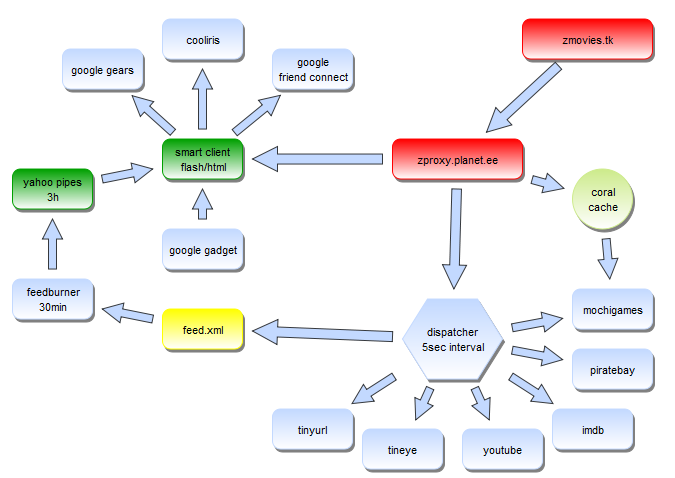
# Project: „MovieAgent“ codename „zmovies“

by Arvo Sulakatko 2009

Project is hosted at <http://code.google.com/p/zmovies/>

## Objective

The primary goal of the project is to provide a rss stream for recent popular torrents with their poster and movie data, including raiting, tagline and genres. The source for data is thepiratebay.org and imdb.com. The project has two parts – one for headless server and one for smart client for presentation. The smart client does not need to be connected to the server and could be hosted on any web server.



## Technology

The sourcecode is written in C#. The application can be ran as a consoleapplication, ASP.NET web application or PHP application. The latter is supported by jsc compiler (<http://jsc.sf.net>) which translates .net binary into php.

## Status

The server has been implemented and is able to output a media stream including the movie poster, raiting, trailer, raiting, genre and the torrent link.

Running on apache: <http://localhost/jsc/zmovies/?Task6_MediaCollector?html>

Running on ASP.NET: <http://localhost:60177/?Task6_MediaCollector?html>

The actual URLs may depend on your server configuration.

The smart client has not yet been finalized and is waiting to be implemented.

Download:

<http://zmovies.googlecode.com/files/20090327_MovieAgentDemo1.zip>

## Dispatcher

The server application is made of tasks. Each task can run total of 5 seconds. This limit is imposed by PHP hosting services. A task has to have work items to be invoked. A task has an interval on which to be invoked. A task can depend on other tasks, just to make it wait for others to complete their work items.

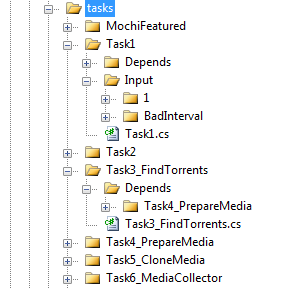


Image Tasks shown in design mode

Visualizing workflow is very important, because the tasks run as a long proccess and you as the administrator want to be informed what is being done. The following picture shows information for one of the tasks. It is grayed out, because it is blocked. It has a log window to display recent decisions. It has a counter to tell how many times has it been invoked. It has a memory indicator to tell how many duplications if any can be detected – a task may want to do the same thing more than once. And finally it displays a list of items it is working on.

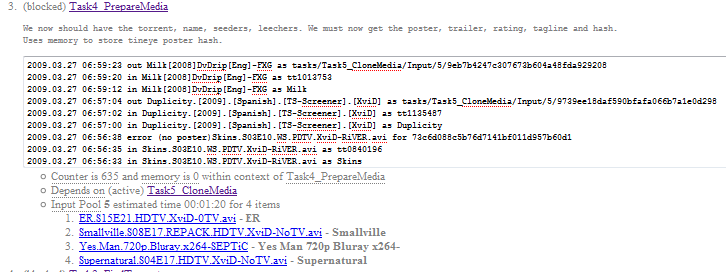
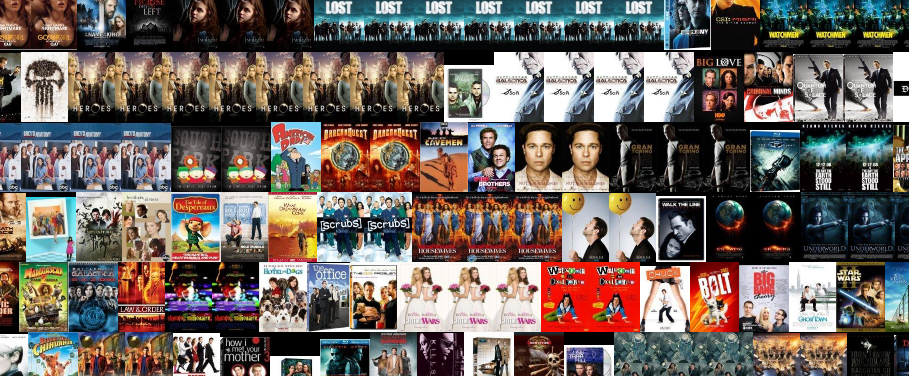


Image Tasks shown at runtime via overview html view

## Result

Once tasks are completed the server application is ready to generate the output rss feed.



The images are not hosted on the server. Bayimg.com free service is used for that. The output could only include latest 8 hours and forget details about older items. Google provides a way to cache the stream, so that the smart client can query historical entries aswell.

Image Visualized output

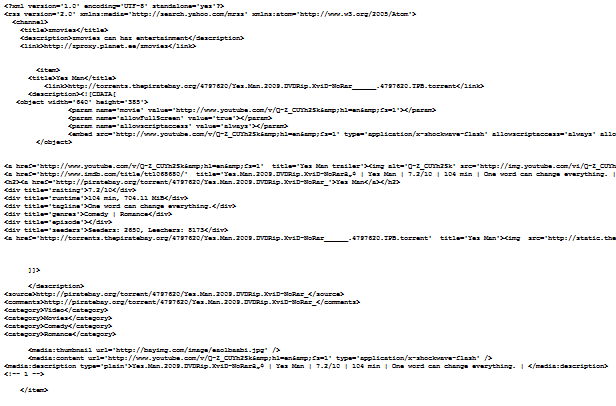


Image Snippet of the media rss

Feed readers may have different way to display the output. Our output feed should be supported in google reader, our smart client, cooliris (<http://cooliris.com>), safari feed reader and opera feed reader.



Image Opera feed reader showing the output

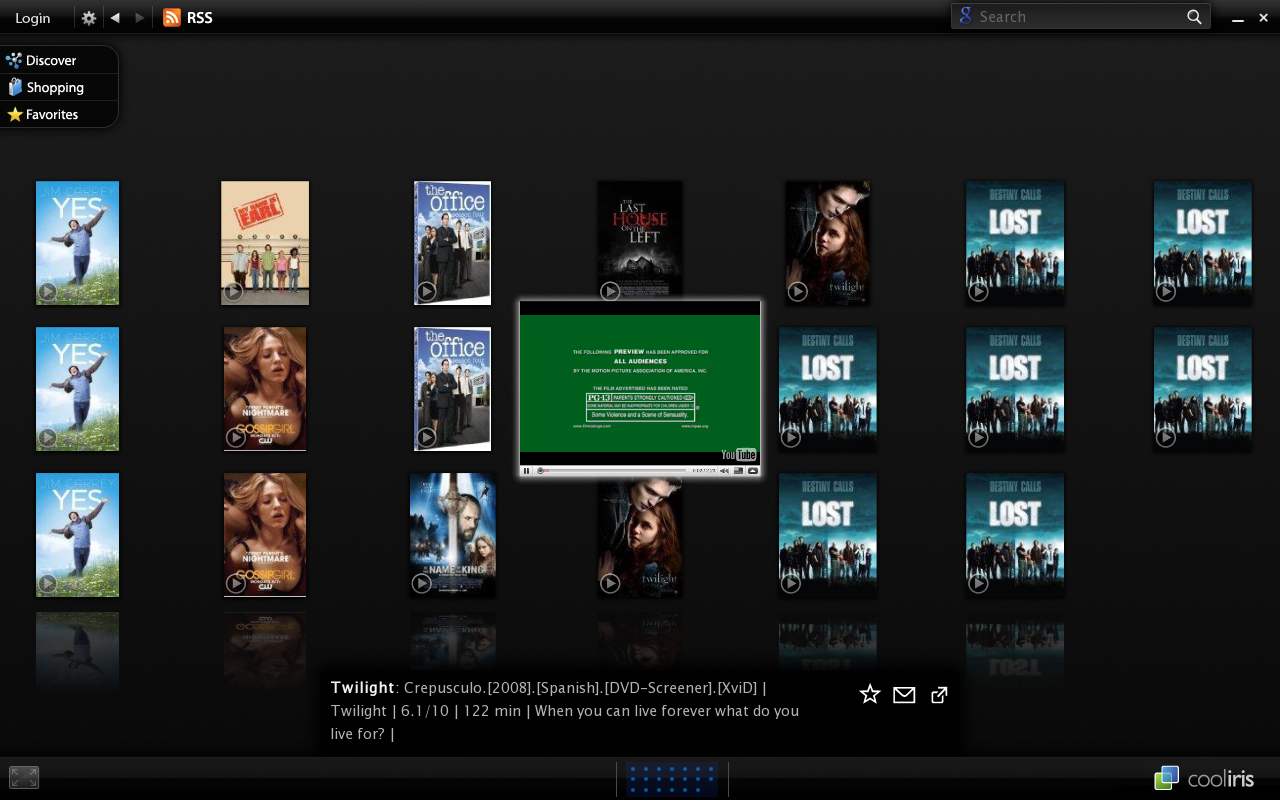


Image Cooliris view

## Smart client

The client side code is packaged in flash object. The source code was also written in C# and then the jsc compiler was used to recompile it to actionscript. The smart client can be described as a Google Gadget. The smart client will query rss items via google feed api. The posters can be clicked to display the movie trailer along with the additional details the item has.



Image HTML source code including rss feed and flash object

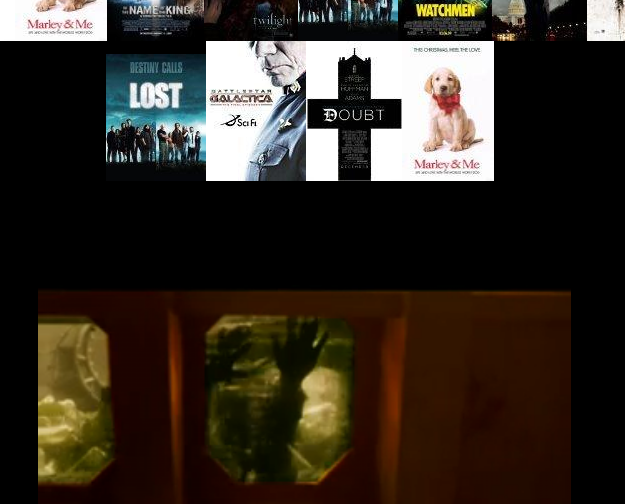


Image Smart Client where posters are above and a running trailer is below