

Australian/New Zealand Standard™

**Information technology—Open
Distributed Processing—Trading
function**

**Part 3: Provision of Trading Function
using OSI Director Service**



AS/NZS ISO/IEC 13235.3:2006

This Joint Australian/New Zealand Standard was prepared by Joint Technical Committee IT-015, Software and Systems Engineering. It was approved on behalf of the Council of Standards Australia on 27 October 2006 and on behalf of the Council of Standards New Zealand on 10 November 2006.
This Standard was published on 11 December 2006.

The following are represented on Committee IT-015:

Australian Computer Society
Australian Electrical and Electronic Manufacturers Association
Australian Society of Technical Communications
Australian Software Metrics Association
Engineers Australia/ACTS Joint Board in Software Engineering
Griffith University
National Association of Testing Authorities Australia
National ICT Australia
New Zealand Organisation for Quality
Software Quality Association, ACT
Software Quality Association, NSW
Systems Engineering Society of Australia
The University of Queensland
University of Auckland, NZ
University of South Australia
University of Technology, Sydney
Vendor Interests, NZ

Keeping Standards up-to-date

Standards are living documents which reflect progress in science, technology and systems. To maintain their currency, all Standards are periodically reviewed, and new editions are published. Between editions, amendments may be issued. Standards may also be withdrawn. It is important that readers assure themselves they are using a current Standard, which should include any amendments which may have been published since the Standard was purchased.

Detailed information about joint Australian/New Zealand Standards can be found by visiting the Standards Web Shop at www.standards.com.au or Standards New Zealand web site at www.standards.co.nz and looking up the relevant Standard in the on-line catalogue.

Alternatively, both organizations publish an annual printed Catalogue with full details of all current Standards. For more frequent listings or notification of revisions, amendments and withdrawals, Standards Australia and Standards New Zealand offer a number of update options. For information about these services, users should contact their respective national Standards organization.

We also welcome suggestions for improvement in our Standards, and especially encourage readers to notify us immediately of any apparent inaccuracies or ambiguities. Please address your comments to the Chief Executive of either Standards Australia or Standards New Zealand at the address shown on the back cover.

This Standard was issued in draft form for comment as DR 06506.

STANDARDS AUSTRALIA/STANDARDS NEW ZEALAND

RECONFIRMATION

OF

AS/NZS ISO/IEC 13235.3:2006

**Information technology—Open Distributed Processing—Trading function
Part 3: Provision of Trading Function using OSI Director Service**

RECONFIRMATION NOTICE

Technical Committee IT-015 has reviewed the content of this publication and in accordance with Standards Australia procedures for reconfirmation, it has been determined that the publication is still valid and does not require change.

Certain documents referenced in the publication may have been amended since the original date of publication. Users are advised to ensure that they are using the latest versions of such documents as appropriate, unless advised otherwise in this Reconfirmation Notice.

Approved for reconfirmation in accordance with Standards Australia procedures for reconfirmation on 30 October 2015.

Approved for reconfirmation in New Zealand on behalf of the Standards Council of New Zealand on 18 May 2016.

The following are represented on Technical Committee IT-015:

Australian Computer Society
Australian Society for Technical Communication (NSW)
Charles Sturt University
Department of Defence (Australian Government)
Griffith University
Institute of IT Professionals New Zealand
IT Service Management Forum Australia
La Trobe University
National Association of Testing Authorities Australia
NEHTA
New Zealand Organisation for Quality
NSW Business Chamber
Quantitative Enterprise Software Performance
Systems Engineering Society of Australia
University of Technology, Sydney

NOTES

Australian/New Zealand Standard™

**Information technology—Open
Distributed Processing—Trading
function**

**Part 3: Provision of Trading Function
using OSI Director Service**

First published as AS/NZS ISO/IEC 13235.3:2006.

COPYRIGHT

© Standards Australia/Standards New Zealand

All rights are reserved. No part of this work may be reproduced or copied in any form or by any means, electronic or mechanical, including photocopying, without the written permission of the publisher.

Jointly published by Standards Australia, GPO Box 476, Sydney, NSW 2001 and Standards New Zealand, Private Bag 2439, Wellington 6020

ISBN 0 7337 7905 0

PREFACE

This Standard was prepared by the Joint Standards Australia/Standards New Zealand Committee IT-015, Software and Systems Engineering.

The objective of this Standard is to provide organizations with a specification which describes how the Open Distributed Processing (ODP) Trading Function can be realised using information entries and support mechanisms of the Open Systems Interconnection (OSI) Directory. This Specification is to be used in conjunction with the ODP Trading Function Standard (ITU-T Rec. X.950. ISO/IEC 13235-1).

This Standard is identical with, and has been reproduced from ISO/IEC 13235-3:1998, *Information technology—Open Distributed Processing—Trading Function—Part 3: Provision of Trading Function using OSI Directory service*.

As this Standard is reproduced from an international standard, the following applies:

- (a) Its number appears on the cover and title page while the international standard number appears only on the cover.
- (b) In the source text ‘this Specification’ should read ‘this Australian/New Zealand Standard’.
- (c) A full point substitutes for a comma when referring to a decimal marker.

References to International Standards should be replaced by references to Australian or Australian/New Zealand Standards, as follows:

<i>Reference to International Standard</i>	<i>Australian/New Zealand Standard</i>
ISO/IEC	AS/NZS
9594 Information technology—Open Systems Interconnection—The Directory:	4019 Information technology—Open Systems Interconnection—The Directory
9594-1 Part 1: Overview of concepts, models and services	4019.1 Part 1: Overview of concepts, models and services
9594-2 Part 2: Models	4019.2 Part 2: Models
9594-3 Part 3: Abstract service definition	4019.3 Part 3: Abstract service definition
9594-4 Part 4: Procedures for distributed operation	4019.4 Part 4: Procedures for distributed operation
9594-5 Part 5: Protocol specifications	4019.5 Part 5: Protocol specifications
9594-6 Part 6: Selected attribute types	4019.6 Part 6: Selected attribute types
9594-7 Part 7: Selected object classes	4019.7 Part 7: Selected object classes
8824 Information technology—Abstract Syntax Notation One (ASN.1)	8824 Information technology—Abstract syntax notation one
8824-1 Part 1: Specification of basic notation	8824.1 Part 1: Specification of basic notation
8824-2 Part 2: Information object specification	8824.2 Part 2: Information object specification
8824-3 Part 3: Constraint specification	8824.3 Part 3: Constraint specification
8824-4 Part 4: Parameterization of ASN.1	8824.4 Part 4: Parameterization of ASN.1
13235 Information technology—Open Distributed Processing—Trading function	13235 Information technology—Open Distributed Processing—Trading function
13235-1 Part 1: Specification	13235.1 Part 1: Specification

Only the normative references that have been adopted as Australian or Australian/New Zealand Standards have been listed.

CONTENTS

	<i>Page</i>
1 Scope and field of application.....	1
2 Normative References.....	1
2.1 Identical Recommendations [International Standards].....	1
3 Definitions.....	2
4 Abbreviations.....	4
5 Overview.....	4
6 Schema.....	5
6.1 General.....	6
6.2 Trader Entry.....	7
6.2.1 commonName.....	7
6.2.2 traderInterface.....	8
6.2.3 dsaName	8
6.2.4 typeRepos	8
6.2.5 defSearchCard.....	8
6.2.6 maxSearchCard.....	8
6.2.7 defMatchCard	9
6.2.8 maxMatchCard.....	9
6.2.9 defReturnCard.....	9
6.2.10 maxReturnCard.....	9
6.2.11 defHopCount.....	10
6.2.12 maxHopCount.....	10
6.2.13 defFollowPolicy	10
6.2.14 maxFollowPolicy	11
6.2.15 maxLinkFollowPolicy	11
6.2.16 supportsModifiableProperties	11
6.2.17 supportsDynamicProperties	11
6.2.18 supportsProxyOffers	12
6.2.19 maxList	12
6.2.20 requestIdStem	12
6.2.21 description.....	12
6.2.22 userPassword	12
6.2.23 Other X.500 attributes.....	12
6.3 Trader Policy Entry	13
6.3.1 commonName.....	13
6.3.2 typeManagementConstraint	13
6.3.3 searchConstraint.....	14
6.3.4 offerAcceptanceConstraint	14
6.3.5 Other X.500 attributes.....	14

	<i>Page</i>
6.4 Service Offer Entry	14
6.4.1 sOfferId	15
6.4.2 serviceInterfaceId	16
6.4.3 serviceTypeId	16
6.4.4 hasDynamicProperties	16
6.4.5 hasModifiableProperties	17
6.4.6 dynamicProps	17
6.4.7 Other X.500 attributes	17
6.5 Trader Link Entry	18
6.5.1 linkName	18
6.5.2 linkId	18
6.5.3 targetTraderInterfaceId	19
6.5.4 defPassOnFollowRule	19
6.5.5 limitingFollowRule	19
6.5.6 Other X.500 attributes	19
6.6 Proxy Offer Entry	20
6.6.1 proxyOfferId	20
6.6.2 proxyLookUpInterfaceId	21
6.6.3 constraintRecipe	21
6.6.4 ifMatchAll	21
6.6.5 Other X.500 attributes	21
6.7 Other X.500 entries used by the T-DUA	22
7 Operations	22
7.1 Initialisation	23
7.2 Client operations	23
7.3 Register operations	23
7.3.1 Export	23
7.3.2 Withdraw	25
7.3.3 Modify	25
7.3.4 Describe	26
7.3.5 Withdraw with constraint	26
7.3.6 Resolve	27
7.4 Look up operations	27
7.4.1 Query operation	27
7.4.2 Policies	28
7.4.3 Searching locally	28
7.4.4 Searching Federated Traders	29
7.4.5 Searching Proxy Offers	29
7.4.6 Service Offer returned	29
7.5 Link operations	29
7.5.1 Add Link	29
7.5.2 Remove Link	30
7.5.3 Modify Link	30
7.5.4 Describe Link	31
7.5.5 List Links	31
7.6 Proxy Offer operations	31
7.6.1 Export Proxy	31
7.6.2 Withdraw Proxy	32
7.6.3 Describe Proxy	32

	<i>Page</i>
7.7 Trader Attribute Operations.....	33
7.8 Administrative operations.....	33
7.8.1 List Offers.....	33
7.8.2 List Proxies	34
7.9 Dynamic Property Evaluation operations	34
7.9.1 EvalDP	34
8 Type Repository	35
8.1 X.500 schema and the Minimal Type Repository	35
9 Dynamic Properties.....	36
9.1 Exporting a Service Offer.....	36
9.2 Importing a Service Offer.....	36
Annex A – Trader definitions schema definition	37
Annex B – Sample service description schema definition	47

INTRODUCTION

The ODP Trading Function (see ITU-T Rec. X.950-Series | ISO/IEC 13235) provides the means to offer a service and the means to discover services that have been offered. ITU-T Rec. X.950 | ISO/IEC 13235-1 defines an enterprise Specification, an information Specification and a computational Specification of this Trading Function. No engineering Specification is defined in ITU-T Rec. X.950 | ISO/IEC 13235-1. This Recommendation | International Standard describes how the Specifications of the Trading Function in ITU-T Rec. X.950 | ISO/IEC 13235-1 can be engineered using OSI Directory Service (see ITU-T Rec. X.500 | ISO/IEC 9594-1) to store information and to provide support mechanisms. This Specification does not prescribe that a trader must be engineered by using OSI Directory. But if OSI Directory is used, this Specification defines standardised templates for information entries (e.g. service offer and link information objects) in the Directory DIT.

Clause 5 gives an overview of how the Trading Function is implemented as a combination of X.500 DUA and DSA. The X.500 DSA is used to store the Trader Information Object and a Trader DUA (T-DUA) implements the functionality required by a Trader, which is difficult, or impossible, to implement using OSI Directory services.

Clause 6 defines the standardised templates for information entries of the Trader Information Object, the information known to a particular Trader.

Clause 7 describes mapping of Trading Function operations to appropriate Directory operations.

Clause 8 specifies a minimal Type Repository Function necessary to enable the correct functioning of the X.500 Directory for Trading.

Clause 9 describes the mechanisms used to enable the handling of dynamic properties of a Trader's service offers.

This Specification contains two annexes.

Annex A is a normative schema definition of Trader definitions.

Annex B is an informative schema definition of a sample service description.

AUSTRALIAN/NEW ZEALAND STANDARD

Information technology—Open Distributed Processing—Trading function

Part 3: Provision of Trading Function using OSI Director Service

1 Scope and field of application

This Specification describes how the ODP Trading Function can be realised using information entries and support mechanisms of the OSI Directory. This Specification is to be used in conjunction with the ODP Trading Function Standard (ITU-T Rec. X.950 | ISO/IEC 13235-1). If there are any discrepancies between the prescriptive statements in ITU-T Rec. X.950 | ISO/IEC 13235-1 and those in this Specification, the prescriptive statements in ITU-T Rec. X.950 | ISO/IEC 13235-1 take precedence.

The scope of this Specification is:

- standardised templates for Trading Function information objects in the DIT;
- descriptions of mapping of Trading Function operations to appropriate Directory operations;
- description of use of other Directory features to provide the support mechanisms for implementing the ODP Trading Function.

This Specification does not prescribe that a trader must be engineered by using OSI Directory. But if OSI Directory is used, this Specification defines standardised templates for information entries (e.g. service offer and link information objects) in the Directory DIT. This Specification does not put any restrictions on where these entries are placed in the Directory DIT. That is, this Specification does not standardise any structure rules. This Specification does describe a mechanism to provide the Trading Function using OSI Directory.

The field of application of this Specification is for the construction of the ODP Trading Function using the OSI Directory, when required.

2 Normative References

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.500 (1993) | ISO/IEC 9594-1:1995, *Information technology – Open Systems Interconnection – The Directory: Overview of concepts, models and services.*
- ITU-T Recommendation X.501 (1993) | ISO/IEC 9594-2:1995, *Information technology – Open Systems Interconnection – The Directory: Models.*
- ITU-T Recommendation X.509 (1993) | ISO/IEC 9594-8:1995, *Information technology – Open Systems Interconnection – The Directory: Authentication framework.*
- ITU-T Recommendation X.511 (1993) | ISO/IEC 9594-3:1995, *Information technology – Open Systems Interconnection – The Directory: Abstract service definition.*
- ITU-T Recommendation X.519 (1993) | ISO/IEC 9594-5:1995, *Information technology – Open Systems Interconnection – The Directory: Protocol specifications.*

- ITU-T Recommendation X.520 (1993) | ISO/IEC 9594-6:1995, *Information technology – Open Systems Interconnection – The Directory: Selected attribute types.*
- ITU-T Recommendation X.521 (1993) | ISO/IEC 9594-7:1995, *Information technology – Open Systems Interconnection – The Directory: Selected object classes.*
- ITU-T Recommendation X.680 (1994) | ISO/IEC 8824-1:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation.*
- ITU-T Recommendation X.681 (1994) | ISO/IEC 8824-2:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Information object specification.*
- ITU-T Recommendation X.682 (1994) | ISO/IEC 8824-3:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Constraint specification.*
- ITU-T Recommendation X.683 (1994) | ISO/IEC 8824-4:1995, *Information technology – Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications.*
- ITU-T Recommendation X.902 (1995) | ISO/IEC 10746-2:1996, *Information technology – Open distributed processing – Reference Model: Foundations.*
- ITU-T Recommendation X.903 (1995) | ISO/IEC 10746-3:1996, *Information technology – Open distributed processing – Reference Model: Architecture.*
- ITU-T Recommendation X.950 (1997) | ISO/IEC 13235-1¹⁾, *Information technology – Open distributed processing – Trading function: Specification.*

3 Definitions

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.902 | ISO/IEC 10746-2:

- activity;
- behaviour;
- client object;
- failure;
- identifier;
- instance;
- interaction;
- interface;
- interface signature;
- name;
- object;
- obligation;
- ODP system;
- policy;
- server object;
- subtype;
- <X> template;
- trading;
- type;
- viewpoint.

¹⁾ To be published.

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.903 | ISO/IEC 10746-3:

- administrator;
- community;
- computational viewpoint;
- engineering interface reference;
- engineering viewpoint;
- enterprise viewpoint;
- exporter;
- importer;
- information viewpoint;
- service export;
- service import;
- service offer;
- technology viewpoint;
- Trading Function;
- Type Repository Function.

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.950 | ISO/IEC 13235-1:

- federated traders;
- iterator;
- link;
- proxy offer;
- service type;
- service property;
- trader;
- trader attribute;
- trading graph.

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.500 | ISO/IEC 9594-1:

- Directory;
- Directory Information Base;
- (Directory) User.

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.501 | ISO/IEC 9594-2:

- attribute;
- attribute type;
- attribute value;
- Directory Information Tree;
- Directory System Agent;
- Directory User Agent;
- distinguished name;
- (Directory) entry;
- filter;

- matching rule;
- (Directory) name;
- name form;
- object;
- object class;
- object entry;
- relative distinguished name;
- structure rule;
- subclass;
- subordinate;
- superclass.

This Recommendation | International Standard makes use of the following operations defined in ITU-T Rec. X.511 | ISO/IEC 9594-3:

- addEntry;
- modifyEntry;
- read;
- removeEntry;
- search.

This Recommendation | International Standard makes use of the following terms defined in ITU-T Rec. X.509 | ISO/IEC 9594-8:

- authentication;
- password.

4 Abbreviations

For the purposes of this Recommendation | International Standard, the following abbreviations apply:

DIB	Directory Information Base
DIT	Directory Information Tree
DN	Distinguished Name
DSA	Directory System Agent
DUA	Directory User Agent
ODP	Open Distributed Processing
OID	Object Identifier
RDN	Relative Distinguished Name
T-DUA	Trader Directory User Agent

5 Overview

In this Specification, the Trading Function is implemented as a combination of X.500 DUA and DSA. As far as possible, the features of X.500 are used to directly implement the Trading Function, but not all Trader features can be directly supported by X.500. For this reason, the Trader (the object that provides the Trading Function) is composed of two components: an X.500 Directory which stores the Trader Information and a Trader DUA (T-DUA) which implements the functionality required by a Trader which is difficult, or impossible, to implement in X.500. The X.500 Directory is used to store the Trader Information Object. Requests from trader clients (importers and exporters) are mapped into operations on the X.500 database. Figure 1 shows the components of a Trader and its interactions with its clients.

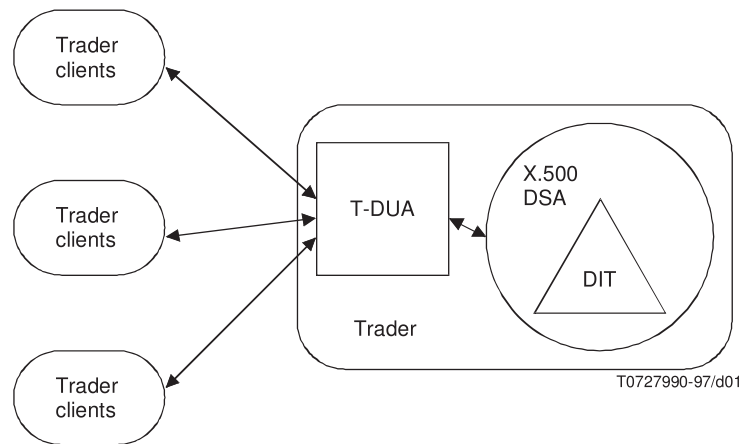


Figure 1 – The trader with its components and clients

The T-DUA and the trader clients (importers and exporters) communicate using a Trader protocol. The Trader protocol is not defined in this Specification. It may be any protocol which implements the functionality specified by ITU-T Rec. X.950 | ISO/IEC 13235-1. The purpose of this Specification is to specify how the T-DUA uses an X.500 Directory to support the functionality specified by ITU-T Rec. X.950 | ISO/IEC 13235-1.

The information stored by the X.500 Directory comprises:

- The Trader Attributes (i.e. information about the Trader itself).
- The Trader Enterprise policies (i.e. rules to determine and guide Trader behaviour).
- The set of Service Offers (i.e. information used by Trader when acting as a server).
- The set of Trader Links (i.e. information used by Trader when acting as a client).
- The set of Proxy Offers (i.e. information used by Trader when acting as a server for Proxy Offers).

X.500 is used to store this information for several reasons:

- The information model required by the ODP Trader is very similar to that provided by X.500.
- X.500 provides significant flexibility in allowing the definition of new X.500 attributes at runtime.
- It makes sense to use the existing investment in X.500 rather than to attempt to create a completely new infrastructure.
- It allows the Trader to use the general X.500 infrastructure to look up presentation addresses of linked Traders and Clients, and to use the security features of X.500 to authenticate users.

NOTE – Details of how to provide the X.500 infrastructure and the security features of X.500 for the Trading Function are outside the scope of this Specification.

It is not possible to implement an ODP Trader completely using X.500 because of the significant differences in the operational model used by the ODP Trader. These include:

- The Trader operations that do not directly map to X.500 operations.
- Distributed operations that are implemented using information stored in Trader Links and Trader Attributes whose meaning differ from the distribution implemented in X.500.

6 Schema

This X.500 schema describes the portion of an X.500 DIT used to store the information known to one Trader. The schema is based on the X.500 Directory model and is given in Annex A.

6.1 General

The information known to a particular Trader (the Trader Information Object) is kept in a subtree of the X.500 DIT. This subtree can be attached anywhere in the global DIT and no Structure Rules are defined for controlling its position. It is expected that the Trader subtree would be commonly attached beneath organisational and organisational unit entries (representing, respectively, the information known to organisational and organisational unit Traders). The information known to each trader is kept separately in the DIT and no attempt is made to map the distribution model used in X.500 to the very different distribution model of federated Traders.

Trader Information is stored in the X.500 DIT as self contained parcels. Each parcel contains the information known to one Trader. In the example shown in Figure 2, there are two Traders: one for the organisation as a whole and a second for a unit within the organisation. Linkages between these two Traders is via the Trader protocol, not via X.500 protocol.

The Trader Information Object (see Figure 3) is composed of five types of entries:

- The Trader Entry contains details about the Trader itself.
- The Trader Policy Entry contains details about the Trader enterprise policies.
- The Service Offer Entries contain details about the Service Offers known to the Trader.
- The Trader Link Entries contain details about the Links with other Traders.
- The Proxy Offer Entries contain details about the Proxy Offers known to the Trader.

NOTE 1 – The structuring of the Service Offers, Links, and Proxy Offers shown in Figure 3 is only one example of possible information structure.

NOTE 2 – In addition to the X.500 attributes listed in each entry, the presence of other attributes in an entry is not a violation of this Specification. Other X.500 attributes may be required for the following reasons:

- if a particular trader application requires specific additional X.500 attributes, they can be defined in that trader application Specification;
- if a particular trader implementation requires specific additional X.500 attributes, they can be defined in the documentation for that implementation.

Additional attributes can be included as Auxiliary Object Classes.

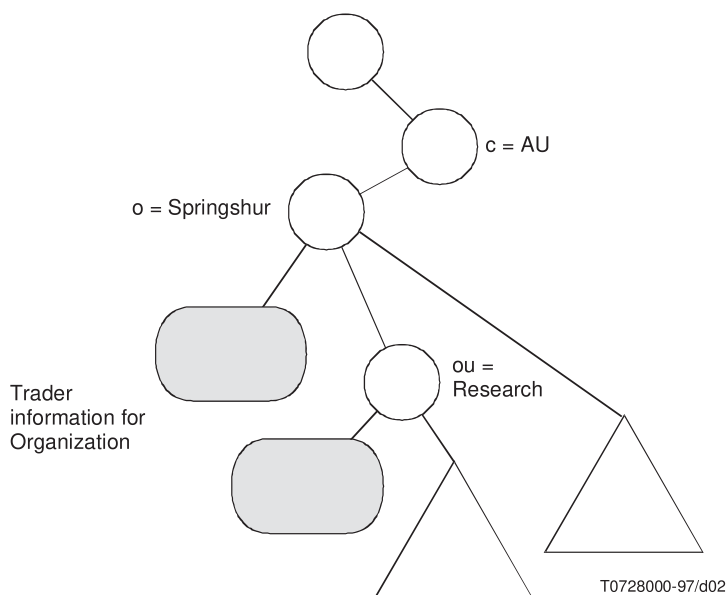


Figure 2 – An example of two traders stored in the X.500 DIT

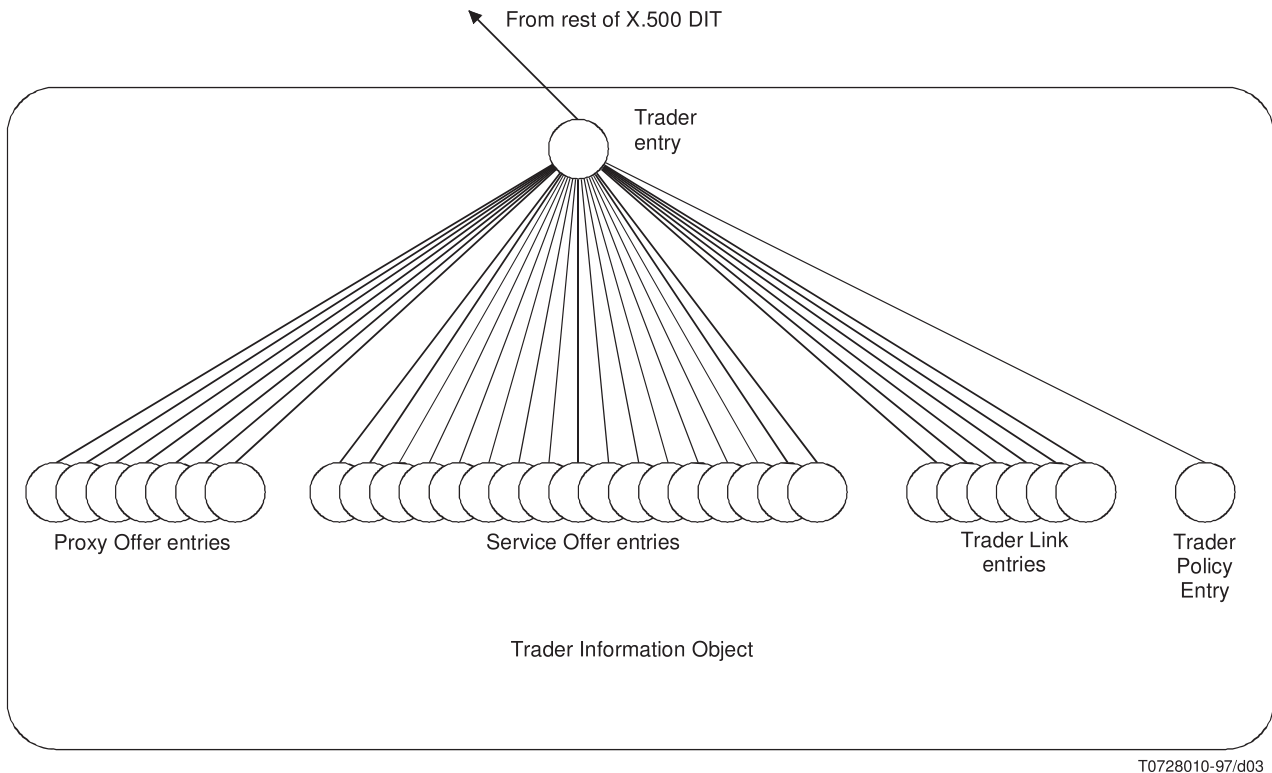


Figure 3 – An example of a Trader Information Object with five types of X.500 entry

6.2 Trader Entry

The root of the Trader subtree is the Trader Entry. This entry contains information about the Trader itself (the Trader Attributes – standardised Trader characteristics and trading policies) and is used as configuration information by the Trader (T-DUA) when it boots. The information is expressed as a set of X.500 attributes which represent Trader Attributes.

```
traderEntry OBJECT-CLASS ::= {
    SUBCLASS OF          {top}
    MUST CONTAIN          {commonName | traderInterface | dsaName | typeRepos |
                           defSearchCard | maxSearchCard | defMatchCard |
                           maxMatchCard | defReturnCard | maxReturnCard |
                           defHopCount | maxHopCount | defFollowPolicy |
                           maxFollowPolicy | maxLinkFollowPolicy |
                           supportsModifiableProperties | supportsDynamicProperties |
                           supportsProxyOffers | maxList | requestIdStem}
    MAY CONTAIN           {description | userPassword}
    ID                    id-trader-oc-traderEntry}
```

6.2.1 commonName

The name of this Trader. The commonName attribute forms the RDN of the Trader Entry. The full name of a Trader is the Distinguished Name of this entry (i.e. the full 'pathname' of the Trader Entry in the global X.500 DIT). The full Distinguished Name uniquely identifies this Trader amongst all other Traders in the X.500 Directory. This is a standard X.500 attribute defined in ITU-T Rec. X.520 | ISO/IEC 9594-6.

6.2.2 traderInterface

The address of the trader. The 'Address' is the Presentation Address at which this Trader can be contacted. This X.500 attribute is used by the Trader when booting as part of its configuration information and also by other Traders when they wish to distribute a Trader import amongst federated Traders.

```
traderInterface ATTRIBUTE ::= {
    SUBTYPE OF          presentationAddress
    SINGLE VALUE        TRUE
    ID                  id-trader-at-traderInterface}
```

6.2.3 dsaName

The name for the DSA associated with the Trader object.

```
dsaName ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-dsaName}
```

6.2.4 typeRepos

The name of the Type Repository used by the Trader for the repository of definitions of Service Types, Interface Types and Service Properties Types.

```
typeRepos ATTRIBUTE ::= {
    SUBTYPE OF          distinguishedName
    SINGLE VALUE        TRUE
    ID                  id-trader-at-typeRepos}
```

6.2.5 defSearchCard

The default upper bound of service offers to be considered before terminating a search. This value is used if none is specified by an importer. It must not exceed the value of maxSearchCard.

```
defSearchCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-defSearchCard }
```

6.2.6 maxSearchCard

The maximum upper bound of service offers a Trader considers before terminating any search.

```
maxSearchCard ATTRIBUTE ::= {
    WITH SYNTAX          INTEGER
    EQUALITY MATCHING RULE integerMatch
    SINGLE VALUE        TRUE
    ID                  id-trader-at-maxSearchCard}
```

6.2.7 defMatchCard

The default upper bound of matched offers found before a Trader terminates a search. This value is used if none is specified by an importer. It must not exceed the value of maxMatchCard.

```
defMatchCard ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE      integerMatch
    SINGLE VALUE                TRUE
    ID                          id-trader-at-defMatchCard}
```

6.2.8 maxMatchCard

The maximum upper bound of matched offers found before a Trader terminates any search.

```
maxMatchCard ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE      integerMatch
    SINGLE VALUE                TRUE
    ID                          id-trader-at-maxMatchCard}
```

6.2.9 defReturnCard

The default upper bound of service offers returned to an importer. This value is used if none is specified by an importer. It must not exceed the value of maxReturnCard.

```
defReturnCard ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE      integerMatch
    SINGLE VALUE                TRUE
    ID                          id-trader-at-defReturnCard}
```

6.2.10 maxReturnCard

The maximum upper bound of service offers returned to an importer.

```
maxReturnCard ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE      integerMatch
    SINGLE VALUE                TRUE
    ID                          id-trader-at-maxReturnCard}
```

6.2.11 defHopCount

The default upper bound of depth of Links to be traversed before terminating a search. This value is used if none is specified by an importer. It must not exceed maxHopCount.

```
defHopCount ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-defHopCount }
```

6.2.12 maxHopCount

The maximum upper bound of depth of Links to be traversed before terminating any search.

```
maxHopCount ATTRIBUTE ::= {
    WITH SYNTAX                INTEGER
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-maxHopCount }
```

6.2.13 defFollowPolicy

The default Link follow behaviour when no link follow behaviour is specified by an importer. The follow behaviour on a Link can be one of the following:

- local_only – Never follow unless explicitly named in an operation;
- if_no_local – Follow only if no local offer can match;
- always – Always follow except when overridden by some policies.

```
defFollowPolicy ATTRIBUTE ::= {
    WITH SYNTAX                FollowOption
    EQUALITY MATCHING RULE     integerMatch
    SINGLE VALUE               TRUE
    ID                          id-trader-at-defFollowPolicy }
```

```
FollowOption ::= ENUMERATED{
    localOnly    (0),
    ifNoLocal    (1),
    always       (2) }
```