

About Attrasoft ImageFinder for Dos

ImageFinder for Dos is a quick and easy programming tool. **ImageFinder for Dos** is not included in the **ImageFinder for Windows** and has to be purchased separately.

ImageFinder for Dos is the **ImageFinder for Windows** without the Graphical User Interface (GUI). Other than this, these two software are exactly the same.

Attrasoft Component-Object Structure consists of three layers:

- Application Layer
- Presentation Layer
- Attrasoft Matching Engine (AME) Layer

PolyApplet is the AME Layer developer tool, **TransApplet** is the Presentation Layer developer tool, and **ImageFinder for Dos** is the Application Layer developer tool.

Attrasoft ImageFinder can be used for:

- **Image Verification (1:1 Matching);**
- **Image Identification (1:N Matching);**
- **Image Search or Retrieval (1:N Matching); and**
- **Multiple Verification or Identification (N:N Matching).**

No commercial use of this software is permitted. No distribution of this software is permitted. This software is for personal use or developmental use only.

Software Requirements

Software Requirements are:

- (1) Windows .Net Framework 1.1.
- (2) J# .Net Redistributable.
- (3) ImageFinder for Windows.
- (4) Internet Explorer.

(1) To get the latest version .Net Framework 1.1, use Internet Explorer, and then click

“Tools\Windows Update”.

(2) To get J# .Net Redistributable, either get it from this Microsoft site directly:

<http://www.microsoft.com/downloads/details.aspx?familyid=E3CF70A9-84CA-4FEA-9E7D-7D674D2C7CA1&displaylang=en>

or get it from the **ImageFinder for Windows** CD with the following path:
CD:\vjredist.exe. Please install it by double clicking it.

Install ImageFinder For Dos

The **ImageFinder for Dos** consists of a **single executable file**, “**imagefinderfordos60.exe**”, plus **several text files**. Copy all files from the CD to a directory and run with the command “**imagefinderfordos60 x**”.

To use the examples in this menu, you must use the image data from the **ImageFinder for Windows**. Please copy file, **imagefinderfordos60.exe**, from the CD to the directory of the **ImageFinder** for Windows. If you have not changed the default directory, it is:

C:\Program Files\Attrasoft\ImageFinder 6.0

You can run the examples for the Dos prompt:

C:\Program Files\Attrasoft\ImageFinder 6.0> imagefinderfordos60 1

Support Information

Attrasoft ImageFinder for Dos
Attrasoft
P. O. Box 13051
Savannah, GA. 31406
USA

<http://attrasoft.com>
imagefinder@attrasoft.com
Phone: (912) 484-1717; (912) 897-1717

© Attrasoft 1998 – 2004

Statement of Copyright Restriction

Attrasoft has copyrighted the Attrasoft program you have purchased, and your rights of ownership are subject to the limitations and restrictions imposed by the copyright laws outlined below.

It is against the law to copy, reproduce or transmit (including, without limitation, electronic transmission over any network) any part of the program except as permitted by the copyright act of the United States (title 17, United States code). However, you are permitted by law to write the contents of the program into the machine memory of your computer so the program may be executed. You are also permitted by law to make a back-up copy of the program subject to the following restrictions:

- Each back-up copy must be treated in the same way as the original copy purchased from Attrasoft;
- **No copy (original, or back-up) may be used while another copy, (original, or back-up) is in use;**
- If you ever sell or give away the original copy of the program, all back-up copies must also be sold or given to the same person, or destroyed.

No commercial use of this software is permitted. No distribution of this software is permitted. This software is for personal use or developmental use only.

© Attrasoft 1998 - 2004

Contents

About Attrasoft ImageFinder for Dos.....	1
Software Requirements	2
Install ImageFinder For Dos.....	3
Support Information.....	4
Statement of Copyright Restriction.....	5
Contents	6
1. Introduction.....	8
1.1 Introduction.....	8
1.2 Why Dos Version ?.....	8
1.3 Software Requirements	9
2 ImageFinder for Windows	10
2.1 ImageFinder Overview	11
2.2 Sample Batch File.....	12
2.3 Creating Batch Codes and Parameter Files.....	15
2.4 Batch Execution Code.....	16
3. ImageFinder for Dos.....	19
3.1 Batch Commands	19
3.2 Start As A Windows Version.....	20
3.3 Example Batch Files	21
4. Examples.....	24
4.1 BioFilter Examples	25
4.1.1 BioFilter Overview	25
4.1.2 BioFilter Input.....	26
4.1.3 Label Identification.....	26
4.1.5 Generating Text Files.....	27
4.1.6 Batch Examples	28
4.2 NeuralFilter Examples	32
4.3 NeuralNet Filter Examples	36
4.3.1 NeuralNet Filter Overview	36
4.3.2 Logo Identification Problem.....	37
4.3.3 Feature Space recognition.....	37
4.3.4 Pixel Space Recognition	38
5. Visual Basic Programming.....	40
5.1 VB 6.0 Programming.....	40
6. AttraID and AttraScan System.....	42

6.1	Camera, Scanner and Driver	42
6.2	Functions Required	43
6.2.1	Initiation	43
6.2.2	Online Enrollment	45
6.2.3	Off-line Enrollment	46
6.2.4	Verification	46
6.2.5	Restrictions	47
6.3	Optional Supporting Buttons	47
6.4	System Design	48
6.5	Init ()	49
6.6	Capture ()	49
6.7	Verify ()	51
6.8	Tuning the System	51

1. Introduction

1.1 Introduction

ImageFinder for Dos is a quick and easy programming tool. **ImageFinder for Dos** is not included in the **ImageFinder for Windows** and has to be purchased separately.

ImageFinder for Dos is the **ImageFinder for Windows** without the Graphical User Interface (GUI). Other than this, the two software are exactly the same.

ImageFinder for Dos is one of several members in the **ImageFinder** family:

- **ImageFinder for Windows;**
- **ImageFinder for Dos;**
- **TransApplet;**
- **PolyApplet.**

Attrasoft Component-Object Structure consists of three layers:

- Application Layer
- Presentation Layer
- AME (Attrasoft Matching Engine) Layer

The available software are:

- **PolyApplet** is the AME (Attrasoft Matching Engine) Layer developer tool;
- **TransApplet** is the Presentation Layer developer tool;
- **ImageFinder for Dos** is the Application Layer developer tool; and
- **ImageFinder for Windows** is the Application Layer software.

1.2 Why Dos Version ?

ImageFinder for Dos is the **ImageFinder for Windows** without the Graphical User Interface (GUI). Other than this, the two software are exactly

the same. The Dos version has many advantages over the Windows version:

- The Dos version can start as a Windows version;
- You can run multiple tasks in a single Dos batch file;
- You can have programming control over the **ImageFinder**;
- You can develop a quick and dirty system for isolated applications or as a proof-of-concept project for major projects.

1.3 Software Requirements

Software Requirements are:

- (1) Windows .Net Framework 1.1;
- (2) J# .Net Redistributable;
- (3) ImageFinder for Windows;
- (4) Internet Explorer.

(1) To get the latest version .Net Framework 1.1, use Internet Explorer, then click:

“Tools\Windows Update”.

(2) To get J# .Net Redistributable, either get it from this Microsoft site directly:

<http://www.microsoft.com/downloads/details.aspx?familyid=E3CF70A9-84CA-4FEA-9E7D-7D674D2C7CA1&displaylang=en>

or get it from this CD with the following path: CD:\vjredist.exe. Please install it by double clicking it.

2 ImageFinder for Windows

You should become familiar with the **ImageFinder** for Windows first. There is one batch command in the **ImageFinder** for Windows:

Batch/Run

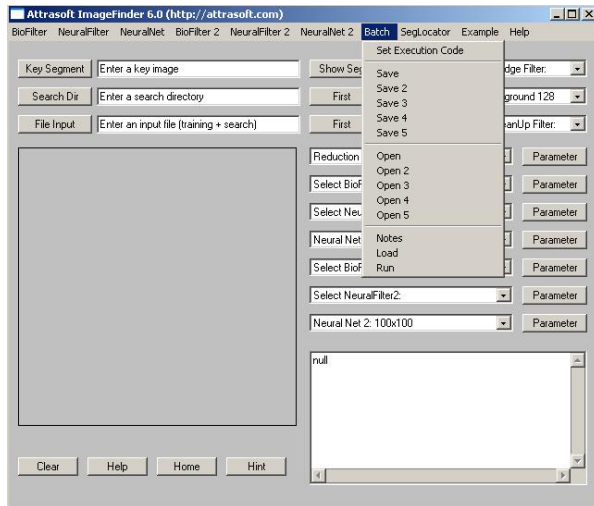


Figure 2.1 ImageFinder Batch Menu.

There are also ten menu items under the menu, “Batch”, in the **ImageFinder** for Windows:

- Batch/Save
- Batch/Save 2
- Batch/Save 3
- Batch/Save 4
- Batch/Save 5

- Batch/Open
- Batch/Open 2
- Batch/Open 3

Batch/Open 4
Batch/Open 5

The first 5 commands create the batch codes and save them to 5 different files. The batch codes can be opened later by the next 5 commands.

ImageFinder for Dos allows you to run these five batch files from the Dos prompt, Dos batch file, Visual Basic program, This allows developers to quickly integrate the **ImageFinder** into their applications.

2.1 *ImageFinder Overview*

This section will explain how the batch code is used in the **ImageFinder**. The procedure is:

1. Create an application using the **ImageFinder**;
2. Save the setting to batch code with the following commands:

Batch/Save
Batch/Save 2
Batch/Save 3
Batch/Save 4
Batch/Save 5

You might find the following online note useful in helping you remember what you saved into these 5 batch files:

Batch/Notes

3. Later you can open the batch file with the following commands:

Batch/Open
Batch/Open 2
Batch/Open 3
Batch/Open 4
Batch/Open 5

4. To load the parameters without running, click:

Batch/Load.

5. To load the parameters and run, click:
Batch/Run.

The Batch/Save command saves the following information:

- Filter selection and their Parameter settings;
- The template file, which contains the records from the images.

2.2 *Sample Batch File*

We assume you are familiar with the **ImageFinder** for Windows. An image matching requires that you specify the following Filters and their Parameters:

Image Preprocessing
Edge Filters;
Threshold Filters; and
Clean Up Filter.

Normalization
Reduction Filter.

Feature Recognition
BioFilter;
Neural Filter.

Pixel Level Recognition
NeuralNet Filter or ABM Filter.

Multi-Layered Pixel Recognition
BioFilter 2;
NeuralFilter 2;
ABM Filter 2.

A typical batch code looks like this:

```
[ImageFinder 6.0]  
  executionCode=1001
```

[Input]
trainFileName=None
searchDirName=None
fileInputName=None

[Image Processing Filters]
edgeFilter=1
thresholdFilter=1
cleanUpFilter=1

[Reduction Filter]
reductionType=0
segmentCut=0
sizeCut=0
borderCut=0
lookAtX=0
lookAtY=0
lookAtXLength=0
lookAtYLength=0

[BioFilter]
bioFilter=10
bioFilterPercent=50
bioFilterMode=2
bioFilterFinal=0
bioFilterCutOff=0

[NeuralFilter]
neuralFilter=0
neuralFilterPercent=20
neuralFilterMode=0
neuralFilterSize=2
neuralFilterFinal=0
neuralFilterCutOff=0

[Neural Net]
neuralNetFilter=0
segmentX=0
segmentY=0
segmentXlength=0
segmentYLength=0
symmetry=3
rotationType=0
translationType=0
scalingType=0
sensitivity=50
blurring=5
internalWeightCut=100
externalWeightCut=100

```
segmentSize=0
imageType=1
fileDisplayType=0
autoSegment=0
neuralNetMode=0
[BioFilter 2]
bioFilter=0
bioFilterPercent=20
bioFilterMode=2
bioFilterFinal=0
bioFilterCutOff=0
[NeuralFilter 2]
neuralFilter=0
neuralFilterPercent=20
neuralFilterMode=0
neuralFilterSize=2
neuralFilterFinal=0
neuralFilterCutOff=0
[Neural Net 2]
neuralNetFilter=0
segmentX=40
segmentY=40
segmentXlength=220
segmentYLength=220
symmetry=3
rotationType=0
translationType=0
scalingType=0
sensitivity=70
blurring=20
internalWeightCut=90
externalWeightCut=10000
segmentSize=0
imageType=1
fileDisplayType=0
autoSegment=0
neuralNetMode=0
```

This batch code has the following sections:

- Batch Execution Code
- Input
- Image Processing Filters
- Reduction Filter

BioFilter
Neural Filter
NeuralNet Filter or ABM Filter
BioFilter 2
NeuralFilter 2
ABM Filter 2

When you create the batch code by command Batch/Save, you will see the above code in the text area. When you open a batch file by command Batch/Open, you will also see the above code in the text area.

2.3 Creating Batch Codes and Parameter Files

Please create the batch file by:

(1) Use the **ImageFinder** to specify the filter selection and their parameter settings first. After you have obtained satisfactory results, go to next step.

(2) Create the batch files via one of the following five commands:

Batch/Save
Batch/Save 2
Batch/Save 3
Batch/Save 4
Batch/Save 5

(3) The produced batch files will be abm60.txt, abm60_2.txt, abm60_3.txt, abm60_4.txt, and abm60_5.txt.

(4) These files, and only these files, will be used by the **ImageFinder for Dos** so please do not rename them.

Beside creating the batch files, the above five commands will also create several parameter files:

Batch/Save
Bf60.txt
Bf260.txt
Nf60.txt
Nf260.txt

Tp60.txt

Batch/Save 2

Bf60_2.txt
Bf260_2.txt
Nf60_2.txt
Nf260_2.txt
Tp60_2.txt

Batch/Save 3

Bf60_3.txt
Bf260_3.txt
Nf60_3.txt
Nf260_3.txt
Tp60_3.txt

...

All of these files will be required for batch run.

2.4 Batch Execution Code

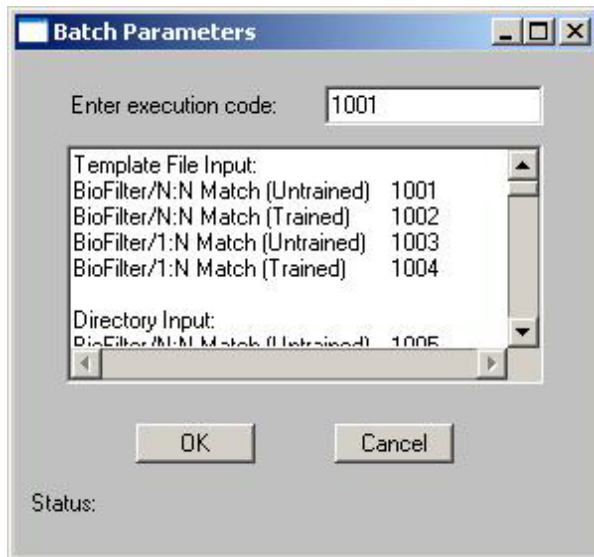


Figure 2.2 Execution Code Window.

There are many commands in the **ImageFinder**. Each command has an integer for identification. This integer is called Batch Execution Code. The “Batch/Run” command uses this code to run the command specified by the batch file. To find the batch code for each command, click:

Batch/Set Execution Code

You will see a textbox (Figure 2.2) and the following codes:

Template File Input:

BioFilter/N:N Match (Untrained)	1001	
BioFilter/N:N Match (Trained)		1002
BioFilter/1:N Match (Untrained)	1003	
BioFilter/1:N Match (Trained)		1004

Directory Input:

BioFilter/N:N Match (Untrained)	1005	
BioFilter/N:N Match (Trained)		1006
BioFilter/1:N Match (Untrained)	1007	
BioFilter/1:N Match (Trained)		1008

File Input:

BioFilter/N:N Match (Untrained)	1009	
BioFilter/N:N Match (Trained)		1010
BioFilter/1:N Match (Untrained)	1011	
BioFilter/1:N Match (Trained)		1012

Template File Input:

NeuralFilter/N:N Match		1013
NeuralFilter/N:(N-1) Match		1014
NeuralFilter/1:N Match		1015

Directory Input:

NeuralFilter/N:N Match		1016
NeuralFilter/N:(N-1) Match		1017
NeuralFilter/1:N Match		1018

File Input:

NeuralFilter/N:N Match		1019
NeuralFilter/N:(N-1) Match		1020
NeuralFilter/1:N Match		1021

Template File Input:

NeuralFilter/Query Set+Target Set/a1+a2 ==> b2	1022
NeuralFilter/Query Set+Target Set/a1+a3 ==> b3	1023
NeuralFilter/Query Set+Target Set/a1+a4 ==> b4	1024
NeuralFilter/Query Set+Target Set/a1+a5 ==> b5	1026

Neural Net:

NeuralNet/1:N Search	1027
NeuralNet/1:N Search+Sort	1028
NeuralNet/N:N Match	1029
NeuralNet/1:N Long-Search	1030
NeuralNet/1:N Long-Search+Sort	1031
NeuralNet/1:N File-Search	1032
NeuralNet/1:N File-Search+Sort	1033
NeuralNet/N:N File Match	1034

BioFilter 2:

BioFilter 2/1:N Match	1035
BioFilter 2/N:N Match	1036

Neural Filter 2:

NeuralFilter 2/1:N Match	1037
NeuralFilter 2/N:N Match	1038

Neural Net 2:

NeuralNet 2/1:N Match	1039
-----------------------	------

The default batch code is 1001. **You must specify the Batch Execution Code for your batch files.** The easiest way is:

- Click Batch/Set Execution Code;
- Enter the Batch Execution Code to the text box and click the OK button.

You can also make changes directly in the batch files. The batch files are abm60.txt, abm60_2.txt, abm60_3.txt, abm60_4.txt, abm60_5.txt.

3. ImageFinder for Dos

3.1 Batch Commands

The **ImageFinder for Dos** commands are:

```
C:\>...\imagefinderfordos60 x  
or  
C:\>...\imagefinderfordos60 x 1
```

Where **x = 1, 2, 3, 4, or 5** (used to specify one of the five batch files) and the file, imagefinderfordos60.exe, will be used.

If you use the command:

```
C:\>...\ imagefinderfordos60 1
```

You will need the following files:

```
Abm60.txt  
Bf60.txt  
Bf260.txt  
Nf60.txt  
Nf260.txt  
Tp60.txt
```

If you use the command:

```
C:\>...\ imagefinderfordos60 2
```

You will need the following files:

```
Abm60_2.txt  
Ba60_2.txt  
Bf260_2.txt  
Nf60_2.txt  
Nf260_2.txt  
Tp60_2.txt
```

If you use the command:

```
C:\>...\imagefinderfordos60 3
```

You will need the following files:

```
Abm60_3txt  
Ba60_3txt  
Bf260_3txt  
Nf60_3txt  
Nf260_3txt  
Tp60_3txt
```

...

The command will stop if:

- It cannot find the file specified by batch file; or
- The file specified by batch file does not agree with the batch format of the **ImageFinder**; or
- The output file specified by the batch file cannot be created; or
- The Execution Code is invalid.

If you want to stop the **ImageFinder** at the Dos prompt,

- Hit “Ctrl+Alt+Del” to open the Windows Task Manager;
- Go to the Processes Tab;
- Stop imagefinderfordos60.exe.

3.2 Start As A Windows Version

For debugging purposes, at the end of a run, you can open the **ImageFinder for Windows** to show the computation process. The command is:

```
C:\>...\imagefinderfordos60 x 1
```

where **x (1, 2, 3, 4, or 5)** is used to specify the batch file.

3.3 Example Batch Files

The **ImageFinder for Windows** has a large number of batch files. You can obtain the settings in these examples as follows:

Click “Example/BioFilter/N:N Match, Untrained, Label Template”
Click Batch/Run
Click Batch/Save

Now the setting in the example, “Example/BioFilter/N:N Match, Untrained, Label Template”, is saved to:

- Abm60.txt
- Bf60.txt
- Bf260.txt
- Nf60.txt
- Nf260.txt
- Tp60.txt

You can get settings for other examples in a similar way. The **ImageFinder for Windows** has the following examples:

Example/BioFilter

Use this menu item to open batch codes for the following examples:

- N:N Match, Untrained, Label Template
- N:N Match, Trained, Label Template
- 1:N Match, Untrained, Label Template
- 1:N Match, Trained, Label Template
- 1:N Match, Trained, Label File
- N:N Match, Trained, Label File
- 1:N Match, Trained, Label Directory
- N:N Match, Trained, Label Directory

Example/NeuralFilter

Use this menu item to open batch codes for the following examples:

- N:N Match, Label Template
- N:N-1 Match, Label Template
- 1:N Match, Label Template
- N:N Match, Label File

N:N-1 Match, Label File
1:N Match, Label File
N:N Match, Label Directory
N:N-1 Match, Label Directory
1:N Match, Label Directory
N:N Match, Logo Template

Example/Neural Net

Use this menu item to open batch codes for the following examples:

1:N Match, Logo, File
N:N Match, Logo, File
United Way - R
Tabasco - R
Mr. Potato - S
Monopoly - S
Compound - RS
Stamp 1
Stamp 2
Long-Search
License Plates
Fingerprint
Fingerprint - 1
Fingerprint - 2
Fingerprint - 34
Fingerprint - 7
Fingerprint - 569

Example/**Segment Locator**

Use this menu item to open batch codes for the following examples:

United Way
Monopoly
Mr. Potato
AAA
Ford
Soup
PointLocator: Feret 20
PointLocator: Feret 100
PointLocator: Results

Example/BioFilter 2

Use this menu item to open batch codes for the following examples:

Label Match Setting

Label Example 1 (1:N)

Label Example 2 (N:N)

Label Example 3 (1:N)

Example/NeuralFilter 2

Use this menu item to open batch codes for the following examples:

Setting

Example 1 (1:N)

Example 2 (N:N)

Example 3 (1:N)

Example/Neural Net 2

Use this menu item to open batch codes for the following examples:

Match 1

Match 2

Match 3

No Match 1

No Match 2

No Match 3

4. Examples

In this chapter, we will run several examples using the **ImageFinder for Dos**. We will use the Dos Prompt. You will need both the **ImageFinder for Windows** and the **ImageFinder for Dos** to complete the examples in this chapter. You will need three things:

- **ImageFinder for Dos** in the correct folder;
- **ImageFinder for Windows** to provide data and text files; &
- **Dos Prompt**.

(1) Please copy **ImageFinder for Dos (imagefinderfordos60.exe)** into the **ImageFinder** folder. If you have not changed the default directory, it is:

C:\Program Files\Attrasoft\ImageFinder 6.0

(2) The Dos version will use images and the following text files:

Abm60.txt
Bf60.txt
Bf260.txt
Nf60.txt
Nf260.txt
Tp60.txt

The **ImageFinder for Windows** will provide both image data and text files.

(3) Start the Dos prompt:

Windows\Start\All programs\Accessories\Command Prompt

You can run the examples for the Dos prompt:

C:\Program Files\Attrasoft\ImageFinder 6.0> imagefinderfordos60 1

In your own application, you will need to write the user interface for both issuing the Dos command and processing the output files.

4.1 BioFilter Examples

4.1.1 BioFilter Overview

The image matching will be done in several steps:

- Initialization
- Converting Images to Records
- Training
- Template Matching

Let us look at each phase.

Initialization

Initialization sets the **ImageFinder** parameters.

Converting Images to Records:

An image is mapped into a record in a Feature Space. This step is slow (several images per second); however:

- This step can be done once for all; and
- This is linear, i.e. the time is directly proportional to the number of images. Therefore, this step does not have much impact on the operating speed.
- The results will be stored in “.\a1.txt”, where “.” is the **ImageFinder** directory.

Training

Training uses the data collected in advance to teach the BioFilter how to match. Training requires two files, a1.txt and match.txt:

- A1.txt is the record file, which contains many records. Each image is converted into a record. A1.txt is produced automatically in the last step. A record represents signatures of an image in a feature space.
- Match.txt is a list of matching pairs. This file will teach the

ImageFinder who will match with whom. You must prepare this file.

Template Matching

The matching speed will be between **100,000 – 1,000,000 comparisons per second**. Both BioFilter and Neural Filter will do the template matching. There are several commands for the matching and all of the commands will be used in this chapter.

4.1.2 BioFilter Input

The Batch Execution can take three types of input:

- Template Input
- File Input
- Directory Input

For the Template Input, the template file is loaded, so the **ImageFinder** does not convert images to records, but rather works on the saved records directly.

For the File Input and Directory Input, the template file is not loaded, so the **ImageFinder** will first convert images to records. The **ImageFinder** will scan through all images in the input file or the input directory.

4.1.3 Label Identification

There are 8 examples for the Label Identification problem:

N:N Match, Untrained, Label Template
N:N Match, Trained, Label Template
1:N Match, Untrained, Label Template
1:N Match, Trained, Label Template
1:N Match, Trained, Label File
N:N Match, Trained, Label File
1:N Match, Trained, Label Directory
N:N Match, Trained, Label Directory

There are 152 pairs or 304 images in this example; they are located at the directory “.\biofilterex1”, where “.\” is the **ImageFinder** software location.

The N:N Match will compare each of the 304 images with all 304 images. A typical image looks like the following Figure.



Figure 4.1 An image in the “.biofilterx1” folder.

The Possible Matches will be $N * N = 304 * 304 = 92,416$.

The Positive Matches will be $N * 2 = 608$.

The Negative Match will be $N * (N - 2) = 91,808$.

4.1.5 Generating Text Files

The first 4 examples use Template Input, which is fast because the conversion from Input Space to Feature Space is already done. This will be followed by 2 examples for File Input and 2 examples for Directory Input.

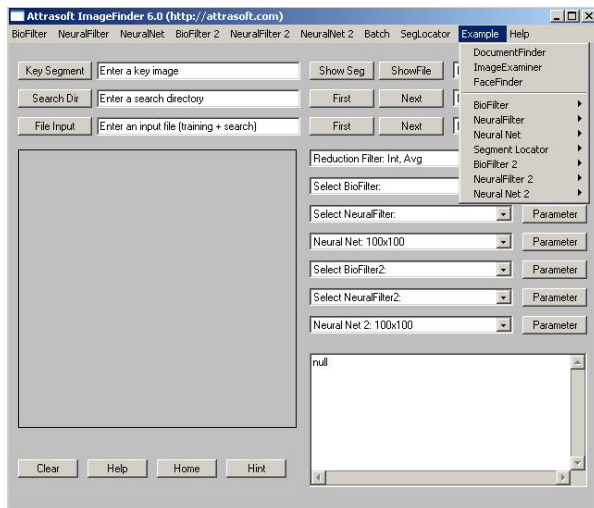


Figure 4.2 Example Menu.

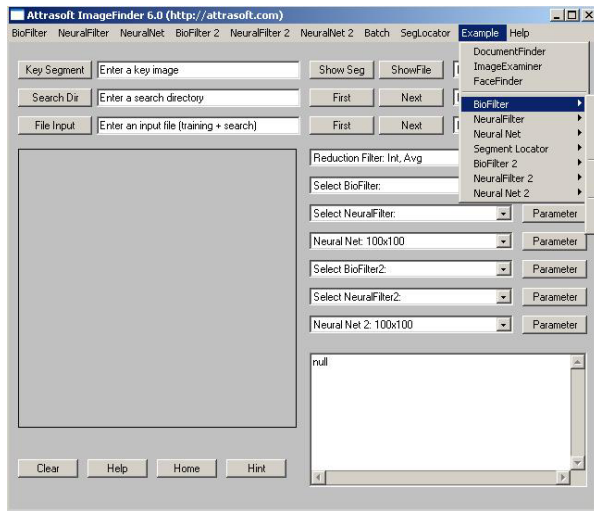


Figure 4.3 Example/BioFilter Menu.

To generate the required text files:

Abm60.txt
 Bf60.txt
 Bf260.txt
 Nf60.txt
 Nf260.txt
 Tp60.txt

Please:

- Click “Example/BioFilter/N:N Match, Untrained, Label Template”
- Click Batch/Run
- Click Batch/Save

If you are working on your own problem, you have to use the trial and error approach to get a satisfactory setting; after that, click Batch/Save.

4.1.6 Batch Examples

Again, you will need:

- **ImageFinder for Windows** to generate text files and provide data for the following example;
- **ImageFinder for Dos** in the correct folder; and
- Dos prompt.

Before working on the examples:

- Start **ImageFinder for Windows**
- Open the Dos Prompt and type in the following text in bold face and enter “Enter”:

```
C:\.\.>cd..
C:\.\.>cd..
C:\>cd “C:\Program Files\Attrasoft\ImageFinder 6.0”
C:\Program Files\Attrasoft\ImageFinder 6.0>
```

Example 1. Untrained N:N Matching, Template Input

ImageFinder for Windows:

To run the example in the Windows version:

- Click “Example\BioFilter\N:N Match, Untrained, Label Template”;
- Click: Batch\Run;
- Click: Batch\Save.

You should see b1.txt opened at this point. Go to end of b1.txt, you should see that 901 matches are found.

Now clear b1.txt (“Edit>Select All”, followed by Del key) and save b1.txt so you can see the **ImageFinder for Dos** results.

To make that sure b1.txt is empty, open it again by either using the Windows Explorer, or clicking “BioFilter\BioFilter Results” command in **ImageFinder for Windows**. You should see a blank file.

Dos Prompt:

In the Dos prompt, type following text in bold face and hit “Enter”:

```
C:\Program Files\Attrasoft\ImageFinder 6.0> imagefinderfordos60 1  
C:\Program Files\Attrasoft\ImageFinder 6.0>
```

You should see the next prompt instantaneously. Now open b1.txt (Windows Explorer, or clicking “BioFilter\BioFilter Results”); you should see the same results as windows version.

Now we can run the Dos version as a Windows version. In the Dos Window, hit F3 key to get the last command and change it to:

```
C:\Program Files\Attrasoft\ImageFinder 6.0> imagefinderfordos60 1 1
```

You should see that the Windows version is activated for debugging purposes. The Unsupervised Learning is not accurate, so we will make an analysis in the next example, Supervised Learning.

Example 2. Trained N:N Matching, Template Input.

ImageFinder for Windows:

To run the example in the Windows version:

- Click “Example\BioFilter\N:N Match, Trained, Label Template”;
- Click: Batch\Run;
- Click: Batch\Save;
- Clear b1.txt.

Dos Prompt:

In the Dos prompt, type following text in bold face and enter “Enter”:

```
C:\Program Files\Attrasoft\ImageFinder 6.0> imagefinderfordos60 1  
C:\Program Files\Attrasoft\ImageFinder 6.0>
```

You should see the next prompt instantaneously. Now open b1.txt

(Windows Explorer, or clicking “BioFilter\BioFilter Results”); you should see the same results as windows version. Go to the end of b1.txt, and see:

“Total Number of Matches = 850”

Analysis:

The Possible Matches will be $N * N = 304 * 304 = 92,416$.

The Positive Matches will be $N * 2 = 608$.

The Negative Matches will be $N * (N - 2) = 91,808$.

Attrasoft Matches = 850.

Positive Matches = 608

False Acceptance = 242 = 850 – 608

False Rejection Rate = 0%

False Acceptance Rate = 0.26 % = 242/91,808.

Example 3. Untrained 1:N Matching, Template Input.

The operation procedure is similar to Example 1 with the following substitute:

- Click “Example\BioFilter\1:N Match, Untrained, Label Template”.

Example 4. Trained 1:N Matching, Template Input.

The operation procedure is similar to Example 1 with the following substitute:

- Click “Example\BioFilter\1:N Match, Trained, Label Template”.

Example 5. Trained 1:N Matching, File Input.

The operation procedure is similar to Example 1 with the following substitute:

- Click “Example\BioFilter\1:N Match, Trained, Label File”.

In this example, the template file is not loaded, so the **ImageFinder** will first convert images to records. There are 8 images in this example. During the Windows version run, you will see the **ImageFinder for Windows** scan through these 8 images. During the Dos run, the **ImageFinder** will run a bit

longer than the last 4 examples. Hit Ctrl/Alt/Del to open Windows Task Manager and go to the Processes Tab, you should see “imagefinderfordos60.exe”.

Example 6. Trained N:N Matching, File Input.

The operation procedure is similar to Example 1 with the following substitute:

- Click “Example\BioFilter\N:N Match, Trained, Label File”.

Example 7. Trained 1:N Matching, Directory Input.

This is the Label Recognition example. This example uses Trained 1:N Matching and uses the directory file.

- Click “Example\BioFilter\1:N Match, Trained, Label Directory”.

Example 8. Trained N:N Matching, Directory Input.

The operation procedure is similar to Example 1 with the following substitute:

- Click “Example\BioFilter\N:N Match, Trained, Label Directory”.

4.2 NeuralFilter Examples

The Neural Filter is similar to the BioFilter. The main purpose of the BioFilter is to make a quick assessment of the problem, while the Neural Filter is the main Feature Space Matching Filter.

The Batch Execution can take three types of input:

- Template Input
- File Input
- Directory Input

For the Template Input, the template file is loaded, so the **ImageFinder** does not convert images to records, but rather works on the records directly. For the File Input and Directory Input, the template file is not loaded, so the

ImageFinder will first convert images to records. The **ImageFinder** will scan through all images in the input file or the input directory.

This section deals with the same Label Identification problem as the last section. There are 9 examples in this section:

N:N Match, Label Template
N:N-1 Match, Label Template
1:N Match, Label Template
N:N Match, Label File
N:N-1 Match, Label File
1:N Match, Label File
N:N Match, Label Directory
N:N-1 Match, Label Directory
1:N Match, Label Directory
N:N Match, Logo Template

The first 3 examples use Template Input, which is fast because the conversion from Input Space to Feature Space is already done. This will be followed by 3 examples for File Input and 3 examples for Directory Input.

Example 1. N:N Matching, Template Input.

ImageFinder for Windows:

To run the example in the Windows version:

- Click “Example\NeuralFilter\N:N Match, Label Template”;
- Click: Batch\Run;
- Click: Batch\Save.

You should see b1.txt opened at this point. Go to end of b1.txt, you should see that 608 matches are found.

Now clear b1.txt (“Edit>Select All”, followed by Del key) and save b1.txt so you can see the **ImageFinder for Dos** results.

To make that sure b1.txt is empty, open it by either using the Windows Explorer, or clicking “NeuralFilter\NeuralFilter Results” command in **ImageFinder** for Windows. You should see a blank file.

Dos Prompt:

In the Dos prompt, type following text in bold face and hit “Enter”:

```
C:\Program Files\Attrasoft\ImageFinder 6.0> imagefinderfordos60 1  
C:\Program Files\Attrasoft\ImageFinder 6.0>
```

You should see the next prompt instantaneously. Now open b1.txt (Windows Explorer, or clicking “NeuralFilter\NeuralFilter Results”); you should see the same results as the windows version.

Now we can run the Dos version as a Windows version. In the Dos Window, hit F3 key to get the last command and change it to:

```
C:\Program Files\Attrasoft\ImageFinder 6.0> imagefinderfordos60 1 1
```

You should see the Windows version is activated for debugging purposes. The Unsupervised Learning is not accurate, so we will make an analysis later.

Analysis:

The Possible Matches will be $N * N = 304 * 304 = 92,416$.

The Positive Matches will be $N * 2 = 608$.

The Negative Match will be $N * (N - 2) = 91,808$.

Attrasoft Matches = 608.

Positive Matches = 608

False Acceptance = 0 = 608 – 608

False Rejection Rate = 0%

False Acceptance Rate= 0 %.

This result is better than the BioFilter result.

Example 2. N:N-1 Matching, Template Input.

The operation procedure is similar to Example 1 with the following substitute:

- Click “Example\NeuralFilter\N:N-1 Match, Label Template”.

Example 3. 1:N Matching, Template Input.

The operation procedure is similar to Example 1 with the following substitute:

- Click “Example\NeuralFilter\1:N Match, Label Template”.

Example 4. N:N Matching, File Input.

The operation procedure is similar to Example 1 with the following substitute:

- Click “Example\NeuralFilter\N:N Match, Label File”.

This example uses an Input File, biofilterex1_input1.txt. This input file lists the first 4 pairs of images in the Label Recognition problem. In this example, the template file is not loaded, so the **ImageFinder** will first convert images to records.

Example 5. N:N-1 Matching, File Input.

The operation procedure is similar to Example 1 with the following substitute:

- Click “Example\NeuralFilter\N:N-1 Match, Label File”.

Example 6. 1:N Matching, File Input.

The operation procedure is similar to Example 1 with the following substitute:

- Click “Example\NeuralFilter\1:N Match, Label File”.

Example 7. N:N Matching, Directory Input.

The operation procedure is similar to Example 1 with the following substitute:

- Click “Example\NeuralFilter\N:N Match, Label Directory”.

This example uses an input directory, “.biofilterex2\”, where “.” is the **ImageFinder** directory. This input directory lists the first 4 pairs of images in the Label Recognition problem. In this example, the template file is not loaded, so the **ImageFinder** will first convert images to records.

Example 8. N:N-1 Matching, Directory Input.

The operation procedure is similar to Example 1 with the following substitute:

- Click “Example\NeuralFilter\N:N-1 Match, Label Directory”.

Example 9. 1:N Matching, Directory Input.

The operation procedure is similar to Example 1 with the following substitute:

- Click “Example\NeuralFilter\1:N Match, Label Directory”.

4.3 NeuralNet Filter Examples

4.3.1 NeuralNet Filter Overview

The **ImageFinder** recognizes images in two phases:

- Feature Space Matching
- Input Space Matching

Feature Space Matching will be done in several steps:

- Initialization
- Converting Images to Records
- Training
- Template Matching

The Input Space Recognition Steps are:

- Initialization
- Pixel Matching
- Results

Let us look at Input Space Matching:

Initialization

Initialization sets the **ImageFinder** parameters.

Entering Data:

The output of the Neural Filter, b1.txt, can be used as the input for the neural net matching. If this file is too large, you can split it into several pieces and match one piece at a time.

Training and Matching

NeuralNet Filter learns the first image in each block and matches it against the rest of the images in the block.

4.3.2 Logo Identification Problem

There are 489 images in this example; they are located in the directory “.\neuralnetex1”, where “.\” is the **ImageFinder** software location.

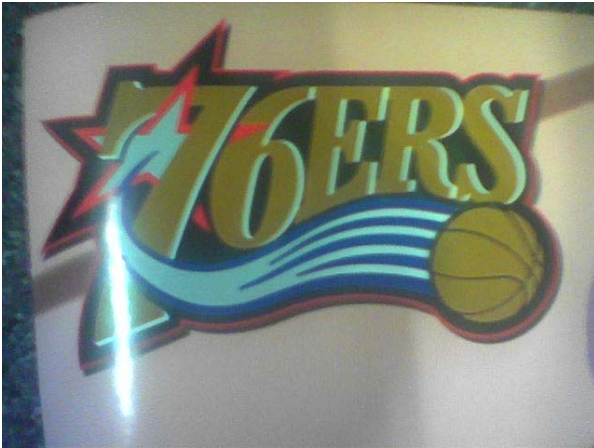


Figure 4.4 An image in “.\neuralnetex1”.

4.3.3 Feature Space recognition

We will first make Feature Space Recognition to eliminate the unmatched images via the Neural Filter. This example uses N:N Matching and Template Input. To duplicate this example in the Windows version:

- Click “Example\NeuralFilter\N:N Match, Logo Template”;
- Click: Batch\Run.

You can repeat this example in the Dos version.

4.3.4 Pixel Space Recognition

This section has Logo Recognition examples, 1:N Matching via file input.

Example 1:N Matching, File Input.

This is the Logo Recognition example. This example uses 1:N Matching.

- Click “Example\Neural Net\1:N Match, Logo, File”;
- Click: Batch\Run.

In this example, the input file is the output file of the Neural Filter, so the results cover both Feature Space recognition (Neural Filter) and Input Space recognition (NeuralNet Filter). The input file, b1.txt, has many blocks of data; in each block, the first line is the Neural Filter input, and the rest of the lines are Neural-Filter output. The NeuralNet Filter will learn the image from the first line and match it against the rest of the line. This will be done for each block of data.

Now we will put the whole thing together:

ImageFinder for Windows:

To run the example in the Windows version:

- Click “Example\NeuralFilter\N:N Match, Logo Template”;
- Click: Batch\Run to generate b1.txt;

You should see b1.txt opened at this point. Go to end of b1.txt, you should see that 13575 matches are found. In the following, rather than using 13,575 matches, we will only use a few matches for this example:

- Click “Example\Neural Net\1:N Match, Logo, File”;

Look at the batch code in the text window of the **ImageFinder for Windows**:

[ImageFinder 6.0]

executionCode=1032

[Input]

trainFileName=Please Specify The Training Image!

searchDirName=Please Specify The Search Directory!

fileInputName=.b1_neuralnetex1_11.txt

...

The input file is not b1.txt, which requires 13,575 matches, but rather b1_neuralnetex1_11.txt, which has a much smaller number of matches. Now,

- Click: Batch\Run to get results.

This will take a few minutes because pixel matching is slower than the feature matching.

- Click: Batch\Save.

Now all of the text files are ready for the Dos version.

Dos Prompt:

In the Dos prompt, type following text in bold face and enter “Enter”:

C:\Program Files\Attrasoft\ImageFinder 6.0> **imagefinderfordos60 1**

C:\Program Files\Attrasoft\ImageFinder 6.0>

5. Visual Basic Programming

In this section, we will use Visual Basic 6.0 to write a simple program.

5.1 VB 6.0 Programming

I. Create a Form:

1. Start by running Visual Basic 6.0. The New Project dialog box will appear. Select “Standard EXE” icon.
2. Click OK and the project will be created.

You will see the default form in the center of the screen, as shown below.

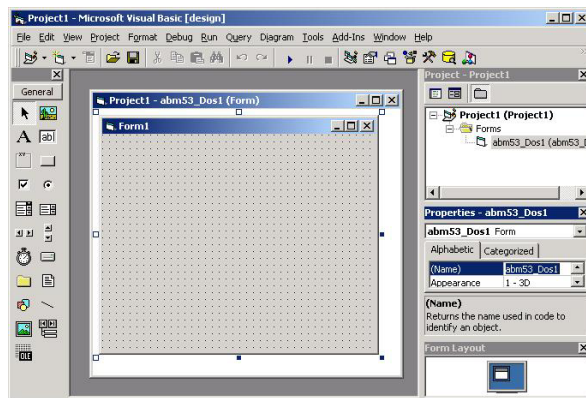


Figure 5.1. The default form appears after your project has been created.

II. Add Button

Add a button to the form, Command1, with caption “Test 1”.

III. Add Code

1. Double click Command1 button to add the code:

```
Private Sub Command1_Click()  
    Dim x As Long  
    x = Shell("imagefinderfordos60 1", 1)  
End Sub
```

2. Click “File/Make Exe File” and make “test1.exe” file.

IV. Test Run

1. Copy the “test1.exe” file to the “C:\Program Files\Attrasoft\ImageFinder 6.0” directory.
2. Go to Dos Prompt: C:\Program Files\Attrasoft\ImageFinder 6.0>.
3. Enter the command in bold face:
C:\Program Files\Attrasoft\ImageFinder 6.0> **test1**
This command will start test1.exe.
4. Click the button on the form to run the **ImageFinder for Dos**.

6. AttraiD and AttraScan System

In this chapter, we will introduce system integration with the **ImageFinder for Dos**.

AttraiD is user interface software, which integrates the **ImageFinder** for Dos and a camera. **AttraScan** is user interface software, which integrates the **ImageFinder** for Dos and a scanner.

The **AttraiD** Technical Specification is:

- Camera: Logitech QuickCam Express;
- Camera Driver: Logitech ImageStudio 7.0.0;
- Interface Software: **AttraiD**;
- Image Matching Engine: **ImageFinder for Dos**.

The **AttraScan** Technical Specification is:

- Camera: Fujitsu fi-4120C scanner;
- Camera Driver: Fujitsu ScanAll 21;
- Interface Software: **AttraScan**;
- Image Matching Engine: **ImageFinder for Dos**

You can use Visual Basic for making similar systems. The system integration will deal with:

- Camera/Scanner;
- Driver;
- Interface Software;
- Image Matching Engine: **ImageFinder for Dos**.

In general, the scanner has higher recognition rates because they impose more restrictions on images.

6.1 Camera, Scanner and Driver

To start a system integration of a Camera/Scanner with the Attrasoft **ImageFinder for Dos**, select a camera or scanner. The **ImageFinder** can be used for any type of images; thus, the integrated system can be used for any type of images.

This chapter introduces programming for systems like the **AttraID** or **AttraScan**. In this chapter, we will use Microsoft Visual Studio 6.0 to write the user interface software.

After selecting a camera/scanner; set up the chosen camera/scanner by following the instructions that comes with the camera/scanner.

6.2 Functions Required

From an operational point of view, there are two modes:

- Enrollment;
- Verification.

The Enrollment procedure takes the pictures of an object and stores them with an ID.

The Verification procedure verifies the object's identity with his ID.

Your system must support Enrollment and Verification. In the following, we will list the jobs needing to be done and their related buttons in your software.

6.2.1 Initiation

The initialization step will start the camera/scanner driver. It will bring up the software similar to Figure 6.1, which is for Logitech QuickCam, or Figure 6.2, which is a scanner driver. This is necessary so your system can control image capture by issuing the keystrokes later.

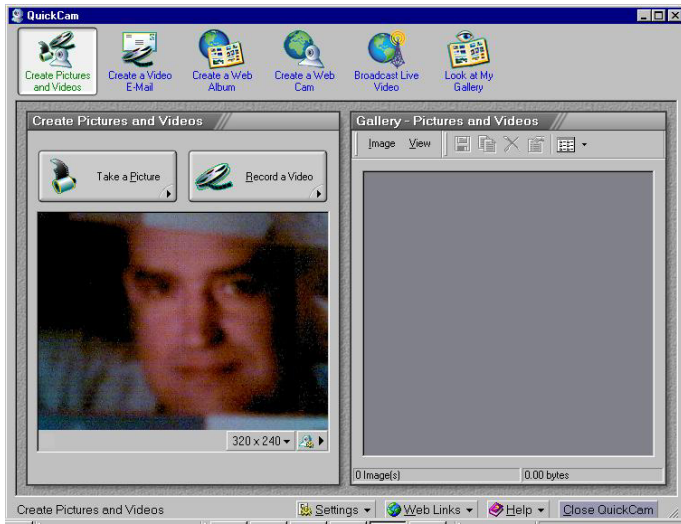


Figure 6.1 AttraID Camera Driver.

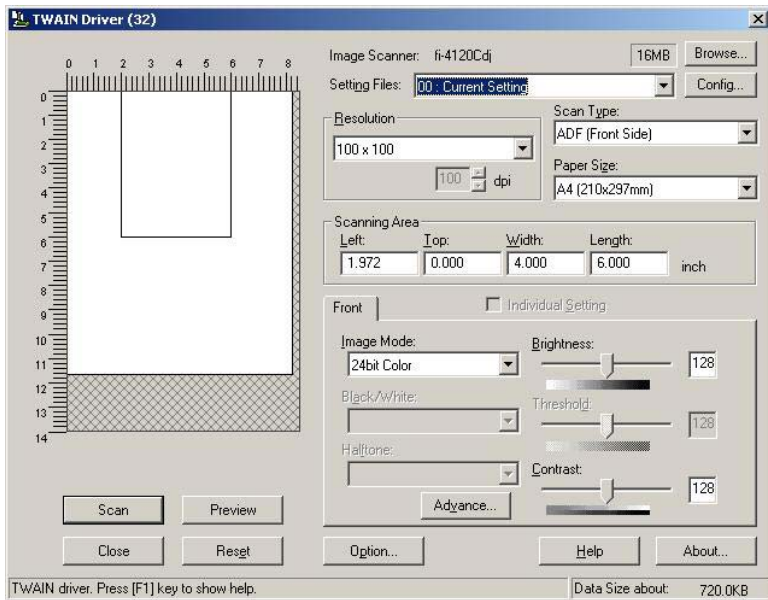


Figure 6.2 AttraScan Scanner Driver.

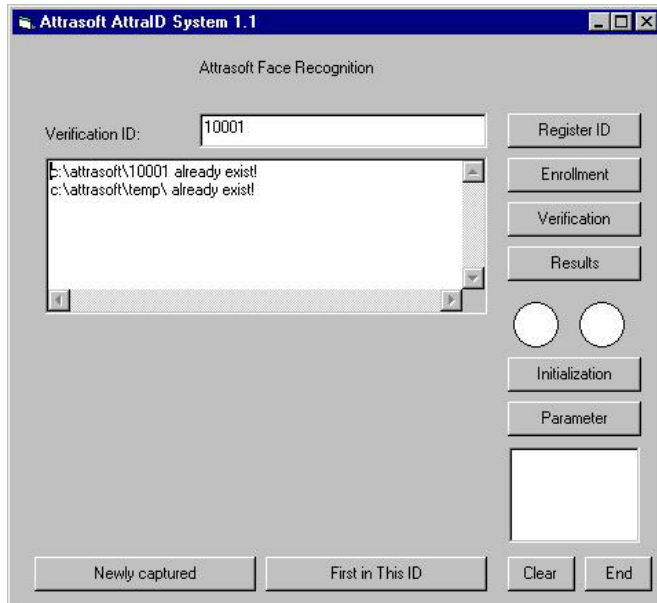


Figure 6.3 AttrID Systems.

6.2.2 Online Enrollment

The Enrollment procedure takes the pictures of an object and stores them with an ID. This can be done on-line or off-line.

The on-line approach is used for **AttrID** and the off-line approach is used for **AttrScan**.

- The on-line approach in **AttrID** uses the **AttrID** to obtain the enrollment data.
- The off-line approach in **AttrScan** prepares the stored data in advance and will not use the **AttrScan** for enrollment purposes.

The On-line Enrollment procedure has two steps:

Step 1. Verification ID

Enter ID to the “Verification ID” box.

Step 2. Enrollment

Click “Enrollment”.

An example is given in Figure 6.3.

6.2.3 Off-line Enrollment

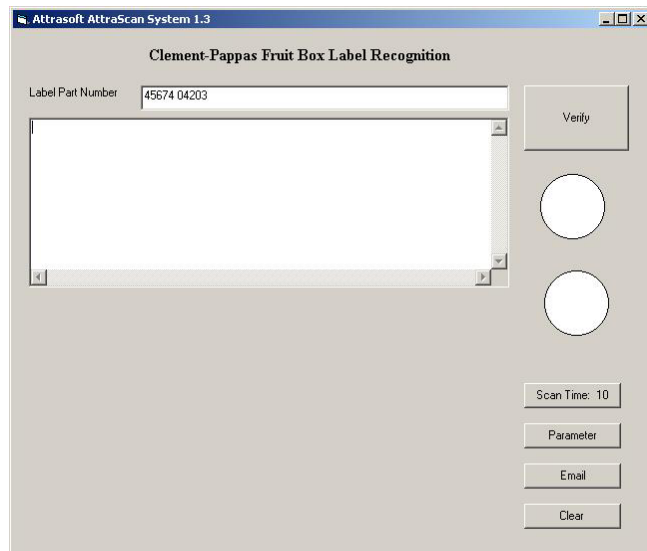


Figure 6.4 Attrascan System.

An off-line enrollment example is **Attrascan** (Figure 6.4) where there is no Enrollment button. The master data is prepared in advance.

6.2.4 Verification

The Verification procedure verifies the object’s identity with it’s ID. The newly entered ID will bring up the previously stored images. They will be compared with the newly captured images.

The Verification procedure has two steps:

Step 1. Verification ID

Enter ID to the “Verification ID” Text Box.

Step 2. Verification

Click “Verification”.

6.2.5 Restrictions

Your **system** might have some restrictions, for example:

- **The system requires a white screen background for cameras;**
- **The distance between the camera and the object is fixed.**

6.3 *Optional Supporting Buttons*

The Enrollment procedure will use the following text box and buttons:

“Verification ID” Text Box;
“Enrollment” Button.

The Verification procedure will use the following text box and buttons:

“Verification ID” Text Box;
“Verification” Button.

Additional Buttons:

Parameter

Use the “Parameter” button to see the parameters used by the your User Interface. The advanced users can change the parameters by clicking this button.

Newly Captured

Use the “Newly Captured” button to see the newly captured image in both the Enrollment and Verification Procedures, if they are not shown automatically.

First in This ID

Use the “First in This ID” button to see the first image associated with the ID in the “Verification ID” box, if there are images associated with the ID.

Next in This ID

Use the “Next in This ID” button to see the next image associated

with the ID in the “Verification ID” box, if there are images associated with the ID.

Clear

Use the “Clear” button to clear the message text area.

Lights

A ‘Green’ Light and a ‘Red’ Light can be added to the user interface. They are normally white. If the last Verification is positive, the ‘Green’ light will be on; if the last Verification is negative, the ‘Red’ light will be on.

6.4 System Design

The system integration will deal with:

Camera/Scanner;
Driver;
Interface Software;
Image Matching Engine: **ImageFinder for Dos**.

Three of the above are off-the-shelf products and you are responsible for integration. Visual Basic will be a good platform for this job.

Your integration will present a User Interface, which will initialize the software.

- The init () function will be necessary to start the Camera/Scanner Driver.
- The capture () function will be necessary to interact with the Camera/Scanner Driver to capture images.
- The verify () function will be necessary to interact with the **ImageFinder for Dos** to match images.

We will discuss these functions in detail in the next three sections.

6.5 *Init ()*

Init() is the function which will be called in your user interface initialization phase. In Visual Basic 6.0, a statement does this:

```
X = Shell(s, 1)
```

Where 's' is the path of the driver. This should start the Camera/Scanner Driver. 'X' is the identification of the driver process, which will be used by your software to go back to this process.

6.6 *Capture ()*

At the heart of your user interface are two functions:

- Capture ();
- Verify ().

The capture () function captures and saves an image to the disk. The function implements the following jobs:

- Go to the Driver;
- Issue the capture command;
- Save the image.

Go to the Driver

In Visual Basic 6.0, a statement does this:

```
AppActivate x, True
```

Issue the capture command

Assuming the software driver uses the following menu item to capture an image:

```
Image/Capture
```

In Visual Basic 6.0, you will need to send a keystroke “Alt I” to get the “Image” menu; following by a keystroke “Alt C” to go to menu item “Image/Capture”; followed by keystroke “Enter”. A few statements will do this:

```
‘ Go to driver with ID x, see the last section for x
AppActivate x, True
SendKeys "%", True
DoEvents

' Image menu
SendKeys "%{I}", True
DoEvents

' Image/Capture menu item
SendKeys "%{C}", True
DoEvents

' Right place, enter
SendKeys "{ENTER}", True
DoEvents
```

Save the Image

Assuming the camera/scanner driver uses the following menu item to save a captured image:

File/Save

In Visual Basic 6.0, you will need to send a keystroke “Alt F” to get the “File” menu; following by a keystroke “Alt S” to go to menu item “File/Save”; followed by keystroke “Enter”. A few statements will do this:

```
‘ Go to driver with ID x, see the last section for x
AppActivate x, True
SendKeys "%", True
DoEvents

' File menu
```

```
SendKeys "%{F}", True  
DoEvents
```

```
' File/Save menu item  
SendKeys "%{S}", True  
DoEvents
```

```
' Right place, enter  
SendKeys "{ENTER}", True  
DoEvents
```

At this point, the “Save As” dialog box should be opened. Please

- Navigate to the right place by sending keystroke, ”Tab”, several times;
- Send the file path and name in a string; and
- Send the keystroke “Enter” to hit the OK button in the “Save As” dialog box.

This should save the newly captured image.

6.7 Verify ()

At this point, you have the newly captured image. Find the previously stored images from the Verification ID and you have all the data necessary for Verification.

The main statement in verify () function will be:

```
Shell(“imagefinderfordos60.exe 1”, 1)
```

6.8 Tuning the System

Your system will consists of:

- Camera/Scanner;
- Attrasoft **ImageFinder for Dos.**

The Attrasoft **ImageFinder for Dos** provides image-matching functions. This module has several parameters. The parameters used by your **system** is in several text files. To change the matching engine setting, you must change the parameters in these files.

To tune the parameters, you must use the **ImageFinder for Windows**. For command “imagefinderfordos60 1”, after tuning, copy these files to a working folder:

- Abm60.txt
- Bf60.txt
- Bf260.txt
- Nf60.txt
- Nf260.txt
- Tp60.txt.

For command “imagefinderfordos60 2”; after tuning, copy these files to a working folder:

- Abm60_2.txt
- Bf60_2.txt
- Bf260_2.txt
- Nf60_2.txt
- Nf260_2.txt
- Tp60_2.txt.

Attrasoft ImageFinder for Dos
Version 6.0

User's Guide and Reference Manual
<http://attrasoft.com>

Attrasoft ImageFinder for Dos
Version 6.0

User's Guide and Reference Manual
<http://attrasoft.com>

Attrasoft
P. O. Box 13051
Savannah, GA. 31406
webmaster@attrasoft.com
imagefinder@attrasoft.com