

DIN EN IEC 62541-12

ICS 25.040.40; 35.200; 35.240.50

Einsprüche bis 2019-10-09

Entwurf

**OPC Unified Architecture –
Teil 12: Erkundung
(IEC 65E/615/CDV:2018);
Englische Fassung prEN IEC 62541-12:2018**

OPC Unified Architecture Specification –
Part 12: Discovery
(IEC 65E/615/CDV:2018);
English version prEN IEC 62541-12:2018
Spécification d'architecture unifiée OPC –
Partie 12: Services globaux et de découverte
(IEC 65E/615/CDV:2018);
Version anglaise prEN IEC 62541-12:2018

Anwendungswarnvermerk

Dieser Norm-Entwurf mit Erscheinungsdatum 2019-08-09 wird der Öffentlichkeit zur Prüfung und Stellungnahme vorgelegt.

Weil die beabsichtigte Norm von der vorliegenden Fassung abweichen kann, ist die Anwendung dieses Entwurfs besonders zu vereinbaren.

Stellungnahmen werden erbeten

- vorzugsweise online im Norm-Entwurfs-Portal von DIN unter www.din.de/go/entwuerfe bzw. für Norm-Entwürfe der DKE auch im Norm-Entwurfs-Portal der DKE unter www.entwuerfe.normenbibliothek.de, sofern dort wiedergegeben;
- oder als Datei per E-Mail an dke@vde.com möglichst in Form einer Tabelle. Die Vorlage dieser Tabelle kann im Internet unter www.din.de/go/stellungnahmen-norm-entwuerfe oder für Stellungnahmen zu Norm-Entwürfen der DKE unter www.dke.de/stellungnahme abgerufen werden;
- oder in Papierform an die DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE, Stresemannallee 15, 60596 Frankfurt am Main.

Die Empfänger dieses Norm-Entwurfs werden gebeten, mit ihren Kommentaren jegliche relevanten Patentrechte, die sie kennen, mitzuteilen und unterstützende Dokumentationen zur Verfügung zu stellen.

Gesamtumfang 100 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Nationales Vorwort

Die Englische Fassung des europäischen Dokuments prEN IEC 62541-12:2018 „OPC Unified Architecture – Teil 12: Erkundung“ (Entwurf in der Umfrage) ist unverändert in diesen Norm-Entwurf übernommen worden.

Die Internationale Elektrotechnische Kommission (IEC) und das Europäische Komitee für Elektrotechnische Normung (CENELEC) haben vereinbart, dass ein auf IEC-Ebene erarbeiteter Entwurf für eine Internationale Norm zeitgleich (parallel) bei IEC und CENELEC zur Umfrage (CDV-Stadium) und Abstimmung als FDIS (en: Final Draft International Standard) bzw. Schluss-Entwurf für eine Europäische Norm gestellt wird, um eine Beschleunigung und Straffung der Normungsarbeit zu erreichen. Dem entsprechend ist das internationale Dokument IEC 65E/615/CDV:2018 „OPC Unified Architecture Specification – Part 12: Discovery“ unverändert in den Entwurf prEN IEC 62541-12:2018 übernommen worden.

Das internationale Dokument wurde vom SC 65E „Devices and integration in enterprise systems“ der Internationalen Elektrotechnischen Kommission (IEC) erarbeitet.

Bei der Abstimmung zu dem Europäischen Schluss-Entwurf bei CENELEC und dem Internationalen Schluss-Entwurf bei IEC [Final Draft International Standard (FDIS)] sind jeweils nur „JA/NEIN“-Entscheidungen möglich, wobei „NEIN“-Entscheidungen fundiert begründet werden müssen. Dokumente, die bei CENELEC als Europäische Norm angenommen und ratifiziert werden, sind unverändert als Deutsche Normen zu übernehmen.

Für dieses Dokument ist das nationale Arbeitsgremium K 931 „Systemaspekte der Automatisierung“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE (www.dke.de) zuständig.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf ein Dokument ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils aktuellste Ausgabe des in Bezug genommenen Dokuments.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe des Dokuments.

Der Zusammenhang der zitierten Dokumente mit den entsprechenden deutschen Dokumenten ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Das Präsidium des DIN hat mit Präsidialbeschluss 1/2004 festgelegt, dass DIN-Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der Informationsverarbeitung gründet, unter bestimmten Bedingungen allein in englischer Sprache veröffentlicht werden dürfen. Diese Bedingungen sind für den vorliegenden Norm-Entwurf erfüllt.

Da sich die Benutzer des vorliegenden Norm-Entwurfs der englischen Sprache als Fachsprache bedienen, wird die englische Fassung der prEN IEC 62541-12 veröffentlicht. Zu deren Abschnitt 1, der den Anwendungsbereich festlegt, und Unterabschnitt 3.1, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA des vorliegenden Norm-Entwurfs hinzugefügt. Für einige der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Dieser Norm-Entwurf steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Es ist vorgesehen, auch bei der entsprechenden zukünftigen Deutschen Norm auf die deutsche Sprachfassung zu verzichten und diese in der englischsprachigen Fassung zu veröffentlichen.

Es wird auf die Möglichkeit hingewiesen, dass einige Elemente dieses Dokuments Patentrechte berühren können. IEC ist nicht dafür verantwortlich, einige oder alle diesbezüglichen Patentrechte zu identifizieren.

Nationaler Anhang NA (informativ)

Auszugsweise Übersetzung

Reihenfolge und Inhalt der folgenden Abschnitte sind identisch mit denen in den jeweiligen Abschnitten der Englischen Fassung.

1 Anwendungsbereich

Dieser Teil legt fest, wie die *Clients* und *Server* der OPC Unified Architecture (OPC UA) in verschiedenen Szenarien mit *DiscoveryServers* interagieren. Er legt die Anforderungen für den *LocalDiscoveryServer*, *LocalDiscoveryServer-ME* und *GlobalDiscoveryServer* fest. Er legt auch die Informationsmodelle für die *Zertifikatverwaltung*, *KeyCredential*-Verwaltung und *Autorisierungsdienste* fest.

3 Begriffe und Konventionen

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach IEC TR 62541-1, IEC TR 62541-2, IEC 62541-3, IEC 62541-4, IEC 62541-6, IEC 62541-9 und die folgenden Begriffe.

3.1.1

ZertifikatVerwaltungsServer

(en: CertificateManagementServer)

Softwareanwendung, die die von *Anwendungen* (en: *Applications*) in einem Verwaltungsbereich verwendeten *Zertifikate* verwaltet

3.1.2

ZertifikatGruppe

(en: CertificateGroup)

Kontext zur Beschreibung der *VertrauensListe* (en: *Trust List*) und des/der *Zertifikate(s)* in Zusammenhang mit einer *Anwendung*

3.1.3

ZertifikatAnforderung

(en: Certificate Request)

PKCS #10-codierte Struktur zum Anfordern eines neuen *Zertifikats* von einer *ZertifizierungsStelle* (en: *CertificateAuthority*)

3.1.4

Schlüsselinformationen

(en: KeyCredential)

eindeutiger und geheimer Bezeichner für den Zugriff auf einen *Server*, einen *AutorisierungDienst* (en: *Authorization Service*) oder einen *Broker*

Anmerkung 1 zum Begriff: Benutzername und Passwort sind Beispiele für Erkennungsmerkmale (en: credentials).

3.1.5

KeyCredentialService

(en: KeyCredentialService)

Softwareanwendung, die *KeyCredentials* für den Zugriff auf einen *Server*, einen *AutorisierungDienst* oder einen *Broker* bereitstellt

3.1.6**DirectoryService**

(en: DirectoryService)

Softwareanwendung oder eine Reihe von Anwendungen, die Informationen über Ressourcen wie Computer oder Dienste speichert und organisiert

3.1.7**DiscoveryServer**

(en: DiscoveryServer)

Anwendung, in der eine Liste von OPC-UA-*Servern* verwaltet wird, die in dem Netzwerk vorhanden sind und Mechanismen zum Abrufen der Liste durch Clients bereitstellt

3.1.8**DiscoveryUrl**

(en: DiscoveryUrl)

URL für einen Netzwerk-*Endpunkt* (en: *Endpoint*), die die Informationen zum Verbinden mit einem *Client* oder *Server* bereitstellt

3.1.9**GlobalDiscoveryServer (GDS)**

(en: GlobalDiscoveryServer (GDS))

DiscoveryServer, auf dem eine Liste von OPC-UA-*Anwendungen* verwaltet wird, die in einem Verwaltungsbereich verfügbar sind

Anmerkung 1 zum Begriff: Ein GDS kann auch Zertifikatverwaltungsdienste bereitstellen.

3.1.10**IPAddress**

(en: IPAddress)

eindeutige Nummer, die einer Netzwerkschnittstelle zugewiesen wird und die es ermöglicht, Internet Protocol (IP)-Anforderungen an diese Schnittstelle weiterzuleiten

Anmerkung 1 zum Begriff: Eine *IPAddress* eines Hosts kann sich nach einer bestimmten Zeit ändern.

3.1.11**LocalDiscoveryServer (LDS)**

(en: LocalDiscoveryServer (LDS))

DiscoveryServer, auf dem eine Liste aller *Server* verwaltet wird, die auf ihm registriert sind

Anmerkung 1 zum Begriff: *Server* registrieren sich üblicherweise mit dem LDS auf demselben Host.

3.1.12**LocalDiscoveryServer-ME (LDS-ME)**

(en: LocalDiscoveryServer(LDS-ME))

LocalDiscoveryServer, der die MulticastExtension beinhaltet

3.1.13**MulticastExtension**

(en: MulticastExtension)

Erweiterung eines *LocalDiscoveryServer*, mit der die Unterstützung für das mDNS-Protokoll hinzugefügt wird

3.1.14**MulticastSubnet**

(en: MulticastSubnet)

Netzwerk, mit dem ermöglicht wird, Multicast-Pakete an alle mit dem Netzwerk verbundene Knoten zu senden

Anmerkung 1 zum Begriff: Ein *MulticastSubnet* ist nicht zwingend das Gleiche wie ein TCP/IP-Subnet.

3.1.15

NetzwerkDienst

(en: NetworkService)

gesicherte Ressource in einem Netzwerk, die Funktionen bereitstellt, die von *Clients* und/oder *Servern* verwendet werden

Anmerkung 1 zum Begriff: Ein *AutorisierungDienst* (en: *Authorization Service (AS)*) ist ein Beispiel für einen *NetzwerkDienst*.

3.1.16

ServerCapabilityIdentifier

(en: ServerCapabilityIdentifier)

kurzer Bezeichner, der eine Reihe von ermittelbaren Fähigkeiten, die von einem Server unterstützt werden, eindeutig identifiziert

Anmerkung 1 zum Begriff: Die Liste der derzeit definierten *ServerCapabilityIdentifiers* befindet sich in Anhang D.

– Leerseite –

CONTENTS

	Page
FIGURES.....	6
TABLES.....	7
FOREWORD.....	9
1 Scope.....	11
2 Normative references	11
3 Terms, definitions, and conventions.....	12
3.1 Terms and definitions.....	12
3.2 Abbreviations and symbols	14
3.3 Conventions for Namespaces.....	14
4 The Discovery Process	15
4.1 Overview.....	15
4.2 Registration and announcement of Applications	15
4.2.1 Overview	15
4.2.2 Hosts with a LocalDiscoveryServer	15
4.2.3 Hosts without a LocalDiscoveryServer	16
4.3 The Discovery Process for Clients to find Servers	16
4.3.1 Overview	16
4.3.2 Security	17
4.3.3 Simple Discovery with a DiscoveryUrl	17
4.3.4 Local Discovery	17
4.3.5 MulticastSubnet Discovery.....	18
4.3.6 Global Discovery	19
4.3.7 Combined Discovery Process for Clients	19
5 Local Discovery Server.....	20
5.1 Overview.....	20
5.2 Security considerations for Multicast DNS.....	21
6 Global Discovery Server	21
6.1 Overview.....	21
6.2 Network Architectures	22
6.2.1 Overview	22
6.2.2 Single MulticastSubnet	23
6.2.3 Multiple MulticastSubnet.....	23
6.2.4 No MulticastSubnet.....	24
6.2.5 Domain Names and MulticastSubnets	24
6.3 Information Model	25
6.3.1 Overview	25
6.3.2 Directory.....	25
6.3.3 DirectoryType	26
6.3.4 FindApplications	26
6.3.5 ApplicationRecordDataType.....	27
6.3.6 RegisterApplication.....	27
6.3.7 UpdateApplication	28
6.3.8 UnregisterApplication	29
6.3.9 GetApplication	29
6.3.10 QueryApplications	30

6.3.11	QueryServers (deprecated).....	31
6.3.12	ApplicationRegistrationChangedAuditEventType.....	32
7	Certificate Management Overview	33
7.1	Overview.....	33
7.2	Pull Management	34
7.3	Push Management	35
7.4	Provisioning	36
7.5	Common Information Model	36
7.5.1	Overview	36
7.5.2	TrustListType.....	36
7.5.3	OpenWithMasks	37
7.5.4	CloseAndUpdate.....	38
7.5.5	AddCertificate	38
7.5.6	RemoveCertificate	39
7.5.7	TrustListDataType	40
7.5.8	TrustListMasks	40
7.5.9	TrustListOutOfDateAlarmType	40
7.5.10	CertificateGroupType.....	41
7.5.11	CertificateType	41
7.5.12	ApplicationCertificateType	42
7.5.13	HttpsCertificateType	42
7.5.14	UserCredentialCertificateType	42
7.5.15	RsaMinApplicationCertificateType	42
7.5.16	RsaSha256ApplicationCertificateType	43
7.5.17	CertificateGroupFolderType.....	43
7.5.18	TrustListUpdatedAuditEventType.....	43
7.6	Information Model for Pull Certificate Management	44
7.6.1	Overview	44
7.6.2	CertificateDirectoryType	44
7.6.3	StartSigningRequest.....	45
7.6.4	StartNewKeyPairRequest	47
7.6.5	FinishRequest	49
7.6.6	GetCertificateGroups	49
7.6.7	GetTrustList.....	50
7.6.8	GetCertificateStatus	51
7.6.9	CertificateRequestedAuditEventType.....	52
7.6.10	CertificateDeliveredAuditEventType.....	52
7.7	Information Model for Push Certificate Management	53
7.7.1	Overview	53
7.7.2	ServerConfiguration.....	53
7.7.3	ServerConfigurationType	54
7.7.4	UpdateCertificate.....	55
7.7.5	ApplyChanges	56
7.7.6	CreateSigningRequest.....	56
7.7.7	GetRejectedList.....	57
7.7.8	CertificateUpdatedAuditEventType	58
8	KeyCredential Management.....	58
8.1	Overview.....	58
8.2	Pull Management	59

8.3	Push Management	59
8.4	Information Model for Pull Management	60
8.4.1	Overview	60
8.4.2	KeyCredentialManagement	61
8.4.3	KeyCredentialServiceType	61
8.4.4	StartRequest	62
8.4.5	FinishRequest	63
8.4.6	Revoke	64
8.4.7	KeyCredentialAuditEventType	64
8.4.8	KeyCredentialRequestedAuditEventType	65
8.4.9	KeyCredentialDeliveredAuditEventType	65
8.4.10	KeyCredentialRevokedAuditEventType	66
8.5	Information Model for Push Management	66
8.5.1	KeyCredentialConfiguration	67
8.5.2	KeyCredentialConfigurationType	67
8.5.3	UpdateCredential	67
8.5.4	DeleteCredential	68
8.5.5	KeyCredentialUpdatedAuditEventType	69
8.5.6	KeyCredentialDeletedAuditEventType	69
9	Authorization Services	69
9.1	Overview	69
9.2	Implicit	70
9.3	Explicit	71
9.4	Chained	72
9.5	Information Model for Requesting Access Tokens	73
9.5.1	Overview	73
9.5.2	AuthorizationServices	74
9.5.3	AuthorizationServiceType	74
9.5.4	RequestAccessToken	75
9.5.5	GetServiceDescription	76
9.5.6	AccessTokenIssuedAuditEventType	76
9.6	Information Model for Configuring Servers	77
9.6.1	Overview	77
9.6.2	AuthorizationServices	77
9.6.3	AuthorizationServiceConfigurationType	77
Annex A (informative)	Deployment and Configuration	79
A.1	Firewalls and Discovery	79
A.2	Resolving references to remote Servers	81
Annex B (normative)	Constants	83
B.1	Numeric Node Ids	83
Annex C (normative)	OPC UA Mapping to mDNS	84
C.1	DNS Server (SRV) Record syntax	84
C.2	DNS Text (TXT) Record syntax	84
C.3	DiscoveryUrl mapping	85
Annex D (normative)	Server Capability Identifiers	86
Annex E (normative)	DirectoryServices	87
E.1	Global Discovery via other directory services	87
E.2	UDDI	87

E.3	LDAP	88
Annex F (normative)	Local Discovery Server	90
F.1	Certificate store directory layout	90
F.2	Installation directories on Windows	90
Annex G (normative)	Application installation process	92
G.1	Provisioning with Pull Management	92
G.2	Provisioning with the Push Management	92
G.3	Setting permissions	93
Annex H (informative)	Comparison with RFC 7030	94
H.1	Overview	94
H.2	Obtaining CA Certificates	94
H.3	Initial enrolment	94
H.4	Client Certificate reissuance	95
H.5	Server key generation	95
H.6	Certificate Signing Request (CSR) attributes request	95

FIGURES

Figure 1 – The Registration process with an LDS	16
Figure 2 – The simple Discovery process	17
Figure 3 – The Local Discovery process	18
Figure 4 – The MulticastSubnet Discovery process	18
Figure 5 – The Global Discovery process	19
Figure 6 – The Discovery Process for Clients	20
Figure 7 – The relationship between GDS and other components	22
Figure 8 – The Single MulticastSubnet architecture	23
Figure 9 – The Multiple MulticastSubnet architecture	24
Figure 10 – The No MulticastSubnet architecture	24
Figure 11 – The Address Space for the GDS	25
Figure 12 – The Pull Certificate management model	34
Figure 13 – The Push Certificate management model	35
Figure 14 – The Certificate Management AddressSpace for the GlobalDiscoveryServer	44
Figure 15 – The AddressSpace for the Server that supports Push Management	53
Figure 16 – The Pull Model for KeyCredential management	59
Figure 17 – The Push Model for KeyCredential management	60
Figure 18 – The Address Space used for Pull KeyCredential management	61
Figure 19 – The Address Space used for Push KeyCredential management	66
Figure 20 – Roles and Authorization Services	70
Figure 21 – Implicit authorization	71
Figure 22 – Explicit authorization	72
Figure 23 – Chained authorization	73
Figure 24 – The Model for Requesting Access Tokens from Authorization Services	74
Figure 25 – The Model for Configuring Servers to use Authorization Services	77
Figure 26 – Discovering Servers outside a firewall	79
Figure 27 – Discovering Servers behind a firewall	80
Figure 28 – Using a Discovery Server with a firewall	81
Figure 29 – Following References to Remote Servers	82
Figure 30 – The UDDI or LDAP Discovery process	87
Figure 31 – UDDI registry structure	88
Figure 32 – Sample LDAP hierarchy	89

TABLES

Table 1 – GDS NamespaceMetadataType Object definition	14
Table 2 – Directory Object definition	25
Table 3 – DirectoryType definition.....	26
Table 4 – FindApplications Method AddressSpace definition.....	27
Table 5 – ApplicationRecordDataType definition	27
Table 6 – RegisterApplication Method AddressSpace definition	28
Table 7 – UpdateApplication Method AddressSpace definition	29
Table 8 – UnregisterApplication Method AddressSpace definition	29
Table 9 – GetApplication Method AddressSpace definition.....	30
Table 10 – QueryApplications Method AddressSpace definition	31
Table 11 – QueryServers Method AddressSpace definition	32
Table 12 – ApplicationRegistrationChangedAuditEventType definition	33
Table 13 – TrustListType definition	36
Table 14 – OpenWithMasks Method AddressSpace definition	37
Table 15 – CloseAndUpdate Method AddressSpace definition	38
Table 16 – AddCertificate Method AddressSpace definition	39
Table 17 – RemoveCertificate Method AddressSpace definition.....	40
Table 18 – TrustListDataType definition.....	40
Table 19 – TrustListMasks values	40
Table 20 – TrustListOutOfDateAlarmType definition.....	40
Table 21 – CertificateGroupType definition	41
Table 22 – CertificateType definition.....	41
Table 23 – ApplicationCertificateType definition.....	42
Table 24 – HttpsCertificateType definition.....	42
Table 25 – UserCredentialCertificateType definition.....	42
Table 26 – RsaMinApplicationCertificateType definition	42
Table 27 – RsaSha256ApplicationCertificateType definition.....	43
Table 28 – CertificateGroupFolderType definition	43
Table 29 – TrustListUpdatedAuditEventType definition	44
Table 30 – CertificateDirectoryType ObjectType definition	45
Table 31 – StartSigningRequest Method AddressSpace definition.....	47
Table 32 – StartNewKeyPairRequest Method AddressSpace definition	48
Table 33 – FinishRequest Method AddressSpace definition	49
Table 34 – GetCertificateGroups Method AddressSpace definition.....	50
Table 35 – GetTrustList Method AddressSpace definition	51
Table 36 – GetCertificateStatus Method AddressSpace definition	52
Table 37 – CertificateRequestedAuditEventType definition	52
Table 38 – CertificateDeliveredAuditEventType definition	52
Table 39 – ServerConfiguration Object definition	53
Table 40 – ServerConfigurationType definition.....	54
Table 41 – UpdateCertificate Method AddressSpace Definition	56
Table 42 – ApplyChanges Method AddressSpace Definition	56

Table 43 – CreateSigningRequest Method AddressSpace definition.....	57
Table 44 – GetRejectedList Method AddressSpace definition.....	58
Table 45 – CertificateUpdatedAuditEventType definition	58
Table 46 – KeyCredentialManagement Object definition	61
Table 47 – KeyCredentialServiceType definition	61
Table 48 – StartRequest Method AddressSpace definition	63
Table 49 – FinishRequest Method AddressSpace definition	64
Table 50 – Revoke Method AddressSpace definition.....	64
Table 51 – KeyCredentialAuditEventType definition	65
Table 52 – KeyCredentialRequestedAuditEventType definition	65
Table 53 – KeyCredentialDeliveredAuditEventType definition	66
Table 54 – KeyCredentialRevokedAuditEventType definition	66
Table 55 – KeyCredentialConfiguration Object definition.....	67
Table 56 – KeyCredentialConfigurationType definition	67
Table 57 – UpdateCredential Method AddressSpace definition	68
Table 58 – DeleteCredential Method AddressSpace definition	69
Table 59 – KeyCredentialUpdatedAuditEventType definition	69
Table 60 – KeyCredentialUpdatedAuditEventType definition	69
Table 61 – AuthorizationServices Object definition	74
Table 62 – AuthorizationServiceType definition.....	74
Table 63 – RequestAccessToken Method AddressSpace definition	76
Table 64 – GetServiceDescription Method AddressSpace definition.....	76
Table 65 – AccessTokenIssuedAuditEventType definition	76
Table 66 – AuthorizationServices Object definition	77
Table 67 – AuthorizationServiceConfigurationType definition	78
Table 68 – Allowed mDNS service names	84
Table 69 – DNS TXT Record string format	85
Table 70 – DiscoveryUrl to DNS SRV and TXT Record Mapping	85
Table 71 – Examples of <i>ServerCapabilityIdentifiers</i>	86
Table 72 – UDDI tModels	88
Table 73 – LDAP object class schema	89
Table 74 – Application Certificate store directory layout.....	90
Table 75 – Verifying that a Server is allowed to provide Certificates	94
Table 76 – Verifying that a Client is allowed to request Certificates	94

INTERNATIONAL ELECTROTECHNICAL COMMISSION

OPC UNIFIED ARCHITECTURE –

Part 12: Discovery and Global Services

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

The main task of IEC technical committees is to prepare International Standards. However, a technical committee may propose the publication of a technical report when it has collected data of a different kind from that which is normally published as an International Standard, for example "state of the art".

IEC 62541-12 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

The text of this technical report is based on the following documents:

Enquiry draft	Report on voting
65E/XX/DTR	65E/XX/RVC

Full information on the voting for the approval of this technical report can be found in the report on voting indicated in the above table.

This is the first edition of edition of IEC 62541-12.

51

52 This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

53 Throughout this document and the referenced other Parts of the series, certain document
54 conventions are used:

55 Italics are used to denote a defined term or definition that appears in the “Terms and definition”
56 clause in one of the parts of the series.

57 Italics are also used to denote the name of a service input or output parameter or the name of a
58 structure or element of a structure that are usually defined in tables.

59 The italicized terms and names are also often written in camel-case (the practice of writing
60 compound words or phrases in which the elements are joined without spaces, with each element's
61 initial letter capitalized within the compound). For example the defined term is AddressSpace
62 instead of Address Space. This makes it easier to understand that there is a single definition for
63 AddressSpace, not separate definitions for Address and Space.

64 A list of all parts of the IEC 62541 series, published under the general title *IEC template*
65 *iecstd.dot*, can be found on the IEC website.

66 The committee has decided that the contents of this publication will remain unchanged until the
67 stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the
68 specific publication. At this date, the publication will be

- 69 • reconfirmed,
- 70 • withdrawn,
- 71 • replaced by a revised edition, or
- 72 • amended.

73

74 The National Committees are requested to note that for this publication the stability date
75 is 2021.

76 THIS TEXT IS INCLUDED FOR THE INFORMATION OF THE NATIONAL COMMITTEES AND WILL BE DELETED AT
77 THE PUBLICATION STAGE.

78 A bilingual version of this publication may be issued at a later date.

79

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

80

81

OPC UNIFIED ARCHITECTURE

Part 12: Discovery and Global Services

1 Scope

This part specifies how OPC Unified Architecture (OPC UA) *Clients* and *Servers* interact with *DiscoveryServers* when used in different scenarios. It specifies the requirements for the *LocalDiscoveryServer*, *LocalDiscoveryServer-ME* and *GlobalDiscoveryServer*. It also defines information models for *Certificate* management, *KeyCredential* management and *Authorization Services*.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC TR 62541-1, *OPC Unified Architecture – Part 1: Overview and Concepts*

IEC TR 62541-2, *OPC Unified Architecture – Part 2: Security Model*

IEC 62541-3, *OPC Unified Architecture – Part 3: Address Space Model*

IEC 62541-4, *OPC Unified Architecture – Part 4: Services*

IEC 62541-5, *OPC Unified Architecture – Part 5: Information Model*

IEC 62541-6, *OPC Unified Architecture – Part 6: Mappings*

IEC 62541-7, *OPC Unified Architecture – Part 7: Profiles*

IEC 62541-9, *OPC Unified Architecture – Part 9: Alarms and Conditions*

IEC 62541-14, *OPC Unified Architecture – Part 14: PubSub*

Auto-IP: Dynamic Configuration of IPv4 Link-Local Addresses

<http://www.ietf.org/rfc/rfc3927.txt>

DNS-Name: Domain Names – Implementation and Specification

<http://www.ietf.org/rfc/rfc1035.txt>

DHCP: Dynamic Host Configuration Protocol

<http://www.ietf.org/rfc/rfc2131.txt>

mDNS: Multicast DNS

<http://www.ietf.org/rfc/rfc6762.txt>

DNS-SD: DNS Based Service Discovery

<http://www.ietf.org/rfc/rfc6763.txt>

RFC 5958: Asymmetric Key Packages

<http://www.ietf.org/rfc/rfc5958.txt>

- 119 PKCS #10: Certification Request Syntax Specification
120 <http://www.ietf.org/rfc/rfc2986.txt>
- 121 PKCS #12: Personal Information Exchange Syntax
122 [http://www.emc.com/emc-plus/rsa-labs/pkcs/files/h11301-wp-pkcs-12v1-1-personal-](http://www.emc.com/emc-plus/rsa-labs/pkcs/files/h11301-wp-pkcs-12v1-1-personal-information-exchange-syntax.pdf)
123 [information-exchange-syntax.pdf](http://www.emc.com/emc-plus/rsa-labs/pkcs/files/h11301-wp-pkcs-12v1-1-personal-information-exchange-syntax.pdf)
- 124 [RFC 7030: Enrollment over Secure Transport](http://www.ietf.org/rfc/rfc7030.txt)
125 <http://www.ietf.org/rfc/rfc7030.txt>
- 126 DI: OPC Unified Architecture for Devices (DI)
127 [https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-](https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-architecture-for-devices-di/)
128 [architecture-for-devices-di/](https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-architecture-for-devices-di/)
- 129 ADI: OPC Unified Architecture for Analyzer Devices (ADI)
130 [https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-](https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-architecture-for-analyzer-devices-adi/)
131 [architecture-for-analyzer-devices-adi/](https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-architecture-for-analyzer-devices-adi/)
- 132 PLCopen: OPC Unified Architecture / PLCopen Information Model
133 [https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-](https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-architecture-plcopen-information-model/)
134 [architecture-plcopen-information-model/](https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-architecture-plcopen-information-model/)
- 135 FDI: OPC Unified Architecture for FDI
136 [https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-](https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-architecture-for-fdi/)
137 [architecture-for-fdi/](https://opcfoundation.org/developer-tools/specifications-unified-architecture/opc-unified-architecture-for-fdi/)
- 138 ISA-95: ISA-95 Common Object Model
139 [https://opcfoundation.org/developer-tools/specifications-unified-architecture/isa-95-](https://opcfoundation.org/developer-tools/specifications-unified-architecture/isa-95-common-object-model/)
140 [common-object-model/](https://opcfoundation.org/developer-tools/specifications-unified-architecture/isa-95-common-object-model/)
- 141 X.500: ISO/IEC 9594-1:2017 – The Directory
142 <https://www.iso.org/standard/72550.html>

143 3 Terms, definitions, and conventions

144 3.1 Terms and definitions

145 For the purposes of this document the following terms and definitions as well as the terms and
146 definitions given in IEC TR 62541-1, IEC TR 62541-2, IEC 62541-3, IEC 62541-4, IEC 62541-
147 6 and IEC 62541-9 apply.

148 3.1.1

149 **CertificateManagement Server**

150 a software application that manages the *Certificates* used by *Applications* in an administrative
151 domain.

152 3.1.2

153 **Certificate Group**

154 a context used to describe the *Trust List* and *Certificate(s)* associated with an *Application*.

155 3.1.3

156 **Certificate Request**

157 a PKCS #10 encoded structure used to request a new *Certificate* from a *Certificate Authority*.

158 3.1.4

159 **KeyCredential**

160 a unique identifier and a secret used to access a *Server*, an *Authorization Service* or a
161 *Broker*.

162 Note 1 to entry: a user name and password is an example of a credential.

163 **3.1.5**164 **KeyCredentialService**

165 a software application that provides *KeyCredentials* needed to access a *Server*, an
166 *Authorization Service* or a *Broker*.

167 **3.1.6**168 **DirectoryService**

169 a software application, or a set of applications, that stores and organizes information about
170 resources such as computers or services.

171 **3.1.7**172 **DiscoveryServer**

173 an *Application* that maintains a list of OPC UA *Servers* that are available on the network and
174 provides mechanisms for *Clients* to obtain this list.

175 **3.1.8**176 **DiscoveryUrl**

177 a URL for a network *Endpoint* that provides the information required to connect to a *Client* or
178 *Server*.

179 **3.1.9**180 **GlobalDiscoveryServer (GDS)**

181 a *DiscoveryServer* that maintains a list of OPC UA *Applications* available in an administrative
182 domain.

183 Note 1 to entry: a GDS may also provide certificate management services.

184 **3.1.10**185 **IPAddress**

186 a unique number assigned to a network interface that allows Internet Protocol (IP) requests to
187 be routed to that interface.

188 Note 1 to entry: An *IPAddress* for a host may change over time.

189 **3.1.11**190 **LocalDiscoveryServer (LDS)**

191 a *DiscoveryServer* that maintains a list of all *Servers* that have registered with it.

192 Note 1 to entry: *Servers* normally register with the LDS on the same host.

193 **3.1.12**194 **LocalDiscoveryServer-ME (LDS-ME)**

195 a *LocalDiscoveryServer* that includes the *MulticastExtension*.

196 **3.1.13**197 **MulticastExtension**

198 an extension to a *LocalDiscoveryServer* that adds support for the mDNS protocol.

199 **3.1.14**200 **MulticastSubnet**

201 a network that allows multicast packets to be sent to all nodes connected to the network.

202 Note 1 to entry: a *MulticastSubnet* is not necessarily the same as a TCP/IP subnet.

203 **3.1.15**204 **Network Service**

205 a secured resource on a network that provides functionality used by *Clients* and/or *Servers*.

Note 1 to entry: an *Authorization Service* (AS) is an example of a *Network Service*.

3.1.16

ServerCapabilityIdentifier

a short identifier which uniquely identifies a set of discoverable capabilities supported by a *Server*.

Note 1 to entry: the list of the currently defined *ServerCapabilityIdentifiers* is in Annex D.

212

3.2 Abbreviations and symbols

API	Application Programming Interface
CA	Certificate Authority
CRL	Certificate Revocation List
CSR	Certificate Signing Request
DER	Distinguished Encoding Rules
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
EST	Enrolment over Secure Transport
GDS	Global Discovery Server
IANA	The Internet Assigned Numbers Authority
LDAP	Lightweight Directory Access Protocol
LDS	Local Discovery Server
LDS-ME	Local Discovery Server with the Multicast Extension
mDNS	Multicast Domain Name System
NAT	Network Address Translation
PEM	Privacy Enhanced Mail
PFX	Personal Information Exchange
PKCS	Public Key Cryptography Standards
SHA1	Secure Hash Algorithm
SSL	Secure Socket Layer
TLS	Transport Layer Security
UA	Unified Architecture
UDDI	Universal Description, Discovery and Integration

3.3 Conventions for Namespaces

This standard uses multiple namespaces to define *Nodes*. The following abbreviations are used in the definitions for these *Nodes*:

CORE	http://opcfoundation.org/UA/
GDS	http://opcfoundation.org/UA/GDS/

The default namespace for each *Node* is defined at the top of the table. All of the *BrowseNames* in the table use the default namespace unless the *BrowseName* is preceded by one of the above abbreviations.

The *NamespaceMetadataType Object* for the GDS namespace is defined Table 1.

Table 1 – GDS NamespaceMetadataType Object definition

Attribute	Value		
BrowseName	http://opcfoundation.org/UA/GDS/		
Namespace	GDS (see 3.3)		
TypeDefinition	NamespaceMetadataType defined in IEC 62541-5.		
References	NodeClass	BrowseName	Value
HasProperty	Variable	NamespaceUri	http://opcfoundation.org/UA/GDS/
HasProperty	Variable	NamespaceVersion	1.04
HasProperty	Variable	NamespacePublicationDate	2016-12-31

251

252

4 The Discovery Process

4.1 Overview

The discovery process allows applications to find other applications on the network and then discover how to connect to them. Note that this discussion builds on the discovery related concepts defined in IEC 62541-4. Discoverable applications are generally *Servers*, however, some *Clients* will support reverse connections as described in IEC 62541-6 and want *Servers* to be able to discover them.

Clients and *Servers* can be on the same host, on different hosts in the same subnet, or even on completely different locations in an administrative domain. The following clauses describe the different configurations and how discovery can be accomplished.

The mechanisms for *Clients* to discover *Servers* are specified in 4.3.

The mechanisms for *Servers* to make themselves discoverable are specified in 4.2.

The *Discovery Services* are specified in IEC 62541-4. They are implemented by individual *Servers* and by dedicated *DiscoveryServers*. The following dedicated *DiscoveryServers* provide a way for applications to discover registered OPC UA applications in different situations:

- A *LocalDiscoveryServer* (LDS) maintains discovery information for all applications that have registered with it, usually all applications available on the host that it runs on.
- A *LocalDiscoveryServer* with the *MulticastExtension* (LDS-ME) maintains discovery information for all applications that have been announced on the local *MulticastSubnet*.
- A *GlobalDiscoveryServer* (GDS) maintains discovery information for applications available in an administrative domain.

LDS and LDS-ME are specified in Clause 5. The GDS is specified in Clause 6.

4.2 Registration and announcement of Applications

4.2.1 Overview

The clause describes how an application registers itself so it can be discovered. Most *Applications* will want other applications to discover them. *Applications* that do not wish to be discovered openly should not register with a *DiscoveryServer*. In this case such *Applications* should only publish a *DiscoveryUrl* via some out-of-band mechanism to be discovered by specific *Applications*.

4.2.2 Hosts with a LocalDiscoveryServer

Applications register themselves with the LDS on the same host if they wish to be discovered. The registration ensures that the applications is visible for local discovery (see 4.3.4) and *MulticastSubnet* discovery if the LDS is a LDS-ME (see 4.3.5).

The OPC UA Standard (IEC 62541-4) defines a *RegisterServer2 Service* which provides additional registration information. All *Applications* and the *LocalDiscoveryServer* shall support the *RegisterServer2 Service* and, for backwards compatibility, the older *RegisterServer Service*. If an *Application* encounters an older LDS that returns a *Bad_ServiceUnsupported* error when calling *RegisterServer2 Service* it shall try again with *RegisterServer Service*.

294 The *RegisterServer2* Service allows the *Application* to specify zero or more *ServerCapabilityIdentifiers*. *ServerCapabilityIdentifiers* are short, string identifiers of well-known OPC UA features. *Applications* can use these identifiers as a filter during discovery.

297 The set of known *ServerCapabilityIdentifiers* is specified in Annex D and is limited to features which are considered to be important enough to report before an application makes a connection. For example, support for the GDS information model or the Alarms information model are *Server* capabilities that have a *ServerCapabilityIdentifier* defined.

301 Before an application registers with the LDS it should call the *GetEndpoints* Service and choose the most secure endpoint supported by the LDS and then call *RegisterServer2* or *RegisterServer*.

304 Registration with LDS or LDS-ME is illustrated in Figure 1.

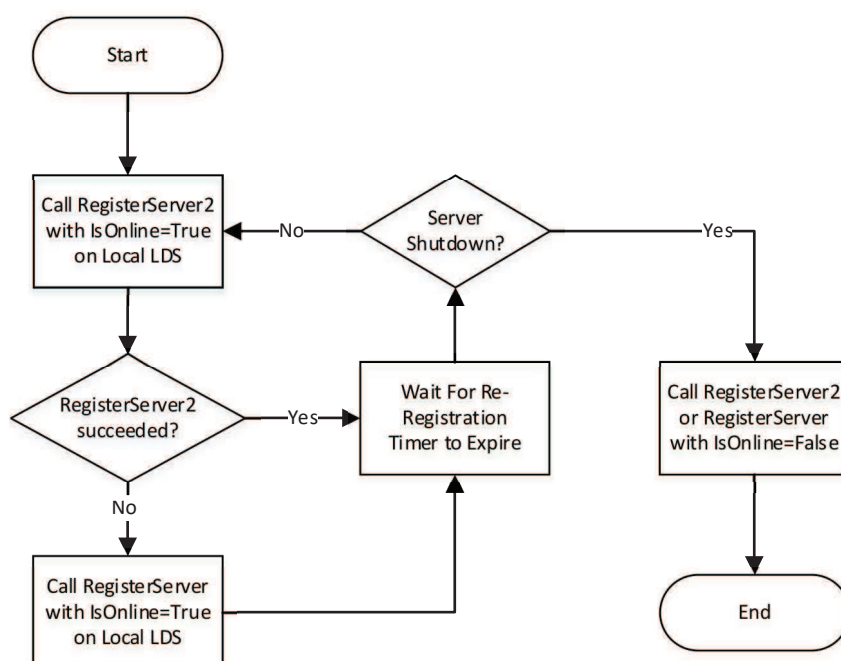


Figure 1 – The Registration process with an LDS

See IEC 62541-4 for more information on the re-registration timer and the *IsOnline* flag.

4.2.3 Hosts without a LocalDiscoveryServer

Dedicated systems (usually embedded systems) with exactly one *Server* installed may not have a separate LDS. Such *Servers* shall become their own LDS or LDS-ME by implementing *FindServers* and *GetEndpoints* Services at the well-known address for an LDS. They should also announce themselves on the *MulticastSubnet* with a basic *MulticastExtension*. This requires a small subset of an mDNS Responder (see mDNS and Annex C) that announces the *Server* and responds to mDNS probes. The *Server* may not provide the caching and address resolution implemented by a full mDNS Responder.

4.3 The Discovery Process for Clients to find Servers

4.3.1 Overview

The discovery process allows *Clients* to find *Servers* on the network and then discover how to connect to them. Once a *Client* has this information it can save it and use it to connect directly to the *Server* again without going through the discovery process. *Clients* that cannot connect

321 with the saved connection information should assume the *Server* configuration has changed
322 and therefore repeat the discovery process.

323 A *Client* has several choices for finding *Servers*:

- 324 • Out-of-band discovery (i.e. entry into a GUI) of a *DiscoveryUrl* for a *Server*;
- 325 • Calling *FindServers* on the LDS installed on the *Client* host;
- 326 • Calling *FindServers* on a remote LDS, where the *HostName* for the remote host is
327 manually entered;
- 328 • Calling *FindServersOnNetwork* (see IEC 62541-4) on the LDS-ME installed on *Client*
329 host;
- 330 • Supporting the LDS-ME functionality locally in the *Client*.
- 331 • Searching for *Servers* known to a *GlobalDiscoveryServer*.

332 The *DiscoveryUrl* provides all of the information a *Client* needs to connect to a
333 *DiscoveryEndpoint* (see 4.3.3).

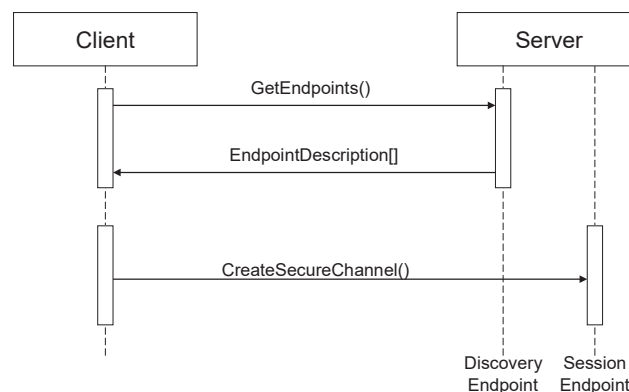
334 4.3.2 Security

335 *Clients* should be aware of rogue *DiscoveryServers* that might direct them to rogue *Servers*.
336 *Clients* can use the SSL/TLS server certificate (if available) to verify that the *DiscoveryServer*
337 is a server that they trust and/or ensure that they trust any *Server* provided by the
338 *DiscoveryServer*. See IEC TR 62541-2 for a detailed discussion of these issues.

339 4.3.3 Simple Discovery with a *DiscoveryUrl*

340 Every *Server* has one or more *DiscoveryUrls* that allow access to its *Endpoints*. Once a *Client*
341 obtains (e.g. via manual entry into a form) the *DiscoveryUrl* for the *Server*, it reads the
342 *EndpointDescriptions* using the *GetEndpoints* Service defined in IEC 62541-4.

343 The discovery process for this scenario is illustrated in Figure 2.



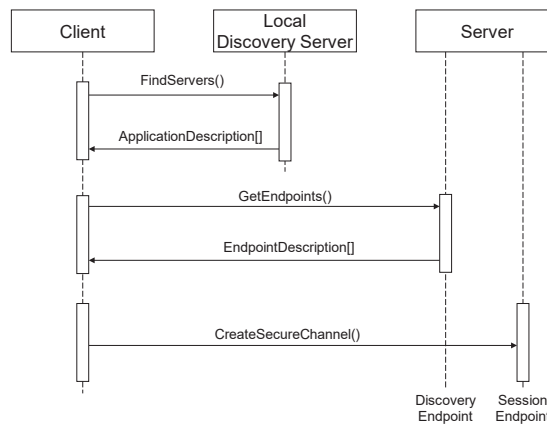
344
345 **Figure 2 – The simple Discovery process**

346 4.3.4 Local Discovery

347 In many cases *Clients* do not know which *Servers* exist but possibly know which hosts might
348 have *Servers* on them. In this situation the *Client* will look for the *LocalDiscoveryServer* on a
349 host by constructing a *DiscoveryUrl* using the Well-Known Addresses defined in IEC 62541-6.

350 If a *Client* finds a *LocalDiscoveryServer* then it will call the *FindServers* Service on the LDS to
351 obtain a list of *Servers* and their *DiscoveryUrls*. The *Client* would then call the *GetEndpoints*

352 service for one of the *Servers* returned. The discovery process for this scenario is illustrated
353 in Figure 3.



354

355

Figure 3 – The Local Discovery process

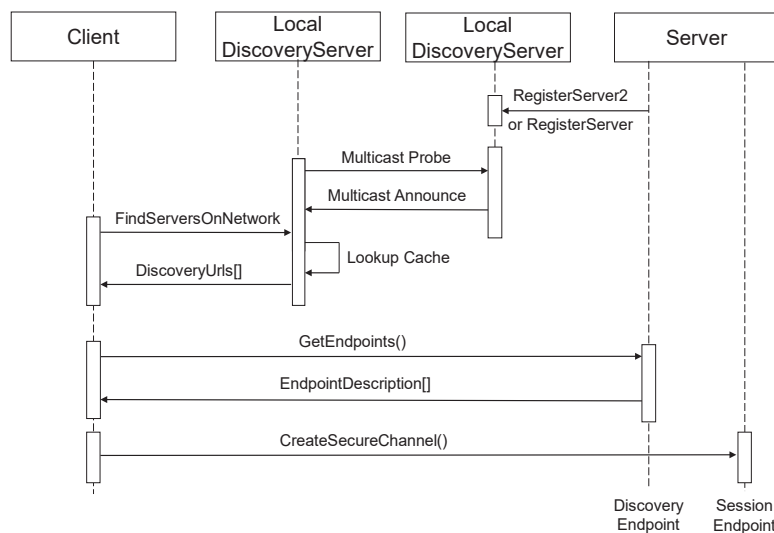
356

4.3.5 MulticastSubnet Discovery

357 In some situations *Clients* will not know which hosts have *Servers*. In these situations the
358 *Client* will look for a *LocalDiscoveryServer* with the *MulticastExtension* on its local host and
359 requests a list of *DiscoveryUrls* for *Servers* and *DiscoveryServers* available on the
360 *MulticastSubnet*.

361

The discovery process for this scenario is illustrated in Figure 4.



362

363

Figure 4 – The MulticastSubnet Discovery process

364 In this scenario the *Server* uses the *RegisterServer2 Service* to tell a *LocalDiscoveryServer* to
365 announce the *Server* on the *MulticastSubnet*. The *Client* will receive the *DiscoveryUrl* and
366 *ServerCapabilityIdentifiers* for the *Server* when it calls *FindServersOnNetwork* and then
367 connects directly to the *Server*. When a *Client* calls *FindServers* it only receives the *Servers*
368 running on the same host as the LDS.

369

Clients running on embedded systems may not have a LDS-ME available on the system,
370 These *Clients* can support an mDNS Responder which understands how OPC UA concepts

are mapped to mDNS messages and maintains the same table of *Servers* as maintained by the LDS-ME. This mapping is described in Annex C.

4.3.6 Global Discovery

A GDS is an OPC UA *Server* which allows *Clients* to search for *Servers* in the administrative domain. It may also provide Certificate Services (see Clause 7). It provides *Methods* that allow applications to search for other applications (See Clause 6). To access the GDS, the *Client* will create a *Session* with the *GDS* and use the *Call* service to invoke the *QueryApplications Method* (see 6.3.11). The *QueryServers Method* is similar to the *FindServers* service except that it provides more advanced search and filter criteria. The discovery process is illustrated in Figure 5.

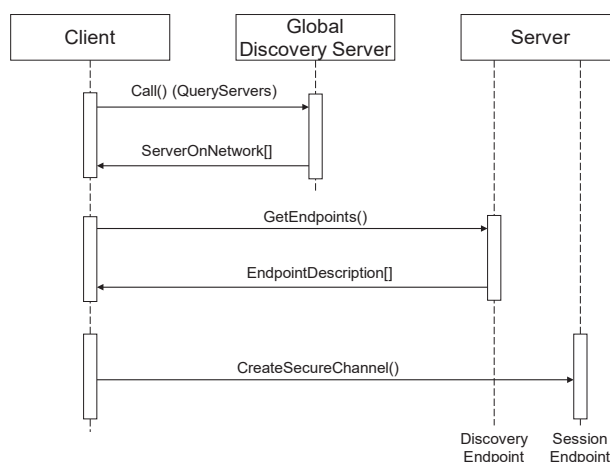


Figure 5 – The Global Discovery process

The GDS may be coupled with any of the previous network architectures. For each *MulticastSubnet*, one or more LDSs may be registered with a GDS.

The *Client* can also be configured with the URL of the GDS using an out of band mechanism.

The complete discovery process is shown in Figure 6.

4.3.7 Combined Discovery Process for Clients

The use cases in the preceding clauses imply a number of choices that have to be made by *Clients* when a *Client* needs to connect to a *Server*. These choices are combined together in Figure 6.

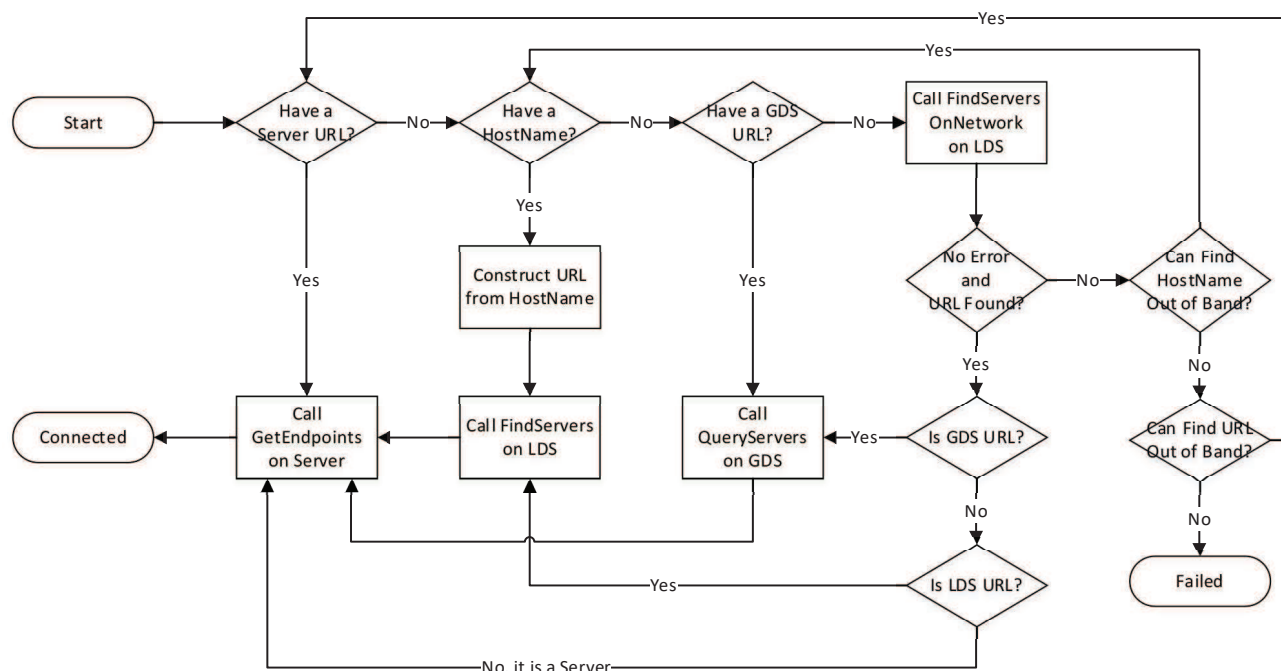


Figure 6 – The Discovery Process for Clients

FindServersOnNetwork can be called on the local LDS, however, It can also be called on a remote LDS which is part of a different *MulticastSubnet*.

An out-of-band mechanism is a way to find a URL or a *HostName* that is not described by this standard. For example, a user could manually enter a URL or use system specific APIs to browse the network neighbourhood.

A *Client* that goes through the discovery process can save the URL that was discovered. If the *Client* restarts later it can use that URL and bypass the discovery process. If reconnection fails the *Client* will have to go through the process again.

5 Local Discovery Server

5.1 Overview

Each host that could have multiple discoverable applications installed should have a standalone *LocalDiscoveryServer* installed. The *LocalDiscoveryServer* shall expose one or more *Endpoints* which support the *FindServers* and *GetEndpoints* services defined in IEC 62541-4 for all applications on the host. In addition, the *LocalDiscoveryServer* shall provide at least one *Endpoint* which implements the *RegisterServer* service for these applications.

In systems (usually embedded systems) with exactly one *Server* installed this *Server* may also be the LDS (see 4.2.3).

An LDS-ME will announce all applications that it knows about on the local *MulticastSubnet*. In order to support this, a *LocalDiscoveryServer* supports the *RegisterServer2 Service* defined in IEC 62541-4. For backward compatibility a *LocalDiscoveryServer* also supports the *RegisterServer Service* which is defined in IEC 62541-4.

Each host with OPC UA Applications (Clients and Servers) installed should have a *LocalDiscoveryServer* with a *MulticastExtension*.

418 The *MulticastExtension* incorporates the functionality of the mDNS Responder described in
419 the Multicast DNS (mDNS) specification (see mDNS). In addition the *LocalDiscoveryServer*
420 that supports the *MulticastExtension* supports the *FindServersOnNetwork Service* described
421 in IEC 62541-4.

422 5.2 Security considerations for Multicast DNS

423 The Multicast DNS (mDNS) specification is used for various commercial and consumer
424 applications. This provides a benefit in that implementations exist, however, system
425 administrators could choose to disable Multicast DNS operations. For this reason,
426 *Applications* shall not rely on Multicast DNS capabilities.

427 Multicast DNS operations are insecure because of their nature; therefore they should be
428 disabled in environments where an attacker could cause problems by impersonating another
429 host. This risk is minimized if OPC UA security is enabled and all *Applications* use *Certificate*
430 *TrustLists* to control access.

431 6 Global Discovery Server

432 6.1 Overview

433 The *LocalDiscoveryServer* is useful for networks where the host names can be discovered.
434 However, this is typically not the case in large systems with multiple servers on multiple
435 subnets. For this reason there is a need for an enterprise wide *DiscoveryServer* called a
436 *GlobalDiscoveryServer*. The *GlobalDiscoveryServer* (GDS) is an OPC UA *Server* which allows
437 *Clients* to search for *Servers* in the administrative domain. It provides methods that allow
438 applications to register themselves and to search for other applications.

439 The essential element of a *GlobalDiscoveryServer* (GDS) is that it can provide the *Certificate*
440 management services defined in Clause 7. These services can simplify *Certificate*
441 management even in medium to small systems, therefore, a GDS can be deployed in smaller
442 systems. Different implementations are expected. Some of them will likely provide a front-end
443 to an existing *DirectoryService* such as LDAP (See Annex E). By standardizing on an OPC UA
444 based interface, OPC UA *Clients* do not need to have knowledge of different
445 *DirectoryServices*.

446 If an administrator registers a *LocalDiscoveryServer* with the GDS, then the GDS shall
447 periodically update its database by calling *FindServersOnNetwork* or *FindServers* on the LDS.
448 Figure 7 shows the relationship between a GDS and the LDS-ME or LDS.

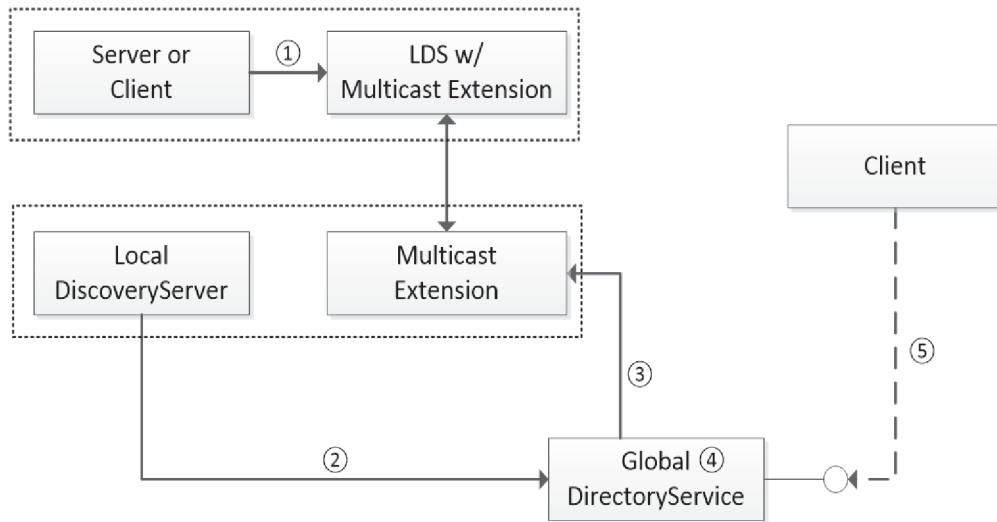


Figure 7 – The relationship between GDS and other components

The steps shown in Figure 7 are:

1	The Server calls <i>RegisterServer2</i> on the LDS running on the same machine.
2	The administrator registers LDS-ME installations with the GDS.
3	The GDS calls <i>FindServersOnNetwork</i> on the LDS-ME to find all applications on the same <i>MulticastSubnet</i> .
4	The GDS creates a record for each application returned by the LDS-ME. These records shall be approved before they are made available to <i>Clients</i> of the GDS. This approval can be obtained from an <i>Administrator</i> or the GDS can connect to the <i>Server</i> and verifies that it has a trusted <i>Certificate</i> .
5	The <i>Client</i> calls <i>QueryServers Method</i> on the GDS to discover applications.

The *Information Model* used for registration and discovery is shown in clause 6.2. Any *Client* shall be able to call the *QueryServers Method* to find applications known to GDS. The complete definitions for each of the types used are described in clause 7.5.

6.2 Network Architectures

6.2.1 Overview

The discovery mechanisms defined in this standard are expected to be used in many different network architectures. The following three architectures are illustrated:

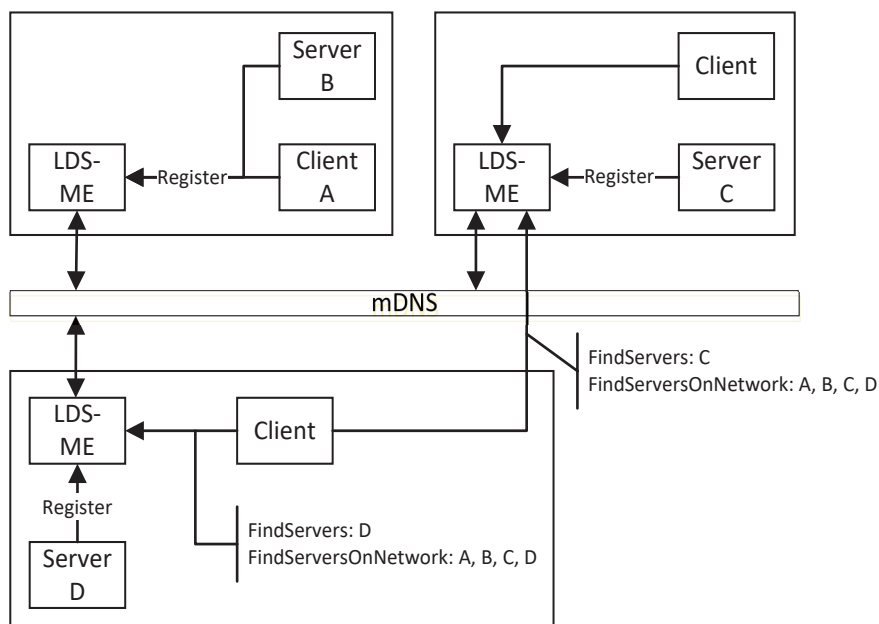
- Single *MulticastSubnet*;
- Multiple *MulticastSubnets*;
- No *MulticastSubnet* (or multiple *MulticastSubnets* with exactly one host each);

A *MulticastSubnet* is a network segment where all hosts on the segment can receive multicast packets from the other hosts on the segment. A physical LAN segment is typically a

464 *MulticastSubnet* unless the administrator has specifically disabled multicast communication.
 465 In some cases multiple physical LAN segments can be connected as a single *MulticastSubnet*

466 6.2.2 Single MulticastSubnet

467 The Single *MulticastSubnet* Architecture is shown in Figure 8.



468
469 **Figure 8 – The Single MulticastSubnet architecture**

470 In this architecture every host has an LDS-ME and uses mDNS to maintain a cache of the
 471 applications on the *MulticastSubnet*. A *Client* can call *FindServersOnNetwork* on any LDS-ME
 472 and receive the same set of applications. When a *Client* calls *FindServers* it only receives the
 473 applications running on the same host as the LDS.

474 6.2.3 Multiple MulticastSubnet

475 The Multiple *MulticastSubnet* Architecture is shown in Figure 9.

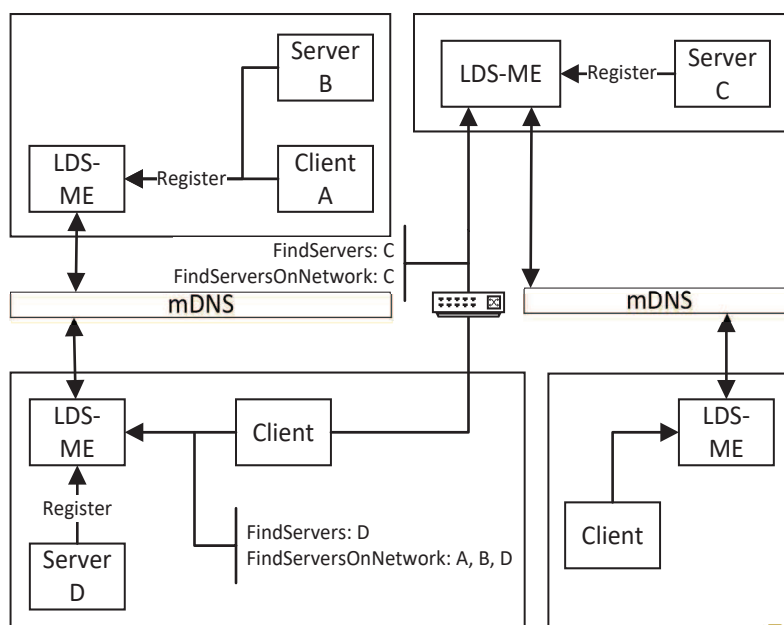


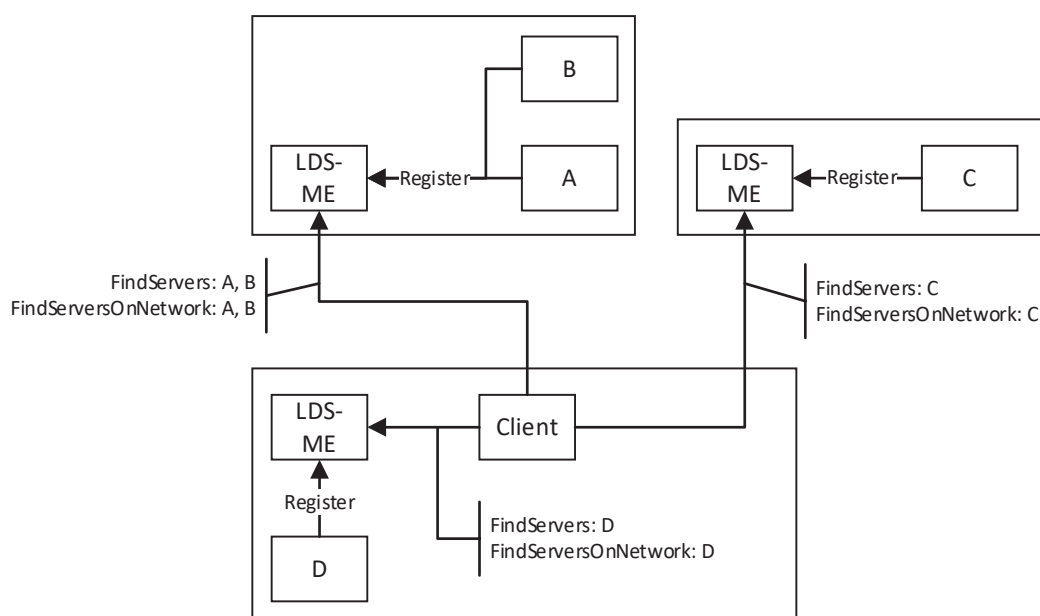
Figure 9 – The Multiple MulticastSubnet architecture

This architecture is the same as the previous architecture except in this architecture the mDNS messages do not pass through routers connecting the *MulticastSubnets*. This means that a *Client* calling *FindServersOnNetwork* will only receive a list of applications running on the *MulticastSubnets* that the LDS-ME is connected to.

A *Client* that wants to connect to a remote *MulticastSubnet* shall use out of band discovery (i.e. manual entry) of a *HostName* or *DiscoveryUrl*. Once a *Client* finds an LDS-ME on a remote *MulticastSubnet* it can use *FindServersOnNetwork* to discover all applications on that *MulticastSubnet*.

6.2.4 No MulticastSubnet

The No *MulticastSubnet* Architecture is shown in Figure 10.

**Figure 10 – The No MulticastSubnet architecture**

In this architecture the mDNS is not used at all because the Administrator has disabled multicast at a network level or by turning off multicast capabilities of each LDS-ME.

A *Client* that wants to discover applications needs to use an out of band mechanism to find the *HostName* and call *FindServers* on the LDS of that host. *FindServersOnNetwork* may also work but it will never return more than what *FindServers* returns.

6.2.5 Domain Names and MulticastSubnets

The mDNS specification requires that fully qualified domain name be announced on the network. If a *Server* is not configured with a fully qualified domain name then mDNS requires that the 'local' top level domain be appended to the domain names. The 'local' top level domain indicates that the domain can only be considered to be unique within the subnet where the domain name was used. This means *Clients* need to be aware that URLs received from any LDS-ME other than the one on the *Client's* machine could contain 'local' domains which are not reachable or will connect to a different machine with the same domain name that happens to be on the same subnet as the *Client*. It is recommended that *Clients* ignore all URLs with the 'local' top level domain unless they are returned from the LDS-ME running on the same machine.

506 System administrators can eliminate this problem by configuring a normal DNS with the fully
507 qualified domain names for all machines which need to be accessed by *Clients* outside the
508 *MulticastSubnet*.

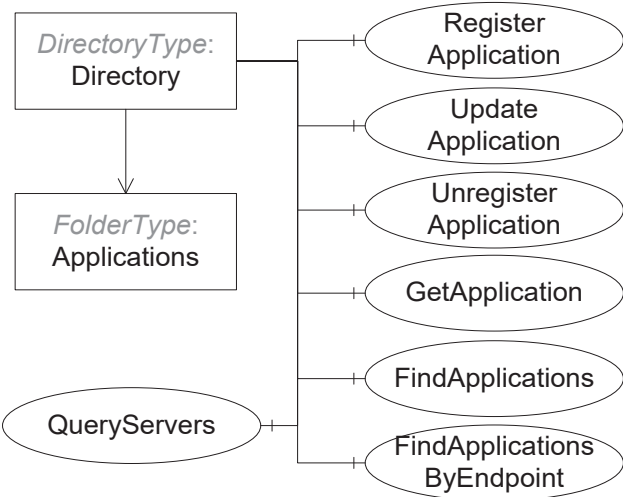
509 *Servers* configured with fully qualified domain names should specify the fully qualified domain
510 name in its *ApplicationInstance Certificate*. *Servers* shall not specify domains with the 'local'
511 top level domain in their *Certificate*. *Clients* using a URL returned from an LDS-ME shall
512 ignore the 'local' top level domain when checking the domain against the *Server Certificate*.

513 Note that domain name validation is a necessary but not sufficient check against rogue
514 *Servers* or man-in-the-middle attacks when *Server Certificates* do not contain fully qualified
515 domain names. The *Certificate* trust relationship established by administrators is the primary
516 mechanism used to protect against these risks.

517 **6.3 Information Model**

518 **6.3.1 Overview**

519 The *GlobalDiscoveryServer Information Model* used for *discovery* is shown in Figure 11. Most
520 of the interactions between the *GlobalDiscoveryServer* and *Application* administrator or the
521 *Client* will be via *Methods* defined on the *Directory* folder.



522

523 **Figure 11 – The Address Space for the GDS**

524 **6.3.2 Directory**

525 This *Object* is the root of the *GlobalDiscoveryServer AddressSpace* and it is the target of an
526 *Organizes* reference from the *Objects* folder defined in IEC 62541-5. It organizes the
527 information that can be accessed into subfolders. The implementation of a GDS can
528 customize and organize the folders in any manner it desires. For example folders may exist
529 for information models, or for optional services or for various locations in an administrative
530 domain. It is defined in Table 2.

531 **Table 2 – Directory Object definition**

Attribute	Value				
BrowseName	Directory				
Namespace	GDS (see 3.3)				
TypeDefinition	DirectoryType defined in 6.3.3.				
References	NodeClass	BrowseName	DataType	TypeDefinition	Model