RADUA
Internet-Draft
Intended status: Experimental
Obsoletes: X660LDAP
Expires: August 27, 2024

J. Coretta

February 29, 2024

The OID Directory: The RA DUA
draft-coretta-oiddir-radua-00.txt

Abstract

In service to the "OID Directory" ID series, this ID covers design strategies, requirements and procedures for the client component of the OID Directory Registration Authority client/server model.

See the RADIR ID for a complete draft series manifest.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 27, 2024.

Copyright Notice

Table of Contents

1. Introduction

The X.500 Directory User Agent represents the client component within the traditional client/server model that interacts with any number of X.500 Directory System Agents for the purposes of information access.

Within the terms of this ID series, the Directory User Agent serves as a client of OID registration and registrant content, as retrieved from a Registration Authority Directory Information Tree.

1.1. Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

1.2. Acronyms Used

See Section 1.3 of the RADIR ID for all acronym references. Also, see Sections 1.7 and 1.8 of the RADIR ID for generalized terms and descriptions of significance to this ID series.

1.2.1. Definitions

The composite acronym "RA DUA" is hereby introduced within this ID. The acronym abbreviates the aforementioned 'Registration Authority Directory User Agent' term, which describes the 'client' component implied within the client/server model relevant to this ID series.

The composite acronym "RA DSA" used throughout this ID is defined in Section 1.2.1 of the RADSA ID.

The composite acronym "RA DIT" used throughout this ID is defined in Section 1.2.1 of the RADIT ID.

1.3. Intended Audience

This ID is intended for application designers, X.500/LDAP architects, and other personnel tasked with supporting or designing components related to the RA client/server model in service to this ID series.

General familiarity with the broad X.500/LDAP specification, as well all supporting IDs cited in Section 2 of the RADIR ID is STRONGLY RECOMMENDED.

1.4. Parameter Abstraction

For simplicity in describing certain request or argument parameters involving either DAP or LDAP operations in this ID, a simplified abstraction of ASN.1 parameters is shown to aid RA DUA adopter.

For example, the following structure may be used to outline the parameters of a Read or Search Operation to be conducted as part of an RA DUA managed procedure.

```
baseObject = dn          ; DN: see RFC4514 and X.501
scope      = 0/1/2       ; eq. X.511 'subset'
typesOnly  = bool or int ; see. X.511 'EntryInformationSelection'
                         ; and RFC4512 'SearchRequest'
filter     = filter      ; see X.511 and RFC4515
attributes = selection(s); see. X.511 'EntryInformationSelection'
                         ; and RFC4512 'AttributeSelector'
```

While the abstraction has favored the use of LDAP-focused parameters derived from [RFC4511], adopters MAY assume similar directives are applicable within the context of DAP unless otherwise indicated.

2. The RA DUA

The RA DUA is a traditional X.500/LDAP client -- supporting most or all of the standard operations defined throughout clauses 9 through

12 of ITU-T Rec. X.511, and throughout Section 4 of [RFC4511] --
that has been OPTIMIZED for use within the terms of this ID series.

2.1. Defined Parameters

The RA DUA is expected to support the directory protocols facilitated by the endpoint RA DSA(s), whether DAP, LDAP or both. Support for connectivity via the OSI networking stack, TCP/IP or IPC socket by the RA DUA is determined by the operational requirements of the RA DSA(s) in question.

Support for parallel X.500 protocols -- such as DOP or DSP -- is not specifically indicated.

No recommendations are made regarding the "appearance" or interactive nature of the RA DUA (i.e.: TUI vs. GUI), nor are any recommendations made regarding the specific language or framework used in its design.

No particular software license applied to the RA DUA is assumed.

The intended application may be for any end user in general, or it may be administratively focused. The RA DUA may be obtained by the general public, or it may be wholly proprietary and for internal use only.

The capabilities of the RA DUA MAY be flexible to suit the end user, or it may be strictly regimented, allowing few variations of behavior in routine operations.

In situations where an RA DUA is designed solely for the query and presentation of entries with no possibility of support for entry modifications, adopters MAY forego implementation of operational capabilities that are Write-focused in nature.

Application designers SHOULD make use of ONLY industry-recognized X.500/LDAP APIs, SDKs or libraries in a manner compliant with all "Best Practices" suggested by both the maintainer(s) and the authors of the standards indicated.

The RA DUA is not necessarily user-managed. An RA DUA may manifest in "clientless" form -- for example, facilitated through a web-based application interface residing on the RA DSA(s) directly, thus acting in the context of an abstract protocol gateway. These strategies may prove useful in reducing both the effort required by the end-user in order to access the service, as well as the costs of supporting the end user.

Regardless of the design and deployment philosophies employed, the primary focus of the RA DUA -- with particular emphasis on any and all proposed optimizations -- is to reduce the tedium of access and administration of potentially large registration and authority bases, and to introduce protective controls meant to ensure integrity of all

relevant content within the RA DIT.

2.2. Procedures

The RA DUA SHALL observe the procedures defined in the following subsections as it pertains to the query, allocation and maintenance of 'registration' and 'registrant' entries within an RA DSA.

2.2.1. Schema Availability

The RA DUA MUST obtain -- or possess complete foreknowledge of -- all schema definitions officially defined in Section 2 of the RASCHEMA ID as well as the schema definitions serving as super types for many of the attribute types defined in Section 2.3 of the RASCHEMA ID.

In addition, the RA DUA MUST both recognize and honor any additional DIT content rules, DIT structure rules and/or (additional) name forms created by the directory architects or administrators after-the-fact.

Obtaining the necessary schema definitions is typically conducted in either of the following manners, shown in order of preference.

- Through a direct Read of the 'subschemaSubentry' of the RA DSA
- Through manual processing of the (approved) schema file(s) based upon the complete contents of the RASCHEMA ID

When obtaining the schema through use of a Read or Search Operation, the schema SHOULD be refreshed at the commencement of a new RA DUA session. This accounts for changes to the schema definitions that may have taken place during runtime.

If the RA DSA has no apparent knowledge of the definitions to be used for the query and/or allocation of registrations and/or registrants within the RA DIT, the RA DUA MUST abandon attempts to interact with the RA DSA. It is RECOMMENDED that, in this case, the RA DUA present the user with error information describing the problem. This could suggest an RA DSA configuration problem, or possibly that the wrong RA DSA has been targeted by the RA DUA.

2.2.2. Configuration

There are two (2) modes of RA DUA configuration: automatic or manual.

Section 2.3 of the RASCHEMA ID introduces a small handful of types intended for "advertisement" by the RA DSA and for consumption by the RA DUA. These attributes are as follows:

- 'rARegistrationBase' ; ex.: ou=Registrations,o=rA
- 'rARegistrantBase' ; ex.: ou=Registrants,o=rA
- 'rADirectoryModel' ; ex.: 1.3.6.1.4.1.56521.101.3.1.3
- 'rAServiceMail' ; ex.: support@ra.example.com

```
    - 'rAServiceURI'       ; ex.: https://ra.example.com
    - 'rADITProfile'       ; ex.: dc=example,dc=com
    - 'rATTL'              ; ex.: 86400
```

These attribute types are extended through use of the 'rADUAConfig' AUXILIARY object class. See Section 2.3.6 of the RADSA ID for usable examples involving this class.

Auto-discovery of these attribute types will require disclosure privileges for the root DSE and any other entries that bear the 'rADUAConfig' object class.

Though the particulars of the root DSE are well outside the scope of this ID series, it is typically accessed by way of the Read Operation executed upon a NULL baseObject.

Retrieval SHOULD be made conditional using the 'rADUAConfig' object class as the filter AVA, and SHOULD involve attribute selection of the types shown below.

```
filter     = objectClass=rADUAConfig   ; Require 'rADUAConfig'
attributes = rARegistrationBase
             rARegistrantBase
             rADirectoryModel
             rAServiceMail
             rAServiceURI
             rADITProfile
             rATTL
```

If zero (0) entries are returned as a result of the Read Operation, this indicates any of the following:

- The RA DSA is not available
- The root DSE is not accessible, possibly due to access rights
- The root DSE is accessible, but lacks the 'rADUAConfig' class

Given any of these conditions, automatic parameter input has failed. The RA DUA has no alternative other than manual parameter input.

If one (1) entry is returned, the root DSE is accessible and has been configured for automatic input. The RA DUA SHOULD choose to proceed with configuration using the values provided.

See Section 2.3.6 of the RADSA ID regarding the possible methods for implementation of this entry with respect to multiple RA DITs served by an RA DSA as opposed to a single RA DIT.

If manual input of configuration values is required, typically this would require foreknowledge of the correct values, or access to an informational resource which makes those values available.

In this scenario, the RA DUA MUST request the user follow a procedure for manual input prior to use. Lack of proper configuration values

precludes any RA DUA session.

2.2.2.1. Processing

Following value input -- whether automatic or manual -- the acquired values MUST be processed and validated.

The following subsections cover each 'rADUAConfig'-extended attribute type in the context of runtime configuration of the RA DUA.

2.2.2.1.1. 'rADITProfile'

The 'rADITProfile' attribute type stores any number of DN values, each acting as a reference to a DIT-housed 'rADUAConfig' entry which contains the standard configuration parameters required by the RA DUA.

The 'rADITProfile' attribute type is a CRITICAL component within any implementation in which the following conditions apply:

- Multiple RA DITs reside on a single RA DSA, with each RA DIT accessed using potentially different configuration values
- Single RA DITs which bear usable configuration settings within DIT entry contexts -- as opposed to storage within the root DSE

In either case, the root DSE SHALL NOT contain any of the attribute types extended by the MAY clauses of the 'rADUAConfig' AUXILIARY object class OTHER THAN the 'rADITProfile', 'rATTL', 'rAServiceMail', and 'rAServiceURI' attribute types.

Similarly, referenced 'rADUAConfig' entries within an RA DIT SHALL NOT bear instances of the 'rADITProfile' attribute type.

Instances where these attribute types are improperly combined within entries is considered a "Duplicate RA Context Error" and represents a serious operational deficiency that MUST be reported to the end user. The RA DUA SHOULD fail the session or (optionally) allow for administrative override if corrective measures are to be taken.

2.2.2.1.2. 'rARegistrationBase'

The 'rARegistrationBase' attribute type is the most CRITICAL of all attribute types related to RA DUA configuration.

The purpose of this multi-valued type is to store the DN(s) in which 'registration' entries are stored. This parameter is REQUIRED, as it prevents the need for inefficient broad-level Search Operations, potentially within a particularly large directory information tree.

The RA DUA MUST handle the instance value(s) as follows:

1. Verify presence and accessibility of entries identified by the respective DN values using the Read Operation
2. Determine whether the given entries bear the 'registration' ABSTRACT object class

2a. If the named entries DO NOT bear the 'registration' class, the RA DUA must interpret the entries as simple organizational containers housing 'registration' entries one (1) level below

2b. If the named entries bear the 'registration' class, this is indicative of an official starting-point for registration content within the "OID Directory"

3. Preserve these DNs for the remainder of the session, as they will influence the various operations that may take place

In the case of condition "2a", a read-only RA DUA MAY opt to fail the session if no 'registration' entries reside exactly one (1) level beneath the apparent "organizational container" entry. The RA DUA MAY allow for administrative override of this behavior, thereby allowing retroactive registration creation within an implementation not yet populated.

2.2.2.1.3. 'rARegistrantBase'

The 'rARegistrantBase' attribute type is OPTIONAL in terms of RA DUA configuration. It identifies one (1) or more DNs which lead to the location of authority-related entries within the RA DIT.

The RA DUA MUST handle the value(s) associated with this type as follows:

1. Verify presence and accessibility of entries identified by the respective DN values using the Read Operation
2. Compare the DN values to those within the 'rARegistrationBase' attribute type instance

2a. If the DN values are identical, this implies use of combined authority/registration entries in a single location within the RA DIT -- a procedure that is generally discouraged

2b. If the DN values are different, this implies use of dedicated authority entries, which bear the 'registrant' STRUCTURAL object class and reside in a location separate from that which houses entries bearing the 'registration' STRUCTURAL Object Class

2c. If no DN values are specified within the 'rARegistrantBase' attribute type instance, the RA DUA MAY interpret this as an indication that no authority information is available within the RA DIT, and associated authority attribute types SHOULD NOT be requested by the RA DUA

In the case of "2a", the RA DUA SHOULD include all attribute types

specified within the 'currentAuthorityContext', 'sponsorContext' and 'firstAuthorityContext' 'MAY' clauses for subsequent Read Operations of 'registration' entries.

In the case of "2b", the RA DUA SHOULD include all attribute types specified within 'currentAuthorityContext', 'sponsorContext' and 'firstAuthorityContext' 'MAY' clauses for subsequent Read Operations of 'registrant' entries.

Dedicated authority entries bearing the 'registrant' STRUCTURAL object class should be located exactly one (1) level below each specified 'rARegistrantBase' DN value within the RA DIT.

Depending on the nature of implementation of this ID series, it may or may not be advisable to populate the 'rARegistrantBase' Attribute Type for consumption by all clients indiscriminately. See Section 5.2 of the RADIT ID for security considerations on this topic.

2.2.2.1.4. 'rADirectoryModel'

The 'rADirectoryModel' type describes the abstract structure of the RA DIT in terms of 'registration' layout and probable DN syntax. The employed model shall have a profound influence on the manner in which the RA DUA shall interact with the RA DIT.

A specified directory model is REQUIRED for proper functioning of the RA DUA, whether directly or indirectly. The decided model specifier, which MUST be a numeric OID, is declared using the 'rADirectoryModel' attribute type.

Sections 3.1.2 and 3.1.3 of the RADIT ID define two (2) official directory models and DN syntax schemes identified by the following numeric OIDs:

- 'twoDimensional' (1.3.6.1.4.1.56521.101.3.1.2)
- 'threeDimensional' (1.3.6.1.4.1.56521.101.3.1.3)

In virtually every case, the 'threeDimensional' model is STRONGLY RECOMMENDED for implementation and use, however RA DUAs SHOULD be prepared to incorporate other models that could be defined in any future extensions to this ID series.

The RA DUA MUST support use of the 'threeDimensional' model without exception and to the letter of every recommendation set forth within this ID series.

As stated clearly and in no uncertain terms within the originating document, the 'twoDimensional' model is STRONGLY DISCOURAGED aside from use in non-standard or particularly unusual scenarios.

The RA DUA MAY support use of the 'twoDimensional' model at the discretion of the application designer. Support for this model is purely OPTIONAL.

The RA DUA MUST handle the value of this instance as follows:

1. Determine whether the 'rADirectoryModel' attribute type has been set with an explicit numeric OID
1a. If NO numeric OID has been specified, use of the RECOMMENDED 'threeDimensional' model is to be enforced by DEFAULT
1b. If a numeric OID has been specified, identify the value as a known and supported model OID and adjust RA DUA behavior in accordance with prescribed procedures of the RADIT ID
1c. If a numeric OID has been specified and is not immediately identifiable within the terms of this ID series -- such as a future extension of this standard -- the RA DUA MAY defer to the specifics of the recommendation or ID from which the OID originates OR the RA DUA MAY choose to fail the session
2. Preserve the value for the remainder of the session, as it will influence the specifics of operations that may occur

In the case of condition "1c", if the RA DUA chooses to fail the session, the RA DUA SHOULD present the user with an error message indicating the encounter with an unknown or as-of-yet unsupported directory model identifier.

2.2.2.1.5. Additional Parameters

The 'rAServiceMail' attribute type, when defined, contains any number of email addresses meant for support or request purposes. Use of this type in the RA DIT is OPTIONAL, but SHOULD be recognized by the RA DUA when present.

The 'rAServiceURI' attribute type, when defined, contains any number of URI values related to the service, such as terms of use, support information, documentation and other resources. Use of this type in the RA DIT is OPTIONAL, but SHOULD be recognized by the RA DUA when present.

The 'rATTL' attribute type, when defined for an entry bearing the 'rADUAConfig' class, imposes a global entry caching TTL value meant for consumption and observance by qualified RA DUA implementations. See Section 2.2.3.4 for details and semantics regarding assignment and precedence strategies for a TTL.

2.2.3. Queries

This subsection covers strictly read-related operations, such as the location and presentation of a given 'registration' entry.

2.2.3.1. Retrieving Entries

The RECOMMENDED procedure for retrieving an entry -- in any directory

model defined in this ID series -- is a Read Operation of the entry by way of its DN. This will return either one (1) entry, or none.

Foreknowledge of the DN is required for a Read Operation attempt of this fashion.

If the entry is a 'registration', the DN may be resolved by way of the associated numeric OID using the appropriate process defined in Section 3.1.

'registration' entries may reference other 'registration' entries using the spatial attribute types defined in Section 2.3 of the RASCHEMA ID and discussed further in Section 2.2.3.3.

'registrant' entries, however, aren't resolvable in any standard manner. These are dedicated authority entries that are normally accessed through references held by any associated 'registration' entries.

- 'firstAuthority' and/or 'c-firstAuthority'
- 'currentAuthority' and/or 'c-currentAuthority'
- 'sponsor' and/or 'c-sponsor'

However, in cases where direct referencing of DNs within the context of a Read Operation is not practical, possibly due to any of the following ...

- Lack of assigned spatial reference types
- An unsupported or incoherent DN syntax is indicated
- Administrative operations are underway

Use of a List Operation or subtree Search Operation may be indicated.

While this method of searching is not generally recommended within the spirit of this ID series, the RA DUA MAY allow this capability where appropriate.

If the RA DUA encounters difficulty relating to a particularly large number of entries retrieved through a Search Operation, support for Simple Paged Results Manipulation by the RA DUA may be indicated if supported by the RA DSA. For details, see clause 7.9 of ITU-T Rec. X.511 and the entirety of [RFC2696].

2.2.3.2. Reading Entries

The following subsections outline the procedures involved in the presentation and analysis of a 'registration' or 'registrant' entry successfully retrieved by way of the Read or Search Operation.

2.2.3.2.1. Entry 'objectClass' Analysis

Given a successfully conducted Read or Search Operation, and assuming

a single entry -- whether 'registration' or 'registrant' -- has been returned, the RA DUA SHOULD first read the 'objectClass' values and make note of all that originate in Section 2.5 of the RASCHEMA ID.

The 'registration' ABSTRACT object class is a required class for any OID allocation. This class MUST only appear on entries which bear the 'rootArc' or 'arc' STRUCTURAL class. It is a sub type of the 'top' ABSTRACT class, defined in Section 2.4.1 of [RFC4512] and in clause 13.3.3 of ITU-T Rec. X.501.

The 'registrant' STRUCTURAL object class is only used for dedicated registrants, which are associated through DN references held by associated 'registration' entries, if any. Appearance upon any other form of entry is a suboptimal or illogical state.

Presence of the 'x660Context' and/or 'x680Context' AUXILIARY classes for a 'registration' entry is permitted. These object classes extend multiple attribute types which conceptually relate to ITU-T Rec. X.660 and X.680 respectively.

Presence of the 'x667Context' AUXILIARY class for a 'registration' entry is only expected in cases where an OID allocation involves a 'registeredUUID' attribute type instance and where the assigned 'n' value is the equivalent unsigned 128-bit integer.

Presence of the 'iTUTRegistration' AUXILIARY class is only permitted for allocations bearing the 'arc' STRUCTURAL class and describe an OID beginning with zero (0). It extends no attribute types.

Presence of the 'iSORegistration' AUXILIARY class is only permitted for allocations bearing the 'arc' STRUCTURAL class and describe an OID beginning with one (1) It extends no attribute types.

Presence of the 'jointISOITUTRegistration' AUXILIARY class is only permitted for allocations bearing the 'arc' STRUCTURAL class and describe an OID beginning with two (2). It extends the 'longArc' attribute type.

Entries SHALL NOT bear more than one (1) of the 'iTUTRegistration', 'iSORegistration' or 'jointISOITUTRegistration' classes.

Presence of the 'spatialContext' AUXILIARY class is only permitted for 'registration' entries. It extends seven (7) spatial reference attribute types used to describe arrangement of allocations within the spectrum. Additional collective incarnations of some of these attribute types may be indicated.

Presence of the 'registrationSupplement' AUXILIARY class is only permitted for 'registration' entries. It extends miscellaneous attribute types which extend from no official standards meant for ease-of-use only.

Collective Attributes are described in [RFC3671].

Presence of the 'firstAuthorityContext', 'currentAuthorityContext', and 'sponsorContext' AUXILIARY classes is permitted for 'registrant' entries and 'registration' entries alike, and would depend upon the 'registrant' model employed within the RA DIT. Each of these classes extends several identity and contact related attribute types for use within the context of sponsorship, current authorities and previous authorities.

Presence of the 'rADUAConfig' AUXILIARY class SHOULD NOT be permitted for 'registration' or 'registrant' entries and indicates a suboptimal or illogical state.

Assuming no violations were perceived as outlined above, the state of the entry's object class stack is apparently copacetic.

2.2.3.2.2. Attribute Types Used

At a minimum, the RA DUA SHOULD only expect Attribute Types defined within Section 2.3 of the RASCHEMA ID, and/or their respective super types defined in [RFC4519], [RFC4524] and [RFC2079], to be assigned to entries within the terms of this ID series.

See Section 3.2 of the RADIT ID for numerous examples regarding various attribute type use cases and requirements.

The RA DIT MAY use other attribute type and object class definitions unrelated to this ID series, for supplemental reasons, however their incorporation would be wholly proprietary and would have no official standing per this ID series.

2.2.3.2.3. Value Syntax

Each attribute type, whether directly or indirectly, is governed via a syntax definition in terms of the allowed value(s) to be set for an entry.

As mentioned in Section 2.1 of the RASCHEMA ID, standard syntaxes do not necessarily align perfectly with the syntactical requirements of the standards upon which this ID series is based -- namely ITU-T Recommendations X.660 and X.680.

To ease the difficulty in implementing an RA DUA which honors the syntactical characteristics of the underlying subject matter -- as opposed to only recognizing the syntax alone -- Section 2.3 of the RASCHEMA ID includes ABNF productions for every attribute type defined, whether by reference or in literal form, intended for use by those tasked with implementation of this ID series in some manner.

The RA DUA MUST recognize these ABNF productions when reading values

that were retrieved through use of the Read or Search Operation.

The RA DUA MAY decide how to handle the case of reading a previously set attribute value of invalid or non-compliant form. The RA DUA may warn the end-user of a value that is not well-formed, or it may opt to omit the value from visibility altogether. If the RA DUA is of an administrative focus, the opportunity for corrective measures MAY be facilitated.

2.2.3.3. Navigating Registration Entries

Some of the more complete RA services, whether public or private, may offer a simple interface to facilitate intuitive incremental movement among registrations that are associated horizontally or vertically in terms of "up", "down", "left", "right", "min", "max" and "top".

Depending on the needs of the intended audience, as well as the manner in which this specification is adopted, this can be an exceptionally difficult feature to implement for the RA DUA, but is one that can dramatically improve the user experience.

The RA DUA SHOULD NOT assume positive support for this practice in all implementations of this ID series.

2.2.3.3.1. Superior Vertical References

During the Read or Search Operation executed in order to obtain a 'registration' entry, the RA DUA MAY request any of the following attribute types:

- 'supArc'
- 'c-supArc'
- 'topArc'
- 'c-topArc'

The 'supArc' and 'c-supArc' attribute types describe the immediate superior (parent) registration in terms of ancestral lineage. Only one (1) of each of these attribute types should be present for any given 'registration' entry, never both.

The 'topArc' and 'c-topArc' attribute types describe the absolute root registration in terms of ancestral lineage. Only one (1) of each of these attribute types should be present for any given 'registration' entry, never both.

Use of Collective attribute types -- namely those prefaced using the requisite 'c-' flag -- is not practical in the two dimensional model and thus need not be requested.

At no point are the above attribute types to be requested for entries

that bear the 'rootArc' STRUCTURAL object class. Root registrations do not have superiors.

2.2.3.3.2. Subordinate Vertical References

Subordinate vertical references tend to be the most challenging among all the various attribute types related to spatial navigation within the terms of this ID series.

The RA DUA MAY request the 'subArc' attribute type as part of the Read or Search Operation used for retrieval of a 'registration' entry from the RA DIT.

At no point should an entry bear the 'subArc' attribute type if it bears an 'isLeafNode' instance with a value of TRUE. This indicates an illogical RA DIT condition.

When defined, the 'subArc' attribute type instance SHOULD reflect a complete manifest of all references for 'registration' entries that are direct subordinates of the bearer. This instance SHOULD be requested in baseObject-scoped context, and is the only multi valued spatial attribute type defined within this ID series.

The RA DUA SHOULD prefer 'subArc' enumeration to a List Operation beneath a 'registration'. This method of searching is a potentially costly request in the face of particularly large sets of subordinate 'registration' entries present within the RA DIT.

The RA DUA SHOULD be prepared to manually sort the set of 'subArc' DN references based on its OID or NumberForm value, however this responsibility may be handled by the RA DSA.

2.2.3.3.3. Horizontal References

Horizontal (sibling) references describe both adjacent as well as minimum and maximum 'registration' contexts.

During a Search Operation intended to obtain a 'registration' entry, the RA DUA SHOULD request the following attribute types:

- 'leftArc'
- 'rightArc'
- 'minArc'
- 'c-minArc'
- 'maxArc'
- 'c-maxArc'

The 'leftArc' attribute type describes the sibling 'registration' entry that is nearest but less than that of the bearer in terms of Number Form ('n') numerical magnitude.

The 'rightArc' attribute type describes the sibling 'registration'

entry that is nearest but greater than that of the bearer in terms of Number Form ('n') numerical magnitude.

The 'minArc' and 'c-minArc' attribute types are used to reference the 'registration' entry bearing the lowest magnitude Number Form ('n') value within a set of siblings. Only one (1) of these Attribute Types should be present for any given 'registration' entry, never both.

The 'maxArc' and 'c-maxArc' attribute types are used to reference the 'registration' entry bearing the highest magnitude Number Form ('n') value within a set of siblings. Only one (1) of these Attribute Types should be present for any given 'registration' entry, never both.

2.2.3.4. Client Entry Caching

For particularly large or busy implementations of this ID series, the RA DUA MAY support basic client-driven entry caching of any retrieved 'registration' and 'registrant' entries by way of either the 'rATTL' or 'c-rATTL' integer attribute types, as defined within Section 2.3 of the RASCHEMA ID.

A valid use case for caching involves serving IANA PEN registrations [PRIVATE], which number in the tens of thousands. Caching may yield tremendous savings in terms of resource utilization associated with particularly large numbers of static entries being retrieved in a repeating manner.

This ID makes no assumptions regarding the design or implementation of the underlying caching subsystem. The only abstract requirement relates to the ability to cache a single directory entry for a span of time equivalent to the effective TTL value in minutes.

The RA DUA SHOULD allow for the user to determine whether an entry being presented is derived from a cache.

2.2.3.4.1. Semantics

An 'rATTL' may be found in the following contexts wherever the 'rADUAConfig', 'registration' or 'registrant' classes are present.

- A root DSE
- An RA DIT context
- An individual leaf-node entry

The collective counterpart attribute type, 'c-rATTL', may be found on all of the above EXCEPT the root DSE.

Presence of an instance of either of these attribute types for any entry serves as an indicator to the RA DUA that a caching directive may be in effect depending on the value.

The significance of value magnitude -- whether collective or not -- is as follows:

- A value less than or equal to "0" is equivalent to a TTL being undefined: NO cached lifespan is specified
- A value greater than or equal to "1" indicates a TTL lifespan expressed by that value (e.g.: "1440" for a 24-hour TTL)

Countdown of a timespan commences at the same time the indicated entry is retrieved and cached by the RA DUA according to the value.

Presence of the 'rATTL' attribute type within the RA DSA's root DSE indicates use of global caching, a condition in which all entries are cached for a fixed amount of time unless they are subjected to an individual override by the 'rATTL' or 'c-rATTL' types.

Presence of the 'rATTL' attribute type within separate 'rADUAConfig' class profile instances indicates context-specific entry caching (for example, a single RA DIT in the midst of others served by a common RA DSA).

The 'c-rATTL' attribute type is only present for entries when served by an RA DSA which supports collective attributes. No instance of the 'c-rATTL' attribute type shall be present within the root DSE.

In the face of multiple overlapping TTLs implied for an entry, these rules of precedence can guide the RA DUA in determining the correct TTL:

- DSE-based TTL overrides nothing (lowest common denominator)
- Contextual TTL overrides DSE TTL
- Collective TTL overrides a subtree of a contextual TTL
- Non-collective leaf entry TTL overrides all of the above

If deemed appropriate within the spirit of an implementation, or if potentially necessary in an administrative context, the RA DUA MAY allow for arbitrary cache bypass, whereas the cached entry may be refreshed ahead of its scheduled TTL expiry.

2.2.4. New Allocations

The following subsections involve considerations and procedures which are related to the incorporation of new registration allocations into an RA DIT.

Each "OID Directory" implementation will almost certainly adopt only certain attribute types for use in entries.

Such restrictions may be exercised based on use of only select object classes within an entry, through observance of any DIT content rules that may be in effect, or through a form of access control.

The RA DUA is expected to honor any policies imposed by the service that would influence or mandate the composition of new entries in a particular way.

2.2.4.1. Verification

Certain preallocation checks MUST be conducted prior to any attempt at creating an allocation. The following subsections describe these procedures. Please note that only the three dimensional directory model is covered due to its STRONGLY RECOMMENDED status over the alternative model.

While the RA DSA may implement 'registration' integrity controls of its own, the RA DUA SHALL NOT rely on such elements to mitigate bogus or ill-advised requests alone. The RA DUA is REQUIRED to submit only well-formed and sanctioned requests, and SHALL NOT be designed under the unfounded assumption that the RA DSA will conduct post-operative or custodial amendments of any kind.

Adopters of the RA DUA construct should remember that the context of an allocation request can greatly influence the perception of the various outcomes discussed in the following subsections.

For example, consider the entry of retroactive 'registration' entries -- obsolete and wholly invalid by today's standards -- by an end user simply to build a thorough and well-formed implementation of the OID spectrum, versus a new registration being created today, and governed by today's standards. The two scenarios have radically different implications, yet the effective action is unchanged.

Depending on the nature and intended audience of an RA DUA solution, this distinction may be particularly important. It may prove useful to allow for effective "overrides" for authorized identities or modes of operating, for example, to allow the creation of registrations beneath an obsolete superior context.

It is important to note that the procedures defined in the following subsections do not account for any internal governance or approval process related to allocation request handling.

2.2.4.1.1. Ancestral Viability

Given an intended registration DN of:

```
n=9999,n=4,n=1,n=6,n=3,n=1,ou=Registrations,o=rA
```

The RA DUA MUST first verify the presence of the intended superior registration DN.

```
n=4,n=1,n=6,n=3,n=1,ou=Registrations,o=rA
```

This search can be conducted using the following SearchRequest parameters:

```
baseObject = n=4,n=1,n=6,n=3,n=1,ou=Registrations,o=rA
scope      = 0
filter     = objectClass=registration
attributes = registrationStatus
             registrationClassification
             isLeafNode
             isFrozen
             objectClass
```

If exactly one (1) entry is returned with no apparent error, the superior entry is confirmed to be present. In this case, the RA DUA should read the values of the requested attribute types.

If the 'registrationStatus' and/or 'registrationClassification' value is OBSOLETE, DEALLOCATED or some other (known) declaration of a state of non-operation, this MUST fail the allocation request unless the attempt was made in (approved) retroactive context. The case folding scheme in effect for these values is not significant.

Similarly, if the 'isFrozen' and/or 'isLeafNode' attribute types bear a Boolean value of TRUE, the allocation request MUST fail unless the attempt was made in (approved) retroactive context.

The error 'noSuchObject' indicates the requested entry was not found. When using LDAP, this bears the resultCode value of "32".

'noSuchObject' SHOULD be investigated by the RA DUA by way of an ancestral traversal -- a process in which each successive ancestor entry is subjected to a similar presence check as described above in incremental fashion. Ordering of ancestral traversal checks is not significant, as either ordering scheme will ultimately lead to the same conclusion.

Each ancestor entry is referenced using a DN value which lacks the leaf-node RDN component of the previously-checked descendant entry.

For instance, continuing with the above superior DN's lineage:

- n=1,n=6,n=3,n=1,ou=Registrations,o=rA
- n=6,n=3,n=1,ou=Registrations,o=rA
- n=3,n=1,ou=Registrations,o=rA
- n=1,ou=Registrations,o=rA

Following this procedure allows the RA DUA or the end user to locate where the apparent ancestral discontinuity begins within the RA DIT.

The terminus of a registration lineage -- such as the one described above -- is determined based on the presence of the STRUCTURAL object class 'rootArc' for an entry (e.g.: "n=1,ou=Registrations,o=rA").

2.2.4.1.2. Number Form Uniqueness

Within a set of sibling 'registration' entries, it is CRITICAL that any new allocation involve use of a horizontally-unique 'n' value.

Using one of the RECOMMENDED DN syntaxes described in Section 3.1, the proper procedure simply involves a Read-based presence check of the intended 'registration' DN.

For instance, if the desired allocation shall bear the DN:

```
n=9999,n=3,n=1,ou=Registrations,o=rA
```

A preemptive Read Operation MUST be executed in the following manner:

```
baseObject = n=9999,n=3,n=1,ou=Registrations,o=rA
scope      = 0
typesOnly  = TRUE
attributes = objectClass
```

Use of the 'typesOnly' option is merely for bandwidth efficiency, as the goal of this request is to see if ANY entry exists by this DN -- not to examine any particular attribute type values.

If zero (0) entries are returned alongside a 'noSuchObject' error, the Number Form is unique.

Any non-zero number of entries that are returned with no apparent error indicates the Number Form is NOT unique and the attempt at allocation MUST fail. The RA DUA SHOULD inform the user of this outcome.

There are no practical override use cases for this condition.

If using a 'registration' DN syntax other than those RECOMMENDED in Section 3.1, use of a filter within a List Operation executed upon the intended parent registration may be required at the risk of performance penalties proportional to the number of entries present:

```
baseObject: = n=3,n=1,ou=Registrations,o=rA
scope       = 1
typesOnly   = TRUE
filter      = n=9999
attributes  = objectClass
```

The same entry count semantics as described above will apply.

See Section 2.2.4.1.3 for considerations regarding the presence of ranges of allocations within the set of sibling registrations.

2.2.4.1.3. Ranged Allocations

During the creation of new registrations, RA DUAs MUST preemptively search for any range-based registrations present that might overlap with the intended 'n' value of the entry. This requirement MAY be relaxed if it is known that ranged allocations are not supported in the target location.

For example, to determine if a ranged allocation resides within the 1.3 OID registration that would overlap with a desired Number Form, the RA DUA MUST perform a List Operation of the following form:

```
baseObject = n=3,n=1,ou=Registrations,o=rA
scope      = 1
typesOnly  = TRUE
filter     = (&(n<=X)
 (|(registrationRange>=X)(registrationRange=-1)))
attributes = registrationRange
```

Given these parameters, where the filter AVA "X" represents the 'n' (Number Form) chosen -- for example "100" -- the search will return zero (0) entries if no range violation is detected involving value "X". This behavior is unchanged in the midst of multiple finite and/or infinite ranges, regardless of contiguity.

Please note that integer "X" MUST be chosen or selected by some means by the RA DUA or its end user for this procedure to work as intended. "X" SHALL NOT be negative.

An infinite range SHALL ONLY appear once in any sibling context, and the associated entry DN SHALL ALWAYS represent the true 'maxArc' or 'c-maxArc' value in the same sibling context, if specified.

If ANY number of entries are returned as a result of this allocation check, the registration MUST fail. The RA DUA MAY opt to make other attempts using another Number Form in place of "X", or it may simply inform the user of an overlapping allocation attempt.

This procedure assumes the preemptive Number Form Uniqueness check procedures described in Section 2.2.4.1.2 have been conducted with a favorable outcome as described.

However, in certain use cases, it may be advantageous to replace the above 'filter' with:

```
(|(&(n<=X)(|(registrationRange>=X)(registrationRange=-1)))(n=X))
```

Doing so accomplishes the goals of this section and Section 2.2.4.1.2 in a single action.

The RA DUA MAY opt to impose a 'typesOnly' value of FALSE to allow the user to manually observe the range structure(s) present within a set of siblings directly (by way of the 'registrationRange' type) if and when an overlapping allocation attempt was made. This allows the user to select an appropriate Number Form value in an informed manner -- as opposed to trial-and-error methodology, for instance.

2.2.4.2. Submission

Assuming the verification procedures covered in Section 2.2.4.1 were of a favorable outcome, the submission of the allocation details may be indicated.

This ID assumes that any requisite review or approval processes that are practiced by those who serve in authority over the RA have been performed.

Creation of a directory entry, based upon decided allocation details, is the last step necessary for a proper submission. Please see the RADIT ID for details and many examples regarding entry composition.

The Add Operation is the intended means for submission of the entry.

If the particular implementation of this ID series does not support this operation in this manner, the RA DSA may be expected to support another means out of scope for this ID series.

Upon submission of the composed entry to the receiving RA DSA, the RA DUA SHOULD retrieve the newly added entry to verify attribute type and object class content is present as intended. This should precede any declarations of successful submission to the end user.

2.2.5. Allocation Updates

Updates to 'registration' entries, as mentioned previously, is often ill-advised outside of extraordinary circumstances, likely involving corrections to entries deemed to be of poor form, or created in bad faith. A general rule-of-thumb suggests that any 'registration' is typically static.

Updates to 'registrant' entries or attributes, however, is far more likely if contact information, which changes over time, is present.

Nevertheless, alterations to the content of entries in either context is facilitated through use of the Modify Operation for any of the following desired actions:

- Addition of object class values
- Addition of attribute type instances or values

- Removal of undesirable or invalid attribute instances or values
- Correction of bogus attribute type values
- Relegation of a currentAuthority to a firstAuthority

In an even more exceptional (and unlikely) scenario, the correction of an invalid Number Form or OID RDN value is facilitated through use of the Modify DN Operation, which is intended for the "renaming" and/or "relocation" of a specified entry. This is almost certainly an administratively focused use case.

The semantics for either of these operations are well outside the scope of this ID series.

3. IANA Considerations

There are no requests to IANA in this document at this time.

4. Security Considerations

The RA DUA MUST possess the capability to honor any authentication and/or confidentiality policies imposed by the RA DSA. This would imply Bind and Unbind capabilities, at the least.

This may involve strategies related to TLS, 2FA, OTP and others. TLS support may include facilities for mutual authentication. Support of password-related operations -- such as those defined in [RFC3062] -- may be indicated.

Certain directory implementations may require these capabilities be exercised prior to interaction with ANY facet of the directory -- no matter how innocuous a request may be perceived (for instance, basic query of the root DSE only). This is an especially common practice in high-security X.500/LDAP implementations. Adopters are advised to be prepared for such conditions.

5. References

5.1. Normative References

RADIR Coretta, J., "The OID Directory: A Technical Roadmap", draft-coretta-oiddir-roadmap, February 2024.

RADIT Coretta, J., "The OID Directory: The RA DIT", draft-coretta-oiddir-radit, February 2024.

RADSA Coretta, J., "The OID Directory: The RA DSA", draft-coretta-oiddir-radsa, February 2024.

RASCHEMA Coretta, J., "The OID Directory: The RA Schema", draft-coretta-oiddir-schema, February 2024.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2696] C. Weider, A. Herron, A. Anantha, Microsoft, T. Howes and Netscape, "LDAP Control Extension for Simple Paged Results Manipulation", RFC 2696, September 1999.

[RFC3671] Zeilenga, K., "Collective Attributes in the Lightweight Directory Access Protocol (LDAP)", RFC 3671, December 2003.

[RFC4511] J. Sermersheim, Ed. "Lightweight Directory Access Protocol (LDAP): The Protocol", RFC 4511, June 2006.

[RFC4512] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): Directory Information Models", RFC 4512, June 2006.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", RFC 8174, May 2017.

[X.501] International Telecommunication Union - Telecommunication Standardization Sector, "The Directory: Models", ITU-T X.501, October 2019.

[X.511] International Telecommunication Union - Telecommunication Standardization Sector, "The Directory: Abstract service definition", ITU-T X.511, October 2019.

5.2. Informative References

[PRIVATE] IANA, "Private Enterprise Numbers", https://www.iana.org/assignments/enterprise-numbers.

[RFC2079] Smith, M., "Definition of an X.500 Attribute Type and an Object Class to Hold Uniform Resource Identifiers (URIs)", RFC 2079, January 1997.

[RFC3062] Zeilenga, K., "LDAP Password Modify Extended Operation", RFC 3062, February 2001.

[RFC4519] Sciberras, Ed., A., "Lightweight Directory Access Protocol (LDAP): Schema for User Applications", RFC 4519, June 2006.

[RFC4524] Zeilenga, K., "Lightweight Directory Access Protocol (LDAP): COSINE LDAP/X.500 Schema", RFC 4524, June 2006

[X.500] International Telecommunication Union - Telecommunication Standardization Sector, "The Directory: Overview of concepts, models and services", ITU-T X.500, October 2019.

[X.660] International Telecommunication Union - Telecommunication Standardization Sector, "General procedures and top arcs of the international object identifier tree", ITU-T X.660, July 2011.

[X.680] International Telecommunication Union - Telecommunication Standardization Sector, "Abstract Syntax Notation One (ASN.1): Specification of basic notation", ITU-T X.680, July 2002.

Author's Address

Jesse Coretta
California, United States

Email: jesse.coretta@icloud.com