LivCos

Cosmos Outline

Table of Contents

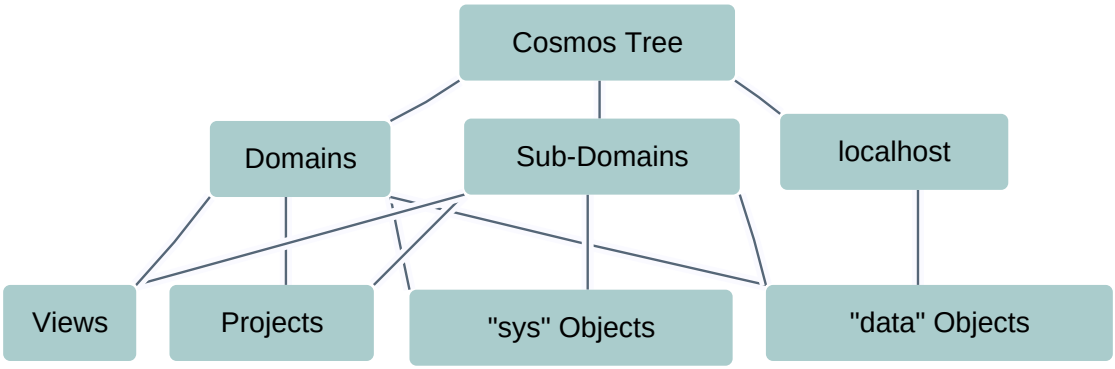
1 Intoduction

While, in theory, the objects within the Cosmos could be placed where ever the author(s) likes them to be, some guidelines and recommendations should help to keep things organized in the practical usage.

Also the system interprets certain objects (domains, sys, data) specially and so imposes some limitations to the outline.

2 Cosmos Tree

The objects within the Cosmos are organized in a tree model with "the Cosmos" as the root.



2.1 Domains

At the first level of the hierarchy we find only domain objects. They need to have a globally unique name to allow any two LivCos instances to work together.

Object names within a domain must be managed to be unique by the people, responsible for that domain.

For domain objects the system automatically defines a text-search index.

2.1.1 Sub-Domains

Sub-domains are interpreted just like normal domains. The only difference is in the naming convention to form the unique name (eg. livcos.org, demo.livcos.org, test.livcos.org are all domains in LivCos).

2.2 Views

The definitions to represent the content data in a certain style, layout, format are called views.

View objects are normal objects, except they usually contain a sys object and a URI resolver. In this sys/resolver object accessing URIs can be transformed into more complex access definitions with parameters,.... URIs can be split into a part to access the view and another for the actual data the view is supposed to show.

2.2.1 Medium, Form

It is recommended to group the different views into the physical target medium first and then into their logical form of presentation. Eg. web/doc represents the data as documents for the Web, page/doc as documents to print page oriented and web/map shows it as a link map on the Web again.

2.2.2 Definition per Namespace

Definition stylesheets (XSLT) to represent entities of a specific namespace.

2.3 "sys" Objects

Sys objects contain system related objects. They can define parameters on how the system should handle the sibling objects.

With a given object URI the system scans the ancestors for a "sys" sibling and applies it's contents to the current activity (resolve URIs, handle exceptions, reporting status,...).

2.3.1 sys/resolver

--content--

2.4 "data" Objects

--content--

2.5 Projects

Project objects contain all the project related objects and are placed directly in the domain. They normally contain project specific views and their content data.

A project object normally contains a data/Intro object to summarize the project description and to provide the entry for the project data (links to the major documents, jobs,...).

2.5.1 Sub-Projects

Project objects can contain other project objects.

2.6 localhost

The "localhost" is a special domain object, representing the server instance or host. It contains system log data, server configurations, access keys, search indexes,....

All the data, contained in localhost, is available locally only and cannot be shared directly with other instances.

3 View Examples

3.1 data

This view represents the data of the Cosmos in it's initial form. Structured content data forms a corresponding entity hierarchy, unstructured content is provided or streamed as binary data.

3.2 sys

The view for the system. Found in different levels within the hierarchy, it provides definitions for the parent view about URL resolvement, job status, -error feedback,....

3.3 cpml

Represents the content in the Common Presentation Markup Language. This intermediate format can be used as a base for Web or page-based views.

3.4 www

Simple Web page view to represent the content in most common Web Browsers.

3.5 web

More advanced Web view, using technologies like AJAX to represent the content on modern Web browsers.

3.5.1 doc

An XHTML view to show the data in a document-like layout (title, toc, chapters,...). It may contain SVG graphics and MathML formulas.

Web Browsers like Firefox 3, Chrome 2, Safari 4 or Opera 9 should support this view.

3.5.1.1 ie

IE specific view for a document-like Web page layout. Basically it takes the parent view and converts XHTML into HTML, SVG into VML, ...

3.5.2 map

An XHTML view to represent the data in a Web Map layout.

The Web Browser support is similar to the "doc" view.

3.5.3 edit

Editor views and controllers for the different entity types.

3.5.4 admin

Web pages for administration tasks and overviews.

3.6 page

View to represent the content in paper-page oriented formats like PDF, RTF,....

3.7 mobile

Web page view, optimized to be viewed on mobile devices (PDA, Smartphone).