Helper app design

Alexey Badalov 2012-09-23

# Overview

The server keeps a list of active sessions and tasks active in each session. The app and the UI are clients. They communicate with the server by querying it for information, downloading resources, and uploading data. Since the server cannot initiate communication with a client, each client keeps up to date with the state of its task by polling the server at regular intervals.

URTaskControl.GetState returns update counts for files and folders. Whenever either of these changes, the server should increment its update count. When a client polls the server for the state, it compares the update count to the last seen. If the new count is greater, it knows there has been a change and reacts correspondingly.

# Resampled file upload

The UI client is capable of adding folders to upload files from and displaying upload progress. When the UI adds a folder, the server updates its state. The next time the app polls for state it changes, it recognizes that a folder has been added and downloads the updated list. The app then proceeds to upload image files from this folder to the server.

Once again, once a file is uploaded, the server updates its state. When the UI polls for changes, it recognizes the change and downloads the updated file list.

## Example

1. The server creates a task ID, consisting of a session ID and task number; it sets all update counts to “0”.
2. The UI adds folder “test”.
3. The server adds the new folder to the folder list and sets the folder update count to “1”.
4. When the app next polls the server, it sees that the update count has changed from “0” to “1”. It
5. The app then downloads the updated folder list and begins uploading files.
6. As each file is uploaded, the server increments the file update count.
7. When the UI next polls the server and sees that the file update count has increased, it downloads the updated file list and updates its progress indicator.

