

Internship Report

Alaska Multiple Listing Service, Inc. (AK MLS)
by Max Konovalov

CS 395
Fall 2005 - Spring 2006

University of Alaska Anchorage

Overview of Alaska Multiple Listing Service, Inc.



Alaska Multiple Listing Service, Inc, commonly known as AK MLS, has been serving the real estate industry in Anchorage since 1956 and statewide since 1996. The service was initially created by several local real estate brokers to provide a way to cooperate on sales of listed properties under a standard set of rules.

The Anchorage Board of REALTORS purchased MLS on October 1, 1969 from the former owners. Nine individuals, being members of that Board, loaned the money for the purchase of the company (\$20,000) and this loan was repaid during the first year of business including 6% interest. The service then functioned as REALTOR MLS with all members being required to be members of the Anchorage Board of Realtors, until June 1976, when due to various reasons, the members elected to become an independent entity called MLS, Inc. In 2002, the members agreed to change the corporate name to Alaska Multiple Listing Service, Inc., and to redesign the corporate logo to better represent the customers and expanded service area of the association; as well as to enable needed trademark protection. In 2003, a decision was made to update the public website, <http://www.alaskarealestate.com>, and this also entailed a new logo to correspond with the aesthetics of this new website. The association was originally intended to operate as a non-profit and was accepted as such when the State of Alaska incorporated it in 1969. However, according to the Federal tax laws, multiple listing services are not considered non-profit, therefore, Alaska MLS operates as a taxed not-for-profit association. Alaska MLS experienced rapid member and financial growth until the mid-1980's when it experienced tremendous losses due to a statewide economic downturn. Thanks to the efforts of dedicated Boards of Directors, and the support of the membership, Alaska MLS recovered as a more efficient association. By the end of the 1990's, the association was again in a challenging state of flux. As a result of rapidly changing member needs, technology, and economics, the Alaska MLS information database and all auxiliary internal systems were completely modernized. The Alaska MLS database is now compatible with not only the present WyldFyre desktop software, but also with the latest Internet browser-based systems, palm units, and third-party software vendors.

Overview of my projects

The biggest project I was involved in during my Internship with Alaska MLS was conversion from WyldFyre to FlexMLS database. The initial stage of the project took about two months. The main idea of the conversion was to map fields in one database to another by using “flat” tables and to find the fastest algorithm to process data from FlexMLS database. Due to my lack of experience in VB/VBA (primary programming languages at Alaska MLS) and 20 hours per week limitation I could not fully participate in this process. At the beginning of the conversion process I carried out mostly supplementary tasks. One of the first of programs I wrote in VBA for MS Access 2000 was database table scanner (see Fig. 1). The program allows traversing every table in a given database and view/change table fields properties (primary key(s), indexes, null, unique, etc.). The program allows quick and easy navigation through database which is very crucial when dealing with a great number of tables.

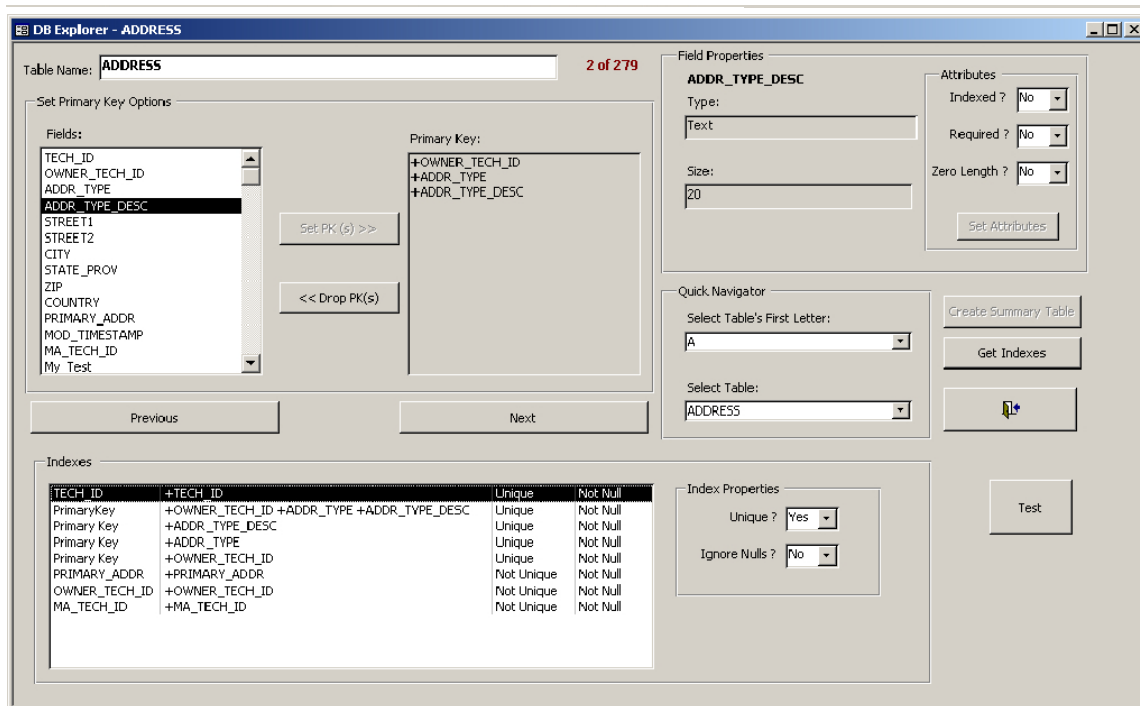


Figure 1. DB tables fields' properties scanner

Another program I wrote within the framework of the project addressed an issue in the differences between garage data representation in FlexMLS and WyldFyre. WyldFyre database distinguished between garage and carport

fields, meanwhile FlexMLS stores data only about carports, which made data mapping and representation extremely difficult. The program I wrote in VB using MS ADO searches currently active listings and displays garage/carport information if available. If the data is incorrect, operator can change the values which will be stored in a separate table for future references. Then, comparing old and new values (as well as modification dates) allows operator to make sure if the data was correctly converted from FlexMLS. In case of any mistakes, the values can be set manually (see Fig. 2). The program allows viewing all the stored data in a separate window and exporting the data into CSV file.

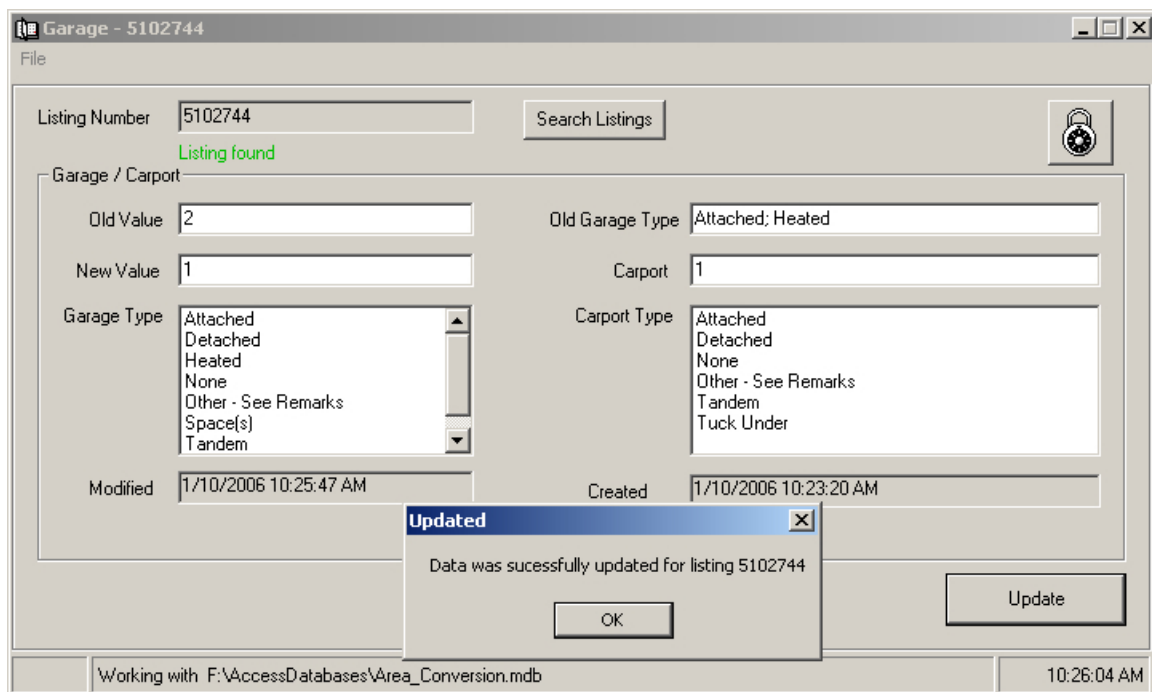


Fig. 2. Garage/Carport listing update

Also I was involved into testing and code review process. As a part of the testing process, I wrote a program in VBA to catch any errors during execution and store the error description as well as other information (time, function that triggered the error, computer, etc.) in a specified table. That allowed executing the rest of the functions written in VBA and organized into MS Access macros without crashing. The data stored in the errors log table can be later analyzed for elimination of the errors.

On the final stage of the data conversion process I was involved into code review. As a result of this, some of the functions were rewritten in order to use SQL instead of recordset operations which increased the runtime of the program. Some of the functions that require recordset operations were combined and rewritten in such a way that they process all the required field operations at a single run without closing and reopening recordset, which also optimized the performance.

Another stage of the database conversion was picture conversions. Originally, all the listings' pictures were stored on FlexMLS server located out of the state. This caused delays in displaying the existing listings. Also due to the established synchronization schedule, the newly added pictures became available only on the next day. As being a part of this project, I wrote a Picture Grabber program which downloads ("grabs") pictures from the specified URL. The program also checks the input file type for correctness (it must be either JPEG or BMP). To make a good usage of the program, I wrote a VBA edition of it for MS Access 2000, which is used as a front end DB interface in the company. However, to make it run on the MS Access 97, which is due to some technical and organizational issues is the main engine for the picture conversion, I had to recreate the program and make a DLL out of it. This DLL is currently used as a part of downloading newly added pictures of the active listings to the local server and store its information in the database. The VB6 project and the source code of the picture grabber DLL is available from <http://www.math.uaa.alaska.edu/~konovalm/cs395.htm>.

Also I was researching possibility of creation of a program or a DLL to generate PDF documents. For example, this program could be used on the website in the ASP environment to generate customer's reports, certificates, etc. Unfortunately, I did not have time to finish the project, since the priorities were changed and I was assigned another project. However, what I was able to achieve is a simple PDF text writer, which creates a PDF file with some specified text effects (50% gray, outlined, filled) as well as font size. Sample program with input is shown on figure 3, and its output is shown on figure 4. I think this is a very interesting project and I am pretty sure I will return to it.

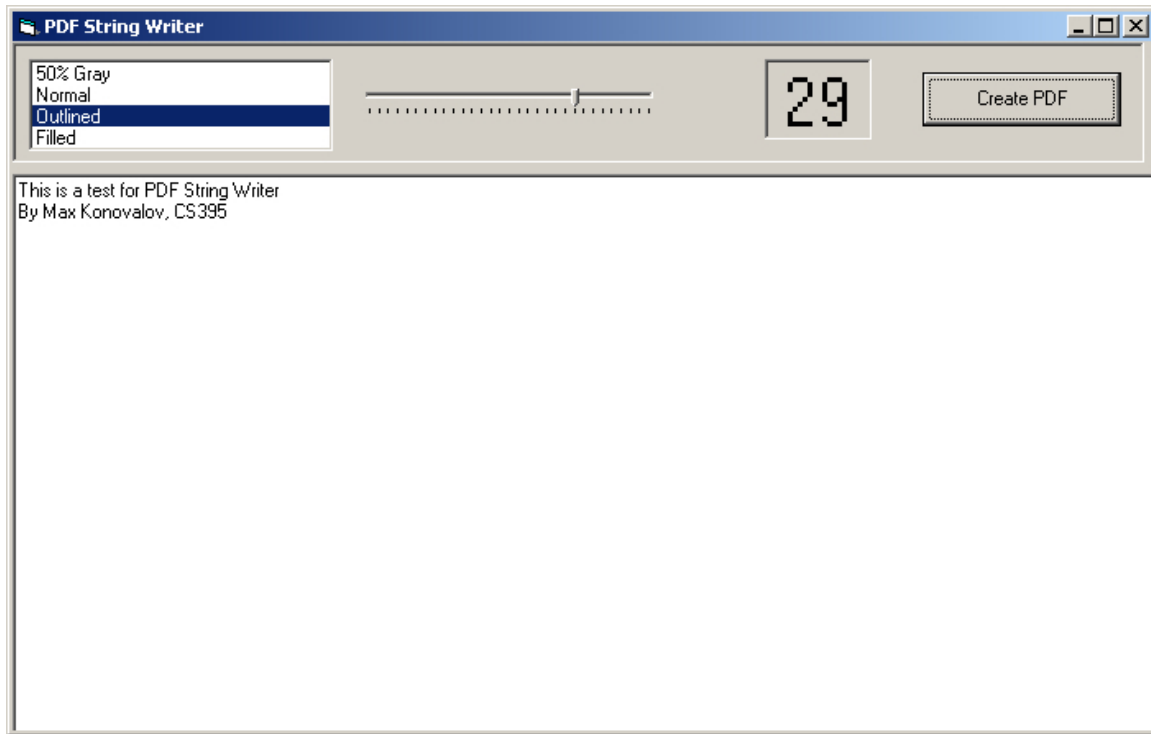


Fig. 3. PDF Writer program with a sample input.

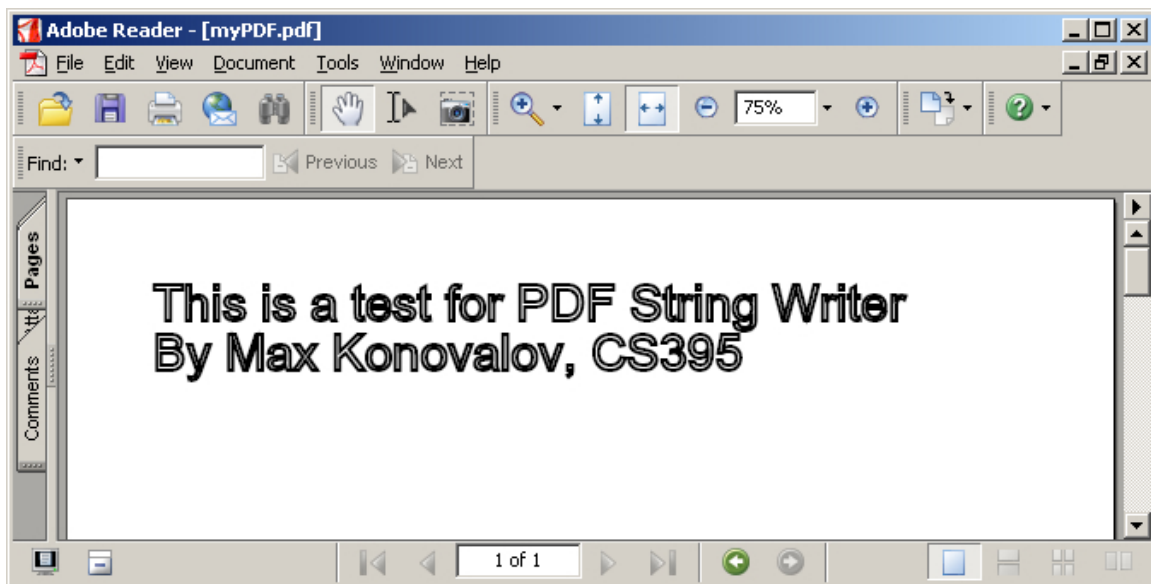


Fig. 4. Output of the PDF writer program.

Another big project I was working on was Thumbnails Generator. Originally, the program was designed to resize all JPEG files in a current folder keeping aspect ratio, and output the thumbnails of the pictures to another folder. Eventually, this project grew up into developing a fully functional program that can be used outside the company. During my work on the project I created

three versions (editions) of the program. The first one was a simple prototype of the second and the third editions. It consisted of a single executable file and could resize only a single folder without subfolders. Also it contained no error checking for input. The second edition of Thumbnails Generator (see fig. 5) was able to process all images in a folder specified as well as in all the subfolders if desired. Also it allowed specifying three different image sizes: large, medium, and thumbnail, assigning prefixes to the output files, specifying padding color of a blank area after resizing, and accepting command line arguments, which was very useful for using the program as a part of a batch job. Working on the second edition was very interesting for me because I was developing custom user control in VB6. This custom user control (see fig. 6) was used in setting image processing options for three different image sizes. The screenshot of the Options setting is shown on figure 7.

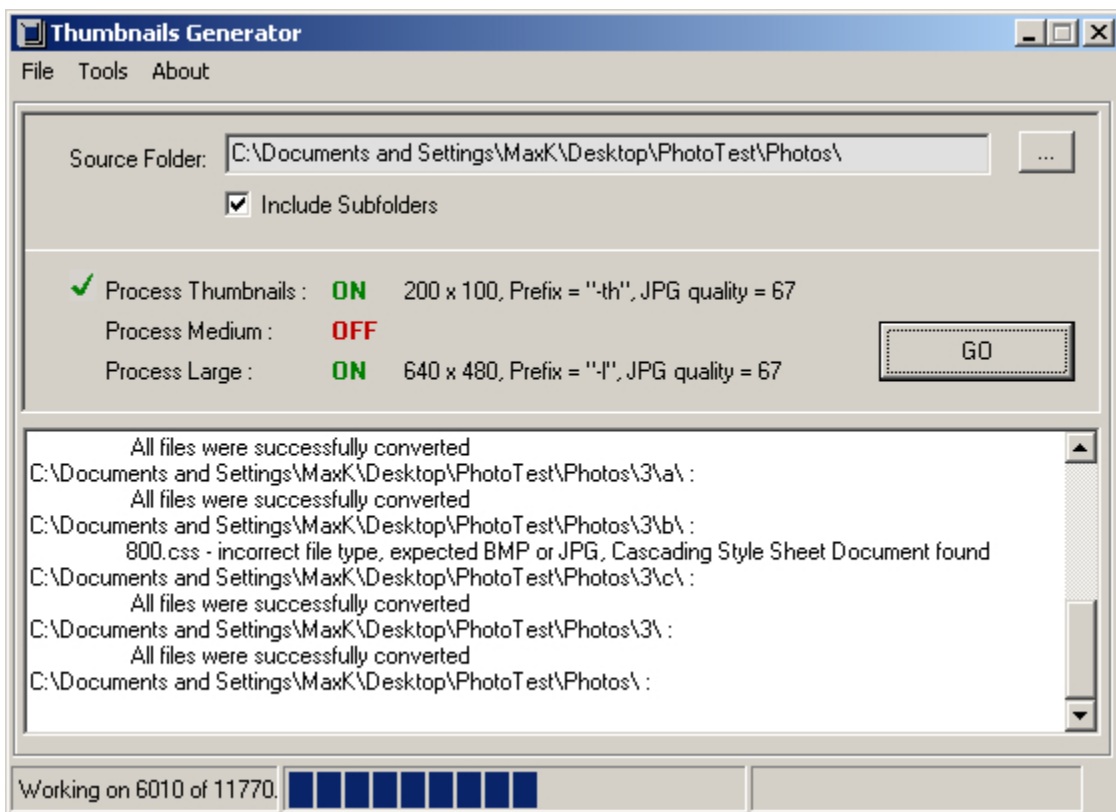


Fig.5. Thumbnails Generator 2 main window

The program also tests an input file for a possible error. First, it checks if the file extension is JPG, JPEG, or BMP. If not, then an error is displayed and program starts working on next file. If the file has one of the allowed extensions, then the file is being checked if the extension and the file format

match. If the file which declared as BMP or JPEG is really a bmp or jpeg file, then it is processed, otherwise, an error is displayed and the program moves to the next file. Upon completion, the program displays statistics on how many files were created, number of errors, type of errors, and time of completion (see fig. 8). The program was tested on a big number of files (up to 15,000 files) with some intentionally planted errors. All the tests were passed successfully.

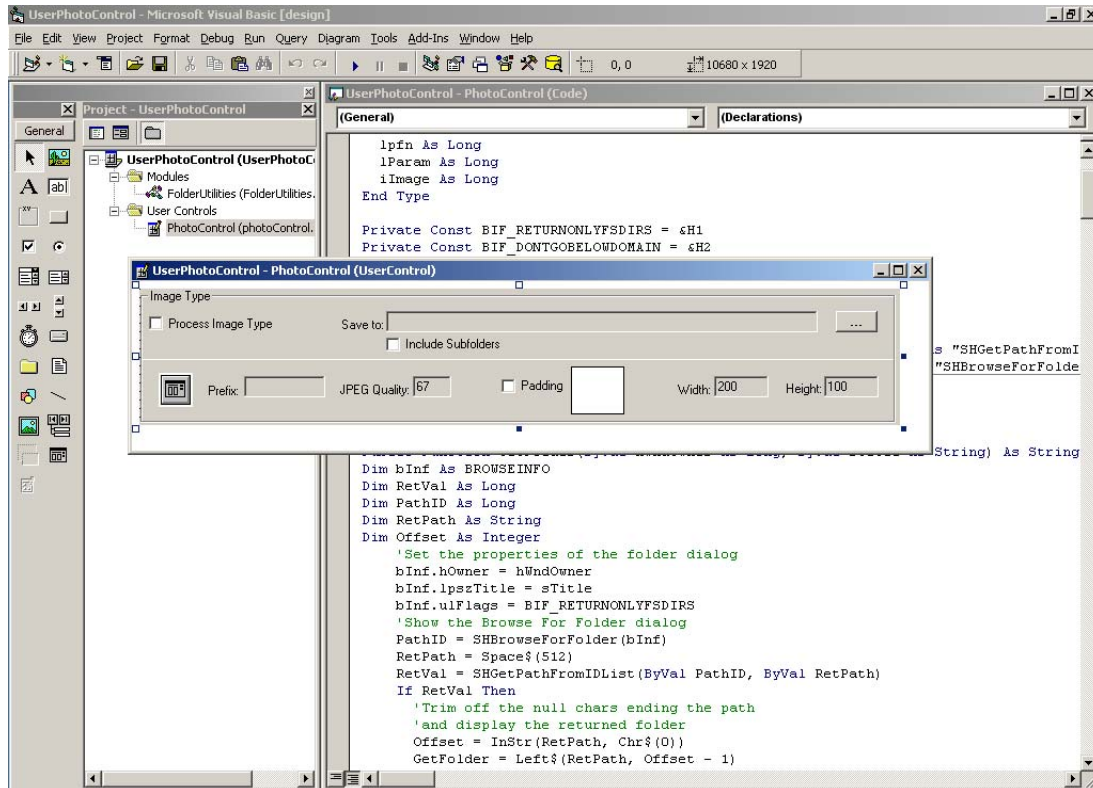


Fig. 6. Custom user control in its development

In the third edition of Thumbnails Generator (**Thumbnails Generator 3**) I decided to implement object orientation approach as long as it is allowed by Visual Basic 6. The program was analyzed and broken down into several "objects", each of which was implemented as a DLL. In addition to the previous versions of the program, Thumbnails Generator 3 saves settings into *.INI file which is read every time program starts. It also allows using not only Resize method to change image sizes, but Resample as well, which runs a little bit slower than Resize but produces better quality of the pictures. Another extra feature of the program allows users adding a watermark to the picture (like copyright mark, date, time, etc.). The source code for Thumbnails Generator 3 and installation package is available at <http://www.math.uaa.alaska.edu/~konovalm/cs395.htm>. Any comments or suggestions are more than welcome.

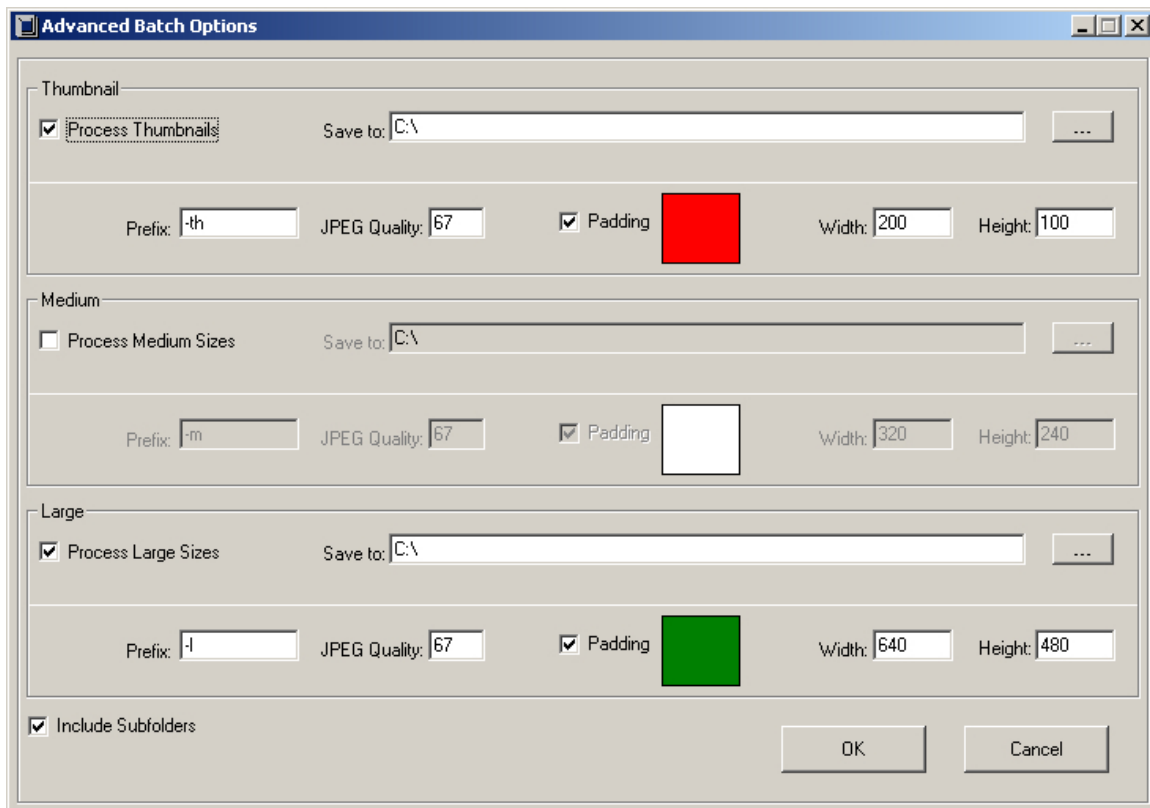


Fig. 7. Custom user control used in the Options dialog box

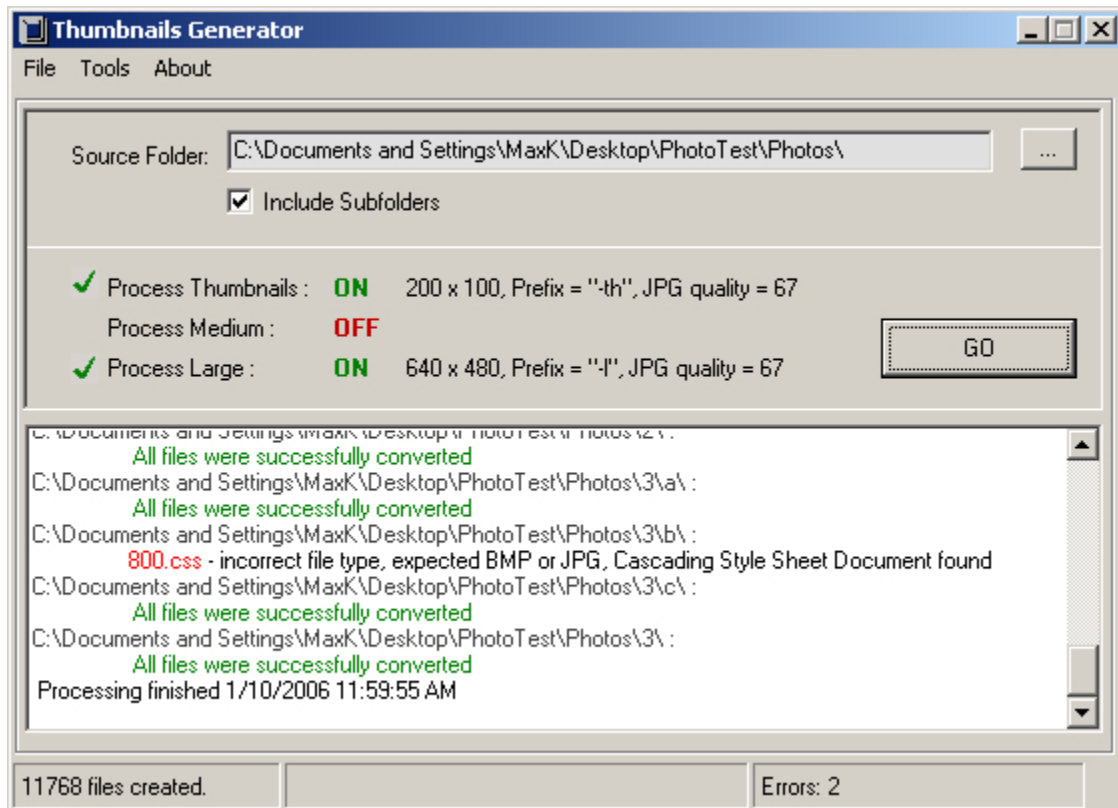


Fig. 8. Thumbnails Generator 2 on job completion screenshot.

As a part of my every day activities, I was involved into Web site design and programming. Some of my programs can be viewed at the following addresses:

http://test.alaskarealestate.com/Search/Property/showlistphotos_basic.asp?ln=5506862

<http://websrvr1.alaskarealestate.com/scripts/Inconverter.htm> - with JavaScript

<http://websrvr1.alaskarealestate.com/scripts/Inconverter.asp> - JavaScript free.

Currently I am developing a prototype for website online management. The intension is to create a web application which would allow designated users to upload files to the website without webmaster. The application will automatically create links under specified sections of the website.

Lessons learned

Overall, the experience I obtained at Alaska MLS, Inc. is very difficult to overestimate. In addition to learning Visual Basic 6, which I did not know before I came to the company, ASP, Microsoft SQL Server, improving my knowledge of SQL and database design principles, I got great real life, hands-on experience of a computer programmer job. Now I understand clearly what this job is consists of, what employers expect from you, and what to expect from the job itself.

The only downside of my experience with Alaska MLS is the .NET technologies are not used there, so I did not have a chance to try them on.

I would like to thank Chris Ulmer, the lead programmer at Alaska MLS, for his patience, assistance, and all the knowledge he shared with me.