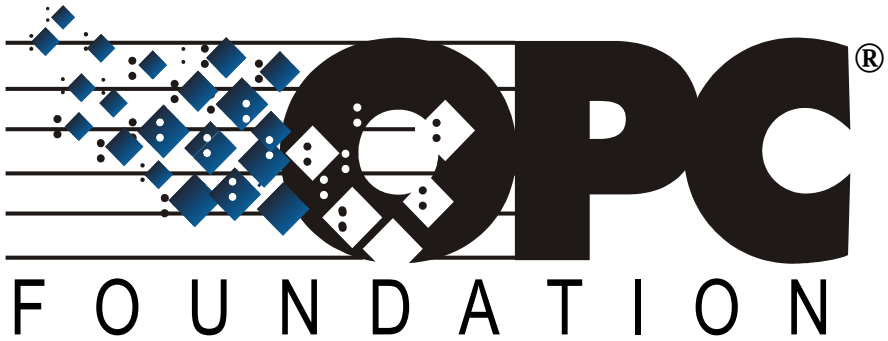

OPC 10000-18

OPC UA Specification

OPC Unified Architecture

Part 18: Role-Based Security

Release 1.05.00

2021-10-15

Specification Type:	Industry Standard Specification	Comments:	Report or view errata: http://www.opcfoundation.org/errata
Document Number	**OPC 10000-18**		
Title:	OPC Unified Architecture Part 18: Role-Based Security	Date:	2021-10-15
Version:	Release 1.05.00	Software:	MS-Word
		Source:	OPC 10000-18 - UA Specification Part 18 - Role-Based Security 1.05.00.docx
Author:	OPC Foundation	Status:	Release

CONTENTS

FIGURES

TABLES

OPC FOUNDATION

UNIFIED ARCHITECTURE –

FOREWORD

This specification is the specification for developers of OPC UA applications. The specification is a result of an analysis and design process to develop a standard interface to facilitate the development of applications by multiple vendors that shall inter-operate seamlessly together.

AGREEMENT OF USE

TRADEMARKS

Most computer and software brand names have trademarks or registered trademarks. The individual trademarks have not been listed here.

GENERAL PROVISIONS

Should any provision of this Agreement be held to be void, invalid, unenforceable or illegal by a court, the validity and enforceability of the other provisions shall not be affected thereby.

This Agreement shall be governed by and construed under the laws of the State of Minnesota, excluding its choice or law rules.

This Agreement embodies the entire understanding between the parties with respect to, and supersedes any prior understanding or agreement (oral or written) relating to, this specification.

ISSUE REPORTING

The OPC Foundation strives to maintain the highest quality standards for its published specifications, hence they undergo constant review and refinement. Readers are encouraged to report any issues and view any existing errata here: http://www.opcfoundation.org/errata

Revision 1.05.00 Highlights

The following table includes the Mantis issues resolved with this revision.

Mantis ID	Summary	Resolution
5528	Make Part 5 Annex F a separate Part	Context of Part 5 Annex F moved to the initial version of this part.
4082	RoleSet and well-known Roles	Added definition of RoleSet including well known roles and recommended default identity mapping.
5135	Missing details or features on how to combine standard and vendor specific handling of user authorization	Added CustomConfiguration Property to 4.4.1 RoleType definition.
5326	Application authentication only not supported in IdentityMappingRuleType	Added criteriaType Application to IdentityMappingRuleType.
5431	Handling of endpointUrl in RoleType::Endpoints needs clarification	Added clarifying text to 4.4.1 RoleType definition.
5554	Identity mapping criteria type enumeration inconsistent with UANodeSet	Changed enumeration DataType name to IdentityCriteriaType
5555	Format of enumeration DataType definitions	Moved definition of IdentityCriteriaType to own chapter and new table format.
5816	Conformance unit assignment	Added conformance unit assignment to type tables
5836	User certificate identification not possible with certificate common name	Added criteriaType X509Subject to IdentityMappingRuleType.
6407	Endpoint verification	Enhanced rules for Endpoint information verification

OPC Unified Architecture Specification

Part 18: Role-Based Security

1 Scope

This part of the OPC Unified Architecture defines an Information Model. The Information Model describes the basic infrastructure to model role-based security.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments and errata) applies.

OPC 10000-1, *OPC Unified Architecture - Part 1: Concepts*

http://www.opcfoundation.org/UA/Part1/

OPC 10000-3, *OPC Unified Architecture - Part 3: Address Space Model*

http://www.opcfoundation.org/UA/Part3/

OPC 10000-4, *OPC Unified Architecture - Part 4: Services*

http://www.opcfoundation.org/UA/Part4/

OPC 10000-5, *OPC Unified Architecture - Part 5: Information Model*
http://www.opcfoundation.org/UA/Part5/

OPC 10000-6, *OPC Unified Architecture - Part 6: Mappings*

http://www.opcfoundation.org/UA/Part6/

OPC 10000-7, *OPC Unified Architecture - Part 7: Profiles*

http://www.opcfoundation.org/UA/Part7/

3 Terms, definitions, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in OPC 10000-1, OPC 10000-3 and OPC 10000-5 apply.

4 Role Model

4.1 General

OPC UA defines a standard approach for implementing role based security. *Servers* may choose to implement part or all of the mechanisms defined here. The OPC UA approach assigns *Permissions* to *Roles* for each *Node* in the *AddressSpace*. *Clients* are then granted *Roles* when they create a *Session* based on the information provided by the *Client*.

Roles are used to separate authentication (determining who a *Client* is with a user token and *Client* application identity) from authorization (*Permissions* determining what the *Client* is allowed to do). By separating these tasks *Servers* can allow centralized services to manage user identities and credentials while the *Server* only manages the *Permissions* on its *Nodes* assigned to *Roles*.

OPC 10000-3 defines the possible *Permissions* and the representation as *Node Attributes*.

Figure 1 depicts the *ObjectTypes*, *Objects* and their components used to represent the *Role* management.

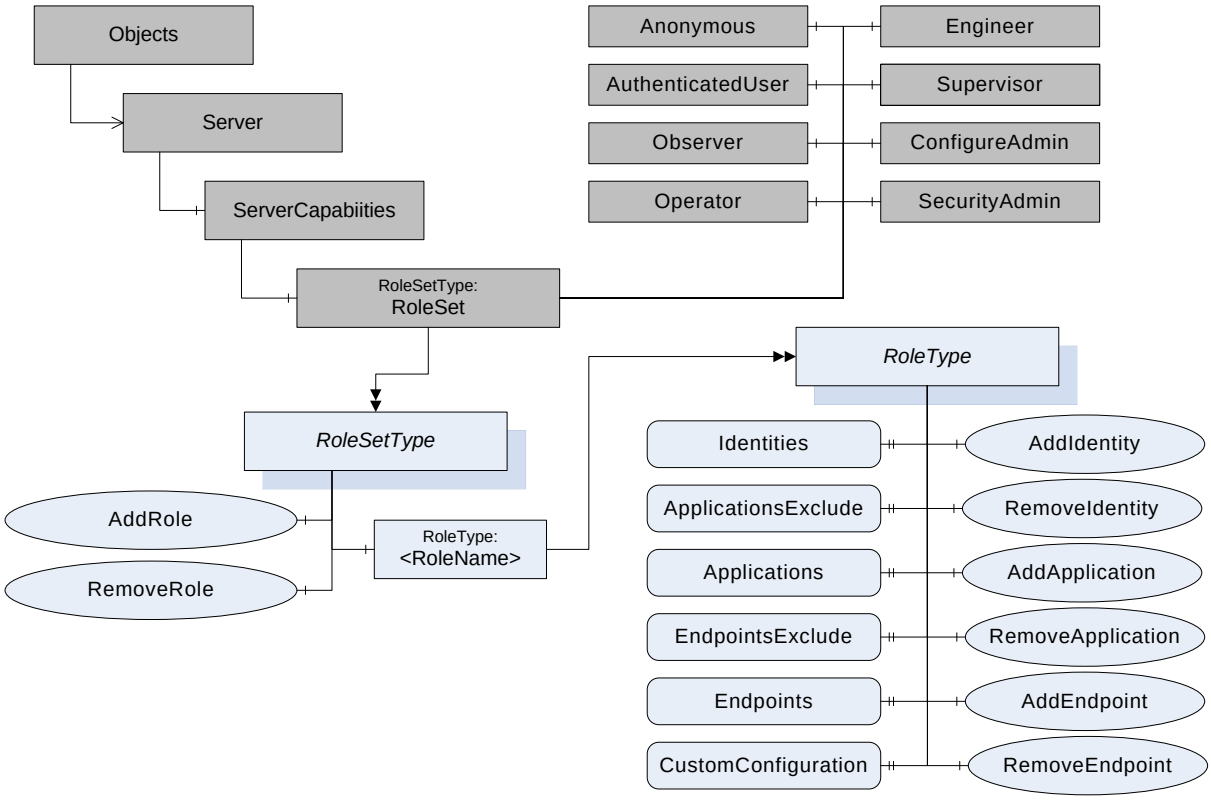

Figure 1 – Role management overview

4.2 RoleSetType

4.2.1 RoleSetType definition

The RoleSet *Object defined in* OPC 10000-5 is a *RoleSetType* which is formally defined in Table 1.

Table 1 – RoleSetType definition

Attribute	*Value*				
BrowseName	RoleSetType				
IsAbstract	False				
References	***Node* Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Subtype of *BaseObjectType* defined in OPC 10000-5					
HasComponent	Object	<RoleName>		RoleType	OptionalPlaceholder
HasComponent	Method	AddRole	Defined in 4.2.2		Mandatory
HasComponent	Method	RemoveRole	Defined in 4.2.3.		Mandatory
Conformance Units					
Base Info ServerType					

The *AddRole Method* allows configuration *Clients* to add a new *Role* to the *Server*.

The *RemoveRole Method* allows configuration *Clients* to remove a *Role* from the *Server*.

4.2.2 AddRole Method

This *Method* is used to add a *Role* to the *RoleSet Object.*

The combination of the NamespaceUri and *RoleName* parameters are used to construct the *BrowseName* for the new *Node*. The BrowseName shall be unique within the *RoleSet Object*.

This *Method* affects security and shall only be browseable and callable by authorized administrators.

OPC 10000-3 defines well-known *Roles*. If this *Method* is used to add a well-known *Role*, the name of the *Role* from OPC 10000-3 is used together with the OPC UA namespace URI. The *Server* shall use the *NodeIds* for the well-known *Roles* in this case. The *NodeIds* for the well-known *Roles* are defined in OPC 10000-6.

Signature

```
AddRole (
   [in]  String           RoleName
   [in]  String           NamespaceUri
   [out] NodeId           RoleNodeId
   );
```

Argument	Description
RoleName	The name of the *Role*.
NamespaceUri	The *NamespaceUri* qualifies the *RoleName*. If this value is null or empty then the resulting *BrowseName* will be qualified by the *Server's NamespaceUri*.
RoleNodeId	The *NodeId* assigned by the *Server* to the new *Node*.

Method Result Codes

ResultCode	Description
Bad_InvalidArgument	The *RoleName* or NamespaceUri is not valid. The text associated with the error shall indicate the exact problem.
Bad_NotSupported	The *Server* does not allow more *Roles* to be added.
Bad_UserAccessDenied	The caller does not have the necessary *Permissions*.

4.2.3 RemoveRole Method

This *Method* is used to remove a *Role* from the *RoleSet Object.*

The *RoleNodeId* is the *NodeId* of the *Role Object* to remove.

The *Server* may prohibit the removal of some *Roles* because they are necessary for the *Server* to function.

If a *Role* is removed all *Permissions* associated with the *Role* are deleted as well. Ideally these changes should take effect immediately; however, some lag may occur.

This Method affects security and shall only be browseable and callable by authorized administrators.

Signature

```
RemoveRole (
   [in]  NodeId RoleNodeId
   );
```

Argument	Description
RoleNodeId	The *NodeId* of the *Role Object*.

Method Result Codes

ResultCode	Description
Bad_NodeIdUnknown	The specified *Role Object* does not exist.
Bad_NotSupported	The *Server* does not allow the *Role Object* to be removed.
Bad_UserAccessDenied	The caller does not have the necessary *Permissions*.
Bad_RequestNotAllowed	The specified Role Object cannot be removed.

4.3 RoleSet

The *RoleSet Object* defined in Table 2 is used to publish all *Roles* supported by the *Server*.

Table 2 – RoleSet definition

Attribute	***Value***				
BrowseName	RoleSet				
References	***Node* Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
ComponentOf the *ServerCapabilities Object* defined in OPC 10000-5					
HasTypeDefinition	ObjectType	RoleSetType			
HasComponent	Object	Anonymous		RoleType	
HasComponent	Object	AuthenticatedUser		RoleType	
HasComponent	Object	Observer		RoleType	
HasComponent	Object	Operator		RoleType	
HasComponent	Object	Engineer		RoleType	
HasComponent	Object	Supervisor		RoleType	
HasComponent	Object	ConfigureAdmin		RoleType	
HasComponent	Object	SecurityAdmin		RoleType	
Conformance Units					
Security Role Server Base 2					

Servers should support the well-known *Roles* which are defined in OPC 10000-3.

The default *Identities* for the *Anonymous Role* should be *Identities* with the *criteriaType IdentityCriteriaType.Anonymous* and the *criteriaType IdentityCriteriaType.AuthenticatedUser*.

The default *Identities* for the *AuthenticatedUser Role* should be an identity with the *criteriaType IdentityCriteriaType.AuthenticatedUser*.

The additional definition for the conformance units of the instances are defined in Table 3.

Table 3 – RoleSet Additional Conformance Units

BrowsePath	Conformance Units
AddRole	Security Role Server Management
RemoveRole	Security Role Server Management
ConfigureAdmin	Security Role Well Known
SecurityAdmin	Security Role Well Known
Anonymous	Security Role Well Known Group 2
AuthenticatedUser	Security Role Well Known Group 2
Observer	Security Role Well Known Group 3
Operator	Security Role Well Known Group 3
Engineer	Security Role Well Known Group 3
Supervisor	Security Role Well Known Group 3
Anonymous AddIdentity	Security Role Server IdentityManagement
Anonymous RemoveIdentity	Security Role Server IdentityManagement
Anonymous ApplicationsExclude	Security Role Server Restrict Applications
Anonymous Applications	Security Role Server Restrict Applications
Anonymous AddApplication	Security Role Server Restrict Applications
Anonymous RemoveApplication	Security Role Server Restrict Applications
Anonymous EndpointsExclude	Security Role Server Restrict Endpoints
Anonymous Endpoints	Security Role Server Restrict Endpoints
Anonymous AddEndpoint	Security Role Server Restrict Endpoints
Anonymous RemoveEndpoint	Security Role Server Restrict Endpoints
AuthenticatedUser AddIdentity	Security Role Server IdentityManagement
AuthenticatedUser RemoveIdentity	Security Role Server IdentityManagement
AuthenticatedUser ApplicationsExclude	Security Role Server Restrict Applications
AuthenticatedUser Applications	Security Role Server Restrict Applications
AuthenticatedUser AddApplication	Security Role Server Restrict Applications
AuthenticatedUser RemoveApplication	Security Role Server Restrict Applications
AuthenticatedUser EndpointsExclude	Security Role Server Restrict Endpoints
AuthenticatedUser Endpoints	Security Role Server Restrict Endpoints
AuthenticatedUser AddEndpoint	Security Role Server Restrict Endpoints
AuthenticatedUser RemoveEndpoint	Security Role Server Restrict Endpoints
Observer AddIdentity	Security Role Server IdentityManagement
Observer RemoveIdentity	Security Role Server IdentityManagement
Observer ApplicationsExclude	Security Role Server Restrict Applications
Observer Applications	Security Role Server Restrict Applications
Observer AddApplication	Security Role Server Restrict Applications

BrowsePath	Conformance Units
Observer RemoveApplication	Security Role Server Restrict Applications
Observer EndpointsExclude	Security Role Server Restrict Endpoints
Observer Endpoints	Security Role Server Restrict Endpoints
Observer AddEndpoint	Security Role Server Restrict Endpoints
Observer RemoveEndpoint	Security Role Server Restrict Endpoints
Operator AddIdentity	Security Role Server IdentityManagement
Operator RemoveIdentity	Security Role Server IdentityManagement
Operator ApplicationsExclude	Security Role Server Restrict Applications
Operator Applications	Security Role Server Restrict Applications
Operator AddApplication	Security Role Server Restrict Applications
Operator RemoveApplication	Security Role Server Restrict Applications
Operator EndpointsExclude	Security Role Server Restrict Endpoints
Operator *Endpoints*	*Security Role Server Restrict Endpoints*
Operator *AddEndpoint*	*Security Role Server Restrict Endpoints*
Operator RemoveEndpoint	Security Role Server Restrict Endpoints
Engineer AddIdentity	Security Role Server IdentityManagement
Engineer RemoveIdentity	Security Role Server IdentityManagement
Engineer ApplicationsExclude	Security Role Server Restrict Applications
Engineer Applications	Security Role Server Restrict Applications
Engineer AddApplication	Security Role Server Restrict Applications
Engineer RemoveApplication	Security Role Server Restrict Applications
Engineer EndpointsExclude	Security Role Server Restrict Endpoints
Engineer Endpoints	Security Role Server Restrict Endpoints
Engineer AddEndpoint	Security Role Server Restrict Endpoints
Engineer RemoveEndpoint	Security Role Server Restrict Endpoints
Supervisor AddIdentity	Security Role Server IdentityManagement
Supervisor RemoveIdentity	Security Role Server IdentityManagement
Supervisor ApplicationsExclude	Security Role Server Restrict Applications
Supervisor Applications	Security Role Server Restrict Applications
Supervisor AddApplication	Security Role Server Restrict Applications

BrowsePath	Conformance Units
Supervisor RemoveApplication	Security Role Server Restrict Applications
Supervisor EndpointsExclude	Security Role Server Restrict Endpoints
Supervisor Endpoints	Security Role Server Restrict Endpoints
Supervisor AddEndpoint	Security Role Server Restrict Endpoints
Supervisor RemoveEndpoint	Security Role Server Restrict Endpoints
ConfigureAdmin AddIdentity	Security Role Server IdentityManagement
ConfigureAdmin RemoveIdentity	Security Role Server IdentityManagement
ConfigureAdmin ApplicationsExclude	Security Role Server Restrict Applications
ConfigureAdmin Applications	Security Role Server Restrict Applications
ConfigureAdmin AddApplication	Security Role Server Restrict Applications
ConfigureAdmin RemoveApplication	Security Role Server Restrict Applications
ConfigureAdmin EndpointsExclude	Security Role Server Restrict Endpoints
ConfigureAdmin Endpoints	Security Role Server Restrict Endpoints
ConfigureAdmin AddEndpoint	Security Role Server Restrict Endpoints
ConfigureAdmin RemoveEndpoint	Security Role Server Restrict Endpoints
SecurityAdmin AddIdentity	Security Role Server IdentityManagement
SecurityAdmin RemoveIdentity	Security Role Server IdentityManagement
SecurityAdmin ApplicationsExclude	Security Role Server Restrict Applications
SecurityAdmin Applications	Security Role Server Restrict Applications
SecurityAdmin AddApplication	Security Role Server Restrict Applications
SecurityAdmin RemoveApplication	Security Role Server Restrict Applications
SecurityAdmin EndpointsExclude	Security Role Server Restrict Endpoints
SecurityAdmin Endpoints	Security Role Server Restrict Endpoints
SecurityAdmin AddEndpoint	Security Role Server Restrict Endpoints
SecurityAdmin RemoveEndpoint	Security Role Server Restrict Endpoints

4.4 RoleType

4.4.1 RoleType definition

Each *Role Object* has the *Properties* and *Methods* defined by the *RoleType* which is formally defined in Table 4.

Table 4 – RoleType definition

Attribute	**Value**				
BrowseName	RoleType				
IsAbstract	False				
References	***Node* Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Subtype of BaseObjectType					
HasProperty	*Variable*	Identities	IdentityMapping RuleType []	PropertyType	Mandatory
HasProperty	Variable	ApplicationsExclude	Boolean	PropertyType	Optional
HasProperty	Variable	Applications	String []	PropertyType	Optional
HasProperty	Variable	EndpointsExclude	Boolean	PropertyType	Optional
HasProperty	Variable	Endpoints	EndpointType []	PropertyType	Optional
HasProperty	Variable	CustomConfiguration	Boolean	PropertyType	Optional
HasComponent	Method	AddIdentity	Defined in 4.4.5.		Optional
HasComponent	Method	RemoveIdentity	Defined in 4.4.6.		Optional
HasComponent	Method	AddApplication	Defined in 4.4.7.		Optional
HasComponent	Method	RemoveApplication	Defined in 4.4.8.		Optional
HasComponent	Method	AddEndpoint	Defined in 4.4.9.		Optional
HasComponent	Method	RemoveEndpoint	Defined in 4.4.10.		Optional
Conformance Units					
Base Info ServerType					

The *Properties* and *Methods* of the *RoleType* contain sensitive security related information and shall only be browseable, writeable and callable by authorized administrators through an encrypted channel.

The *Identities Property* specifies the currently configured rules for mapping a *UserIdentityToken* to the *Role*. If this Property is an empty array and *CustomConfiguration* is not *TRUE*, then the *Role* cannot be granted to any *Session*.

The *Role* shall only be granted to the *Session* if all of the following conditions are true:

- The *UserIdentityToken* complies with *Identities.*
- The *Applications Property* is not configured or the *Client Certificate* complies with the *Applications* settings.
- The *Endpoints Property* is not configured or the *Endpoint* used complies with the *Endpoints* settings.

The *ApplicationsExclude Property* defines the *Applications Property* as an include list or exclude list. If the *ApplicationsExclude Property* is not provided or has a value of *FALSE* then only *Application Instance Certificates* included in the *Applications Property* shall be included in this *Role*. All other *Application Instance Certificates* shall not be included in this *Role*. If this *Property* has a value of *TRUE* then all *Application Instance Certificates* included in the *Applications Property* shall be excluded from this *Role*. All other *Application Instance Certificates* shall be included in this *Role*. If the *Applications Property* is provided with an empty array and all *Application Instance Certificates* should be included, the *ApplicationsExclude Property* shall be present and the value must be *TRUE*.

The *Applications Property* specifies the *Application Instance Certificates* of *Clients* which shall be included or excluded from this *Role*. Each element in the array is an *ApplicationUri* from a *Client Certificate* which is trusted by the *Server*.

The *EndpointsExclude Property* defines the *Endpoints Property* as an include list or exclude list. If this *Property* is not provided or has a value of *FALSE* then only *Endpoints* included in the *Endpoints Property* shall be included in this *Role*. All other *Endpoints* shall not be included in this *Role*. If this *Property* has a value of *TRUE* then all *Endpoints* included in the *Endpoints Property* shall be excluded from this *Role*. All other *Endpoints* shall be included in this *Role*. If

the *Endpoints Property* is provided with an empty array and all endpoints should be included, the *EndpointsExclude Property* shall be present and the value must be *TRUE*.

The *Endpoints Property* specifies the *Endpoints which shall be included or excluded from this* Role. Each element in the array is an *EndpointType* that contains an *Endpoint* description. The *EndpointUrl* and the other *Endpoint* settings are compared with the configured *Endpoint* that is used by the *SecureChannel* for the *Session*. The *EndpointType DataType* is defined in 4.4.2. Fields that have default values as defined in the *EndpointType DataType* are ignored during the comparison.

The *CustomConfiguration Property* indicates that the configuration of the *Role* and the assignment of the *Role* to *Sessions* is vendor specific. *Roles* are required to support the *RolePermissions Attribute*. If a *Server* want to support *RolePermissions* but is not able to support the standard *Role* functionality, it can indicate this with the *CustomConfiguration Property*. If *CustomConfiguration* is *TRUE*, the *Server* may hide the configuration options completely or the *Server* may provide additional vendor specific configuration options.

The *AddIdentity Method* adds a rule used to map a *UserIdentityToken* to the *Role*. If the *Server* does not allow changes to the mapping rules, then the Method is not present. A *Server* should prevent certain rules from being added to particular *Roles*. For example, a *Server* should refuse to allow an ANONYMOUS_5 (see 4.4.2) mapping rule to be added to *Roles* with administrator privileges.

The *RemoveIdentity Method* removes a mapping rule used to map a *UserIdentityToken* to the *Role*. If the *Server* does not allow changes to the mapping rules, then the *Method* is not present.

The *AddApplication Method* adds an *Application Instance Certificate* to the list of. If the *Server* does not enforce application restrictions or does not allow changes to the mapping rules for the *Role* the Method is not present.

The *RemoveApplication Method* removes an *Application Instance Certificate* from the list of applications. If the *Server* does not enforce application restrictions or does not allow changes to the mapping rules for the *Role* the *Method* is not present.

4.4.2 EndpointType

This *structure describes an Endpoint.* The *EndpointType* is formally defined in Table 5.

Table 5 – EndpointType Structure

Name	Type	Description
EndpointType	structure	
endpointUrl	String	The URL for the *Endpoint*.
securityMode	MessageSecurityMode	The type of message security. The type *MessageSecurityMode* type is defined in OPC 10000-4. The default value is *MessageSecurityMode Invalid.* The field is ignored for comparison if the default value is set.
securityPolicyUri	String	The URI of the *SecurityPolicy*. The default value is an empty or null *String.* The field is ignored for comparison if the default value is set.
transportProfileUri	String	The URI of the *Transport Profile*. The default value is an empty or null *String.* The field is ignored for comparison if the default value is set.

The *EndpointType Structure* representation in the *AddressSpace* is defined in Table 6.

Table 6 – EndpointType definition

Attributes	**Value**			
BrowseName	EndpointType			
IsAbstract	False			
References	**NodeClass**	**BrowseName**	**IsAbstract**	**Description**
Subtype of Structure defined in OPC 10000-5.				
Conformance Units				
Base Info ServerType				

4.4.3 IdentityMappingRuleType

The *IdentityMappingRuleType* structure defines a single rule for selecting a *UserIdentityToken*. The structure is described in Table 7.

Table 7 – IdentityMappingRuleType

Name	**Type**	**Description**
IdentityMappingRuleType	Structure	Specifies a rule used to map a *UserIdentityToken* to a *Role*.
criteriaType	Enumeration IdentityCriteriaType	The type of criteria contained in the identity mapping rule. The *IdentityCriteriaType* is defined in 4.4.4.
criteria	String	The criteria which the *UserIdentityToken* must meet for a *Session* to be mapped to the *Role*. The meaning of the criteria depends on the *criteriaType*. The *criteria* are a "" for *Anonymous* and *AuthenticatedUser*.

If the *criteriaType* is *UserName*, the *criteria* is a name of a user known to the *Server*, For example, the user could be the name of a local operating system account.

If the *criteriaType* is *Thumbprint*, the *criteria* is a thumbprint of an immediate user *Certificate* or an issuer *Certificate* in its chain which is trusted by the *Server*. For the criteria, the thumbprint shall be encoded as a hexadecimal string with upper case characters and without spaces.

If the *criteriaType* is *Role*, the *criteria* is a name of a restriction found in the *Access Token*. For example, the *Role* "subscriber" may only be allowed to access *PubSub* related *Nodes*.

If the *criteriaType* is *GroupId*, the *criteria* is a generic text identifier for a user group specific to the *Authorization Service.* For example, an *Authorization Service* providing access to an Active Directory may add one or more Windows Security Groups to the *Access Token*. OPC 10000-6 provides details on how groups are added to *Access Tokens*.

If the *criteriaType* is *Anonymous*, the *criteria* is a null string which indicates no user credentials have been provided.

If the *criteriaType* is *AuthenticatedUser*, the *criteria* is a null string which indicates any valid user credentials have been provided.

If the *criteriaType* is *Application*, the *criteria* is the *ApplicationUri* from the *Client Certificate* used for the *Session*. The *Client Certificate* shall be trusted by the *Server*. This criteria type is used if a *Role* should be granted to a *Session* for *Application Authentication* with *Anonymous UserIdentityToken*. If a *Role* should be granted to a *Session* for *Application Authentication* combined with *User Authentication*, the *Applications Property* on the *RoleType* is combined with the *Identities Property* on the *RoleType* as defined in 4.4.1.

If the *criteriaType* is *X509Subject*, the criteria is the X509 subject name of a *Certificate* of a user which is trusted by the *Server*. The format of the subject name criteria consists of a sequence of name value pairs separated by a ‘/’. The name shall be one of entries in Table 8 and shall be followed by a '=' and then followed by the value, which is always enclosed in double quotes (""). The order shall be by the order shown in Table 8 with the lowest number first. Every value from Table 8 present in the *Certificate* shall be included in the criteria, others must not be included. The value may be any printable character except for "". For example: CN="User Name"/O="Company". Table 8 contains all subject name attributes where support is required by X509 and some commonly used attributes where support is optional. Additional fields may be added in the future. If one name is used multiple times in the certificate, the name is also

repeated in the criteria. The entries with the same name are entered in the order they appear in the *Certificate*. All names listed in Table 8 that are included in the X509 subject name shall match the content of the criteria *String*. Names not included in Table 8 are ignored.

Table 8 – Order for subject name criteria

Order	Name	Value
1	CN	Common Name
2	O	Organization
3	OU	Organization Unit
4	DC	Domain Component
5	L	Locality
6	S	State
7	C	Country
8	dnQualifier	Distinguished name qualifier
9	serialNumber	Serial number

The *IdentityMappingRuleType Structure* representation in the *AddressSpace* is defined in Table 9.

Table 9 – IdentityMappingRuleType definition

Attributes	**Value**			
BrowseName	IdentityMappingRuleType			
IsAbstract	True			
References	**NodeClass**	**BrowseName**	**IsAbstract**	**Description**
Subtype of Structure defined in OPC 10000-5.				
Conformance Units				
Base Info ServerType				

4.4.4 IdentityCriteriaType

The *IdentityCriteriaType* Enumeration is defined in Table 10.

Table 10 – IdentityCriteriaType Values

Name	Value	Description
UserName	1	The rule specifies a UserName from a *UserNameIdentityToken*.
Thumbprint	2	The rule specifies the *Thumbprint* of a user or CA *Certificate*.
Role	3	The rule is a *Role* specified in an *Access Token.*
GroupId	4	The rule is a user group specified in the *Access Token.*
Anonymous	5	The rule specifies *Anonymous UserIdentityToken.*
AuthenticatedUser	6	The rule specifies any non *Anonymous UserIdentityToken.*
Application	7	The rule specifies the combination of an application identity and an *Anonymous UserIdentityToken.*
X509Subject	8	The rule specifies the X509 subject name of a user or CA *Certificate*.

Its representation in the *AddressSpace* is defined in Table 11.

Table 11 – IdentityCriteriaType Definition

Attribute	**Value**				
BrowseName	IdentityCriteriaType				
IsAbstract	False				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the Enumeration type defined in OPC 10000-5					
HasProperty	Variable	EnumValues	EnumValueType []	PropertyType	
Conformance Units					
Base Info ServerType					

4.4.5 AddIdentity Method

This *Method* is used to add an identity mapping rule to a *Role*.

The *Client* shall use an encrypted channel and shall provide user credentials with administrator rights when invoking this *Method* on the *Server*.

Signature

```
AddIdentity (
   [in]  IdentityMappingRuleType Rule
   );
```

Argument	Description
Rule	The rule to add.

Method Result Codes

ResultCode	Description
Bad_InvalidArgument	The rule is not valid.
Bad_RequestNotAllowed	The rule cannot be added to the *Role* because of *Server* imposed restrictions.
Bad_NotSupported	The rule is not supported by the *Server*.
Bad_AlreadyExists	An equivalent rule already exists.

4.4.6 RemoveIdentity Method

This *Method* is used to remove an identity mapping rule from a *Role*.

The *Client* shall provide user credentials with administrator rights when invoking this *Method* on the *Server*.

Signature

```
RemoveIdentity (
   [in]  IdentityMappingRuleType Rule
   );
```

Argument	Description
Rule	The Rule to remove.

Method Result Codes

ResultCode	Description
Bad_NotFound	The rule does not exist.
Bad_UserAccessDenied	The session user is not allowed to configure the object.

4.4.7 AddApplication Method

This *Method* is used to add an application mapping rule to a *Role*.

The *Client* shall provide user credentials with administrator rights when invoking this *Method* on the *Server*.

Signature

```
AddApplication (
   [in]  String ApplicationUri
   );
```

Argument	Description
ApplicationUri	The *ApplicationUri* for the application.

Method Result Codes

ResultCode	Description
Bad_InvalidArgument	The *ApplicationUri* is not valid.
Bad_RequestNotAllowed	The mapping cannot be added to the *Role* because of *Server* imposed restrictions.
Bad_AlreadyExists	The ApplicationUri is already assigned to the *Role*.
Bad_UserAccessDenied	The session user is not allowed to configure the object.

4.4.8 RemoveApplication Method

This *Method* is used to remove an application mapping rule from a *Role*.

The *Client* shall provide user credentials with administrator rights when invoking this *Method* on the *Server*.

Signature

```
RemoveApplication (
   [in]  String ApplicationUri
   );
```

Argument	Description
ApplicationUri	The *ApplicationUri* for the application.

Method Result Codes

ResultCode	Description
Bad_NotFound	The ApplicationUri is not assigned to the *Role*.
Bad_UserAccessDenied	The session user is not allowed to configure the object.

4.4.9 AddEndpoint Method

This *Method* is used to add an endpoint mapping rule to a *Role*.

The *Client* shall provide user credentials with administrator rights when invoking this *Method* on the *Server*.

Signature

```
AddEndpoint (
   [in]  EndpointType Endpoint
   );
```

Argument	Description
Endpoint	The *Endpoint to add*.

Method Result Codes

ResultCode	Description
Bad_InvalidArgument	The *EndpointUrl* is not valid.
Bad_RequestNotAllowed	The mapping cannot be added to the *Role* because of *Server* imposed restrictions.
Bad_AlreadyExists	The *Endpoint* with the passed settings is already assigned to the *Role*.
Bad_UserAccessDenied	The session user is not allowed to configure the object.

4.4.10 RemoveEndpoint Method

This *Method* is used to remove an endpoint mapping rule from a *Role*.

The *Client* shall provide user credentials with administrator rights when invoking this *Method* on the *Server*.

Signature

```
RemoveEndpoint (
    [in]  EndpointType Endpoint
    );
```

Argument	Description
Endpoint	The *Endpoint to remove*.

Method Result Codes

ResultCode	Description
Bad_NotFound	The *EndpointUrl* is not assigned to the *Role*.
Bad_UserAccessDenied	The session user is not allowed to configure the object.

4.5 RoleMappingRuleChangedAuditEventType

This *Event* is raised when a mapping rule for a *Role* is changed.

This is the result of calling any of the add or remove *Methods* defined on the *RoleType*.

It shall be raised when the *AddIdentity, RemoveIdentity, AddApplication, RemoveApplication, AddEndpoint* or *RemoveEndpoint* Method causes an update to a *Role*.

Its representation in the *AddressSpace* is formally defined in Table 12.

Table 12 – RoleMappingRuleChangedAuditEventType definition

Attribute	Value				
BrowseName	RoleMappingRuleChangedAuditEventType				
IsAbstract	True				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of the *AuditUpdateMethodEventType* defined in OPC 10000-5					
Conformance Units					
Security Role Server Base Eventing					

This *EventType* inherits all *Properties* of the *AuditUpdateMethodEventType*. Their semantics are defined in OPC 10000-5.

_____