entry's DN matches or contains one of the Password Policy Exclusions values. The target entry of an operation is the bind requestor or the DN of an entry whose userPassword is to be modified.

The default is no value.

Notes:

- The Password Storage Scheme (see attributes Password Storage Scheme and Password Storage Scheme Level above) does not depend on Password Policy Exclusions; that is, the userPassword of all users is treated in the same way with respect to the storage scheme.

#### 3.1.17.26.1. Abbreviation

PPEXCL

#### 3.1.17.26.2. LDAP Name(s)

pwdExclusions

#### 3.1.17.26.3. Syntax (DAP)

Distinguished Name

#### 3.1.17.26.4. Syntax (LDAP)

Distinguished Name

#### 3.1.17.26.5. Example (DAP)

```
{PPEXCL={/O=My-Company/CN=Admin}}
```

If this attribute value is found in a password policy subentry, the entry named /O=My-Company/CN=Admin is not subject to password policy rules.

#### 3.1.17.26.6. Example (LDAP)

```
{pwdExclusions={ou=development,o=my-company}}
```

If this attribute value is found in the password policy subentry, all entries located below the organizational unit “development” are not subject to password policy rules.

### 3.1.18. Attributes of the Proxied Authorization Control Subentry

The Proxied Authorization Control subentry stores the policy for operations that carry the Proxied Authorization control according to RFC 4370.

The Proxied Authorization control provides a mechanism for specifying an authorization identity on a per-operation basis. It allows an LDAP client to request that an operation be performed under a given authorization identity instead of under the current authorization identity associated with the session. This mechanism enables a user to perform operations efficiently on behalf of multiple users without the need to invoke an individual bind operation for each user. The authorization identity (authID) is the key data for the DirX Directory DSA’s access control decision operation in the context of each operation. As a result, proper administration of the privilege to use the Proxied Authorization control is very important.

Proxied authorization uses a single sign-on model of trust: the LDAP client (typically a service-related application) uses a method outside the scope of LDAP (for example HTTPS) to authenticate the user. In subsequent LDAP accesses to the directory service, the LDAP client uses the Proxied Authorization control to transport the authID - which must be represented in DN format - to the DirX Directory DSA. The DSA grants the proxy request 1) if the LDAP client is authenticated and 2) the policy stored in the Proxied Authorization subentry permits the LDAP client to present the specified authID.

There are two entities associated with a Proxied Authorization control:

- The proxyControlOwner - the user who has performed the bind operation that initiates the session in whose context the operation that contains the Proxied Authorization control occurs
- The proxiedUser - the user named in the Proxied Authorization control on whose behalf the operation is to be performed

Here is an example of a Proxied Authorization control as it might be sent along with an LDAP request:

```
Type:          2.16.840.1.113730.3.4.18
Criticality: TRUE
Value:         "dn:cn=abele,ou=sales,o=my-company"
```

The "Type" argument specifies the OID that identifies the control as a Proxied Authorization control, the "Criticality" value is mandatory, and the "Value" argument specifies the proxiedUser.

The Proxied Authorization Control subentry provides two attributes for defining Proxied Authorization control policies:

- The proxyControlOwner attribute, which specifies **who** (the entity that performs the bind operation to the directory service) is granted proxy rights.
- The subtreeSpecification attribute, which defines **for whom** the proxyControlOwner can act as a proxy.

If a request carries the a Proxied Authorization control and the DSA recognizes by the policy stored in the subentry that

- The user who originated the bind is not a valid proxyControlOwner or
- The proxiedUser is outside the range of valid authIDs for the particular proxyControlOwner

The DSA returns an error result that is mapped onto the LDAP error code **123** for that particular request.

A Proxied Authorization Control subentry can be created below an administrative point (ADM-Point) if the Administrative Role of the ADM-Point has the value **ProxyAuthenticationSpecificArea (PASA)**.

### 3.1.18.1. Proxy Control Owner

The Proxy Control Owner attribute is a multi-valued, mandatory attribute that specifies distinguished names (DNs) of entries to be granted proxy rights via Proxied Authorization controls. The distinguished names must point to leaf entries; that is, every entry that is granted proxy rights must be specified individually.

If there is no Proxied Authorization Control subentry, no entry is granted proxy rights.

#### 3.1.18.1.1. Abbreviation

PCO

#### 3.1.18.1.2. LDAP Name

proxyControlOwner

#### 3.1.18.1.3. Syntax (DAP)

Distinguished Name

#### 3.1.18.1.4. Syntax (LDAP)

Distinguished Name (LDAP style)

#### 3.1.18.1.5. Example (DAP)

```
PCO={/O=My-Company/CN=Admin}
```

If this attribute value is found in the Proxied Authorization Control subentry, the entry named `/O=My-Company/CN=Admin` is the only entry that is granted proxy rights.

#### 3.1.18.1.6. Example (LDAP)

```
proxyControlOwner={cn=manager,o=my-company};{cn=admin,o=my-company}
```

If these attribute values are found in the Proxied Authorization Control Subentry, the two entries `cn=manager,o=my-company` and `cn=admin,o=my-company` are the only entries that are granted proxy rights.

### 3.1.18.2. Subtree Specification

The Proxied Authorization Control subentry inherits the Subtree Specification attribute from its superior Objectclass subentry as a mandatory attribute. The Subtree Specification attribute restricts the range of authIDs that the entry specified in the Proxy Control Owner attribute is granted to use. The entire range of Subtree Specification values can be used to describe this restriction. (See **Subtree-Specification** in section **X.500 Directory Operational Attributes** above and in chapter **DirX Directory String Representation for DAP Binds** for details).

The following examples illustrate the effects of the Subtree Specification attribute values of this subentry.

#### 3.1.18.2.1. Examples (DAP)

```
SS={DEF=TRUE}
```

The entry specified by the Proxy Control Owner attribute can act as a proxy for every entry in the entire subtree below the Administrative Point.

```
SS={BAS={/OU=Sales}}
```

If the ADM-Point is /O=my-company, the entry specified by the proxyControlOwner attribute can act as a proxy for every entry in subtree below /O=my-company/OU=sales. If the Proxy Control Owner attempts to present, for example, the authID “dn:cn=mayer,o=development,o=my-company” as a proxiedUser in a Proxied Authorization control request, the operation fails and returns the error code **123**.

```
SS={BAS={/OU=Sales}, SF={ITEM=IPER}}
```

This example represents an further restriction of SS={BAS={/OU=Sales}}. Here the Proxy Control Owner can act as a proxy only for every inetOrgPerson entry in the subtree below /O=my-company/OU=sales.

## 3.2. X.500 Directory Operational Attributes

Directory operational attributes represent control information, such as access control information, or other information provided by the Directory, such as when an entry was modified. These attributes can be viewed and manipulated using **dirxcp**.

### 3.2.1. Attributes for Access Control

The following directory operational attributes are used to grant or deny users access to directory information.

#### 3.2.1.1. Access Control Scheme

The Access Control Scheme attribute specifies the means by which access to the Directory information and potentially to access rights themselves may be controlled within an access control-specific area. The attribute is placed in the administrative entry for the corresponding administrative point. Only administrative entries for access control-specific points contain the Access Control Scheme attribute.

The Access Control Scheme attribute can have the following values:

- **BACS** basic-access-control scheme
- **SACS** simplified-access-control scheme

##### 3.2.1.1.1. Abbreviation

ACS

##### 3.2.1.1.2. Syntax (DAP)

Object Identifier

##### 3.2.1.1.3. Example (DAP)

```
ACS=SACS
```

### 3.2.1.2. Entry ACI

The Entry ACI attribute specifies the local access control requirements applicable to a single entry. (See **ACI-Item** syntax for details.)

#### 3.2.1.2.1. Abbreviation

EACI

#### 3.2.1.2.2. Syntax (DAP)

ACI-Item

#### 3.2.1.2.3. Example (DAP)

```
EACI={ID=Authenticated user can Modify his passwords,
      PR=254,
      AL={BL={L=SIMPLE}},
      UF={UC={TH=TRUE},
          UP={PI={E=TRUE},
              GAD=grantModify};
          {PI={AT=UP,
              AAV=UP},
              GAD=grantAdd+grantRemove}}}
```

### 3.2.1.3. Prescriptive ACI

The Prescriptive ACI attribute is an attribute of a subentry that specifies access control information applicable to entries within a subentry's scope. (See **ACI-Item** syntax for details.)

#### 3.2.1.3.1. Abbreviation

PACI

#### 3.2.1.3.2. Syntax (DAP)

ACI-Item

#### 3.2.1.3.3. Example (DAP)

```
PACI={ID=Some authenticated users can Read all entries,
      PR=50,
      AL={BL={L=SIMPLE}},
```



```

UF={UC={N={DN={/O=My-Company/CN=admin}}};
      {DN={/O=My-Company/OU=Sales/CN=Mayer}};
      {DN={/O=My-Company/OU=Manufacturing/CN=Ketzer}};
      SUB={BAS={/O=My-Company/OU=Development}}},
UP={PI={E=TRUE},
    GAD=grantRead+grantBrowse+grantReturnDN};
    {PI={AUATV=TRUE},
    GAD=grantRead+grantCompare+grantFilterMatch};
    {PR=51,
    PI={AT=UP},
    GAD=denyRead+denyFilterMatch}}};
{ID=Authenticated users can Modify their passwords,
  PR=100,
  AL={BL={L=SIMPLE}}},
UF={UC={TH=TRUE},
    UP={PI={E=TRUE},
        GAD=grantModify};
        {PI={AT=UP,
            AAV=UP},
        GAD=grantAdd+grantRemove}}};
{ID=The administrator has additional permissions,
  PR=254,
  AL={BL={L=SIMPLE}}},
UF={UC={N={DN={/O=My-Company/CN=admin}}},
    UP={PI={E=TRUE},
        GAD=grantAdd+grantRemove+grantModify};
        {PI={AUATV=TRUE},
        GAD=grantAdd+grantRemove};
        {PI={AT=CRN;CRT;MN;MT},
        GAD=grantRead}}
}

```

### 3.2.1.4. Subentry ACI

The Subentry ACI attribute is an attribute of an administrative point that specifies access control information applicable to each of the subentries associated with the administrative point. (See **ACI-Item** syntax for details.)

#### 3.2.1.4.1. Abbreviation

SACI

#### 3.2.1.4.2. Syntax (DAP)

ACI-Item

#### 3.2.1.4.3. Example (DAP)

```
SACI={ID=admin: enable handling of subentries,  
      PR=254,  
      AL={BL={L=SIMPLE}},  
      UF={UC={N={DN={/O=My-Company/CN=admin}}},  
          UP={PI={E=TRUE},  
              GAD=grantAdd+grantRemove+grantModify};  
          {PI={AUATV=TRUE},  
              GAD=grantAdd+grantRemove};  
          {PI={AT=CRN;CRT;MN;MT},  
              GAD=grantRead}}  
      }
```

### 3.2.2. Attributes of Information Framework

The following directory operational attributes are used for controlling aspects of the DirX Directory information framework.

#### 3.2.2.1. Administrative Role

The Administrative Role attribute specifies the administrative role of an entry representing an administrative point; that is, the type of an administrative point. It is also used to regulate the subentries permitted to be subordinate to an administrative entry.

Valid values for the Administrative Role attribute are:

- **AA** autonomous area
- **ACSA** access control specific area
- **ACIA** access control inner area
- **CAIA** collective attribute inner area
- **CASA** collective attribute specific area
- **PASA** proxy authorization specific area
- **PPSA** password policy specific area
- **SASA** subschema administrative area

##### 3.2.2.1.1. Abbreviation

AR

#### 3.2.2.1.2. Syntax (DAP)

Object Identifier

#### 3.2.2.1.3. Example (DAP)

```
AR=AA  
{AR=AA;ACSA;SASA}
```

#### 3.2.2.2. Collective Exclusions

The Collective Exclusions attribute specifies the collective attribute types that should be excluded from an entry. The values are object identifiers for the collective attributes that are to be excluded. The value **XACA** (OID: **2.5.18.0**) excludes all collective attributes from an entry.

##### 3.2.2.2.1. Abbreviation

CE

##### 3.2.2.2.2. Syntax (DAP)

Object Identifier

##### 3.2.2.2.3. Example (DAP)

```
CE=XACA          excludes all collective attributes  
{CE=LC;TTIC;TXNC} excludes collective locality, ttx-terminal-  
                  identifier and telex-number
```

#### 3.2.2.3. Creators Name

The Creators Name attribute specifies the distinguished name of the Directory user that created an entry.

##### 3.2.2.3.1. Abbreviation

CRN

##### 3.2.2.3.2. Syntax (DAP)

Distinguished-Name

##### 3.2.2.3.3. Example (DAP)

```
CRN={/O=My-Company/CN=Administrator}
```

#### 3.2.2.4. Creation Time

The Creation Time attribute specifies the time that an entry was created.

##### 3.2.2.4.1. Abbreviation

CRT

##### 3.2.2.4.2. Syntax (DAP)

Generalized-Time

##### 3.2.2.4.3. Example (DAP)

```
CRT=19970101123000Z
```

#### 3.2.2.5. DirX Directory Entry UUID

The DirX Directory Entry UUID attribute specifies a unique identifier that the DirX Directory DSA creates at entry creation time.

##### 3.2.2.5.1. Abbreviation

DUUID

##### 3.2.2.5.2. Syntax (DAP)

PrintableString

##### 3.2.2.5.3. Example (DAP)

```
DUUID= 1b005b45-2ace-4de8-9ff1-b2b25b0b3d45
```

#### 3.2.2.6. Modifiers Name

The Modifiers Name attribute specifies the distinguished name of the Directory user that last modified an entry.

##### 3.2.2.6.1. Abbreviation

MN

##### 3.2.2.6.2. Syntax (DAP)

Distinguished-Name

##### 3.2.2.6.3. Example (DAP)

```
MN={/O=My-Company/CN=Administrator}
```

### 3.2.2.7. Modification Time

The Modification Time attribute specifies the time that an entry was last modified.

#### 3.2.2.7.1. Abbreviation

MT

#### 3.2.2.7.2. Syntax (DAP)

Generalized-Time

#### 3.2.2.7.3. Example (DAP)

```
MT=19970102123700Z
```

### 3.2.2.8. Number of Subordinates

The Number of Subordinates attribute specifies the number of immediate subordinates of an entry. This is a virtual attribute. It must not be used in search filters. If it is used in a search filter, an **Unwilling to Perform** error will be returned.

#### 3.2.2.8.1. Abbreviation

NOS

#### 3.2.2.8.2. LDAP Name(s)

numSubordinates

#### 3.2.2.8.3. Syntax (DAP)

Integer

#### 3.2.2.8.4. Example (DAP)

```
NOS=212
```

#### 3.2.2.8.5. Example (LDAP)

```
numSubordinates=212
```

### 3.2.2.9. Number of All Subordinates

The Number of All Subordinates attribute specifies the number of all subordinates of an entry. This is a virtual attribute. It must not be used in search filters. If it is used in a search filter, an **Unwilling to Perform** error will be returned.

#### 3.2.2.9.1. Abbreviation

NOAS

#### 3.2.2.9.2. LDAP Name(s)

numAllSubordinates

#### 3.2.2.9.3. Syntax (DAP)

Integer

#### 3.2.2.9.4. Example (DAP)

```
NOS=21234
```

#### 3.2.2.9.5. Example (LDAP)

```
numAllSubordinates=21234
```

### 3.2.2.10. Subtree Specification

The Subtree Specification attribute specifies subtrees. It is a mandatory attribute of a subentry. (See **Subtree-Specification** syntax for details.)

This attribute is not supported for LDAP protocol.

#### 3.2.2.10.1. Abbreviation

SS

#### 3.2.2.10.2. Syntax (DAP)

Subtree-Specification

#### 3.2.2.10.3. Example (DAP)

```
SS={DEF=TRUE}  
SS={BAS={/OU=Research/OU=Metallurgy Department}}  
SS={BAS={/OU=Research/OU=Metallurgy Department},SF={ITEM=ORP}}  
SS={SF={ITEM=2.5.6.7}}
```

### 3.2.3. Attributes for Schema Administration

The following directory operational attributes are used to control aspects of DirX Directory schema administration.

#### 3.2.3.1. Governing Structure Rule

The Governing Structure Rule attribute specifies the governing structure rule of an entry.

##### 3.2.3.1.1. Abbreviation

GSR

##### 3.2.3.1.2. Syntax (DAP)

Integer

##### 3.2.3.1.3. Example (DAP)

```
GSR=1
```

#### 3.2.3.2. Structural Object Class

The Structural Object Class attribute specifies the structural object class of an entry. This is a virtual attribute. It must not be used in search filters. If it is used in a search filter, an **Unwilling to Perform** error will be returned.

##### 3.2.3.2.1. Abbreviation

SOC

##### 3.2.3.2.2. Syntax (DAP)

Object Identifier

##### 3.2.3.2.3. Example (DAP)

```
SOC=ORP
```

### 3.2.4. Attributes for Database Configuration

The following directory operational attributes are used for controlling DirX Directory database configuration.

#### 3.2.4.1. Attribute Index

The Attribute Index attribute is a proprietary DirX Directory attribute that provides information about the actual index configuration of an attribute type. It is an attribute of

the database configuration subentry **CN=DB-Configuration-Subentry** and is used only for display purposes.

#### 3.2.4.1.1. Abbreviation

ATIDX

#### 3.2.4.1.2. Syntax (DAP)

Directory-String {32768}

#### 3.2.4.1.3. Example (DAP)

```
ATIDX= ( 2.5.4.3 NAME cn INDEX NO-PRESENT INITIAL FINAL NO-CONTAINS  
NO-UNIQUE )
```

### 3.2.4.2. Attribute Outsourcing

The Attribute Outsourcing attribute is a proprietary DirX Directory attribute that is used to configure outsourced attributes. It is an attribute of the database configuration subentry **CN=DB-Configuration-Subentry**. This attribute should not be changed directly via modify operations. Instead use the **dirxadm attr** operations. See the **attr** reference page in the *DirX Directory Administration Reference* for details.

#### 3.2.4.2.1. Abbreviation

ATOUT

#### 3.2.4.2.2. Syntax (DAP)

Directory-String {32768}

#### 3.2.4.2.3. Example (DAP)

```
ATOUT= (2.5.4.39 NAME certificateRevocationList OUTSOURCE )
```

## 3.2.5. Password Policy Attributes

In order to support the password policy procedures specified by the relevant password policy subentry, each user entry contains a set of operational attributes that provides information about the current status of the user's password. This set of attributes provides state information to support procedures for

- password checking
- password aging
- account lockout
- password administration



- storing login information

### 3.2.5.1. Attributes Supporting Password Checking

To support password-checking procedures for reusing old passwords, each user entry contains the Password History attribute. The following section describes this attribute.

#### 3.2.5.1.1. Password History Attribute

The Password History attribute is a multi-valued attribute saving the last recently used password values. The password values are stored in an internal format containing the password modification timestamp, the object identifier of the attribute, and the SHA-1 hashed password value.

This attribute is neither written to LDIF files nor replicated via DISP.

This attribute is maintained by the DSA. Read or write access should not be granted even to administrative users.

#### 3.2.5.1.2. Abbreviation

PPOHIST

#### 3.2.5.1.3. LDAP Name(s)

pwdHistory

#### 3.2.5.1.4. Syntax (DAP)

Octet String

#### 3.2.5.1.5. Example (DAP)

```
PPOHIST=20040401123024Z#2.5.4.35#+{SHA}+xser47sdwertrtz11==
```

#### 3.2.5.1.6. Example (LDAP)

```
pwdHistory=20040401123024Z#2.5.4.35#+{SHA}+xser47sdwertrtz11==
```

### 3.2.5.2. Attributes Supporting Password Aging

To support password-aging procedures for password expiration, each user entry contains the Password Changed Time, the Password Expiration Warned, and the Password Grace Use Time attributes. The following sections describe these attributes.

#### 3.2.5.2.1. Password Changed Time Attribute

The Password Changed Time attribute is a single-valued attribute that stores the timestamp at which the entry's password was changed.

This attribute is written to LDIF files and replicated via DISP.

It is maintained by the DSA during the modify operation. Read or write access should not be granted even to administrative users.

#### 3.2.5.2.2. Abbreviation

PPOCHTI

#### 3.2.5.2.3. LDAP Name(s)

pwdChangedTime

#### 3.2.5.2.4. Syntax (DAP)

Generalized-Time

#### 3.2.5.2.5. Syntax (LDAP)

Generalized-Time

#### 3.2.5.2.6. Example (DAP)

```
PPOCHTI=20020101123000Z
```

#### 3.2.5.2.7. Example (LDAP)

```
pwdChangedTime=20020101123000Z
```

#### 3.2.5.2.8. Password Expiration Warned

The Password Expiration Warned attribute is a single-valued attribute that saves the timestamp at which the password expiration warning was initially sent. The password expires Password Expire Warning seconds after the first warning has been sent. The Password Expire Warning attribute is an attribute of the relevant Password Policy subentry.

It is maintained by the DSA during the bind operation, by default on the local DSA only. Therefore, in a shadowing scenario, a user's entry may have different values for this password policy-related operational attribute on different DSAs.

Setting the environment variable **DIRX\_GLOBAL\_PPO\_STATES** to **1** directs the DSA to distribute and replicate changes to the Password Expiration Warned operational attribute to all DSAs in a shadowing scenario, creating a globally consistent password policy state for the user.

Read or write access should not be granted even to administrative users.

#### 3.2.5.2.9. Abbreviation

PPOEXWA

#### 3.2.5.2.10. LDAP Name(s)

pwdExpirationWarned

#### 3.2.5.2.11. Syntax (DAP)

Generalized-Time

#### 3.2.5.2.12. Syntax (LDAP)

Generalized-Time

#### 3.2.5.2.13. Example (DAP)

```
PPOEXWA=20020101123000Z
```

#### 3.2.5.2.14. Example (LDAP)

```
pwdExpirationWarned=20020101123000Z
```

#### 3.2.5.2.15. Password Grace Use Time

The Password Grace Use Time attribute is a multi-valued attribute saving the timestamps of grace-logins, which are logins that are granted even though an expired password has been used.

It is maintained by the DSA during the bind operation, by default on the local DSA only. Therefore, in a shadowing scenario, a user's entry may have different values for this password policy-related operational attribute on different DSAs.

Setting the environment variable **DIRX\_GLOBAL\_PPO\_STATES** to **1** directs the DSA to distribute and replicate changes to the Password Grace Use Time operational attribute to all DSAs of a shadowing scenario, creating a globally consistent password policy state for the user.

Read or write access should not be granted even to administrative users.

The Password Grace Login Limit attribute of the relevant Password Policy subentry specifies the maximum number of times an expired password can be used to authenticate.

#### 3.2.5.2.16. Abbreviation

PPOGRUT

#### 3.2.5.2.17. LDAP Name(s)

pwdGraceUseTime

#### 3.2.5.2.18. Syntax (DAP)

Generalized-Time

#### 3.2.5.2.19. Syntax (LDAP)

Generalized-Time

#### 3.2.5.2.20. Example (DAP)

```
PPOGRUT=20020101123000Z
```

#### 3.2.5.2.21. Example (LDAP)

```
{pwdGraceUseTime=20020101123000Z;20020102123000Z;20020103123000Z}
```

### 3.2.5.3. Attributes Supporting Account Lockout

To support password account lockout procedures, each user entry contains the Password Account Locked Time and the Password Failure Time attributes. The following sections describe these attributes.

#### 3.2.5.3.1. Password Account Locked Time Attribute

The Password Account Locked Time attribute is a single-valued attribute saving the timestamp the account was locked that is the password can no longer be used to authenticate.

The DSA locks an account during the bind operation if the lock conditions are met (specified in the attributes Password Lockout, Password Maximum Failure and Password Failure Counter Interval of the relevant Password Policy subentry). By default, the Account Locked Time is set on the local DSA only. Therefore, in a shadowing scenario, a user's entry may have different values for this password policy-related operational attribute on different DSAs.

Setting the environment variable **DIRX\_GLOBAL\_PPO\_STATES** to **1** directs the DSA to distribute and replicate changes to Password Account Locked Time to all DSAs in a shadowing scenario, creating a globally consistent account lock state for the user.

Usually the account is unlocked when performing a successful bind operation and at least Password Lockout Duration seconds has elapsed after the Password Account Locked Time. The Password Lockout Duration is an attribute of the relevant Password Policy subentry.

A value of **19700101000000Z** (in GMT format) indicates that the account has been locked permanently and only an administrator can unlock the account by deleting this attribute;

for example, with a **dirxcp** modify entry operation.

The administrator can lock an account by adding this attribute to a user's entry.

If the administrator has locked or unlocked an account on the master DSA of the user's entry, the value of this attribute is replicated; that is, the account is locked or unlocked on all DSAs participating in the directory system. If the account has been locked on a master or shadow entry while performing a bind or compare operation, the value of this attribute is not replicated; that is, the account is locked only on a single DSA.

Administrative users should be granted read and write permissions.

#### 3.2.5.3.2. Abbreviation

PPOALOT

#### 3.2.5.3.3. LDAP Name(s)

pwdAccountLockedTime

#### 3.2.5.3.4. Syntax (DAP)

Generalized-Time

#### 3.2.5.3.5. Syntax (LDAP)

Generalized-Time

#### 3.2.5.3.6. Example (DAP)

```
PPOALOT=20020101123000Z
```

#### 3.2.5.3.7. Example (LDAP)

```
pwdAccountLockedTime=20020101123000Z
```

#### 3.2.5.3.8. Password Failure Time

The Password Failure Time attribute is a multi-valued attribute saving the timestamps of consecutive authentication failures.

It is maintained by the DSA during the bind operation, by default on the local DSA only. Therefore, in a shadowing scenario, a user's entry may have different values for this password policy- related operational attribute on different DSAs.

Setting the environment variable **DIRX\_GLOBAL\_PPO\_STATES** to **1** directs the DSA to distribute and replicate changes to Password Failure Time to all DSAs in a shadowing scenario, creating a globally consistent password policy state for the user.

Read or write access should not be granted even to administrative users.

If the Password Lockout policy is enabled (the value of the Password Lockout attribute of the relevant Password Policy subentry is **TRUE**), the user's account is locked after Password Maximum Failure consecutive failed bind attempts have occurred within the period of time specified in Password Failure Counter Interval. The account is locked for Password Lockout Duration seconds. The Password Maximum Failure, Password Lockout and the Password Failure Counter Interval attribute are attributes of the relevant Password Policy subentry.

As long as an account is not locked:

- All Password Failure Time values are deleted when a successful bind occurs.
- All Password Failure Time values that are older than Password Failure Counter Interval are deleted when an unsuccessful bind occurs.

#### 3.2.5.3.9. Abbreviation

PPOFATI

#### 3.2.5.3.10. LDAP Name(s)

pwdFailureTime

#### 3.2.5.3.11. Syntax (DAP)

Generalized-Time

#### 3.2.5.3.12. Syntax (LDAP)

Generalized-Time

#### 3.2.5.3.13. Example (DAP)

```
{PPOFATI=200201011123000Z;200201011124000Z}
```

#### 3.2.5.3.14. Example (LDAP)

```
{pwdFailureTime=200201011123000Z;200201011124000Z}
```

### 3.2.5.4. Attributes Supporting Password Administration

To support password administration procedures each user entry contains the Password Reset attribute. The following sections describe this attribute.

#### 3.2.5.4.1. Password Reset Attribute

The Password Reset attribute is a single-valued attribute specifying whether (value **TRUE**) or not (value **FALSE**) a user must change his password after he has performed a successful

bind operation.

An administrator usually forces a user to change his password by modifying or adding this attribute with the value **TRUE** together with resetting the user's password; for example, because the user has forgotten his password.

The user may use the re-set password to authenticate during the bind operation. However, all operations except a modify password operation are prohibited. The DSA rejects these operations with the error code **Unwilling to perform** and the error message **Password Modification is the only operation allowed due to Password Policy**. Perform the following **dirxcp** operation to modify the password:

```
obj modify [distinguished_name] [-bindid bid]
  -replaceattr {userPassword=new_user_password}
```

For example:

```
obj modify {cn=mueLLer peter,ou=board,o=my-company} \
  -replaceattr {userPassword=xWersT}
```

The Password Safe Modify (PPSAMO, pwdSafeModify) attribute must also be **FALSE**. It is not possible to use another kind of modify operation to change the user password.

#### 3.2.5.4.2. Abbreviation

PPORSET

#### 3.2.5.4.3. LDAP Name(s)

pwdReset

#### 3.2.5.4.4. Syntax (DAP)

Boolean

#### 3.2.5.4.5. Syntax (LDAP)

Boolean

#### 3.2.5.4.6. Example (DAP)

```
PPORSET=TRUE
```

#### 3.2.5.4.7. Example (LDAP)

```
pwdReset=TRUE
```

### 3.2.5.5. Attributes Supporting Storing of Login Information

To support the storing of login information, each user entry contains the Password Number of Failed Logins and the Password Last Login Time operational attributes. The following sections describe these attributes.

#### 3.2.5.5.1. Password Number of Failed Logins Attribute

The Password Number of Failed Logins attribute is a single-valued attribute recording the number of unsuccessful bind operations. The value of this attribute is incremented when a bind operation fails due to an **Invalid Credentials** error if the password policy Password Store Login Info (pwdStoreloginInfo) attribute has a value of **1** or **3**.

Disabling this feature by setting the Password Store Login Info attribute in the relevant password policy subentry to a value of **0** or **2** does not delete or modify the counter.

The purpose of this attribute is only informational; it is not evaluated by any password policy procedure. The scope of this operational attribute is local; that is, the value of the Password Number of Failed Logins attribute in a replicated environment represents the number of failed logins at the local DSA. This attribute is independent from the setting of the environment variable **DIRX\_GLOBAL\_PPO\_STATES**.

Administrative users should be granted read permissions.

#### 3.2.5.5.2. Abbreviation

PPONOFLL

#### 3.2.5.5.3. LDAP Name(s)

pwdNumberOfFailedLogins

#### 3.2.5.5.4. Syntax (DAP)

Integer

#### 3.2.5.5.5. Syntax (LDAP)

Integer

#### 3.2.5.5.6. Example (DAP)

```
PPONOFLL=20
```

#### 3.2.5.5.7. Example (LDAP)

```
pwdNumberOfFailedLogins=20
```



#### 3.2.5.5.8. Password Last Login Time Attribute

The Password Last Login Time attribute is a single-valued attribute recording the time of the most recent successful bind operation. The value of this attribute is set as a result of a successful bind operation if the password policy Password Store Login Info (pwdStoreloginInfo) attribute has a value of **2** or **3**.

Disabling the feature by setting the Password Store Login Info attribute in the relevant password policy subentry to a value of **0** or **1** does not delete or modify the value of this operational attribute.

The purpose of this attribute is only informational; it is not evaluated by any password policy procedure. The scope of this operational attribute is local; that is, the value of the Password Last Login Time attribute in a replicated environment represents the time of the most recent successful bind to the local DSA. This attribute is independent from the setting of the environment variable **DIRX\_GLOBAL\_PPO\_STATES**.

Administrative users should be granted read permissions.

#### 3.2.5.5.9. Abbreviation

PPOLL0T

#### 3.2.5.5.10. LDAP Name(s)

pwdLastLoginTime

#### 3.2.5.5.11. Syntax (DAP)

Generalized-Time

#### 3.2.5.5.12. Syntax (LDAP)

Generalized-Time

#### 3.2.5.5.13. Example (DAP)

```
PPOLL0T=20130401123000Z
```

#### 3.2.5.5.14. Example (LDAP)

```
pwdLastLoginTime=20130401123000Z
```

### 3.2.6. Attributes for Synchronous Shadowing

The following operational attributes are used by a DSA when it is part of a synchronous shadowing configuration. For more information about synchronous shadowing, see the **dirxadm sob create** operation in the *DirX Directory Administration Reference* and the chapter “Creating a Synchronous Shadow DSA” in the *DirX Directory Administration Guide*.

### 3.2.6.1. DirX Directory Backup Type

The DirX Directory Backup Type attribute is generated or modified by the supplier DSA when a total update by media is configured and a database backup is started. The attribute value contains the name of the backup type which was initiated. The administrator can choose between two database backup types:

- **Dirxbackup**, which requires binary compatibility between the supplier DSA and the consumer DSA.
- **ldif\_dump**, which dumps the database in LDIF format with **dirxadm** and does not require binary compatibility between the supplier DSA and the consumer DSA.

This attribute is part of the backup signature. See “Creating a Synchronous Shadow DSA” -> “Performing a Total Update by Media” in the *DirX Directory Administration Guide* for more information on this topic.

#### 3.2.6.1.1. Abbreviation

DRXBTYPE

#### 3.2.6.1.2. Name (LDAP)

dirxBackupType

#### 3.2.6.1.3. Syntax

Directory String

#### 3.2.6.1.4. Example (DAP)

```
DRXBTYPE=dirxbackup
```

#### 3.2.6.1.5. Example (LDAP)

```
dirxBackupType=ldif_dump
```

### 3.2.6.2. DirX Directory Backup Time

The DirX Directory Backup Time attribute is generated or modified by the supplier DSA when a total update by media is configured and a backup is started. It contains the time stamp of the start of the backup.

This attribute is part of the backup signature. See “Creating a Synchronous Shadow DSA” -> “Performing total update by media” in the *DirX Directory Administration Guide* for more information on this topic.

#### 3.2.6.2.1. Abbreviation

DRXBTIME

#### 3.2.6.2.2. Name (LDAP)

dirxBakupTime

#### 3.2.6.2.3. Syntax

Generalized-Time

#### 3.2.6.2.4. Example (DAP)

```
DRXBTIME=20170328102028.374Z
```

#### 3.2.6.2.5. Example (LDAP)

```
dirxBakupTime=20170328102028.374Z
```

### 3.2.6.3. DirX Directory Backup MSN

The DirX Directory Backup MSN attribute is generated respectively modified by the supplier DSA when a total update by media is configured and a backup is started and contains the value of attribute DirX Directory Recent MSN at the time of the start of the backup.

This attribute is part of the backup signature. See “Creating a Synchronous Shadow DSA” -> “Performing total update by media” in the *DirX Directory Administration Guide* for more information on this topic.

#### 3.2.6.3.1. Abbreviation

DRXBMSN

#### 3.2.6.3.2. Name (LDAP)

dirxBakupMsn

#### 3.2.6.3.3. Syntax

Numeric-String

#### 3.2.6.3.4. Example (DAP)

```
DRXBMSN=28560
```

#### 3.2.6.3.5. Example (LDAP)

```
dirxBackupMsn=28650
```

#### 3.2.6.4. DirX Directory Backup DSAName

The DirX Directory Backup DSAName attribute is generated or modified by the supplier DSA when a total update by media is configured and a backup is started. It contains the name of the supplier DSA as specified in the environment variable DIRX\_DSA\_NAME.

This attribute is part of the backup signature. See “Creating a Synchronous Shadow DSA” -> “Performing total update by media” in the *DirX Directory Administration Guide* for more information on this topic.

##### 3.2.6.4.1. Abbreviation

DRXBDSAN

##### 3.2.6.4.2. Name (LDAP)

dirxBackupDsaName

##### 3.2.6.4.3. Syntax

Directory-String

##### 3.2.6.4.4. Example (DAP)

```
DRXDSAN=cn=DirX-DSA-myhost
```

##### 3.2.6.4.5. Example (LDAP)

```
dirxBackupDsaName=cn=DirX-DSA-myhost
```

#### 3.2.6.5. DirX Directory Entry MSN

The DirX Directory Entry MSN attribute specifies the modification sequence number (MSN) of the last update operation of the DSE. It is an attribute of the DSE. The value of this attribute is equal to the value of the DirX Directory Recent MSN attribute at the time of the update operation.

##### 3.2.6.5.1. Abbreviation

EMSN

#### 3.2.6.5.2. Name (LDAP)

dirxEntryMSN

#### 3.2.6.5.3. Syntax (LDAP)

Numeric String

#### 3.2.6.5.4. Example (LDAP)

```
EMSN=28560
```

### 3.2.6.6. DirX Directory In Sync

The DirX Directory In Sync (CINS) attribute shows the status of data synchronicity in a synchronous shadowing configuration. It is an attribute of the subentry **/CN=DirXDBVersionSubentry**. On the master DSA, the value of this attribute is always **TRUE**. On a consumer DSA, the attribute can have a value of **FALSE** if the shadowing agreement has been disabled, a total update is running, or an error has occurred. In this case, you can use the **dirxadm sob show** command with the **-supplier** option on the supplier DSA to display more information about the status of the shadowing agreement. See the **dirxadm** section in the *DirX Directory Administration Reference* for details on the **sob show** command.

#### 3.2.6.6.1. Abbreviation

CINS

#### 3.2.6.6.2. Name (LDAP)

dirxInSync

#### 3.2.6.6.3. Syntax (LDAP)

Boolean

#### 3.2.6.6.4. Example (LDAP)

```
CINS=FALSE
```

### 3.2.6.7. DirX Directory Recent DN

The DirX Directory Recent DN attribute shows the distinguished name (DN) of the last update operation. It is an attribute of the subentry **CN=DirXDBVersionSubentry**.

#### 3.2.6.7.1. Abbreviation

RECDN

#### 3.2.6.7.2. Name (LDAP)

dirxRecentDN

#### 3.2.6.7.3. Syntax (LDAP)

Distinguished Name (LDAP style)

#### 3.2.6.7.4. Example (LDAP)

```
RECDN=/O=My-Company/OU=Sales/CN=Abele
```

### 3.2.6.8. DirX Directory Recent MSN

The DirX Directory Recent MSN (RSMN) attribute specifies a modification sequence number (MSN) that uniquely identifies the version of a DSA database. It is an attribute of the subentry **/CN=DirXDBVersionSubentry**. The master DSA increments the DirX Directory Recent MSN attribute each time it performs an update operation and stores the value in the local subentry. At each incremental refresh, the master DSA propagates the value of the DirX Directory Recent MSN attribute to all synchronous shadow consumer DSAs.

#### 3.2.6.8.1. Abbreviation

RMSN

#### 3.2.6.8.2. Name (LDAP)

dirxRecentMSN

#### 3.2.6.8.3. Syntax (LDAP)

Numeric string

#### 3.2.6.8.4. Example (LDAP)

```
RMSN=28560
```

### 3.2.6.9. DirX Directory Recent MSN Time Stamp

The DirX Directory Recent MSN Time Stamp attribute holds the last modification time stamp of the last update operation. It is an attribute of the subentry **/CN=DirXDBVersionSubentry**.

#### 3.2.6.9.1. Abbreviation

RMSNTS

#### 3.2.6.9.2. Name (LDAP)

dirxRecentMSNTimeStamp

#### 3.2.6.9.3. Syntax (LDAP)

Numeric String

#### 3.2.6.9.4. Example (LDAP)

```
RMSN=28560
```

### 3.2.6.10. DirX Directory Recent Operation

The DirX Directory Recent Operation attribute shows the type of request of the last update operation. It is an attribute of the subentry **/CN=DirXDBVersionSubentry**.

#### 3.2.6.10.1. Abbreviation

RECOP

#### 3.2.6.10.2. Name (LDAP)

dirxRecentOperation

#### 3.2.6.10.3. Syntax (LDAP)

Directory String {32768}

#### 3.2.6.10.4. Example (LDAP)

```
RECOP=modify
```

## 3.2.7. Attributes for Nested and Dynamic Groups

The following operational attributes can be used by the administrator for nested and dynamic group management.

### 3.2.7.1. DirX Directory Member

The DirX Directory Member attribute can be requested in search operations. It is used to show all members of a groupOfNames object and of a nested group object (structural object class groupOfNames plus auxiliary class dirxImportGroup), including all the members that come from the imported groups. It is also used to show all members from a dynamic group object (structural object class dirxGroupOfUrls). The attribute can also be specified in a search filter and is used for matching in nested groups and dynamic groups.

#### 3.2.7.1.1. Abbreviation

DRXMEM

#### 3.2.7.1.2. LDAP Name(s)

dirxMember

#### 3.2.7.1.3. Syntax (DAP)

Distinguished Name

#### 3.2.7.1.4. Syntax (LDAP)

Distinguished Name (LDAP style)

#### 3.2.7.1.5. Example (DAP)

```
DRXMEM={/O=My-Company/CN=Admin}
```

#### 3.2.7.1.6. Example (LDAP)

```
dirxMember= cn=Admin ,o=My-Company
```

### 3.2.7.2. DirX Directory Member Of

The DirX Directory Member Of attribute can be requested in search operations. It shows the DNs of groupOfNames entries in which the resulting entry is a direct member of the group or an indirect member through its inclusion in a nested group or a member of a dynamic group. The attribute can also be specified in a search filter and returns the DNs of the nested and dynamic groups of which the entry is a member.

#### 3.2.7.2.1. Abbreviation

DRXMEMOF

#### 3.2.7.2.2. LDAP Name(s)

dirxMemberOf

#### 3.2.7.2.3. Syntax (DAP)

Distinguished Name

#### 3.2.7.2.4. Syntax (LDAP)

Distinguished Name (LDAP style)



#### 3.2.7.2.5. Example (DAP)

```
DRXMEMOF={/O=My-Company/CN=AdminGroup}
```

#### 3.2.7.2.6. Example (LDAP)

```
dirxMemberOf= cn=AdminGroup,o=My-Company
```

### 3.2.7.3. DirX Directory Member Url

The DirX Directory MemberUrl attribute specifies an LDAP search expression. It is used to define a dynamic group.

#### 3.2.7.3.1. Abbreviation

DRXMURL

#### 3.2.7.3.2. LDAP Name(s)

dirxMemberUrl

#### 3.2.7.3.3. Syntax (DAP)

Directory-String, format DirX Directory-Member-Url

#### 3.2.7.3.4. Syntax (LDAP)

Directory-String, format DirX Directory-Member-Url

#### 3.2.7.3.5. Example (DAP)

```
DRXMURL=ldap:///o=My-Company??sub?(&(l=sunnyvale)(title=senior*))
```

#### 3.2.7.3.6. Example (LDAP)

```
dirxMemberUrl=ldap:///o=My-  
Company??sub?(&(l=sunnyvale)(title=senior*))
```

### 3.2.7.4. DirX Directory Imported Groups

The DirX Directory Imported Groups attribute can be requested in search operations. It shows the DNs of all imported groups of a nested group entry.

#### 3.2.7.4.1. Abbreviation

IMPGROUP

#### 3.2.7.4.2. LDAP Name(s)

dirxImportedGroups

#### 3.2.7.4.3. Syntax (DAP)

Directory String

#### 3.2.7.4.4. Syntax (LDAP)

Directory String (LDAP Style)

#### 3.2.7.4.5. Example (DAP)

```
Imported-Groups : 1- cn=Employees1,o=My-Company
                  : 2-- cn=SalesGroup,ou=Sales,o=My-Company
                  : 2-- cn=DevelopmentGroup,ou=Development,o=My-Company
```

#### 3.2.7.4.6. Example (LDAP)

```
dirxImportedGroups : 1- cn=Employees1,o=My-Company
                    : 2-- cn=SalesGroup,ou=Sales,o=My-Company
                    : 2-- cn=DevelopmentGroup,ou=Development,o=My-
Company
```

### 3.2.7.5. DirX Directory Export To

The DirX Directory Export To attribute is automatically maintained by the DSA in an imported group and refers to the nested group entry that imports the group.

#### 3.2.7.5.1. Abbreviation

EXPTO

#### 3.2.7.5.2. LDAP Name(s)

dirxExportTo

#### 3.2.7.5.3. Syntax (DAP)

Distinguished Name

#### 3.2.7.5.4. Syntax (LDAP)

Distinguished Name (LDAP style)

#### 3.2.7.5.5. Example (DAP)

```
EXPT0={/O=My-Company/CN=UserGroups}
```

#### 3.2.7.5.6. Example (LDAP)

```
dirxExportTo=cn=UserGroups,o=My-Company
```

## 3.3. X.500 Distributed Operational Attributes

Operational attributes are attributes representing operational and/or administrative information. Distributed operational attributes are used by several DSAs as part of distributed operations; for example, knowledge reference attributes.

### 3.3.1. Knowledge Reference Attributes

This section describes distributed operational attributes related to knowledge references.

#### 3.3.1.1. Non Specific Knowledge

The Non Specific Knowledge attribute specifies access points for the master DSAs of one or more naming contexts and/or shadow DSAs for the same or more naming contexts. The context prefixes of the naming context(s) is (are) not known. The immediate superior is known and the access point information is associated with its name. (See **Master-And-Shadow-Access-Points** syntax for details.) The attribute is multi-valued and managed by the DSA itself. It is accessible only by using **dirxadm**.

The attribute is held in a DSE of type **NSSR** and represents non-specific subordinate references.

The DSA may use the attribute information when constructing a continuation reference returned in a DAP or DSP referral, when performing chaining, and when constructing shadowed DSA specific entries of type **NSSR** provided in the DISP.

Non-specific subordinate references are not supported by the DirX Directory DSAs.

##### 3.3.1.1.1. Abbreviation

NSK

##### 3.3.1.1.2. Syntax (DAP)

Master-And-Shadow-Access-Point

### 3.3.1.1.3. Example (DAP)

```
NSK={MOS={MSAP={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa1},
                PSAP={TS=Server19,
                      NA='TCP/IP!internet=123.45.67.89+port=102'}}
```

### 3.3.1.2. Specific Knowledge

The Specific Knowledge attribute specifies access points for the master DSAs of a naming context and/or shadow DSAs for that naming context. The context prefix of the naming context is known and associated with the access point information. (See **Master-And-Shadow-Access-Points** syntax for details.) The attribute is single-valued. It is accessible only by using **dirxadm**.

The attribute is held in a DSE of type **SUBR** (representing a specific subordinate reference), **IMM\_SUPR** (representing a immediate superior reference), or **XR** (representing a cross reference).

The DSA may use the attribute information when constructing a continuation reference returned in a DAP or DSP referral, when performing chaining, and when constructing shadowed DSA specific entries of type **SUBR**, **IMM\_SUPR**, or **XR** provided in the DISP.

To return LDAP referrals to LDAP clients, the AE component can contain the value of the special naming attribute LDAP URL (abbreviation LURL and syntax Directory-String {32768}). The referral to another LDAP server is returned as LDAP URL. In this case, the presentation address (PSAP) is ignored.

#### 3.3.1.2.1. Abbreviation

SK

#### 3.3.1.2.2. Syntax (DAP)

Master-And-Shadow-Access-Point

#### 3.3.1.2.3. Example (DAP)

```
SK={MOS={MSAP={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa1},
                PSAP={TS=Server19,
                      NA='TCP/IP!internet=123.45.67.89+port=102'}}
```

```

    {MSAP={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa2},
      PSAP={TS=Server19,
        NA='TCP/IP!internet=139.88.99.1+port=102'}}
```

```

    }
  SK={MOS={MSAP={AE={/LURL=ldap://\ldapservice.dec.de:389},
    PSAP={TS=X,NA='TCP/IP!internet=127.0.0.1+port=1'}}}}
```

## 3.4. DirX Directory DSA Operational Attributes

Operational attributes are attributes representing operational and/or administrative information. DSA operational attributes are used only locally to the DSA.

### 3.4.1. Authentication

This section describes DirX Directory DSA operational attributes related to authentication.

#### 3.4.1.1. DirX Directory-Administrators

The DirX Directory-Administrators attribute specifies the distinguished names of the users that the DSA will recognize as an authorized user of **dirxadm**. It is an attribute of the root DSE. (See the **dse bind** operation of the **dse (dirxadm)** section in the *DirX Directory Administration Reference* for details.)

Because this attribute is only accessible by using **dirxadm**, no access using LDAP or DAP protocol is possible.

##### 3.4.1.1.1. Abbreviation

DADM

##### 3.4.1.1.2. Syntax

Distinguished-Name

##### 3.4.1.1.3. Example (dirxadm)

```

DADM={/O=My-Company/CN=Administrator}; \
      {/O=NSA/CN=NSA-Administrator}
```

### 3.4.2. Knowledge Attributes

This section describes DirX Directory DSA operational attributes related to replication configurations.

### 3.4.2.1. Consumer Knowledge

The Consumer Knowledge attribute specifies information about shadow consumers for a particular naming context held by a shadow supplier DSA. It is a multi-valued attribute associated with the context prefix of a naming context. The information identifies the access points of the consumer DSAs and also the identifiers for the corresponding shadowing agreements. (See **Consumer-Information** syntax for details.)

All shadow supplier DSAs contain a value of this attribute for each shadowing agreement in which they act as a supplier.

The **AGR** (agreement id) component of the attribute is required in all DISP operations.

#### 3.4.2.1.1. Abbreviation

CK

#### 3.4.2.1.2. Syntax (DAP)

Consumer-Information

#### 3.4.2.1.3. Example (DAP)

```
CK={CON={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa},  
  
PSAP={TS=Server19,NA='TCP/IP!internet=123.45.67.89+port=102'},  
      AGR={ID=10,VERS=20}  
}
```

### 3.4.2.2. Superior Knowledge

The Superior Knowledge attribute is used by non-first level DSAs to specify its superior reference. It is a single-valued attribute of the root-DSE and accessible only by using **dirxadm**. All non-first level DSAs shall hold this attribute. The attribute is managed by the DSA itself. (See **Access-Point** syntax for details.)

The DSA may use the attribute information when constructing a continuation reference returned in a DAP or DSP referral or when performing chaining.

#### 3.4.2.2.1. Abbreviation

SPK

#### 3.4.2.2.2. Syntax

Access-Point

#### 3.4.2.2.3. Example (dirxadm)

```
SPK={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa},  
      PSAP={TS=Server19,NA='TCP/IP!internet=123.45.67.89+port=102'}}
```

#### 3.4.2.3. Supplier Knowledge

The Supplier Knowledge attribute specifies information about the shadow supplier DSA in a shadow agreement for a particular naming context held by a shadow consumer DSA. It is a multi-valued attribute associated with the context prefix of a shadowed naming context. The attribute consists of the ID and version number that identify the shadow agreement between the supplier and the consumer, the access point for the supplier of the copy of the naming context, and an indication of whether this supplier holds the master copy of the naming context. In addition, the information can optionally include the access point for the master if the supplier does not hold the master copy. (See **Supplier-Information** syntax for details.)

All shadow consumer DSAs hold a value of this attribute for each shadowing agreement in which they act as a consumer.

The DSA may use the attribute information when constructing a continuation reference returned in a DAP or DSP referral.

The **AGR** (agreement id) component of the attribute is required in all DISP operations.

##### 3.4.2.3.1. Abbreviation

SUK

##### 3.4.2.3.2. Syntax (DAP)

Supplier-Information

##### 3.4.2.3.3. Example (DAP)

```
SUK={SUP={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa1},  
          PSAP={TS=Server19,  
                NA='TCP/IP!internet=123.45.67.89+port=102' }},  
      AGR={ID=1,  
          VERS=1},  
      ISM=TRUE,  
      NSM={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa2},  
          PSAP={TS=Server19,  
                NA='TCP/IP!internet=139.88.99.1+port=102' }},  
          {AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa3},  
          PSAP={TS=Server19,
```

```
}
    NA= ' TCP/IP!internet=139.88.99.2+port=102' }}
```

#### 3.4.2.4. Secondary Shadows

The Secondary Shadows attribute specifies the suppliers of secondary shadows and their consumers. It is a multi-valued attribute associated with the context prefix of a shadowed naming context. The attribute consists of the access point of a shadow supplier and a list of its direct consumers. (See **Supplier-And-Consumers** syntax for details.)

##### 3.4.2.4.1. Abbreviation

SES

##### 3.4.2.4.2. Syntax (DAP)

Supplier-And-Consumers

##### 3.4.2.4.3. Example (DAP)

```
SES={SUP={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa1},
    PSAP={TS=Server19,
        NA= ' TCP/IP!internet=123.45.67.89+port=102' }},
    CON={AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa2},
        PSAP={TS=Server19,
            NA= ' TCP/IP!internet=139.88.99.2+port=102' }},
    {AE={/O=my-company/OU=asw/CN=dsa/CN=sni-dsa3},
        PSAP={TS=Server19,
            NA= ' TCP/IP!internet=139.88.99.3+port=102' }}}
}
```

### 3.4.3. Policy Attributes

This section describes DirX Directory DSA operational attributes for establishing policies.

#### 3.4.3.1. Audit Policy

The Audit Policy attribute specifies the auditing of DirX Directory transactions. It is an attribute of the root-DSE and is managed by the **audit (dirxadm)** command. (See the **audit operation** in the *DirX Directory Administration Reference* for details.)

Because this attribute is only accessible by using **dirxadm**, no access using LDAP and DAP protocols is possible.



#### 3.4.3.1.1. Abbreviation

AUDP

#### 3.4.3.1.2. Syntax

Audit-Info

#### 3.4.3.1.3. Example (dirxadm)

See the **audit (dirxadm)** command in the *DirX Directory Administration Reference*.

### 3.4.3.2. DSA Policy

The DSA Policy attribute specifies information that defines bilateral agreements between the local DSA and other remote DSAs known to it. It is an attribute of the root DSE and is managed by using **dirxadm**. It supplies information about the authentication procedures for DSP and DISP operations. The attribute is multi-valued. Each value is assigned to a specific DSA or / and contains one or more policy options. (See **DSA-Policies** syntax for details.)

Because this attribute is only accessible by using **dirxadm**, no access using LDAP and DAP protocols is possible.

#### 3.4.3.2.1. Abbreviation

DSAP

#### 3.4.3.2.2. Syntax

DSA-Policies

#### 3.4.3.2.3. Example (dirxadm)

**Error! Cannot open file.**

### 3.4.3.3. Local Scope Policy

The Local Scope Policy attribute specifies the interpretation of the local-scope service control. It is a single-valued attribute of the root-DSE and is managed by using **dirxadm**. (See **Local Scope Policy** syntax for details.)

Because this attribute is only accessible by using **dirxadm** no access using LDAP and DAP protocols is possible.

Valid values for the Local Scope Policy attribute are:

- **DSA** - The local scope is the scope of the DSA. The DSA responds on the basis of information that it holds itself, and does not chain or create referrals or continuation references
- **LOCAL-DOMAIN** - The local scope is the scope of the local domain of the DSA. The DSA chains or creates referrals or continuation references only to the group of DSAs in the

local domain. A DSA belongs to the same local domain as the local DSA if a DSA policy exists for this DSA with the value **OPT=LOCAL-DOMAIN**

- **DMD** - The local scope is the scope of the Directory Management Domain (DMD) of the DSA. The DSA will chain or create referrals or continuation references only within the group of DSAs configured within the DMD. A DSA belongs to the same DMD as the local DSA if a DSA policy exists for this DSA with the value **OPT=DMD**

The default value is **DSA**.

#### 3.4.3.3.1. Abbreviation

LSP

#### 3.4.3.3.2. Syntax

Local Scope Policy

#### 3.4.3.3.3. Example (dirxadm)

```
LSP=DMD
LSP=DSA
LSP=LOCAL-DOMAIN
```

#### 3.4.3.4. Paging Policy

The Paging Policy attribute specifies the configuration parameters for paged results of search requests. It is an attribute of the root DSE and is managed by using **dirxadm**. The attribute is single-valued. It is valid for all search operations. (See **Paging-Policies** syntax for details.)

Because this attribute is only accessible by using **dirxadm**, no access using LDAP and DAP protocols is possible.

When you want to use simple paging with your client, the administrator should set this global policy.

When one of the specified limits has been exceeded, the DSA returns the error **Unwilling To Perform**.

#### 3.4.3.4.1. Abbreviation

PAGP

#### 3.4.3.4.2. Syntax

Paging-Policies

#### 3.4.3.4.3. Example (dirxadm)

In the following example, resume information kept for paged results is released when no subsequent request has been received for a period of 180 seconds.

Search operations are rejected with the error "Unwilling To Perform" when

- a search operation containing a paged result request requires to allocate resume information while resume information is already kept for 15 other search operations
- the resume information exceeds the maximum amount of 2 MB

```
PAGP={PTO=180,  
      MAXNUM=15,  
      MAXMEM=2048}
```

#### 3.4.3.5. Partial Entry Shadowing Policy

The Partial Entry Shadowing Policy attribute specifies the behavior of the DSA when performing DISP protocol on supplier / initiator side; that is, when generating update information messages for shadowing or when writing LDIF files.

If this attribute is not specified or set to **FALSE**, the total update message segments contain only complete entries. If there is a huge entry to be shadowed or to be written to an LDIF file, for example, an entry that contains a group with huge number of members, it may happen that the system cannot provide enough memory to prepare the update segment and shadowing or writing an LDIF file fails.

If this attribute is set to **TRUE**, the DSA is able to process partial entries and generate segmented messages or LDIF file records that contain only a part of an entry.

An administrator may set this attribute to **TRUE** when all the following conditions are met:

- The DSA acts as a shadow supplier or writes LDIF content files.
- Total update is performed by sending protocol messages and not done by media.
- The DSA contains huge entries that cannot be loaded from the database to the memory as a whole; for example, entries that contain groups with a huge number of members.
- An administrator should set the PESHA attribute to **TRUE** *only* when failing TOTAL updates occur because of memory problems.

DSAs providing this policy cannot interoperate with DSAs older than DirX Directory EE V2.0C00 or DirX Directory V7.0B00 or with other vendor implementations. When running your DSA in such interoperation scenarios, you must set this attribute to **FALSE**.

This attribute is an attribute of the root-DSE and is managed by using **dirxadm**. The attribute is single-valued.

Because this attribute is only accessible by using **dirxadm**, no access using LDAP and DAP protocols is possible.

#### 3.4.3.5.1. Abbreviation

PESHA

#### 3.4.3.5.2. Syntax

Boolean

#### 3.4.3.5.3. Example (dirxadm)

```
PESHA=TRUE
```

#### 3.4.3.6. Search Base Policy

The Search Base Policy attribute specifies the topmost level in the DIT where a search operation can be started. It is an attribute of the root DSE and is managed by using **dirxadm**.

DAP and DSP search requests for subtrees are rejected with the service error `administrativeLimitExceeded` if the base object contains fewer RDNs than specified in the Search Base Policy attribute.

Because this attribute is only accessible by using **dirxadm**, no access using LDAP and DAP protocols is possible.

##### 3.4.3.6.1. Abbreviation

SBP

##### 3.4.3.6.2. Syntax

Integer

##### 3.4.3.6.3. Example (dirxadm)

```
SBP=2    DSA rejects all search operations started from root and  
country  
        level
```

#### 3.4.3.7. Time Limit Policy

The Time Limit Policy attribute specifies the maximum time limit in seconds for search operations that is accepted by the DSA in user requests. It is an attribute of the root DSE and is managed by using **dirxadm**.

If the time limit is exceeded, an incomplete result is returned.

Because this attribute is only accessible by using **dirxadm**, no access using LDAP and DAP protocols is possible.

#### 3.4.3.7.1. Abbreviation

TLP

#### 3.4.3.7.2. Syntax

Integer

#### 3.4.3.7.3. Example (dirxadm)

```
TLP=25
```

### 3.4.3.8. User Policy

The User Policy attribute specifies policies for DAP users, discriminating between them by name. It is an attribute of the root DSE and is managed by using **dirxadm**. It supplies information about the user's required authentication method, whether the users can access the chaining and modify services of the DSA, the maximum number of entries of a retrieval result, and the maximum length of attribute values returned in a retrieval result. The attribute is multi-valued and each value is assigned to a specific user or a group of users. (See **User-Policies** syntax for details.)

Because this attribute is only accessible by using **dirxadm**, no access using LDAP and DAP protocols is possible.

#### 3.4.3.8.1. Abbreviation

USP

#### 3.4.3.8.2. Syntax

User-Policies

#### 3.4.3.8.3. Example (dirxadm)

```
USP={USN={/O=my-company/OU=asw},
      OPT=CHAINING+MODIFY,
      AP={AMB={AM=SIMPLE;SIMPLE-WITH-PWD},
          OPWD=my-pwd,
          CV=300}};
{USN={/O=my-company/OU=com},
  OPT=CHAINING,
  AP={AMB={AM=SIMPLE},
      OPWD=pwd}}
}
```

### 3.4.4. Miscellaneous Attributes

This section describes miscellaneous DirX Directory DSA operational attributes.

#### 3.4.4.1. Cooperating DSA

The Cooperating DSA attribute specifies information about the administered operational bindings with partner DSAs. It is an attribute of a subentry of the root-DSE and is used only for display purposes. The name of the subentry is **/CN=Cooperating-DSAs-Subentry**. The attribute value should not be modified by the user. It can be accessed by **dirxadm** only. Use the **dirxadm lob** and **sob show** operation to display the value. The **dirxadm lob** and **sob** functions must be used to create, modify, or delete the information displayed by this attribute. (See **Cooperating-DSA** syntax for details.)

The master and supplier DSA of the cooperating DSAs subentry is the DSA that is displayed in the Supplier Knowledge (SUK) attribute of the agreement propagating the cooperating DSA subentry to all participating DSAs. This agreement is created and administrated automatically by the DSA when the administrator creates a shadowing agreement. The supplier of the cooperating DSA is modified by the DSA when the administrator performs a **sob switch** operation. The cooperating DSA subentry must be the same on all DSAs participating in a network.

Because this attribute is only accessible by using **dirxadm**, no access using LDAP and DAP protocols is possible.

##### 3.4.4.1.1. Abbreviation

CDSA

##### 3.4.4.1.2. Syntax

Cooperating-DSA

##### 3.4.4.1.3. Example (dirxadm)

1. In the following example the CDSA (Cooperating-DSA) attribute displays the attribute values of two LDIF agreements.

```
CDSA={SUPP={AE={/O=My-Company/CN=DSA1},

PSAP={TS=DSA1,NA=TCP/IP_IDM!internet=123.45.67.89+port=21100}},

LDIF agreement 1:
    SOB={SAI={SS={AREA={CP={/O=My-Company},
        RA={BAS={/}}},ATT={DEF=TRUE}},
        UM={SI={S={PS={WS=1000,UI=1}}}},
        AGR={ID=14057,VERS=0},
        OBMV={VF=030401064822Z},
```

```
OBS=NONCOOPERATIVE,
RPOL={SUPP={MAXLOC=1024},
      LDSUPP={BINO=NONE, MAXROW=200,
              CS=UTF8, CHANGE0=FALSE}}};
```

LDIF agreement 2:

```
{SAI={SS={AREA={CP={/O=My-Company2}}, },
      RA={BAS={/}}}, ATT={DEF=TRUE}},
  UM={SI={S={PS={WS=1000, UI=1}}}},
  AGR={ID=16058, VERS=0},
  OBMV={VF=030401064822Z},
  OBS=NONCOOPERATIVE,
  RPOL={LDSUPP={CS=LATIN1, CHANGE0=FALSE}}}
```

2. In the following example the CDSA (Cooperating-DSA) attribute displays the attribute values of two shadowing agreements.

```
{CDSA={SUPP={AE={/O=My-Company/CN=DSA1}
```

```
PSAP={TS=DSA1, NA=TCP/IP_IDM!internet=123.45.67.89+port=21100}},
```

```
CONS={AE={/O=My-Company/CN=DSA2},
```

```
PSAP={TS=DSA2, NA=TCP/IP_IDM!internet=123.45.67.88+port=21100}},
```

agreement 1:

```
SOB={SAI={SS={AREA={CP={/O=My-Company},
      RA={BAS={/}}}, ATT={DEF=TRUE}},
      UM={SI={S={PS={WS=1000, UI=1}}}},
      AGR={ID=0, VERS=0},
      OBMV={VF=030401065354Z},
      OBS=NONCOOPERATIVE,
      RPOL={INI={ATU=TRUE},
            SUPP={MAXLOC=100},
            CONS={MAXLOC=100, REPLS=TRUE}}};
```

automatically generated agreement to propagate agreement information to consumer DSA:

```
{SAI={SS={AREA={CP={/CN=Cooperating-DSAS-Subentry},
      RA={DEF=TRUE}}, ATT={DEF=TRUE}},
  UM={SI={OC=TRUE}}},
```

```

AGR={ID=1,VERS=0},
OBMV={VF=030401065354Z},
OBS=COOPERATIVE};

agreement 2:
{SAI={SS={AREA={CP={/O=My-Company2},
RA={BAS={/}}},ATT={DEF=TRUE}},
UM={SI={S={PS={WS=1000,UI=1}}}},
AGR={ID=16058,VERS=0},
OBMV={VF=030401065354Z},
OBS=NONCOOPERATIVE,
RPOL={SUPP={US=TOTAL},CONS={US=TOTAL}}}}

```

### 3.4.4.2. DSE Type

The DSE Type attribute specifies the role that a DSE plays in regard to an entry and its distribution. A DSE can have multiple DSE types. A combination of DSE type values is specified by using the + (plus sign). Every DSE in a DSA has this attribute, which is accessible using **dirxadm** only. (See **DSE Type Syntax** for details.)

Because this attribute is only accessible by using **dirxadm**, no access using LDAP and DAP protocols is possible.

Valid values for the DSE Type attribute are:

- **ROOT** - Used exclusively by the DSE that represents the root of the DIT in a particular DSA.
- **GLUE** - A DSE that represents a name only. A DSA uses glue DSEs to hold the names of the superiors of the context prefixes and cross references where there is no other knowledge associated with them. You cannot assign any other DSE types to a DSE typed as GLUE.
- **CP** - A DSE that represents the context prefix of a naming context.
- **ENTRY** - A DSE that represents the entry itself, normally including the user attributes. If the SHADOW bit is not set, the DSE represents the master copy of the entry and holds all the attributes associated with the entry.
- **ALIAS** - A DSE that represents an alias entry.
- **SUBR** - A DSE that represents a specific subordinate reference and holds a corresponding knowledge attribute.
- **NSSR** - A DSE that represents a non-specific subordinate reference, and holds a corresponding knowledge attribute.
- **SUPR** - A root DSE that holds a knowledge attribute that represents a specific superior reference. (See **ROOT** value.)
- **XR** - A DSE that represents a cross-reference and holds a corresponding knowledge attribute.



- **ADM\_POINT** - A DSE that represents an administrative point.
- **SUBENTRY** - A DSE that represents a subentry.
- **SHADOW** - A DSE that represents shadowed information.
- **IMM\_SUPR** - A DSE that represents an immediate superior reference and holds a corresponding knowledge attribute.
- **RHOB** - A DSE that holds policy information (either administrative point or subentry information), which has been received from a superior DSA as the result of a hierarchical operational binding.
- **SA** - An additional characteristic of a subordinate reference DSE (that is, of type **SUBR**), which is used to indicate that the context prefix pointed to by the reference is an alias and not a normal entry.

The following list gives the most commonly used types and combinations in the absence of shadowing:

- **ROOT+SUPR**
- **GLUE** (must be used alone)
- **ENTRY+CP+ADM\_POINT**
- **ALIAS+CP**
- **SUBENTRY** (used alone)
- **IMM\_SUPR** (used alone)

#### 3.4.4.2.1. Abbreviation

DSET

#### 3.4.4.2.2. Syntax

DSE-Type

#### 3.4.4.2.3. Example (dirxadm)

```
DSET=ENTRY+ADM_POINT
DSET=GLUE
```

#### 3.4.4.3. Enable Unique Index

The Enable Unique Index attribute enables or disables unique constraint checking of attribute type values. It is an attribute of the root DSE. By default, this feature is disabled and the Enable Unique Index (ENUI) attribute is not present in the root DSE. To enable this feature, create the Enable Unique Index (ENUI) attribute with the value TRUE in the root DSE.

#### **3.4.4.3.1. Abbreviation**

ENUI

#### **3.4.4.3.2. LDAP Name**

enableUniqueIndex

#### **3.4.4.3.3. Syntax**

Boolean

#### **3.4.4.3.4. Example (dirxadm)**

ENUI=TRUE

# Appendix A: Country Codes

This appendix lists countries and their 2-letter ISO 3166 country code. This code is used in the Country Name simple syntax.

| Country                | Code |
|------------------------|------|
| AFGHANISTAN            | AF   |
| ALBANIA                | AL   |
| ALGERIA                | DZ   |
| AMERICAN SAMOA         | AS   |
| ANDORRA                | AD   |
| ANGOLA                 | AO   |
| ANGUILLA               | AI   |
| ANTARCTICA             | AQ   |
| ANTIGUA AND BARBUDA    | AG   |
| ARGENTINA              | AR   |
| ARMENIA                | AM   |
| ARUBA                  | AW   |
| AUSTRALIA              | AU   |
| AUSTRIA                | AT   |
| AZERBAIJAN             | AZ   |
| BAHAMAS                | BS   |
| BAHRAIN                | BH   |
| BANGLADESH             | BD   |
| BARBADOS               | BB   |
| BELARUS                | BY   |
| BELGIUM                | BE   |
| BELIZE                 | BZ   |
| BENIN                  | BJ   |
| BERMUDA                | BM   |
| BHUTAN                 | BT   |
| BOLIVIA                | BO   |
| BOSNIA AND HERZEGOVINA | BA   |
| BOTSWANA               | BW   |
| BOUVET ISLAND          | BV   |
| BRAZIL                 | BR   |

| Country                               | Code |
|---------------------------------------|------|
| BRITISH INDIAN OCEAN TERRITORY        | IO   |
| BRUNEI DARUSSALAM                     | BN   |
| BULGARIA                              | BG   |
| BURKINA FASO                          | BF   |
| BURUNDI                               | BI   |
| CAMBODIA                              | KH   |
| CAMEROON                              | CM   |
| CANADA                                | CA   |
| CAPE VERDE                            | CV   |
| CAYMAN ISLANDS                        | KY   |
| CENTRAL AFRICAN REPUBLIC              | CF   |
| CHAD                                  | TD   |
| CHILE                                 | CL   |
| CHINA                                 | CN   |
| CHRISTMAS ISLAND                      | CX   |
| COCOS (KEELING) ISLANDS               | CC   |
| COLOMBIA                              | CO   |
| COMOROS                               | KM   |
| CONGO                                 | CG   |
| CONGO, THE DEMOCRATIC REPUBLIC OF THE | CD   |
| COOK ISLANDS                          | CK   |
| COSTA RICA                            | CR   |
| COTE D'IVOIRE                         | CI   |
| CROATIA                               | HR   |
| CUBA                                  | CU   |
| CYPRUS                                | CY   |
| CZECH REPUBLIC                        | CZ   |
| DENMARK                               | DK   |
| DJIBOUTI                              | DJ   |
| DOMINICA                              | DM   |
| DOMINICAN REPUBLIC                    | DO   |
| ECUADOR                               | EC   |
| EGYPT                                 | EG   |
| EL SALVADOR                           | SV   |

| Country                           | Code |
|-----------------------------------|------|
| EQUATORIAL GUINEA                 | GQ   |
| ERITREA                           | ER   |
| ESTONIA                           | EE   |
| ETHIOPIA                          | ET   |
| FALKLAND ISLANDS (MALVINAS)       | FK   |
| FAROE ISLANDS                     | FO   |
| FIJI                              | FJ   |
| FINLAND                           | FI   |
| FRANCE                            | FR   |
| FRENCH GUIANA                     | GF   |
| FRENCH POLYNESIA                  | PF   |
| FRENCH SOUTHERN TERRITORIES       | TF   |
| GABON                             | GA   |
| GAMBIA                            | GM   |
| GEORGIA                           | GE   |
| GERMANY                           | DE   |
| GHANA                             | GH   |
| GIBRALTAR                         | GI   |
| GREECE                            | GR   |
| GREENLAND                         | GL   |
| GRENADA                           | GD   |
| CUADELOUPE                        | GP   |
| GUAM                              | GU   |
| GUATEMALA                         | GT   |
| GUINEA                            | GN   |
| GUINEA-BISSAU                     | GW   |
| GUYANA                            | GY   |
| HAITI                             | HT   |
| HEARD ISLAND AND MCDONALD ISLANDS | HM   |
| HOLY SEE (VATICAN CITY STATE)     | VA   |
| HONDURAS                          | HN   |
| HONG KONG                         | HK   |
| HUNGARY                           | HU   |
| ICELAND                           | IS   |

| Country                                    | Code |
|--------------------------------------------|------|
| INDIA                                      | IN   |
| INDONESIA                                  | ID   |
| IRAN, ISLAMIC REPUBLIC OF                  | IR   |
| IRAQ                                       | IQ   |
| IRELAND                                    | IE   |
| ISRAEL                                     | IL   |
| ITALY                                      | IT   |
| JAMAICA                                    | JM   |
| JAPAN                                      | JP   |
| JORDAN                                     | JO   |
| KAZAKHSTAN                                 | KZ   |
| KENYA                                      | KE   |
| KIRIBATI                                   | KI   |
| KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF     | KP   |
| KOREA, REPUBLIC OF                         | KR   |
| KUWAIT                                     | KW   |
| KYRGYZSTAN                                 | KG   |
| LAO PEOPLE'S DEMOCRATIC REPUBLIC           | LA   |
| LATVIA                                     | LV   |
| LEBANON                                    | LB   |
| LESOTHO                                    | LS   |
| LIBERIA                                    | LR   |
| LIBYAN ARAB JAMAHIRIYA                     | LY   |
| LIECHTENSTEIN                              | LI   |
| LITHUANIA                                  | LT   |
| LUXEMBOURG                                 | LU   |
| MACAO                                      | MO   |
| MACEDONIA, THE FORMER YUGOSLAV REPUBLIC OF | MK   |
| MADAGASCAR                                 | MG   |
| MALAWI                                     | MW   |
| MALAYSIA                                   | MY   |
| MALDIVES                                   | MV   |
| MALI                                       | ML   |
| MALTA                                      | MT   |

| Country                         | Code |
|---------------------------------|------|
| MARSHALL ISLANDS                | MH   |
| MARTINIQUE                      | MQ   |
| MAURITANIA                      | MR   |
| MAURITIUS                       | MU   |
| MAYOTTE                         | YT   |
| MEXICO                          | MX   |
| MICRONESIA, FEDERATED STATES OF | FM   |
| MOLDOVA, REPUBLIC OF            | MD   |
| MONACO                          | MC   |
| MONGOLIA                        | MN   |
| MONTSERRAT                      | MS   |
| MOROCCO                         | MA   |
| MOZAMBIQUE                      | MZ   |
| MYANMAR                         | MM   |
| NAMIBIA                         | NA   |
| NAURU                           | NR   |
| NEPAL                           | NP   |
| NETHERLANDS                     | NL   |
| NETHERLANDS ANTILLES            | AN   |
| NEW CALEDONIA                   | NC   |
| NEW ZEALAND                     | NZ   |
| NICARAGUA                       | NI   |
| NIGER                           | NE   |
| NIGERIA                         | NG   |
| NIUE                            | NU   |
| NORFOLK ISLAND                  | NF   |
| NORTHERN MARIANA ISLANDS        | MP   |
| NORWAY                          | NO   |
| OMAN                            | OM   |
| PAKISTAN                        | PK   |
| PALAU                           | PW   |
| PALESTINIAN TERRITORY, OCCUPIED | PS   |
| PANAMA                          | PA   |
| PAPUA NEW GUINEA                | PG   |

| Country                                      | Code |
|----------------------------------------------|------|
| PARAGUAY                                     | PY   |
| PERU                                         | PE   |
| PHILIPPINES                                  | PH   |
| PITCAIRN                                     | PN   |
| POLAND                                       | PL   |
| PORTUGAL                                     | PT   |
| PUERTO RICO                                  | PR   |
| QATAR                                        | QA   |
| REUNION                                      | RE   |
| ROMANIA                                      | RO   |
| RUSSIAN FEDERATION                           | RU   |
| RWANDA                                       | RW   |
| SAINT HELENA                                 | SH   |
| SAINT KITTS AND NEVIS                        | KN   |
| SAINT LUCIA                                  | LC   |
| SAINT PIERRE AND MIQUELON                    | PM   |
| SAINT VINCENT AND THE GRENADINES             | VC   |
| SAMOA                                        | WS   |
| SAN MARINO                                   | SM   |
| SAO TOME AND PRINCIPE                        | ST   |
| SAUDI ARABIA                                 | SA   |
| SENEGAL                                      | SN   |
| SERBIA AND MONTENEGRO                        | CS   |
| SEYCHELLES                                   | SC   |
| SIERRA LEONE                                 | SL   |
| SINGAPORE                                    | SG   |
| SLOVAKIA                                     | SK   |
| SLOVENIA                                     | SI   |
| SOLOMON ISLANDS                              | SB   |
| SOMALIA                                      | SO   |
| SOUTH AFRICA                                 | ZA   |
| SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS | GS   |
| SPAIN                                        | ES   |
| SRI LANKA                                    | LK   |



| Country                              | Code |
|--------------------------------------|------|
| SUDAN                                | SD   |
| SURINAME                             | SR   |
| SVALBARD AND JAN MAYEN               | SJ   |
| SWAZILAND                            | SZ   |
| SWEDEN                               | SE   |
| SWITZERLAND                          | CH   |
| SYRIAN ARAB REPUBLIC                 | SY   |
| TAIWAN, PROVINCE OF CHINA            | TW   |
| TAJIKISTAN                           | TJ   |
| TANZANIA, UNITED REPUBLIC OF         | TZ   |
| THAILAND                             | TH   |
| TIMOR-LESTE                          | TL   |
| TOGO                                 | TG   |
| TOKELAU                              | TK   |
| TONGA                                | TO   |
| TRINIDAD AND TOBAGO                  | TT   |
| TUNISIA                              | TN   |
| TURKEY                               | TR   |
| TURKMENISTAN                         | TM   |
| TURKS AND CAICOS ISLANDS             | TC   |
| TUVALU                               | TV   |
| UGANDA                               | UG   |
| UKRAINE                              | UA   |
| UNITED ARAB EMIRATES                 | AE   |
| UNITED KINGDOM                       | GB   |
| UNITED STATES                        | US   |
| UNITED STATES MINOR OUTLYING ISLANDS | UM   |
| URUGUAY                              | UY   |
| UZBEKISTAN                           | UZ   |
| VANUATU                              | VU   |
| VENEZUELA                            | VE   |
| VIET NAM                             | VN   |
| VIRGIN ISLANDS, BRITISH              | VG   |
| VIRGIN ISLANDS, U.S.                 | VI   |

| Country           | Code |
|-------------------|------|
| WALLIS AND FUTUNA | WF   |
| WESTERN SAHARA    | EH   |
| YEMEN             | YE   |
| ZAMBIA            | ZM   |
| ZIMBABWE          | ZW   |

# DirX Product Suite

The DirX product suite provides the basis for fully integrated identity and access management; it includes the following products, which can be ordered separately.



## DirX Identity

DirX Identity provides a comprehensive, process-driven, customizable, cloud-enabled, scalable, and highly available identity management solution for businesses and organizations. It provides overarching, risk-based identity and access governance functionality seamlessly integrated with automated provisioning. Functionality includes lifecycle management for users and roles, cross-platform and rule-based real-time provisioning, web-based self-service functions for users, delegated administration, request workflows, access certification, password management, metadirectory as well as auditing and reporting functionality.



## DirX Directory

DirX Directory provides a standards-compliant, high-performance, highly available, highly reliable, highly scalable, and secure LDAP and X.500 Directory Server and LDAP Proxy with very high linear scalability. DirX Directory can serve as an identity store for employees, customers, partners, subscribers, and other IoT entities. It can also serve as a provisioning, access management and metadirectory repository, to provide a single point of access to the information within disparate and heterogeneous directories available in an enterprise network or cloud environment for user management and provisioning.



## DirX Access

DirX Access is a comprehensive, cloud-ready, scalable, and highly available access management solution providing policy- and risk-based authentication, authorization based on XACML and federation for Web applications and services. DirX Access delivers single sign-on, versatile authentication including FIDO, identity federation based on SAML, OAuth and OpenID Connect, just-in-time provisioning, entitlement management and policy enforcement for applications and services in the cloud or on-premises.



## DirX Audit

DirX Audit provides auditors, security compliance officers and audit administrators with analytical insight and transparency for identity and access. Based on historical identity data and recorded events from the identity and access management processes, DirX Audit allows answering the “what, when, where, who and why” questions of user access and entitlements. DirX Audit features historical views and reports on identity data, a graphical dashboard with drill-down into individual events, an analysis view for filtering, evaluating, correlating, and reviewing of identity-related events and job management for report generation.

For more information: [support.dirx.solutions/about](https://support.dirx.solutions/about)



Eviden is a registered trademark © Copyright 2025, Eviden SAS – All rights reserved.

#### Legal remarks

On the account of certain regional limitations of sales rights and service availability, we cannot guarantee that all products included in this document are available through the Eviden sales organization worldwide. Availability and packaging may vary by country and is subject to change without prior notice. Some/All of the features and products described herein may not be available locally. The information in this document contains general technical descriptions of specifications and options as well as standard and optional features which do not always have to be present in individual cases. Eviden reserves the right to modify the design, packaging, specifications and options described herein without prior notice. Please contact your local Eviden sales representative for the most current information. Note: Any technical data contained in this document may vary within defined tolerances. Original images always lose a certain amount of detail when reproduced.