CNS 3670 Image Server Application Protocol

The image server is on **http://love.uvsc.edu** socket, 10000. Application dialog is as defined below.

The Application Protocol consists of a set of simple XML Documents (strings). Each XML Document ends with a null byte (i.e., An XML document is treated as a single string. A given XML Document String will always contain an XML element (tag) pair (e.g., <Error>error text</Error>) or a beginning element with an abbreviated ending (e.g., <Complete/>). Both types of tags can be parsed by XML DOM methods (.NET XMLDocument Class).

Typical application protocol exchanges are as follows:

Client	Direction	Server
Get Category List Command Exchange Follows		
Request Image DB Client Dialog Session	====> Session Request	Sends Client Dialog Session Number back to client.
Receives Session number	<==== Session Number	
Sent Category List Request	====> Category List Request	1. Receive list request 2. Reads list from database 3. Creates xml document 4. Sends category list xml document
Receives Category List and adds to list selection box	<==== Category List XML Document	
5. Send Request Complete 6. Close Connection	====> Complete	1. Closes out Client Dialog Socket 2. Go back to Accept on Listen Socket.
Get Image List Command Exchange Follows		
Request Image DB Client Dialog Session	====> Session Request	Sends Client Dialog Session Number back to client.
Receives Session number	<==== Session Number	

Client	Direction	Server
Send Image List Request	====> list request	1. Receive Image List Request 2. Retrieve List From Data Base 3. Creates an XML Document with beginning tag, list item tags, and list ending tag. 4. Sends Image List XML Document.
1. Clear list display 2. Receive XML Document 3. Parse XML Document 4. Display items	<==== Image List XML Document	
7. Send Request Complete 8. Close Connection	====> Complete	1. Closes out Client Dialog Socket 2. Go back to Accept on Listen Socket.
Get Image Command Exchange Follows		
Repeat Client Dialog Session Request above		
Send Get Image	====> Get(DB Index)	Receives request and looks up image
Create Image Display	<==== Begin Img(length)	Send Begin Image
1. Receive image bytes and place in a buffer. 2. When done, create a new image from the buffered data, and 3. display it in a PictureBox.	<==== binary byte stream	For size of Image: Send image bytes.
1. Send Request Complete 2. Close Connection	====> Complete	3. Closes out Client Dialog Socket 4. Go back to Accept on Listen Socket.
Miscellaneous Commands:		
Cancel request	====> Cancel	Resets any activity to it's initial state
terminates any display activity and starts over	<==== Reset	Reset
Fail	<==== Fail	Application Protocol Error on Client's messages

Client	Direction	Server
Receive Error, Display Error Report	<==== Error Report	Server Application Error while processing messages

Exact XML document command strings are as follows (Text in Bold must be sent exactly as specified, text in Italics is replaced with data from database.):

Message	XML Command String Syntax
Session Request	**<ReqSession>***student name***</ReqSession>**\0
Socket Number	**<Session host="***love.uvsc.edu***" number="***nnn***"/>**\0
Fail	**<Fail/>**\0
Category List Request	**<CategoryListRequest/>**\0
Category List XML Document	**<CategoryList>** **<Category name="***category name 1***" index="***nnn 1***"/>** **<Category name="***category name 2***" index="***nnn 2***"/>** ••• **<Category name="***category name n***" index="***nnn n***"/>** **</CategoryList>**\0
Image List Request	**<ImageListRequest category="***nnn***"/>**\0 nnn is the category index number
Image List XML Document	**<ImageList category="***nnn***">** **<Item name="***image name 1***" index="***nnn 1***"/>** **<Item name="***image name 2***" index="***nnn 2***"/>** ••• **<Item name="***image name n***" index="***nnn n***"/>** **</ImageList>**\0
Get Image	**<GetImage index="***nnn***"/>**\0
Begin Image	**<ImageBegin index="***nnn***" length="***nnn***"/>**\0
binary byte stream	A stream of bytes of the length specified in the ImageBegin tag above. Each byte containing unsigned binary data (0 to 255).
complete	**<Complete/>**\0
Cancel	**<Cancel/>**\0
Reset	**<Reset/>**\0
Error Report	**<Error>***error description***</Error>**\0 error description includes a copy of the command that was bad and the internal error message it generated.

Notes:

1. \0 as used above represents a null byte (Hex 00) command string terminator.
2. Image streams have no delimiter, end is determined by the length attribute in the

ImageBegin Tag.
3. Use standard ASCII Encoding for send/receive stream
4. All protocol tags follow standard XML format.
5. Case Matters, bolded items must be exactly as they are above.
6. Italics denote actual data content.
7. *student name* should be your given name and last name (e.g., Kirk Love)
8. *nnn* represents an integer of any length (will fit into a 32 bit signed integer data field).
9. An image name may be from 1 to 50 characters long.
10. The number of images in an Image List is undefined: may be from 0 to any number.

Image Data Base:

Categories Table		
Field Name	**Data Type**	**Comments**
CIndex	4 byte integer	primary key, auto increment, read only
Category	String	

Images Table		
Field Name	**Data Type**	**Comments**
IIndex	4 byte integer	primary key, auto increment, read only
CIndex	4 byte integer	foreign key to Categories Table
FileName	String	name of original image file
Description	String	
ImageDate	datetime	file date from original image file
Format	String	“jpg”, “gif”, etc.
Height	4 byte integer	
Width	4 byte integer	
Image	variable length	,compressed image BLOB