UBIGRAPH XMLRPC Quick Reference

BASIC API FUNCTIONS

```
void clear();
```
Delete all vertices and edges, reset styles.

```
int new_vertex();
```
Create a new vertex. Its vertex-id is returned.

```
int new_edge(int x, int y);
```
Create a new edge from vertex x to vertex y. Returns edge-id.

```
int remove_vertex(int x);
```
Delete a vertex. Any adjacent edges are removed.

```
int remove_edge(int e);
```
Delete an edge.

```
int new_vertex_w_id(int id);
int new_edge_w_id(int id, int x, int y);
```
Variants: create vertex or edge with a specified id. Returns 0 on success, -1 if requested id is in use.

VERTEX STYLE ATTRIBUTES

```
int set_vertex_attribute(int vertex,
        string attribute, string value);
```

colour/color	"#0000ff" (hex rgb triple)
shape	sphere, cone, cube, torus, dodecahedron, icosahedron, octahedron, tetrahedron, none
shapedetail	Only meaningful for sphere, cone, torus. Useful range 4-40. Default: 0 (auto-adjust).
label	String. Default none.
labelpos	Position of label relative to vertex. Default "[0,1.2,0]". Scaled by size.
size	Size of vertex. Default "1.0"
fontcolor	"#ffffff" (hex rgb triple)
fontfamily	Helvetica, Times Roman
fontsize	10, 12, 18, 24.
visible	If false, vertex not drawn.

EDGE STYLE ATTRIBUTES

```
int   ubigraph_set_edge_attribute(int vertex,
        string attribute, string value);
```

colour/color	"#0000ff" (hex rgb triple)
label, fontcolor, fontfamily, fontsize	See vertex style attributes.
spline	If "true", draw curved edge.
strength	How much the edge will pull its vertices together. Default "1.0". Use "0.0" for edges that do not affect layout.
oriented	If "true", the edge tries to point downward.
stroke	solid, dashed, dotted
width	Default "1.0"
arrow	Draw arrowheads. Default "false".
showstrain	Default "false". If true, draw long edges red and short edges blue.
visible	If false, edge not drawn.

STYLES

Style 0 is always the default vertex/edge style.

Vertex Styles

```
int new_vertex_style(int parent_style);
int new_vertex_style_w_id(int style,
      int parent_style);
```
Create a new vertex style based on the specified parent style.

```
int set_vertex_style_attribute(int style,
        string attribute, string value);
```
Set an attribute of a vertex style.

```
int change_vertex_style(int vertex, int style);
```
Set the style of a vertex.

Edge Styles

Replace "vertex" with "edge"' above.

LANGUAGE BINDINGS

Python

```
import xmlrpclib
server = xmlrpclib.Server(
  'http://localhost:20738/RPC2')
G = server.ubigraph
x = G.new_vertex()
y = G.new_vertex()
G.new_edge(x,y)
```

Ruby

See also rubigraph API in Ruby/rubigraph.

```
require 'xmlrpc/client'
server = XMLRPC::Client.new2(
   "http://localhost:20738/RPC2")
x = server.call("ubigraph.new_vertex")
y = server.call("ubigraph.new_vertex")
server.call("ubigraph.new_edge", x, y)
```

C

```
#include <UbigraphAPI.h>
int x = ubigraph_new_vertex();
int y = ubigraph_new_vertex();
ubigraph_new_edge(x,y);
```

Link: `-lubigraphclient`, plus xmlrpc-c and libwww libraries.

C++

```
extern "C" {
  #include <UbigraphAPI.h>
}
```

Java

```
import org.ubiety.ubigraph.UbigraphClient;
...
UbigraphClient G = new UbigraphClient();
int x = G.newVertex();
int y = G.newVertex();
G.newEdge(x,y);
```

Need Apache XML-RPC. Place ubigraph.jar in the classpath.