KEA Documentation

Table of Contents

CXL: Concept Mapping Extensible Language

CXL is a publicly available XML-based language for describing the content of Cmaps. CXL is defined by the XML Schema http://cmap.ihmc.us/xml/cmap.xsd. The structure of a Cmap in CXL is the following:

```
<cmap xmlns="http://cmap.ihmc.us/xml/cmap/"
      xmlns:dc="http://purl.org/dc/elements/1.1/"
      xmlns:dcterms="http://purl.org/dc/terms/"
      xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#">
  <res-meta>
    <dc:contributor> <!-- Last user that modified the resource -->
       <vcard:FN>Full name (First [Middle|Initial] Last)</vcard:FN>
       <vcard:EMAIL>Email address</vcard:EMAIL>
       <vcard:ORG>
          <vcard:OrgName>Organization name</vcard:OrgName>
       </vcard:ORG>
    </dc:contributor>
    <dc:creator> <!-- Original author -->
       <vcard:FN>Full name (First [Middle|Initial] Last)</vcard:FN>
       <vcard:EMAIL>Email address</vcard:EMAIL>
       <vcard:ORG>
          <vcard:OrgName>Organization name</vcard:OrgName>
       </vcard:ORG>
    </dc:creator>
    <dc:description>Focus question of the Cmap</dc:description>
    <dc:format>x-cmap/text-xml</dc:format>
    <dc:identifier>HTTP URL with resource id (e.g.
    http://cmap.ihmc.us/rid=10002929_292992_19)</dc:identifier>
    <dc:language>Language code (RFC1766) (e.g. en_US)</dc:language>
    <dc:publisher>IHMC CmapTools v. 4.X</dc:publisher>
    <dc:relation>HTTP URL, location based (e.g.
    http://cmap.ihmc.us/plants.cxl)</dc:relation>
    <dc:source>cmap:<server-id>:<folder-id>:<resource-id></dc:source>
    <dc:subject>Keyword list, comma delimited</dc:subject>
    <dc:title>Name of the Cmap</dc:title>
    <dcterms:created>Date when the Cmap was created (ISO Date yyyy-MM-
    dd'T'HH:mm:ss'Z' GMT Time Zone)</dcterms:created>
    <dcterms:extent>size (integer [space] units)</dcterms:extent>
    <dcterms:modified>Date of last modification (ISO Date yyyy-MM-
    dd'T'HH:mm:ss'Z' GMT Time Zone)</dcterms:modified>
    <dcterms:rightsHolder> <!-- Cmap owner -->
       <vcard:FN>Full name (First [Middle|Initial] Last)</vcard:FN>
       <vcard:EMAIL>Email address</vcard:EMAIL>
       <vcard:ORG>
          <vcard:OrgName>Organization name</vcard:OrgName>
       </vcard:ORG>
    </dcterms:rightsHolder>
  </res-meta>
   <map>
      <concept-list>
         <concept/>
         ...
      </concept-list>
      <linking-phrase-list>
         <linking-phrase/>
         ...
      </linking-phrase-list>
```

```
<connection-list>
    <connection/>
    ...
</connection-list>
<resource-group-list>
    <resource-group>
        <resource/>
        ...
    </resource-group>
    ...
</resource-group-list>
<proposition-list>
    <proposition>
        <prop-conn/>
        ...
    </proposition>
    ...
</proposition-list>
<not-a-proposition-list>
    <proposition>
        <prop-conn/>
        ...
    </proposition>
    ...
</not-a-proposition-list>
<concept-appearance-list>
    <concept-appearance>
        <localized-style/>
        ...
    </concept-appearance>
    ...
</concept-appearance-list>
<linking-phrase-appearance-list>
    <linking-phrase-appearance>
        <localized-style/>
        ...
    </linking-phrase-appearance>
    ...
</linking-phrase-appearance-list>
<connection-appearance-list>
    <connection-appearance>
        <control-point/>
        ...
    </connection-appearance>
    ...
</connection-appearance-list>
<resource-appearance-list>
    <resource-appearance/>
    ...
</resource-appearance-list>
<style-sheet-list>
    <style-sheet>
        <map-style/>
        <concept-style/>
        <linking-phrase-style/>
        <connection-style/>
        <resource-style/>
    </style-sheet>
    ...
</style-sheet-list>
<extra-properties-list>
    <property-list>
        <property/>
```

```
            ...
         </property-list>
         ...
      </extra-properties-list>
      <extra-graphical-properties-list>
         <property-list>
            <property/>
            ...
         </property-list>
         ...
      </extra-graphical-properties-list>
      <image-list>
         <image/>
         ...
      </image-list>
   </map>
</cmap>
```

cmap

This is the main element of a CXL definition. This element encloses the definition of a Concept map in CXL.

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
res-meta	0	1
map	0	1

map

Defines the structure of the map.

Attributes

Name	XML Schema Type	Required	Description
root-id	string	No	The id of the root concept of the map
header	string	No	The text of the header
footer	string	No	The text of the footer
width	int	No	the minimum width of the map's canvas
height	int	No	the minimum height of the map's canvas
default-stylesheet-id	string	No	the unique id of the maps current default named stylesheet

Child Elements

Name	Min. Occurrences	Max. Occurrences
concept-list	0	1
linking-phrase-list	0	1
connection-list	0	1
resource-group-list	0	1
proposition-list	0	1
not-a-proposition-list	0	1
concept-appearance-list	0	1
linking-phrase-appearance-list	0	1
connection-appearance-list	0	1
resource-appearance-list	0	1
style-sheet-list	0	1
extra-properties-list	0	1
extra-graphical-properties-list	0	1
image-list	0	1

concept-list

List of concepts in the map.

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
concept	0	Unbounded

linking-phrase-list

List of linking phrases in the map.

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
linking-phrase	0	Unbounded

connection-list

List of connections in the map.

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
connection	0	Unbounded

resource-group-list

List of resource groups in the map.

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
resource-group	0	Unbounded

proposition-list

List of propositions in the map.

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
proposition	0	Unbounded

not-a-proposition-list

A List of propositions that the user has defined as not valid propositions in this map.

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
proposition	0	Unbounded

concept-appearance-list

List of concept appearances.

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
concept-appearance	0	Unbounded

linking-phrase-appearance-list

List of linking phrase appearances.

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
linking-phrase-appearance	0	Unbounded

connection-appearance-list

List of connection appearances.

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
connection-appearance	0	Unbounded

resource-appearance-list

List of resource appearances.

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
resource-appearance	0	Unbounded

style-sheet-list

List of named style sheets

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
style-sheet	0	Unbounded

extra-properties-list

List of extra properties lists

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
property-list	0	Unbounded

extra-graphical-properties-list

List of extra graphical properties lists

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
property-list	0	Unbounded

image-list

List of images

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
image	0	Unbounded

concept

Attributes

Name	XML Schema Type	Required	Description
id	string	No	The unique Id of the Node. Must be unique for all nodes concepts and linking phrases
parent-id	string	No	The unique Id of the parent container if not the map (usually the Id of another Node)
label	string	Yes	The text displayed in the node on screen. Used as the reference id if no id is provided
short-comment	string	No	Extra concise text info about this node
long-comment	string	No	Extra text info about this node
uri	string	No	Unique identifier for this entity

Child Elements

None

linking-phrase

Attributes

Name	XML Schema Type	Required	Description
id	string	No	The unique Id of the Node. Must be unique for all nodes concepts and linking phrases
parent-id	string	No	The unique Id of the parent container if not the map (usually the Id of another Node)
label	string	Yes	The text displayed in the node on screen. Used as the reference id if no id is provided
short-comment	string	No	Extra concise text info about this node
long-comment	string	No	Extra text info about this node
uri	string	No	Unique identifier for this entity

Child Elements

None

connection

Attributes

Name	XML Schema Type	Required	Description
id	string	No	The unique Id of the Connection (required only if appearance is defined)
parent-id	string	No	The unique Id of the parent container if not the map (usually the Id of another Node)
from-id	string	Yes	The unique Id of the Node where the connection starts
to-id	string	Yes	The unique Id of the Node where the connection ends
isBidirectional	boolean	No	true if the connection is bidirectional (defaults to false)

Child Elements

None

resource-group

Attributes

Name	XML Schema Type	Required	Description
parent-id	string	Yes	the id of the node this group is connected to
group-type	string	No	the type of this group (unknown, image, soup, html, cmap, text, <text-and-image>, audio, video, executable, discussion-thread, organizer, email, shortcut)

Child Elements

Name	Min. Occurrences	Max. Occurrences
resource	0	Unbounded

resource

Description of a resource linked to by this map.

Attributes

Name	XML Schema Type	Required	Description
id	string	No	the unique id for this resource description (only required if specifying appearance)
label	string	Yes	the Text displayed to the viewer if needed
description	string	No	a meaningful description of this resource
resource-name	string	No	the name of the linked resource
resource-mimetype	string	Yes	the MIME type of the linked resource
resource-server-id	string	No	The unique Id of the server containing this resource (required if no url specified)
resource-folder-id	string	No	The unique Id of the folder containing this resource(required if no url specified)
resource-id	string	No	The unique Id of the linked resource (required if no url specified)
resource-url	string	No	The url to the linked resource (required if sid,pid,rid triplet not specified)
focus-entity-id	string	No	The unique Id of the entity in the specified resource that is to be focused on

Child Elements

None

proposition

Each proposition is an ordered list of connections that should start and end with a concept with at least one linking phrase in between each concept in the proposition.

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
prop-conn	1	Unbounded

prop-conn

Attributes

Name	XML Schema Type	Required	Description
conn-id	string	Yes	Id of the connection

Child Elements

None

concept-appearance

Defines the individual appearance information for a concept

Attributes

Name	XML Schema Type	Required	Description
id	string	Yes	the unique id of the concept to which this appearance is applied
x	int	No	the center x location of this node (default is 10)
y	int	No	the center y location of this node (default is 10)
width	int	No	the minimum width of this node. -1 means to stretch to text visual size (default is -1)
height	int	No	the minimum height of this node. -1 means to stretch to text visual size (default is -1)
expanded	boolean	No	if true and this is a nested node then it should be drawn as expanded (defalut is false)
draw-order	int	No	index into draw order relative to parent (z-order)
font-name	string	No	the font family's name
font-size	decimal	No	the font point size
font-style	string	No	the font style (<plain>|italic|bold|underlined)
font-color	string	No	font color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
text-margin	int	No	size of whitespace around node text in pixels
text-alignment	string	No	text alignment in node bounds (, top, bottom, left, right, top-left, top-right, bottom-left, bottom-right)
background-color	string	No	background color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
background-image	string	No	the background image id reference. If "none" then no background image
background-image-style	string	No	How to draw the background image (<full>, scaled, cropped, or tiled)
border-color	string	No	border color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
border-style	string	No	border line style (none, <solid>, dotted, dashed, dash-dot, dash-dot-dot)
border-thickness	int	No	border line thickness (0-6 where 0 = no border and 6 is the largest)

border-shape	string	No	the border shape (rectangle, <rounded-rectangle>, or oval)
shadow-color	string	No	shadow color (an RGBA value [0-255, 0-255, 0-255, 0 or 255] or "none" if no shadow)
stylesheet-id	string	No	style info to apply to this node (default is _default_)

Child Elements

Name	Min. Occurrences	Max. Occurrences
localized-style	0	Unbounded

linking-phrase-appearance

Defines the individual appearance information for a linking phrase

Attributes

Name	XML Schema Type	Required	Description
id	string	Yes	the unique id of the linking phrase to which this appearance is applied
x	int	No	the center x location of this node (default is 10)
y	int	No	the center y location of this node (default is 10)
width	int	No	the minimum width of this node. -1 means to stretch to text visual size (default is -1)
height	int	No	the minimum height of this node. -1 means to stretch to text visual size (default is -1)
expanded	boolean	No	if true and this is a nested node then it should be drawn as expanded (defalut is false)
draw-order	int	No	index into draw order relative to parent (z-order)
font-name	string	No	the font family's name
font-size	decimal	No	the font point size
font-style	string	No	the font style (<plain>|italic|bold|underlined)
font-color	string	No	font color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
text-margin	int	No	size of whitespace around node text in pixels
text-alignment	string	No	text alignment in node bounds (, top, bottom, left, right, top-left, top-right, bottom-left, bottom-right)
background-color	string	No	background color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
background-image	string	No	the background image id reference. If "none" then no background image
background-image-style	string	No	How to draw the background image (<full>, scaled, cropped, or tiled)
border-color	string	No	border color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
border-style	string	No	border line style (none, <solid>, dotted, dashed, dash-dot, dash-dot-dot)
border-thickness	int	No	border line thickness (0-6 where 0 = no border and 6 is the largest)

border-shape	string	No	the border shape (rectangle, <rounded-rectangle>, or oval)
shadow-color	string	No	shadow color (an RGBA value [0-255, 0-255, 0-255, 0 or 255] or "none" if no shadow)
stylesheet-id	string	No	style info to apply to this node (default is _default_)

Child Elements

Name	Min. Occurrences	Max. Occurrences
localized-style	0	Unbounded

localized-style

Localized styles are applied to the text in the specified begin/end region if begin is not specified it is assumed to be 0, and if end is not specified it is assumed to be the end of the text. So by not setting begin and end you can apply the style change to the whole text. Order matters and if localized style bounds overlap the last specified will win.

Attributes

Name	XML Schema Type	Required	Description
begin	int	No	start index into the node's text where the specified change overrides the default
end	int	No	end index into the node's text where the specified change overrides the default (end not included)
font-name	string	No	Localized font name
font-size	decimal	No	Localized font point size
font-style	string	No	Localized font style (<plain>|italic|bold|underlined)
font-color	string	No	Localized font color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])

Child Elements

None

connection-appearance

Defines the individual appearance information for a connection

Attributes

Name	XML Schema Type	Required	Description
id	string	Yes	the unique id of the connection to which this appearance is applied
from-pos	string	No	connect to (, top, bottom, left, right, top-left, top-right, bottom-left, bottom-right)
to-pos	string	No	connect to (, top, bottom, left, right, top-left, top-right, bottom-left, bottom-right)
draw-order	int	No	index into draw order relative to parent
color	string	No	line color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
style	string	No	line style (none, <solid>, dotted, dashed, dash-dot, dash-dot-dot)
thickness	int	No	line thickness (0-6 where 0 = not visible and 6 is the largest)
type	string	No	the line type (straight, vector, bezier-3pt, bezier-4pt, or spline)
arrowhead	string	No	how to draw the arrow head on a connection yes - always draw the arrow head on the end or ends if bidirectional no - never draw the arrow head if-to-concept - draw if ending at a concept if-to-concept-and-slopes-up - draw if ending at a concept and in an upward slope
stylesheet-id	string	No	style info to apply to this connection (default is _default_)

Child Elements

Name	Min. Occurrences	Max. Occurrences
control-point	0	Unbounded

control-point

Control points are used to define how the connection looks. Order matters! If the connection type is straight, these points are ignored. If it is spline or Vector then all points are used. If it is bezier-3pt then the first one is used. If it is bezier-4pt the first 2 are used.

Attributes

Name	XML Schema Type	Required	Description
x	int	Yes	x coordinate of the connection control point
y	int	Yes	y coordinate of the connection control point

Child Elements

None

resource-appearance

Defines the individual appearance information for a resource

Attributes

Name	XML Schema Type	Required	Description
id	string	Yes	the unique id of the resource to which this appearance is applied
font-name	string	No	the font family's name
font-size	decimal	No	the font point size
font-style	string	No	the font style (<plain>|italic|bold|underlined)
font-color	string	No	font color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
background-color	string	No	background color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
stylesheet-id	string	No	style info to apply to this node (default is _default_)

Child Elements

None

style-sheet

A Style Sheet contains specific style information about the entities of a map.

Attributes

Name	XML Schema Type	Required	Description
id	string	Yes	The unique Id of the Style sheet (_Default_ and _LatestChanges_ are reserved as special ids)
name	string	No	the unique name of this style sheet. (_Default_ and _LatestChanges_ are reserved as special names)
parent-id	string	No	the parent style sheet id (if a style is not defined here then the parent is checked.)

Child Elements

Name	Min. Occurrences	Max. Occurrences
map-style	0	1
concept-style	0	1
linking-phrase-style	0	1
connection-style	0	1
resource-style	0	1

map-style

map-style defines the appearance of the map background. None of these fields are required. If you wish to reposition the background image then set the image-top-left value. If you wish to scale the background image, then use both the image-top-left and image-bottom-right values.

Attributes

Name	XML Schema Type	Required	Description
background-color	string	No	the background color of the map defined in an RGBA value (0-255, 0-255, 0-255, 0 or 255)
background-image	string	No	the background image id reference. If "none" empty then no background image
image-style	string	No	Styles are not-drawn, full, or tiled
image-top-left	string	No	Defines the (x,y) position of the top left corner of the background image
image-bottom-right	string	No	Defines the (x,y) position of the bottom-right corner of the background image

Child Elements

None

concept-style

concept-style defines the appearance of a concept

Attributes

Name	XML Schema Type	Required	Description
font-name	string	No	the font family's name
font-size	decimal	No	the font point size
font-style	string	No	the font style (<plain>|italic|bold|underlined)
font-color	string	No	font color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
text-margin	int	No	size of whitespace around node text in pixels
text-alignment	string	No	text alignment in node bounds (, top, bottom, left, right, top-left, top-right, bottom-left, bottom-right)
background-color	string	No	background color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
background-image	string	No	the background image id reference. If "none" then no background image
background-image-style	string	No	How to draw the background image (<full>, scaled, cropped, or tiled)
border-color	string	No	border color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
border-style	string	No	border line style (none, <solid>, dotted, dashed, dash-dot, dash-dot-dot)
border-thickness	int	No	border line thickness (0-6 where 0 = no border and 6 is the largest)
border-shape	string	No	the border shape (rectangle, <rounded-rectangle>, or oval)
shadow-color	string	No	shadow color (an RGBA value [0-255, 0-255, 0-255, 0 or 255] or "none" if no shadow)

Child Elements

None

linking-phrase-style

linking-phrase-style defines the appearance of a linking phrase

Attributes

Name	XML Schema Type	Required	Description
font-name	string	No	the font family's name
font-size	decimal	No	the font point size
font-style	string	No	the font style (<plain>|italic|bold|underlined)
font-color	string	No	font color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
text-margin	int	No	size of whitespace around node text in pixels
text-alignment	string	No	text alignment in node bounds (, top, bottom, left, right, top-left, top-right, bottom-left, bottom-right)
background-color	string	No	background color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
background-image	string	No	the background image id reference. If "none" then no background image
background-image-style	string	No	How to draw the background image (<full>, scaled, cropped, or tiled)
border-color	string	No	border color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
border-style	string	No	border line style (none, <solid>, dotted, dashed, dash-dot, dash-dot-dot)
border-thickness	int	No	border line thickness (0-6 where 0 = no border and 6 is the largest)
border-shape	string	No	the border shape (rectangle, <rounded-rectangle>, or oval)
shadow-color	string	No	shadow color (an RGBA value [0-255, 0-255, 0-255, 0 or 255] or "none" if no shadow)

Child Elements

None

connection-style

connection-style defines the appearance of a connection.

Name	XML Schema Type	Required	Description
color	string	No	line color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
style	string	No	line style (none, <solid>, dotted, dashed, dash-dot, dash-dot-dot)
thickness	int	No	line thickness (0-6 where 0 = not visible and 6 is the largest)
type	string	No	the line type (straight, vector, bezier-3pt, bezier-4pt, or spline)
arrowhead	string	No	how to draw the arrow head on a connection yes - always draw the arrow head on the end or ends if bidirectional no - never draw the arrow head if-to-concept - draw if ending at a concept if-to-concept-and-slopes-up - draw if ending at a concept and in an upward slope

Child Elements

None

resource-style

resource-style defines the appearance of a resource link.

Attributes

Name	XML Schema Type	Required	Description
font-name	string	No	the font family's name
font-size	decimal	No	the font point size
font-style	string	No	the font style (<plain>|italic|bold|underlined)
font-color	string	No	font color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])
background-color	string	No	background color (an RGBA value [0-255, 0-255, 0-255, 0 or 255])

Child Elements

None

property-list

Attributes

Name	XML Schema Type	Required	Description
id	string	Yes	the unique id of the connection to which this appearance is applied

Child Elements

Name	Min. Occurrences	Max. Occurrences
property	0	Unbounded

property

Attributes

Name	XML Schema Type	Required	Description
key	string	Yes	Key of the property
value	string	Yes	Value of the property

Child Elements

None

image

Attributes

Name	XML Schema Type	Required	Description
id	string	Yes	the unique id of this image
resource-server-id	string	No	The unique Id of the server containing this resource (required if no url specified)
resource-folder-id	string	No	The unique Id of the folder containing this resource(required if no url specified)
resource-id	string	No	The unique Id of the linked resource (required if no url specified)
resource-url	string	No	The url to the linked resource (required if sid, pid, rid triplet not specified)
bytes	base64Binary	Yes	Base 64 encoded image bytes

Child Elements

None

res-meta

Defines the metadata of a Cmap or of any resource inside the CmapTools environment. This element makes use of elements defined in the following namespaces:

Namespace	Prefix	Schema Location
http://purl.org/dc/elements/1.1/	dc	**dc.xsd**
http://purl.org/dc/terms/	dcterms	**dcterms.xsd**

Attributes

Name	XML Schema Type	Required	Description
content-type	string	No	If the referenced resource is a container then this attribute will tell what type of elements it contains inside the **content** element. It can either be **res-meta-list** or **part-list**

Child Elements

Name	Min. Occurrences	Max. Occurrences
dc:contributor	0	1
dc:creator	0	1
dc:description	0	1
dc:format	0	1
dc:identifier	0	1
dc:language	0	1
dc:publisher	0	1
dc:relation	0	1
dc:source	0	1
dc:subject	0	1
dc:title	0	1
dcterms:created	0	1
dcterms:extent	0	1
dcterms:modified	0	1
dcterms:rightsHolder	0	1
content	0	1

dc:contributor

Last user that modified the resource. This element makes use of elements defined in the following namespaces:

Namespace	Prefix	Schema Location
http://www.w3.org/2001/vcard-rdf/3.0#	vcard	No schema definition found yet

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
vcard:FN	0	1
vcard:EMAIL	0	1
vcard:ORG	0	1

dc:creator

Original author. This element makes use of elements defined in the following namespaces:

Namespace	Prefix	Schema Location
http://www.w3.org/2001/vcard-rdf/3.0#	vcard	No schema definition found yet

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
vcard:FN	0	1
vcard:EMAIL	0	1
vcard:ORG	0	1

dc:description

Focus question in the case of Cmaps or description of the resource otherwise.

Attributes

None

Child Elements

A string

dc:format

MIME type of the resource.

Attributes

None

Child Elements

A string

dc:identifier

HTTP URL with resource id (e.g. http://cmap.ihmc.us/rid=10002929_292992_19)

Attributes

None

Child Elements

A string

dc:language

Language code (**RFC1766**) (e.g. en_US)

Attributes

None

Child Elements

A string

dc:publisher

e.g. "IHMC CmapTools v. 4.X"

Attributes

None

Child Elements

A string

dc:relation

HTTP URL, location based (e.g. http://cmap.ihmc.us/plants.cxl)

Attributes

None

Child Elements

A string

dc:source

cmap:<server-id>:<folder-id>:<resource-id>

Attributes

None

Child Elements

A string

dc:subject

Keyword list, comma delimited

Attributes

None

Child Elements

A string

dc:title

Name of the resource

Attributes

None

Child Elements

A string

dcterms:created

Date when the resource was created (ISO Date yyyy-MM-dd'T'HH:mm:ss'Z' GMT Time Zone)

Attributes

None

Child Elements

A string

dcterms:extent

size (integer [space] units)

Attributes

None

Child Elements

A string

dcterms:modified

Date of last modification (ISO Date yyyy-MM-dd'T'HH:mm:ss'Z' GMT Time Zone)

Attributes

None

Child Elements

A string

dcterms:rightsHolder

Cmap owner. This element makes use of elements defined in the following namespaces:

Namespace	Prefix	Schema Location
http://www.w3.org/2001/vcard-rdf/3.0#	vcard	No schema definition found yet

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
vcard:FN	0	1
vcard:EMAIL	0	1
vcard:ORG	0	1

vcard:FN

Full name (First [Middle|Initial] Last)

Attributes

None

Child Elements

A string

vcard:EMAIL

Email address

Attributes

None

Child Elements

A string

vcard:ORG

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
vcard:OrgName	0	1

vcard:OrgName

Organization name.

Attributes

None

Child Elements

A string

content

Attributes

None

Child Elements

This element can either contain one **`res-meta-list`** element or a **`part-list`** element.

res-meta-list

Attributes

Name	XML Schema Type	Required	Description
count	unsignedInt	No	Total number of children currently in the list
total-count	unsignedInt	No	Total number of children in the source data set
start-index	unsignedInt	No	Index within the source data set of the first child in the list
end-index	unsignedInt	No	Index within the source data set of the last child in the list
sort-as	string	No	alpha if sorted alphabetically, or numeric if sorted numerically
sort-dir	string	No	Direction of the sort, either ascending or descending
sort-by	string	No	XPath expression identifying the element or attribute compared for sorting

Child Elements

Name	Min. Occurrences	Max. Occurrences
res-meta	0	Unbounded

part-list

Attributes

Name	XML Schema Type	Required	Description
count	unsignedInt	No	Total number of children currently in the list
total-count	unsignedInt	No	Total number of children in the source data set
start-index	unsignedInt	No	Index within the source data set of the first child in the list
end-index	unsignedInt	No	Index within the source data set of the last child in the list
sort-as	string	No	alpha if sorted alphabetically, or numeric if sorted numerically
sort-dir	string	No	Direction of the sort, either ascending or descending
sort-by	string	No	XPath expression identifying the element or attribute compared for sorting

Child Elements

Name	Min. Occurrences	Max. Occurrences
part	0	Unbounded

part

Defines the metadata of a Cmap or of any resource inside the CmapTools environment. This element makes use of elements defined in the following namespaces:

Namespace	Prefix	Schema Location
http://purl.org/dc/elements/1.1/	dc	**dc.xsd**

Attributes

Name	XML Schema Type	Required	Description
content-type	string	No	If the referenced resource is a container then this attribute will tell what type of elements it contains inside the **content** element. It can either be **res-meta-list** or **part-list**

Child Elements

Name	Min. Occurrences	Max. Occurrences
dc:format	0	1
dc:identifier	0	1
dc:relation	0	1

Cmap Web Service

The Cmap Web Service allows access to resources on a Cmap Server. It allows browsing the contents of the Cmap Server as well as saving and retrieving resources. The service definition is available as a WSDL 1.1 file at http://cmap.ihmc.us/xml/CmapWebService.wsdl.

Method Summary

Method Name	Input	Output	Description
createCmap	**res-meta**, **folder-url**, **cmap**, [**account-list**]	**res-meta**	Saves a new Cmap in the specified folder
createFolder	**res-meta**, **parent-folder-url**, [**account-list**]	**res-meta**	Creates a new folder inside the specified parent folder
createResource	**res-meta**, **folder-url**, **resource**,	**res-meta**	Saves a new resource in the specified folder

	[**account-list**]		
delete	**resource-url**, [**account-list**]		Deletes the specified resource
getCmap	**resource-url**, [**account-list**]	**cmap**	Returns the specified Cmap
getResource	**resource-url**, [**account-list**]	**resource**	Returns the specified resource
getResourceMeta	**resource-url**, [**account-list**]	**res-meta**	Returns the metadata of the specified resource
getResourceMetaList	**folder-url**, [**account-list**]	**res-meta-list**	Returns the metadata of the elements contained by the specified folder
getRootResourceMeta	[**account-list**]	**res-meta**	Returns the metadata of the root folder
getRootResourceMetaList	[**account-list**]	**res-meta-list**	Returns the metadata of the elements of the root folder
saveCmap	**resource-url**, **cmap**, [**account-list**]	**res-meta**	Saves an existing Cmap
saveResource	**resource-url**, **resource**, [**account-list**]	**res-meta**	Saves an existing resource
setResourceMeta	**res-meta**, [**account-list**]	**res-meta**	Modifies the metadata of a resource

account-list

This list is used for authentication on the server side, to grant access to resources in the context of the specified operation.

Attributes

None

Child Elements

Name	Min. Occurrences	Max. Occurrences
account	0	Unbounded

account

This element represents a credential to be used on the server for authentication.

Attributes

Name	XML Schema Type	Required	Description
user-id	string	Yes	The user-id to be used for authentication
password	string	Yes	The password corresponding to the user-id

Child Elements

None

resource

A Base64 string representing binary data. This element is defined to be of type base64Binary from the namespace **http://www.w3.org/2005/05/xmlmime**.

Attributes

Name	XML Schema Type	Required	Description
contentType	string	No	MIME type of the binary data

Child Elements

A base64Binary string

resource-url

URL for the resource. This URL can be the value of the **dc:identifier** or **dc:relation** inside the **res-meta** of the resource.

Attributes

None

Child Elements

A string

folder-url

URL for the folder. This URL can be the value of the **dc:identifier** or **dc:relation** inside the **res-meta** of the folder.

Attributes

None

Child Elements

A string

parent-folder-url

URL for the parent folder. This URL can be the value of the **dc:identifier** or **dc:relation** inside the **res-meta** of the parent folder.

Attributes

None

Child Elements

A string

Error Reporting

Errors are returned through the SOAP fault mechanism. The detail element in the SOAP Fault might contain an **error** element.

error

Attributes

Name	XML Schema Type	Required	Description
code	string	Yes	The code for the error

Values for the code attribute

Value	Description
ParseError	Error parsing the request
AccessDenied	The provided credentials don't allow you to perform the operation. You must provide different credentials.
ResourceAlreadyLocked	The resource is locked by someone else.
ResourceNotFound	The resource specified was not found
WriteFailed	Error writing
ReadFailed	Error reading
DeleteFailed	Could not delete the resource
IOFailed	Error during Input or Output operations
RenameFailed	Error renaming the resource
InvalidCredentials	The provided credentials are invalid
CopyFailed	Failed copying the resource
ConfigError	Configuration error
InternalError	Server internal error

Child Elements

Name	Min. Occurrences	Max. Occurrences
message	0	Unbounded

message

Attributes

Name	XML Schema Type	Required	Description
lang	language	No	The language for the message

Child Elements

A string

Layout Web Service

The Layout Web Service allows lying out Concept Maps. The Layout Web Service provides access to the latest and most advanced automated graphical layout algorithms, to enable users to better view, organize, and understand their Cmaps. The Layout Web Service encapsulates the GraphViz library (http://www.graphviz.org) of layout algorithms developed by AT&T. This library is open source, written in C and C++, and is actively developed.

The service definition for the Layout Web Service is available as a WSDL 1.1 file at http://cmap.ihmc.us/xml/LayoutWebService.wsdl.

Method Summary

Method Name	Input	Output	Description
layout	**cmap**	**cmap**	Lays out a Cmap using GraphViz

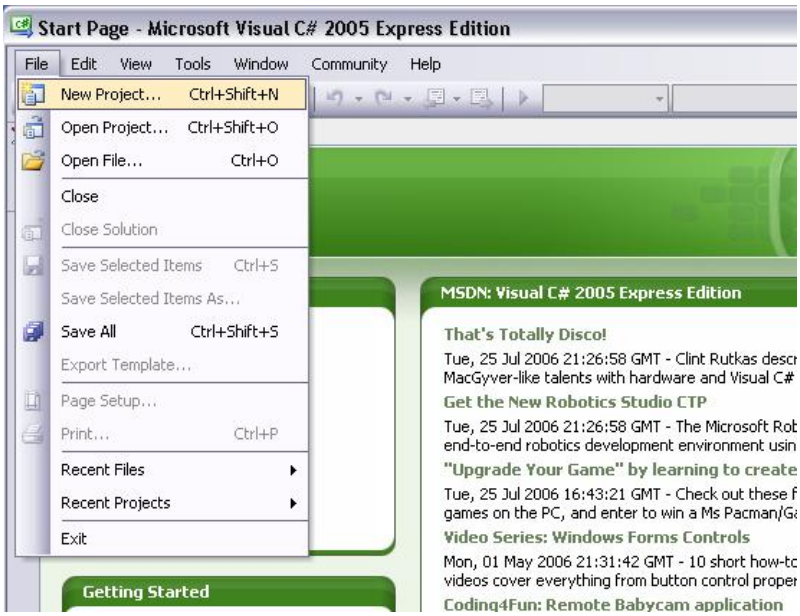
Client Example in Microsoft .NET from Scratch

This section demonstrates:

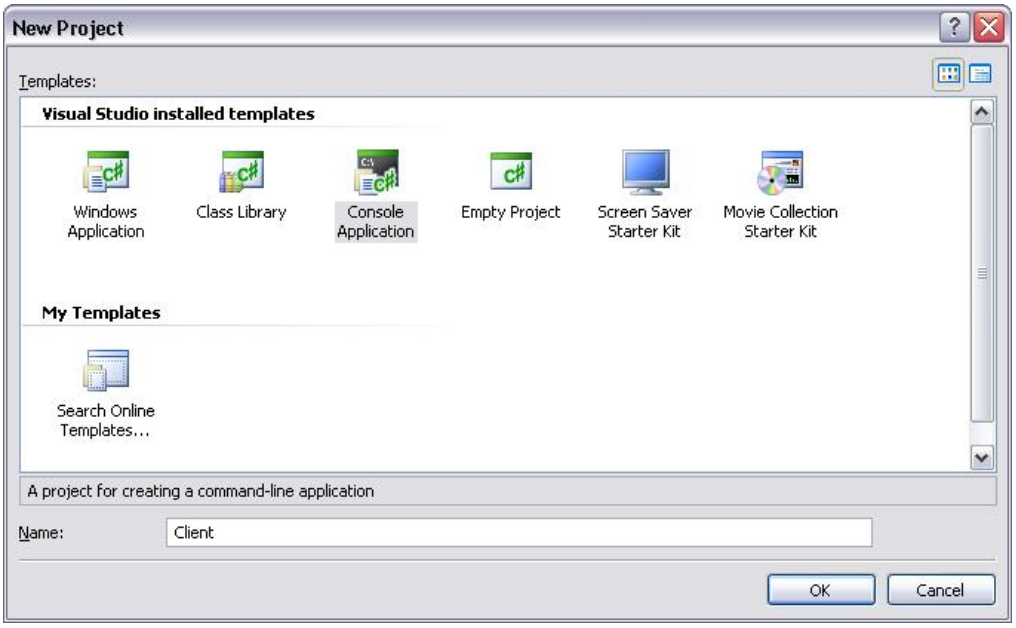
- How to generate a client stub in Microsoft Visual C# 2005 from the WSDL of the Cmap Web Service and Layout Web Service
- How to use the generated stub.

Create a project

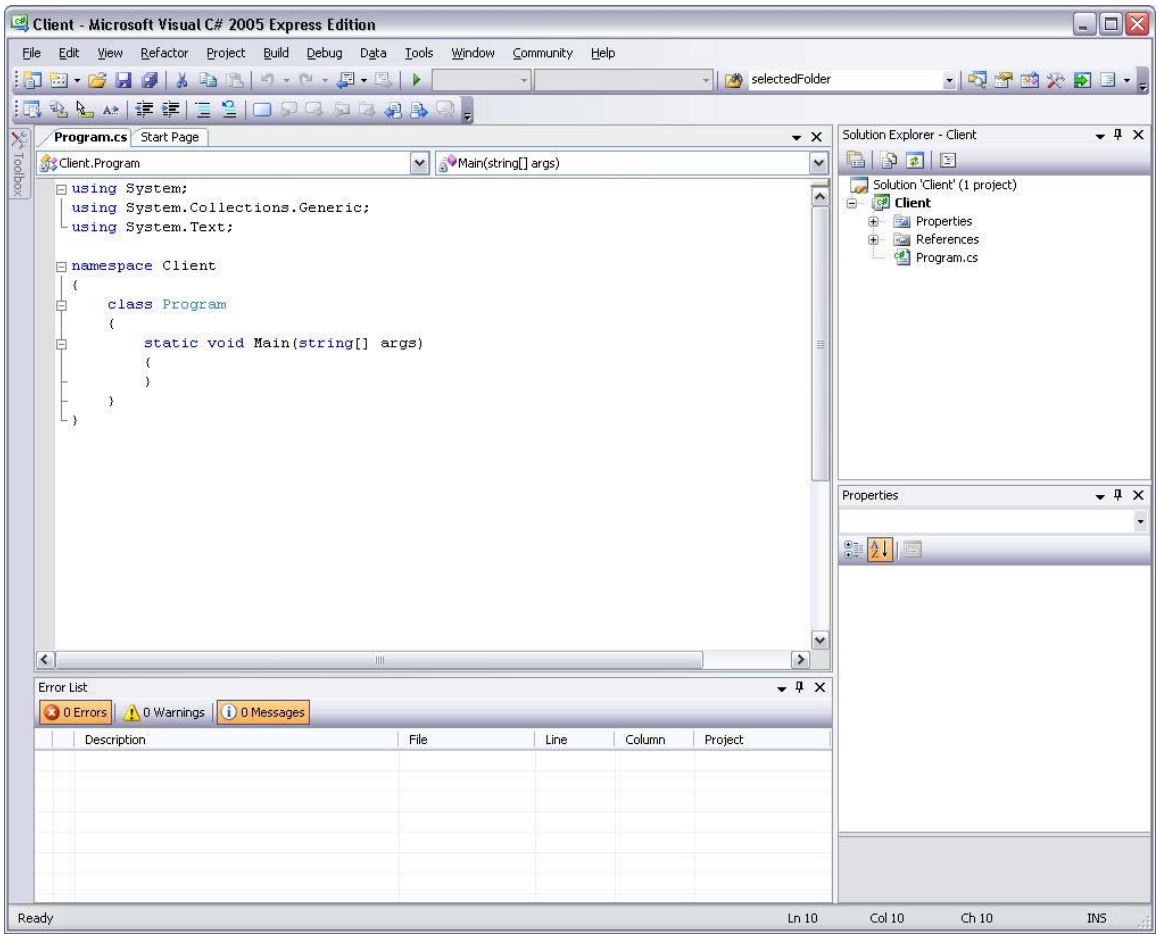
Open Microsoft Visual C# 2005 and in the File menu, select the option "New Project...":



In our example we will create a simple console application called "Client". Select the template "Console Application" and type "Client" for the name of the project in the textbox labeled "Name" and click the "OK" button:

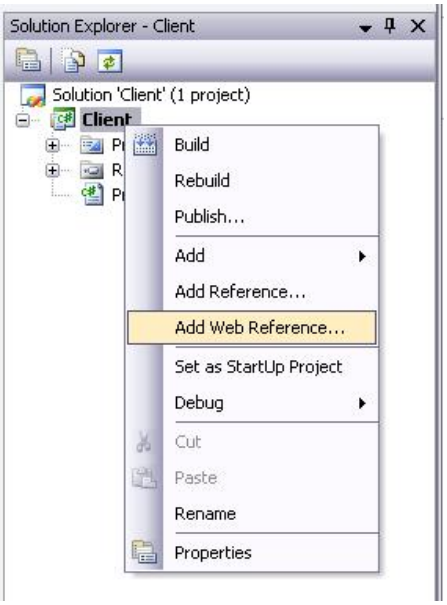


Your Microsoft Visual C# 2005 application will look as follows:

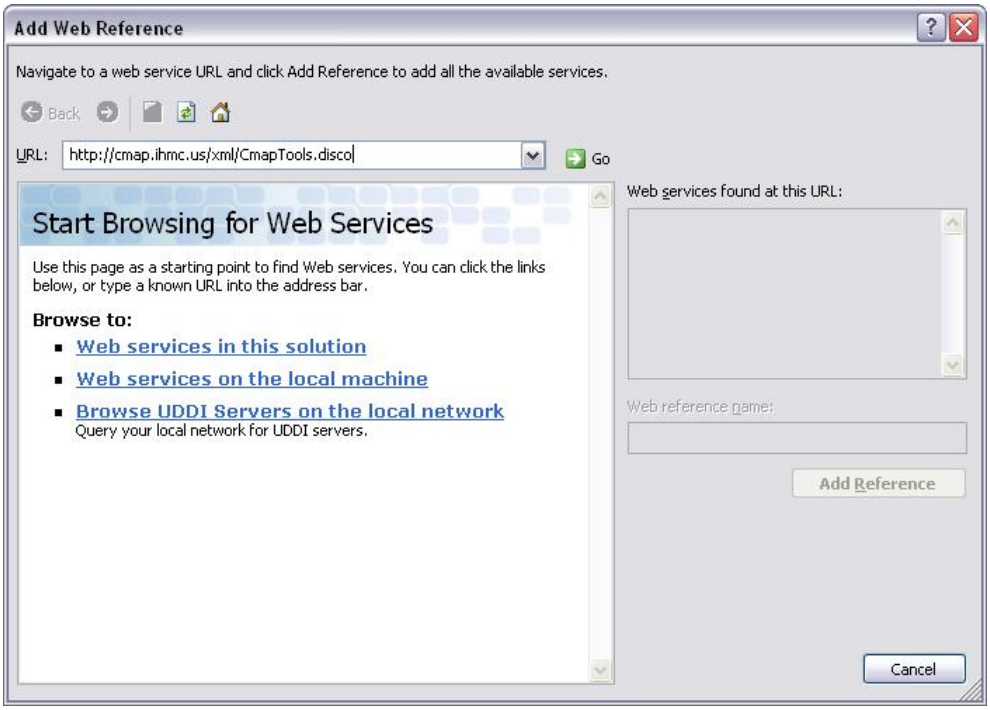


Generate the stub

In the Solution Explorer Frame, right click over the Client Project and select the option "Add Web Reference...":

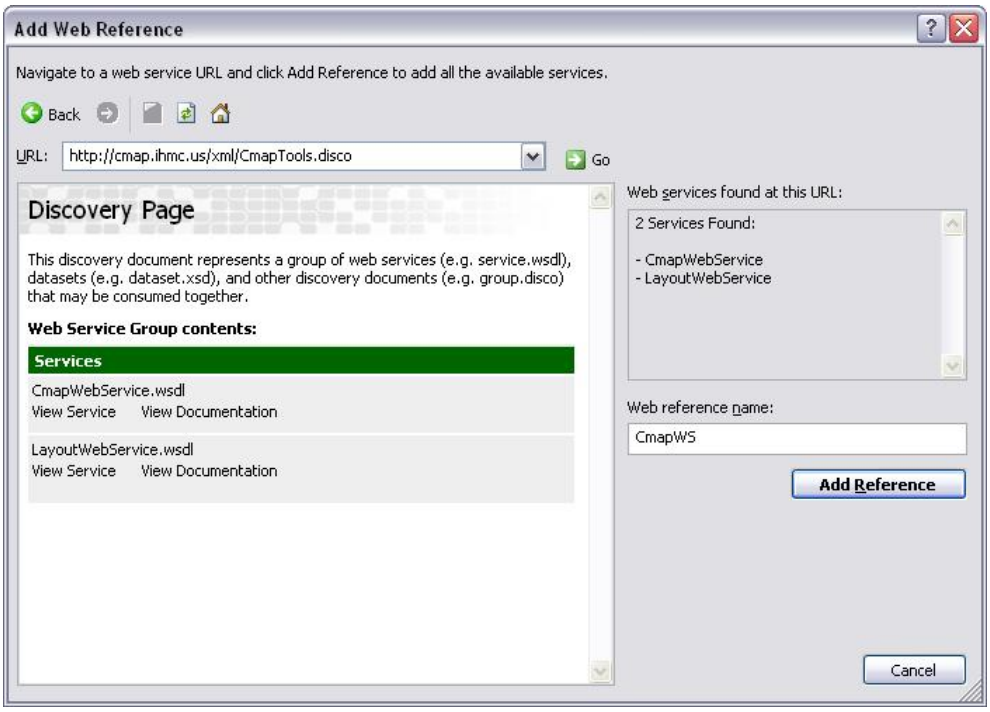


A window will pop up. Enter http://cmap.ihmc.us/xml/CmapTools.disco in the text field labeled "URL" and click "Go":

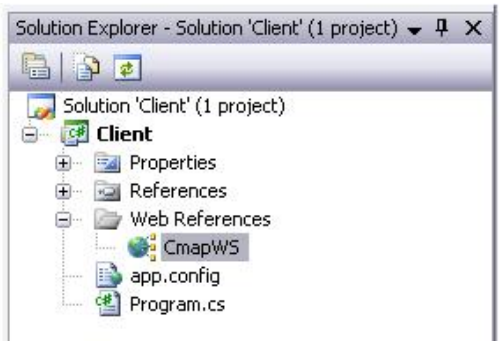


The list of services defined in the Disco File will appear. Next, in the textbox labeled "Web reference name", enter a label for the namespace that you want to assign to the generated

stub (in our example, we chose the label "CmapWS"). Then click the button "Add Reference":



In the Solution Explorer you should now see the reference to the Web Service:



Use the stub

In the Program.cs code window, add a using line to reference the stub namespace (in our case Client.CmapWS).

```
using Client.CmapWS;
```

In the main method of the same window declare and create an object of type CmapWebServiceBinding, which is the service stub for the Cmap Web Service, and an object of

type LayoutWebServiceBinding, which is the service stub for the Layout Web Service, and set the appropriate URL to each of the services (replace the <server-address> with the address of the server where the services are hosted):

```
CmapWebServiceBinding cmapWs = new CmapWebServiceBinding();
cmapWs.Url = "http://<server-address>/services/CmapWebService";

LayoutWebServiceBinding layoutWs = new LayoutWebServiceBinding();
cmapWs.Url = "http://<server-address>/services/LayoutWebService";
```

Create an account list for using as credentials when calling the methods in the Cmap Web Service (replace the <user-id> and <password> with an user-id and password of an user with write permissions over the root folder, because we are going to write back a Cmap in that folder):

```
account acc = new account();
acc.userid = "<user-id>";
acc.password = "<password>";

account[] accountList = new account[] {acc};
```

Browse for a Cmap

Get the list of resources in the root folder:

```
resmetalist rootResMetaList =
                    cmapWS.cmapWs.getRootResourceMetaList(accountList);
```

Iterate through the list to find a Cmap:

```
resmeta cmapResMeta = null;
foreach (resmeta rm in cmapResMeta.resmeta)
{
    if (rm.format.Text[0] == "x-cmap/x-cxl")
    {
        cmapResMeta = rm;
        break;
    }
}
```

Get the Cmap, Lay it out and Save it back to the server

Now if a Cmap was found, retrieve it from the server, then lay it out with the Layout Web Service, and finally save it back to the server:

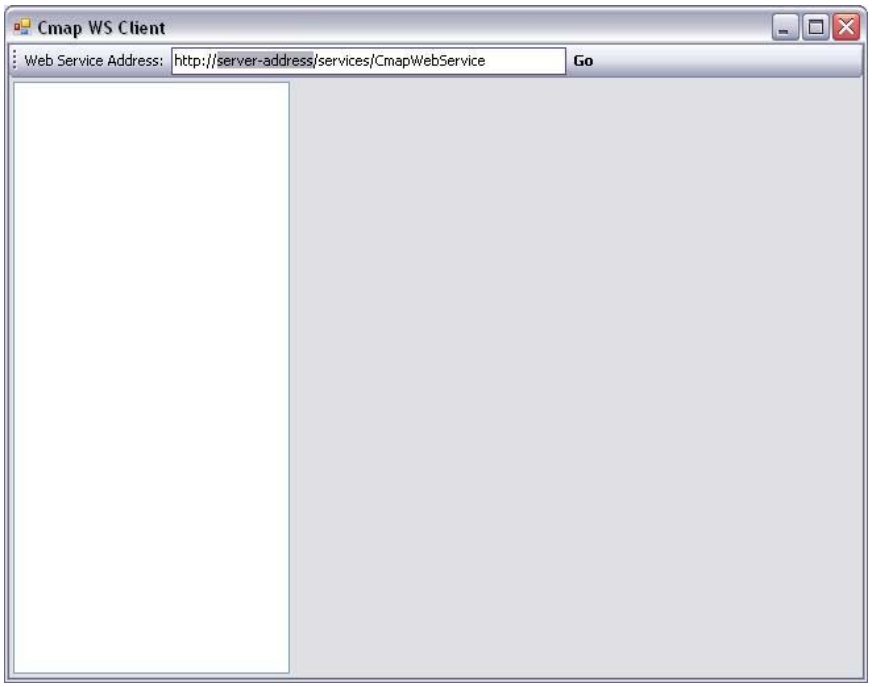
```
if (cmapResMeta != null)
{
      cmap map = cmapWs.getCmap (cmapResMeta.identifier.Text[0], accountList);
      layoutWs.layout(ref map);
      cmapWs.saveCmap(cmapResMeta.identifier.Text[0], map, accountList);
}
```

Tutorial for Client Example in Microsoft .Net

This section illustrates the use of the Client Example Application in Microsoft .NET 2.0, available in source code form from http://cmap.ihmc.us/xml/src/CmapWSClient.zip, and in binary form from http://cmap.ihmc.us/xml/bin/CmapWSClient.zip. You must have installed the Microsoft .Net Framework 2.0 in order to run the example.

Starting the application

For starting the application you need to double click the CmapWSClient.exe file. You will get the following window:



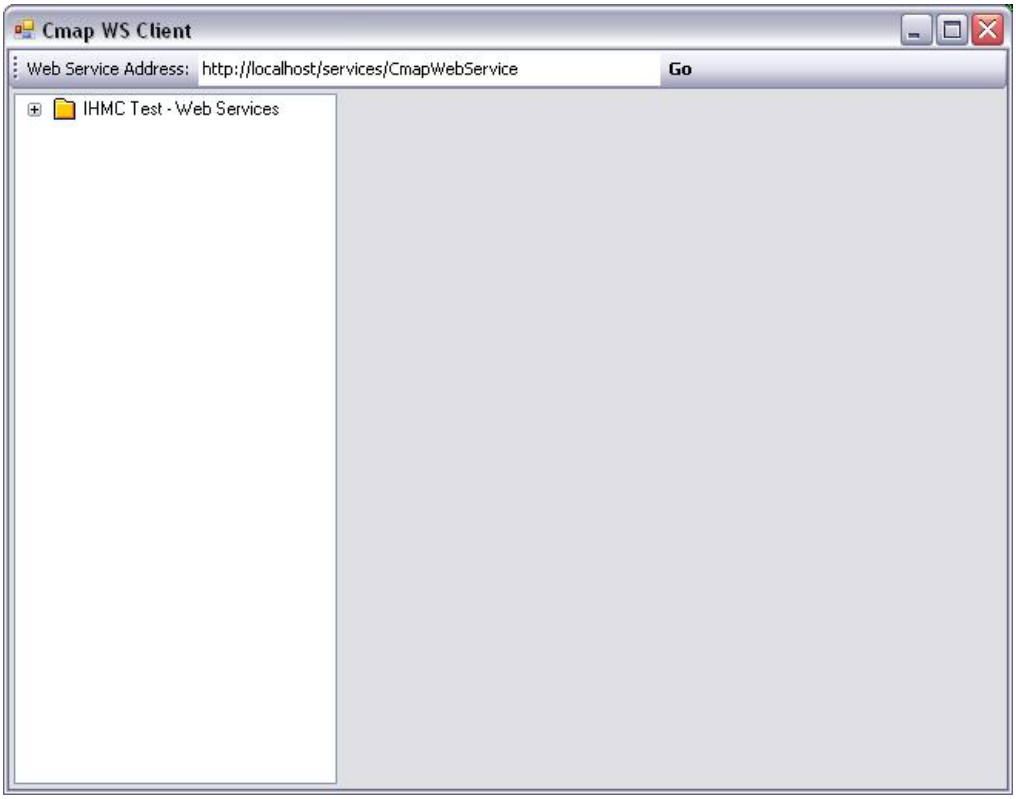
Contacting a Cmap Web Service

You have to replace the server-address text, in the text box labeled "Web Service Address" by the address of the server you want to access. In our case, we have a Cmap Server running on the local machine, so we are going to use localhost as the server address. This address can be

either an IP Address or a DNS name like cmapspublic.ihmc.us. If the server is not running on the default HTTP port (Port 80), you should specify the server port by appending a colon and the port number to the server address (cmapspublic.ihmc.us:8080).

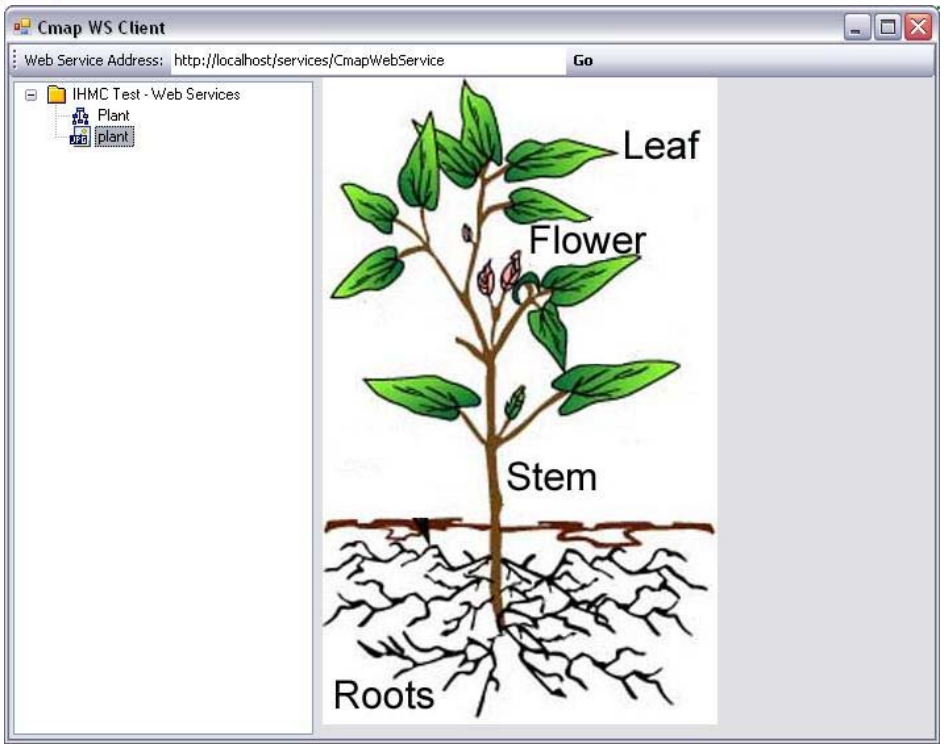
A Cmap Server might have several web services, and for targeting a specific web service, we have to append an URL Path to the server address. In our case the path services/CmapWebService identifies the Cmap Web Service, and the path services/LayoutWebService identifies the Layout Web Service.

After typing the server address click on the "Go" button. If the server you are targeting is running and accessible and it has the web service running, the window should look something like this:

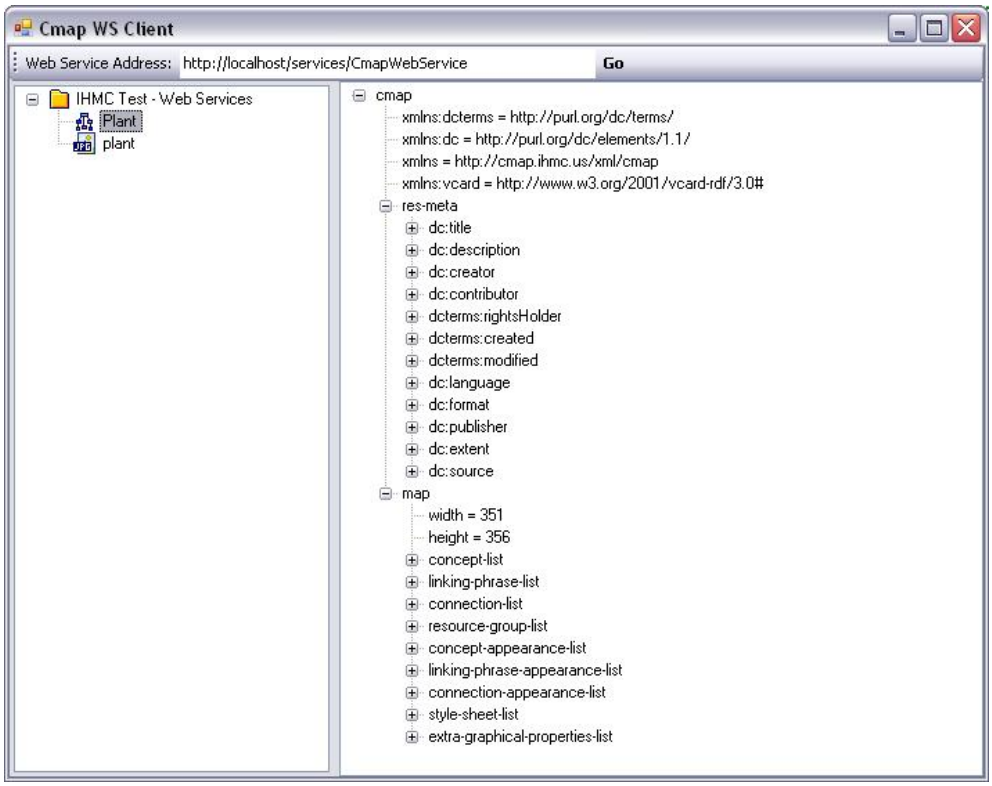


Browsing the Cmap Server

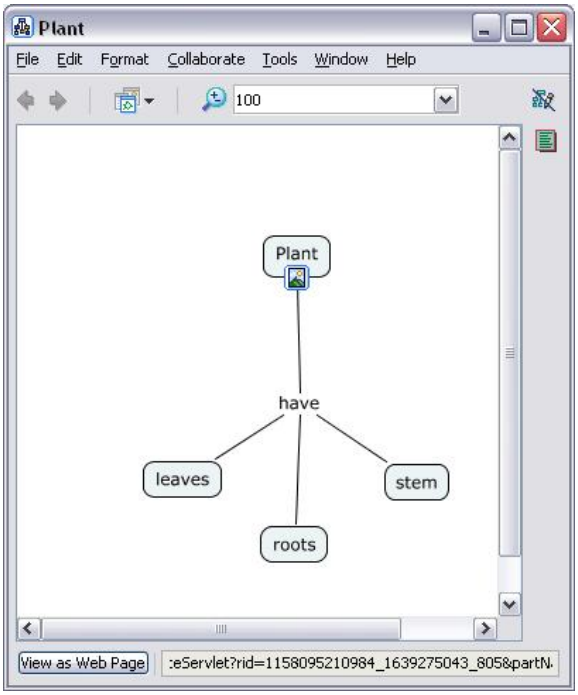
In the window just shown you must see a folder named "IHMC Test - Web Services". That is the name of our server, and it represents the root folder of the server we are accessing. If you click on the plus (+) symbol at the left of the folder icon you will get the list of resources in the root folder of the Cmap Server. In our case we only have two (2) resources: a Cmap named Plant, and a image with the same name. If you click in the image, the program will download the image and display it in the right side of the window, like this:



If you click in the Cmap, you will get the XML tree representing the Cmap in CXL:

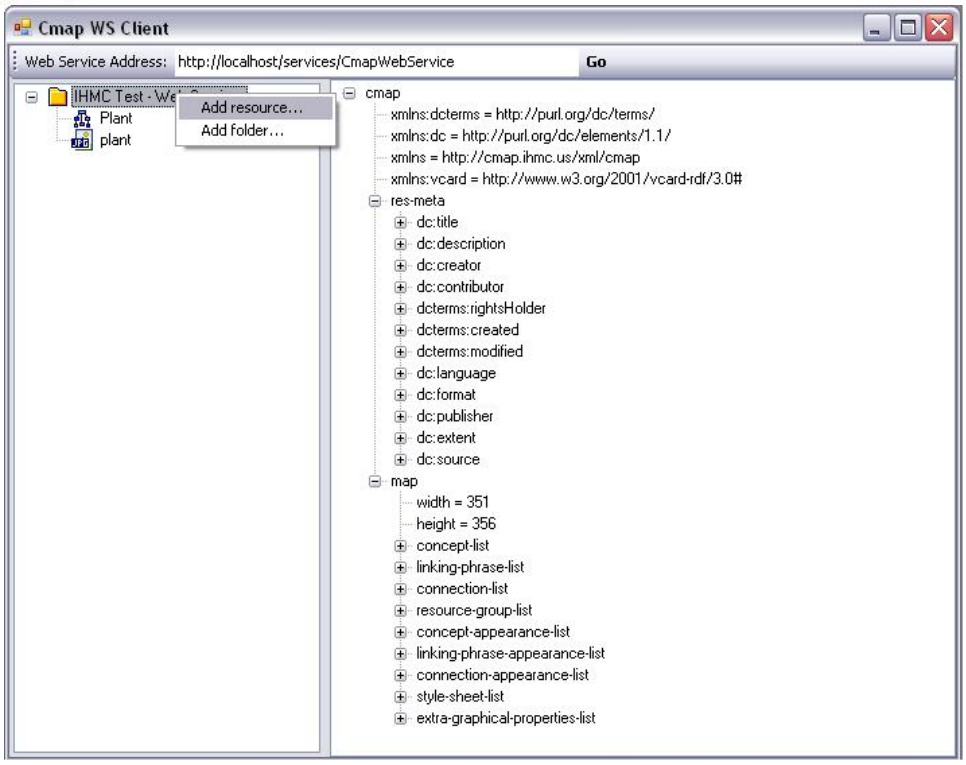


This same map accessed from CmapTools looks like this:

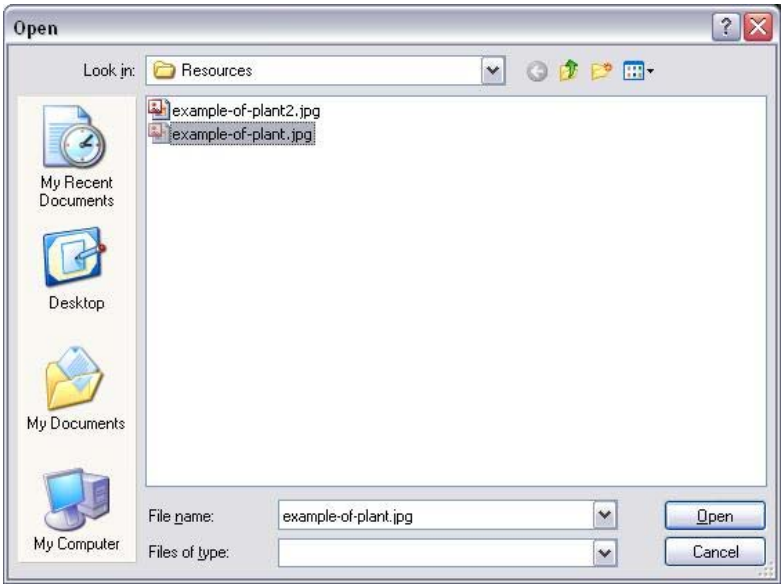


Adding a resource

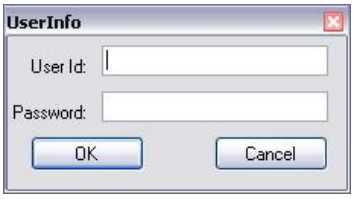
For adding a resource, you must right click in a folder and select the "Add Resource..." option in the pop-up menu that shows up:



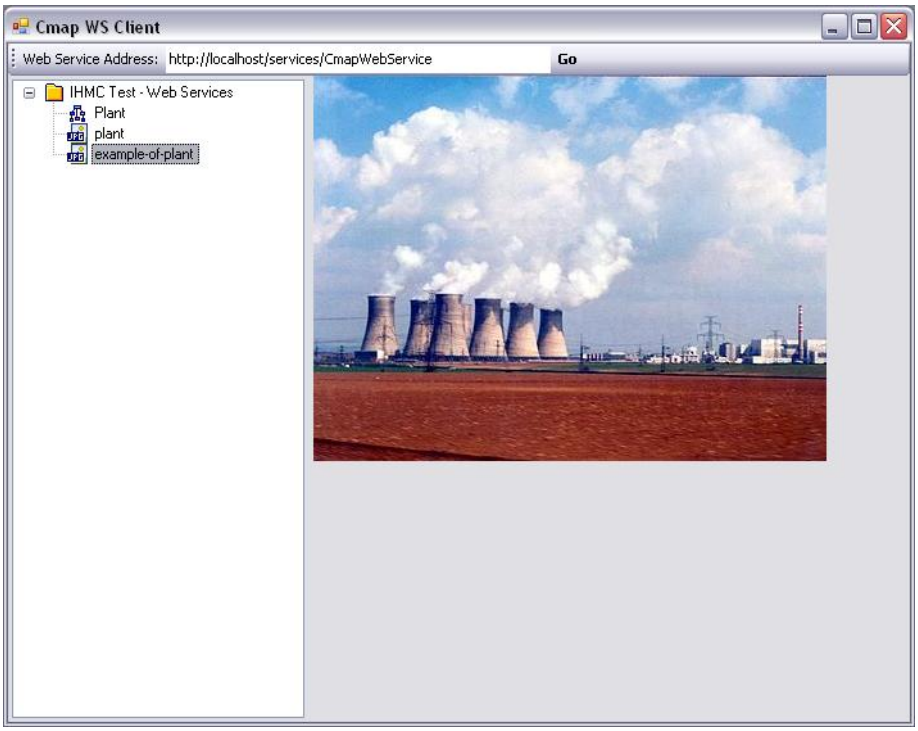
Then you have to browse and select the resource you want to add, in our case we are going to add the file plant-example.jpg:



If you need permissions to write into the selected folder, you will be prompted for an user id and password:

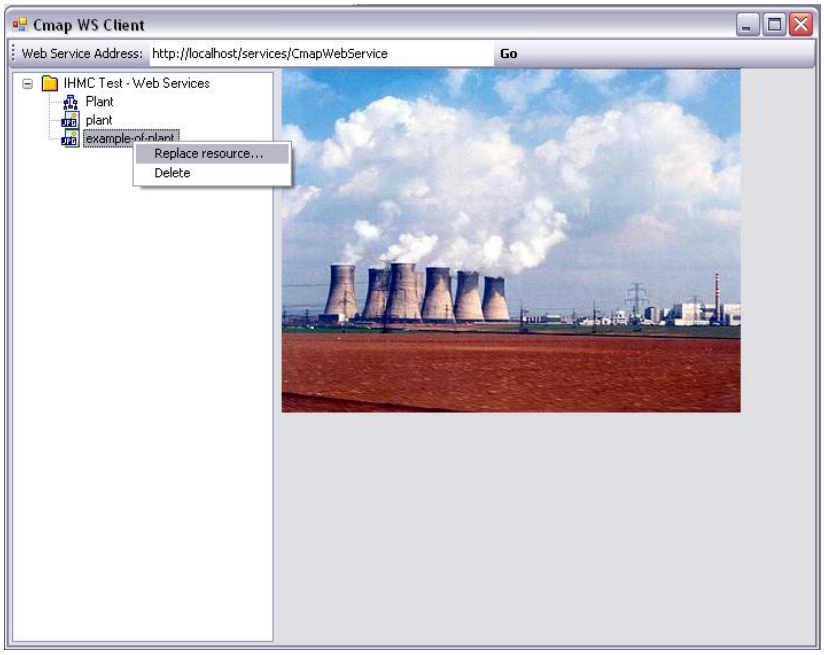


After adding the resource you should see the new resource added under the folder that you initially selected, and if the Resource is an image or a Cmap you will see the resource displayed in the right side of the window:

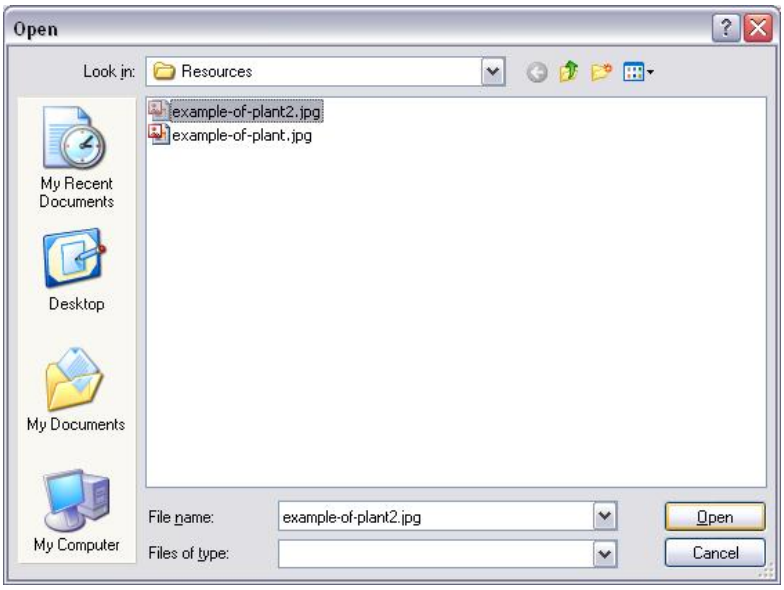


Replacing a resource

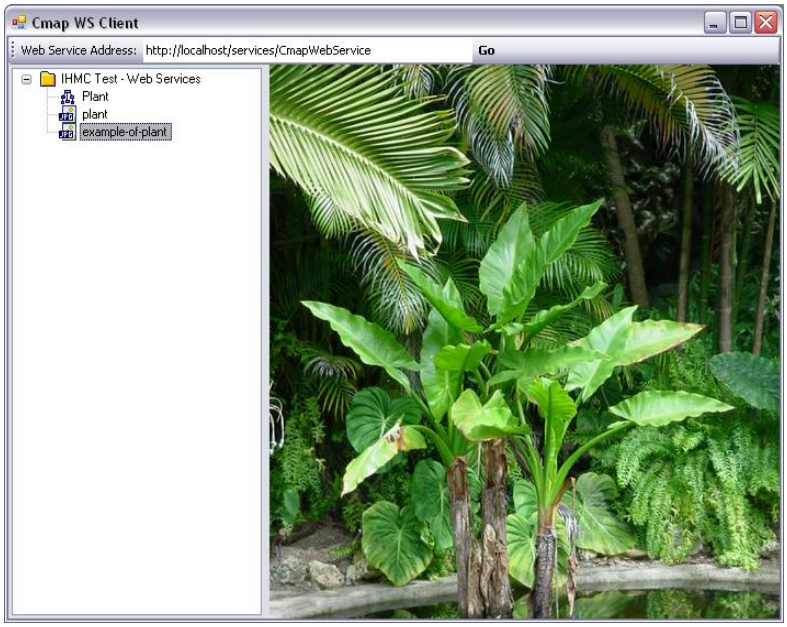
The example of plant image that we just added is not exactly the kind of plant we are referring to in our Cmap, so we need to replace it by right clicking in the plant-example.jpg resource, and then selecting the "Replace Resource..." option in the pop-up menu that shows up:



Then you have to browse and select the resource you want for replacing the old resource, in our case we are going to use the file plant-example2.jpg:

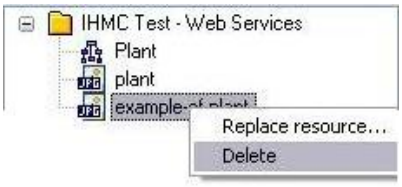


After replacing the resource, if the Resource is an image or a Cmap, you will see it displayed in the right side of the window:

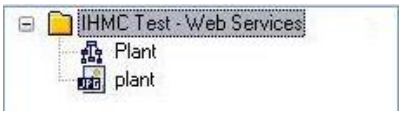


Deleting a resource

Now, we have decided to remove the resource just added by right clicking in the plant-example.jpg resource, and then selecting the "Delete" option in the pop-up menu that shows up:



After deleting the resource you will notice that it has been removed from the folder that contained it:

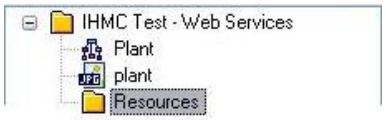


Adding a folder

For adding a folder you must right click in folder where you want to add the new folder and select the "Add Folder..." option in the pop-up menu that shows up:

Then, you will be prompted to enter a name for the folder:

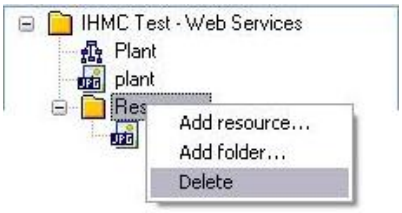
After selecting "OK" in the dialog, and assuming that the folder doesn't already exist and that you use valid characters for the folder name, you should see the new folder added to the folder that you initially selected:



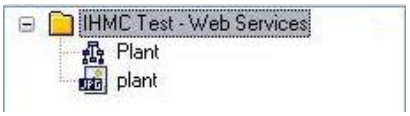
Now we can add resources to this folder:

Deleting a folder

To delete a folder you just need to right click over the folder you want to delete and select the "Delete" option in the pop-up menu that shows up:

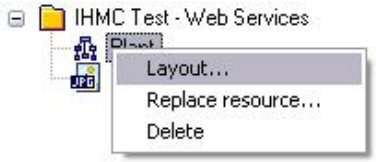


After deleting the folder you will notice that it has been removed from the folder that contained it:

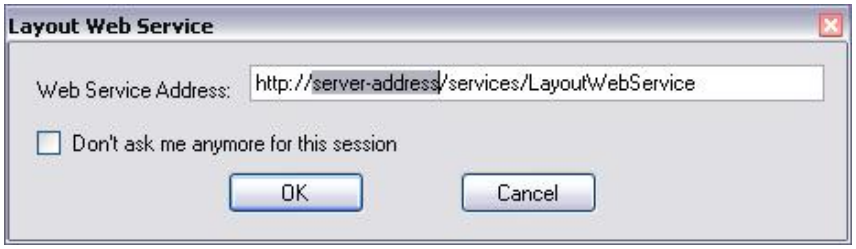


Laying out a Cmap

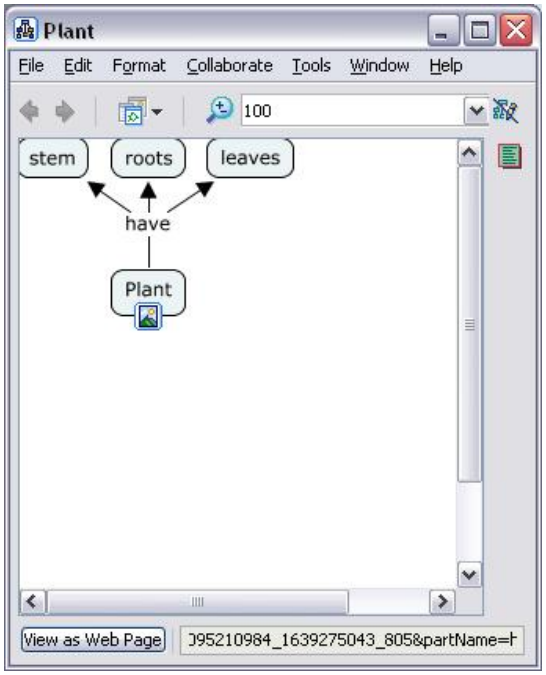
For laying out a Cmap, using the Layout Web Service, you must right click over the Cmap that you want to lay out and then select the "Layout..." option of the pop-up menu that shows up:



Then, you will be prompted for the Layout Web Service address:



The same rules for the Cmap Web Service address apply here, though you don't need to point to the same server that contains the Cmap Web Service, you could decide to use a different Layout Web Service in a different server. After entering the address, and assuming that Layout Web Service is available in the server you are targeting, the Cmap will be saved back to the server. If you use CmapTools to display the map, it might look like this:


Plant
File Edit Format Collaborate Tools Window Help
100
stem
roots
leaves
have
Plant
View as Web Page
J95210984_1639275043_805&partName=F