

## SECTION 3: THE APIs



The API documentation below is organized first by object, and then by CRUD operation. Therefore, to look up **deleteContentClass**, for example, first look up the **ContentClass** object, and then look under the **Delete** operation.

### 3.1 CONTENTCLASS

ContentClass objects are essentially lists of Element references. You can use them to specify database schemas, web forms, search interfaces, and so on.

ContentClasses are arranged in a single-inheritance tree, starting from the global root. One branch descending from the global root is reserved for extension property definitions. Classes within that branch are used to define extension property structure for all extensible objects, such as Elements and Terms. These classes can be extended as well, for example to specify sets of extension properties for Terms in different Vocabularies. These classes have predefined GUIDs (see Predefined Constants).

When creating a Content Class, you need to specify the parent class ID (all Content Classes reside in one tree, so each class you create must have a parent. You can find some predefined values in Predefined Constants).

ContentClass children inherit from their parents in a complex way. Any Elements associated with a given ContentClass are also implicitly associated with all descendant ContentClasses. There is no masking or negation of inherited Elements. However, certain properties of the Element can be overridden. When the aggregate definition of the ContentClass in question is rendered, locally overridden properties take precedence over the native values of the properties set on the Element or overridden by ancestor ContentClasses.

Element properties that can be overridden include the following:

- ◆ name
- ◆ minOccurs
- ◆ maxOccurs
- ◆ defaultValue
- ◆ vocViewId

If a user has ownership rights to a ContentClass, that user has ownership rights to all descendant ContentClasses.

### 3.1.1 Properties

PROPERTY	TYPE	DESCRIPTION
<code>createdAt</code>	integer	Read-only. This is the date and time of creation. It is set only once, when the object is created. Its value is the number of milliseconds that have elapsed since the Java "epoch," which is 01/01/1970 00:00 GMT.
<code>description</code>	string	Human-readable description of the object.
<code>extensionProperties</code>	Set	Extension properties. The extension properties can currently be returned only in an "all or nothing" manner. Picking individual extension properties by using a finer grained projection is planned for a future release.
<code>id</code>	Identifier	GUID of the object.
<code>lastModified</code>	integer	Read-only. This is the date and time of the last modification of the Term. Its value is the number of milliseconds that have elapsed since the Java "epoch", which is 01/01/1970 00:00 GMT. If the object has been created and never modified, the value is null.
<code>level</code>	integer	Read-only. Determined by the position of the ContentClass in the tree. The root class is level 1, its immediate children are level 2, their children are level 3, and so on.
<code>localizedValues</code>	Set	Localized (internationalized) translations of ContentClass values and descriptions.
<code>name</code>	string	Should be unique.
<code>parentId</code>	Identifier	The ID of the parent of the ContentClass. The root class has no parent, so its parentId is null.

### 3.1.2 Relationships

#### ELEMENTS (M – M, OPTIONAL)

ContentClasses may have one or more Elements associated with them. Elements may be associated with the ContentClass directly, or inherited from parent ContentClasses.

Each relationship between a ContentClass and Element is modified by a ClassElement object providing contextual properties that override or alter the use of the Element definition.

Unlike XML, SchemaServer does not support embedding definitions of sub-Elements within the encompassing structure (in this case, the ContentClass). All sub-Element definitions are drawn from a globally visible set of Elements. This encourages definition reuse.

#### PARENT CLASS (1-M, REQUIRED)

All ContentClasses other than the built-in root class have one parent ContentClass. This relationship specifies the class inheritance mechanism.

### 3.1.3 ContentClass Operations

#### CREATE

*Identifier* **createContentClass** (*ContentClass cc*, *Identifier parentID*)

Creates a ContentClass object and returns its ID.

##### Parameters:

- ♦ **cc** – The ContentClass object to be created.
- ♦ **parentID** – The ID of the parent ContentClass of the object to be created.

##### Returns:

*Identifier* – the ID of the ContentClass object that was created.

##### Examples

```
CreateContentClass.cs  
CreateContentClass.java
```

#### READ

*ContentClass* **getContentClass** (*Identifier ccID*)

Returns the ContentClass object with the specified ID.

##### Parameters:

- ♦ **ccID** – The ID of the ContentClass object to return.

##### Returns:

*ContentClass* – The ContentClass object with the specified ID, or null if it does not exist.

##### Examples

```
GetContentClass.cs  
GetContentClass.java
```

#### UPDATE

*boolean* **updateContentClass** (*ContentClass cc*, *FieldMask fm*)

Updates the fields of a ContentClass object according to the fields specified by a FieldMask object.

##### Parameters:

- ♦ **cc** – The ContentClass object to update.
- ♦ **fm** – A FieldMask object specifying the fields to update.

##### Returns:

*boolean* – True if the ContentClass was updated successfully, otherwise false.

##### Examples

```
UpdateContentClass.cs
```