

Version 0.3

# Content Management Interoperability Services

Part I – Namespace Proposal

## CONTENTS

Contents .....	2
Namespaces.....	4
Motivation .....	4
Proposed Changes .....	5

## Change Summary

Version 0.1 (Jens Hübel, Florian Müller)

- Initial draft

Version 0.2 (Jens Hübel, Florian Müller, Julian Reschke)

- Clarification prefix/namespaces
- Added globally unique name to property definitions

Version 0.3 (Jens Hübel, Florian Müller, Julian Reschke)

- Added globally unique name to type definitions

## NAMESPACES

### MOTIVATION

The current CMIS specification draft handles properties as a flat list of key-value pairs. That may cause name conflicts between CMIS base property names and type specific property names. The same holds true type names. It is likely that some repositories already use the type names “Document” or “Folder”.

During the face-to-face meetings and in various discussions proposals have come up to extend this naming to a more powerful mechanism by introducing namespaces. The following proposal is an approach to extend the CMIS specification with a simple concept of namespaces, being widely compatible with the existing specification. The proposal touches type names and type query names as well as property names and property query names.

Furthermore, many other systems use globally unique property identifiers in order to promote distributed extensibility. Examples are RDF (which uses IRIs) and JCR and WebDAV (which use XML expanded names, consisting of a namespace IRI and a local name). If these types of systems are accessed over CMIS, it is desirable to expose the “native” property identifier to clients. On the other hand, when CMIS based systems are accessed through JCR or over WebDAV, it would be useful if the repository can expose a globally unique identifier as well.

Therefore, we propose to make that globally unique property name an optional field of the property definition.

Introducing namespaces provides the following advantages:

1. The CMIS specification defines a set of standard properties. An existing repository may have its own properties with the same names but with different semantics or different sets of constraints. A namespace allows clearly distinguishing between CMIS properties and repository or type specific properties.
2. Certain repositories allow using the same name of a property to be used at more than one place with a different meaning. Common property names like “Name” or “Invoice” might exist in different flavors and in different application contexts.
3. Future versions of the CMIS specifications may define additional standard properties (e.g. for Records Management). By introducing a separate namespace for the CMIS standard we can guarantee backwards compatibility between a newer version of the specification and an existing repository or application implementation.
4. A namespace can be the foundation to introduce extended properties (e.g. hierarchies). Here are some relationships to other upcoming proposals, for example Aspects/Mixins that should be synchronized.

A namespace is used to group properties within a type. A namespace is expressed by a prefix to the property name. A prefix on the object type level helps to distinguish the CMIS “Document” type from a repository “Document” type.

Using prefixes is optional with the exception of the CMIS standard properties and types. A repository that does not support namespaces can provide the same implementation as for the current CMIS spec and is still CMIS conformant.

## PROPOSED CHANGES

### Proposed changes for type prefixes

In “2.7.2.1 Attributes common to ALL Object-Type Definitions”, add to the definition of “typeId” and “queryName”:

The type Id and/or the query name MAY have one or more prefixes. Prefixes SHALL conform to the syntax rules for SQL identifiers. A colon (“:”) separates a prefix from the local name and another prefix. The local name (behind the last colon reading from left-to-right) is always a valid identifier. The prefix “cmis:” identifies a reserved namespace and MUST NOT be used by any repository except for names that are defined in this specification.

The length of the property name plus prefix is not limited.

Add the prefix “cmis:” to all base types.

### Proposed changes for property prefixes

In “2.7.3.2 Attributes common to ALL Object-Type Property Definitions”, add to the definition of “name”:

The property name MAY have one or more prefixes. Prefixes SHALL conform to the syntax rules for SQL identifiers. A colon (“:”) separates a prefix from the local name or another prefix. The local name (behind the last colon reading from left-to-right) is always a valid identifier. Examples are “myrepository:name” or “MyRepository:Finance:Order:OrderNumber”. The prefix “cmis:” identifies a reserved namespace and MUST NOT be used by any repository except for names that are defined in this specification.

The length of the property name plus prefix is not limited.

Replace all predefined property names so that they are prefixed with “cmis:”. For example, change “CreatedBy” to “cmis:CreatedBy”.

### Proposed changes of query grammar

Change the following lines in the BNF of the CMIS query grammar from:

```
<correlation name> ::= <identifier>  
<table name> ::= <identifier>  
<column name> ::= <identifier>
```

to:

<correlation name> ::= <prefixedidentifier>  
<table name> ::= <prefixedidentifier>  
<column name> ::= <prefixedidentifier>

Add line:

<prefixedidentifier> ::= [ ( <prefixedidentifier> | <identifier> ) ":" ] <identifier>

unchanged:

<identifier> ::= !! As defined by SQL-92 grammar

---

## Proposed changes for globally unique type id

In "2.7.2.1 Attributes common to ALL Object-Type Definitions", add:

Name: globallyUniqueName

Type: string

Function: A globally unique name for this object type. This is either a IRI (RFC 3987, Section 2.2), or an XML NS Expanded Name (XMLNS, Section 2.1), consisting of a namespace IRI and a local name. The lexical representation for the first form is just the IRI, for the second form it is the Clark Notation ("{" + namespace + "}" + localname)".

---

## Proposed changes for globally unique property id

Part I, Section 2.7.3.2:

Add the following attribute:

Name: globallyUniqueName

Type: string

Function: A globally unique name for this property. This is either a URI IRI (RFC 3987, Section 2.2), or an XML NS Expanded Name (XMLNS, Section 2.1), consisting of a namespace IRI and a local name. The lexical representation for the first form is just the IRI, for the second form it is the Clark Notation ("{" + namespace + "}" + localname) of the expanded name.