



Extending the Object Model for the Cisco Collaboration Cloud

Developer Technical Note

Cisco Systems, Inc., WebEx Technology Group
3979 Freedom Circle, Santa Clara, CA 95054

Corp: +1-408-435-7000 Sales: +1.877.50.WEBEX

[Hwww.webex.com](http://www.webex.com)



Introduction

The Cisco Collaboration Cloud Platform offers flexible and robust means for integrating customer and partner applications. A core feature of the platform is the back-end object model provided by the Collaboration Cloud Platform Services API layer. Applications can not only interact and interrogate this data model, but can securely extend the model in a private namespace. This allows the developer to store and manage custom data associated with the customer organization, individual users, and other objects in the system. Existing objects can be extended by adding custom fields, and entirely new and independent object can be defined as needed.

Cisco Collaboration Cloud Platform Services

The Cisco Collaboration Cloud Platform Services (CCCPS) constitute a web-services interface to the back-end object model and services that provide the backbone of the platform. These include the following:

- The core object model supporting private (namespace-scoped) extensibility, as well as entirely custom objects known as “things”
- Authentication services
- Policy model
- User and identity management

For complete information on accessing this data and these services, refer to the *Cisco Collaboration Cloud Platform Server* reference guide found on the Cisco Developer Network (CDN) web site.

Objects are represented as XML entities with a well defined schema for each type, and are classified into five categories:

- Namespace objects provide a method of distinguishing among identically named objects belonging to different organizations, groups, users, and workspaces.
- Affiliation objects organize users into organizations, groups, and define the user’s roles and privileges with the affiliated entities, typically based on policies.
- Information objects encapsulate knowledge in the form of documents, calendars and events.
- Communication objects represent methods for users to interact, such as persistent chat.
- WebEx objects represent customary WebEx services, such as Meetings and Sessions.

Basic Entry Points

The following are just some of the actions that are supported by the CCCPS API:

- *login / logout* – authentication and session management
- *create / delete* – instantiate or remove objects
- *get / set* – retrieve or define specific properties of an object
- *add / remove* – insert or remove a recurring element in an object
- *shape / validate* – shape returns the elements and attributes of an object, allowing dynamic discovery of an object’s structure; validate tests that an object conforms to its definition
- *commit / rollback* - manage updates to the database

Built-in Objects

The table below lists pre-defined object within CCCPS, and provides a brief description. Each of these can be extended as discussed later in this tech note.

<i>Object Type</i>	<i>Description</i>
User	Information about a specific user.

Group	Information about a specific group. Groups are the general container object in the Collaboration Cloud Platform. They can contain other groups, users, applications, documents, and custom objects.
Organization	The outer most group that corresponds to an organization. Unaffiliated users belong to a degenerate organization (called 'General Public'). The organization object is where organizational policies can be placed.
Namespace	Namespaces are used to avoid name collisions in the platform. For example, a user name needs to be unique only within the namespace. A namespace will map to only one organization, but an organization can have multiple namespaces.
Role	In the platform, a role is a container for one or more privileges. Roles are assigned to users in some context (such as a Group or a Space). In making the assignment, you are essentially saying that this user, in this context, has these privileges. However, it is up to policy rules to determine what the privileges really mean.
Privilege	A capability that a user can have. The interpretation of this capability is done by one or more policies executing in context.
Policy	A policy (rule) is a set of directives that define how access control and security is to be implemented. Policies can be associated with organizations, groups, applications, documents, custom objects. Policies essentially add the meaning to roles and privileges.
Thing	A custom object, whose definition is entirely up to the developer.

Extending the Model

Each object may be extended within an `<ext></ext>` stanza, allowing partners and developers to customize objects in order to meet particular needs. All custom extensions are attached here. These extensions however are not returned via the Shape command.

Extending Built-in Objects

Under the `<ext>` element is one or more DOM sub-trees. The root element of the sub-tree is the namespaceID of the owning organization. Depending on your context, you will probably only 'see' the custom extensions for your organization. Extensions for other organizations will generally be hidden unless they have exposed them for your use.

The XSD for each standard object will accept any "well-formed" XML within the `<ext>` stanza. The nature and structure of the extension and its content is entirely determined by the application that invoked the `<ext>` stanza, which is responsible for assuring the consistency and validity of the extension.

Below is an example using three different extensions for the user object:

```

<user>
  <userID/>
  <email/>
  <name/>
  <homeGroupID/>
  <groups/>
  <ext>
    <N1234142568124526843985217>
    ... extension DOM for namespace N1234142
    </N1234142568124526843985217>
    <NAFAEIFE568124526843985217>
    ... extension DOM for namespace NAFAEIFE
    </NAFAEIFE568124526843985217>
  </ext>
</user>

```

```

<N12AASQQ568124526843985217>
... extension DOM for namespace N12AASQQ
</N12AASQQ568124526843985217>
</ext>
</user>

```

Defining Custom Objects

The thing object is a type of custom object or resource. Examples include documents, recordings, and custom objects. For custom objects, the user is allowed to provide a validating DOM structure. The `<thingType/>` field will contain the `guid` of the validating DOM. The validating DOM is stored as another 'thing'.

<i>Thing Type</i>	<i>Description</i>
doc	Documents
GUID	A custom object type where the type is represented by the <code>thingID</code> of the definition.
none (or omitted)	Generic custom object with no validating DOM.

Conclusion

The Cisco Collaboration Cloud Platform not only provides a rich set of collaboration and content management services, but it also is fundamentally a robust infrastructure upon which to extend existing applications, and build entirely new applications.

©2011 Cisco Systems, Inc. All rights reserved. The Cisco WebEx logo is a registered trademark of Cisco Systems, Inc. All other trademarks are the property of their respective owners.