ES CLIPBOARD MONITOR ENGINE USER MANUAL

Overview

The ES Clipboard Monitor Engine is a program that stays resident in memory, waiting for an image to be copied to the clipboard. It utilizes major components from the ES Image Printer Driver, minus the native driver itself. The Clipboard Monitor Engine is simply a layer on top that utilizes the core functions.

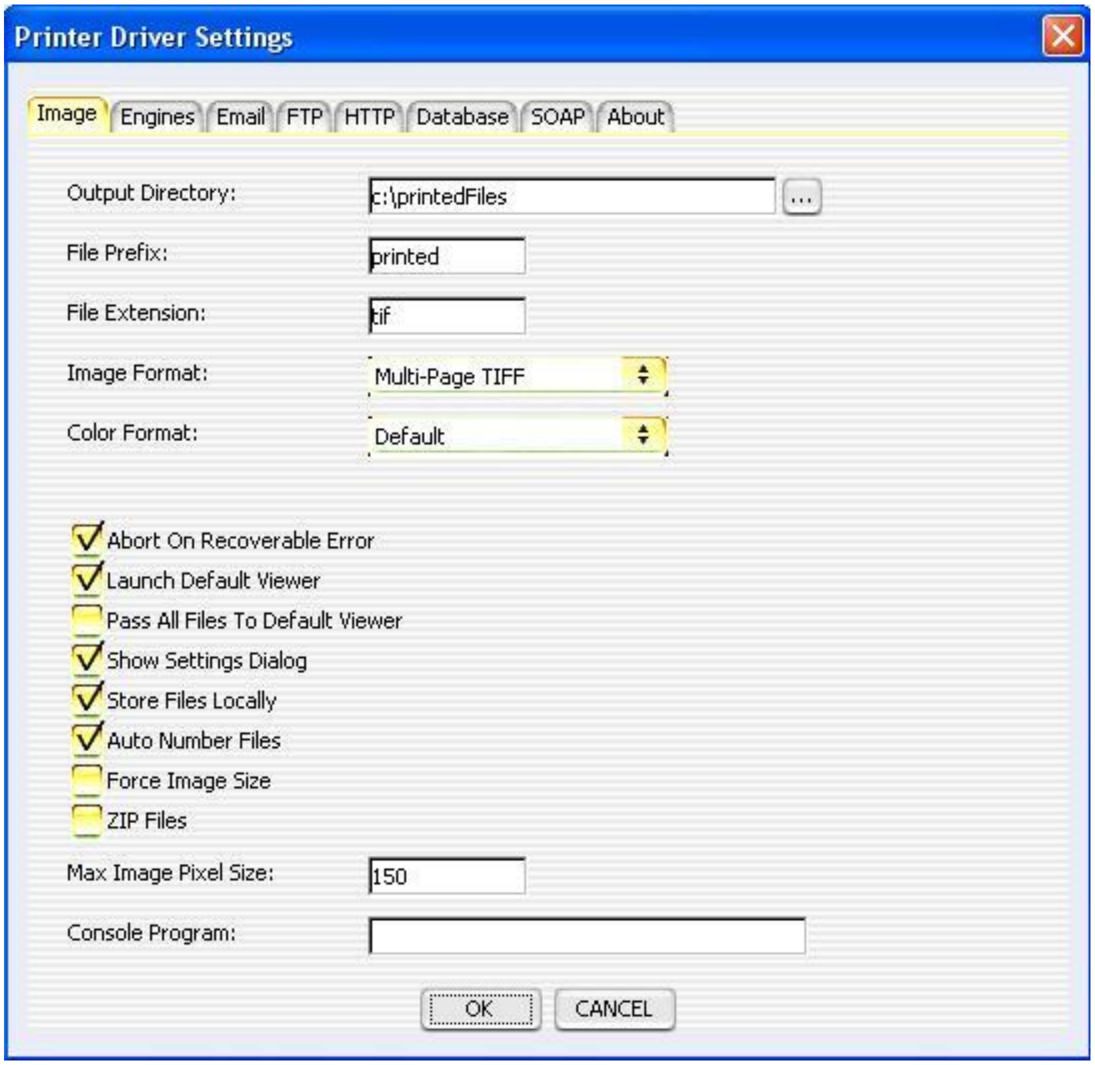
The monitor is great for easily capturing screenshots and enabling instant email, FTP, etc. It can save tremendous time for troubleshooting and sending information.

If an image is copied to the clipboard, the monitor will popup a dialog asking whether to process the image through the engine or not. The user may choose to ignore the copy or send the image through the engine. The clipboard is left undisturbed regardless of whether the image was processed or not.

On Windows, the clipboard monitor is a program named 'es_clipboard_monitor.exe' and can be terminated at any time through Windows Task Manager, or disabled permanently through the 'Disable ES Clipboard Monitor Engine' shortcut in the programs menu.

On non-Windows systems, execution of the Clipboard Engine is possible by executing the 'es_clipboard_monitor_engine.jar' file. On most Operating Systems, this is possible by double clicking the jar file.

Images are produced and stored on a specified drive and location with incremental numbers. The engine is flexible and supports several image file formats such as PNG, BMP, JPEG, JPEG 2000, TIFF, Multi-Page TIFF, PNM, and PDF.



Printer Driver Settings

Output Directory: c:\printedFiles

File Prefix: printed

File Extension: tif

Image Format: Multi-Page TIFF

Color Format: Default

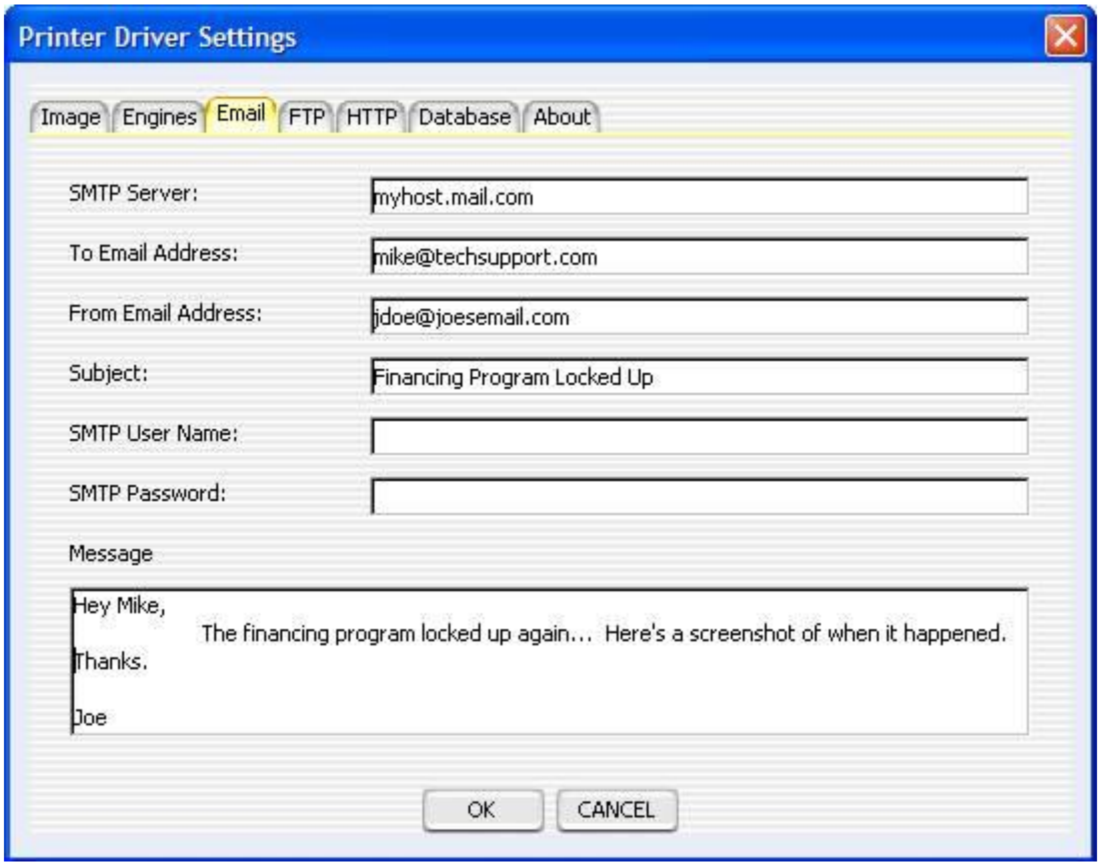
[✓] Abort On Recoverable Error
[✓] Launch Default Viewer
[ ] Pass All Files To Default Viewer
[✓] Show Settings Dialog
[✓] Store Files Locally
[✓] Auto Number Files
[ ] Force Image Size
[ ] ZIP Files

Max Image Pixel Size: 150

Console Program:

FIGURE 1

It also supports sending an email to anyone via SMTP. The images captured are added as attachments to the email, and users may type in a full text message for the body of the email. This is accomplished by displaying a graphical user interface at capture time, which is optional. This interface can be suppressed at each capture.



SMTP Server:	myhost.mail.com
To Email Address:	mike@techsupport.com
From Email Address:	jdoe@joesemail.com
Subject:	Financing Program Locked Up
SMTP User Name:	
SMTP Password:	

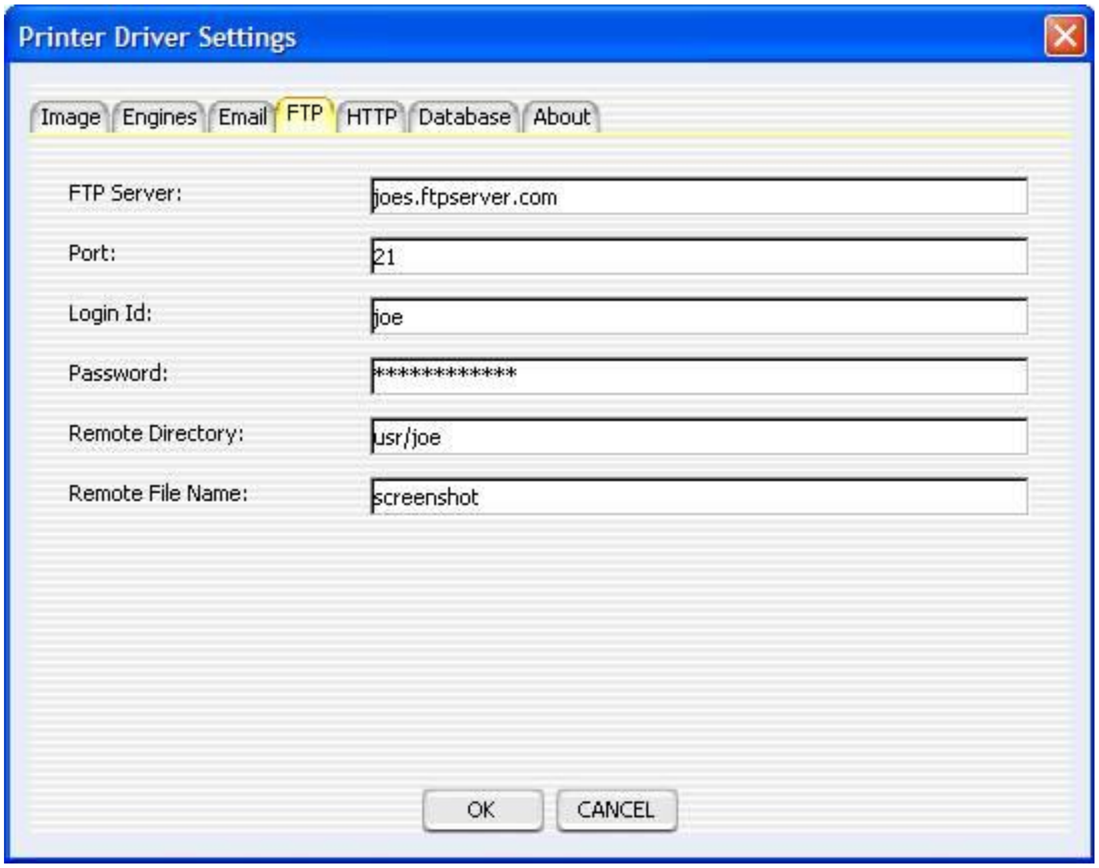
Message

Hey Mike,
The financing program locked up again... Here's a screenshot of when it happened.
Thanks.

Joe

FIGURE 2

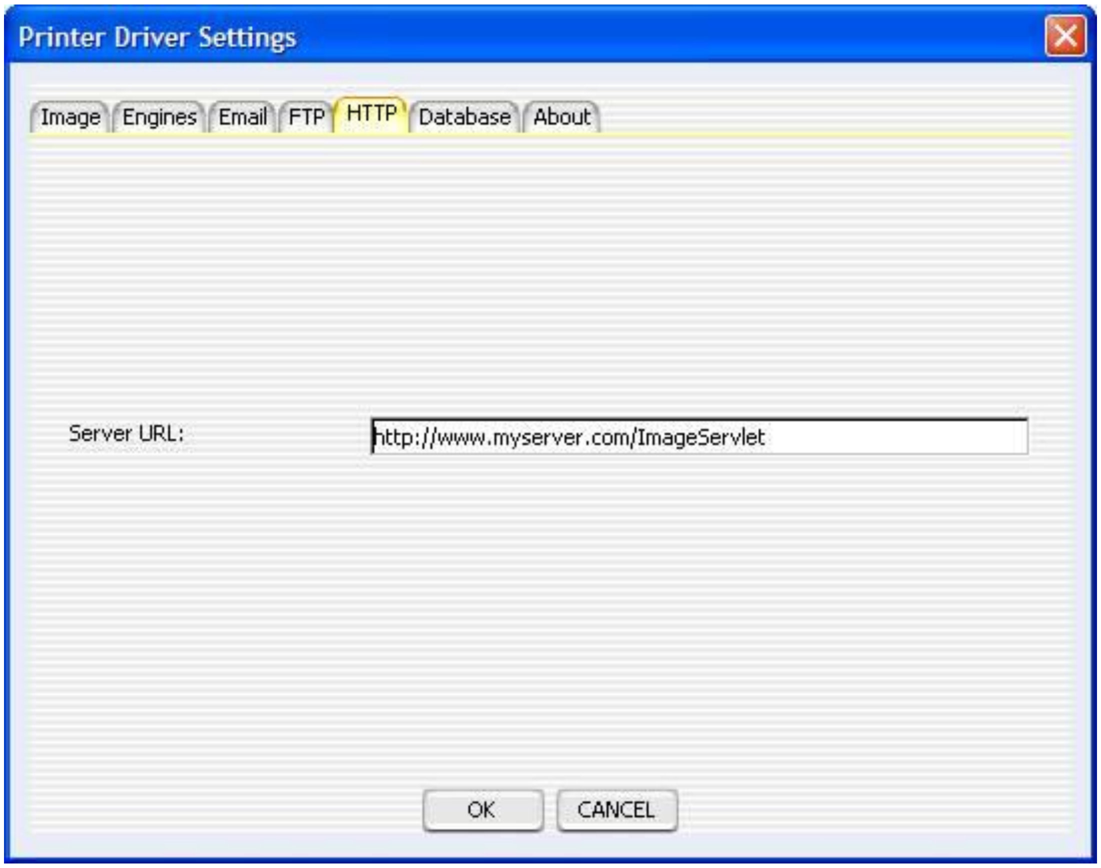
FTP sending is also an option with the engine. The images that are produced can automatically be uploaded to a FTP server on any TCP/IP network (Intranet, Internet, etc).



FTP Server: joes.ftpserver.com
Port: 21
Login Id: joe
Password: ************
Remote Directory: usr/joe
Remote File Name: screenshot

FIGURE 3

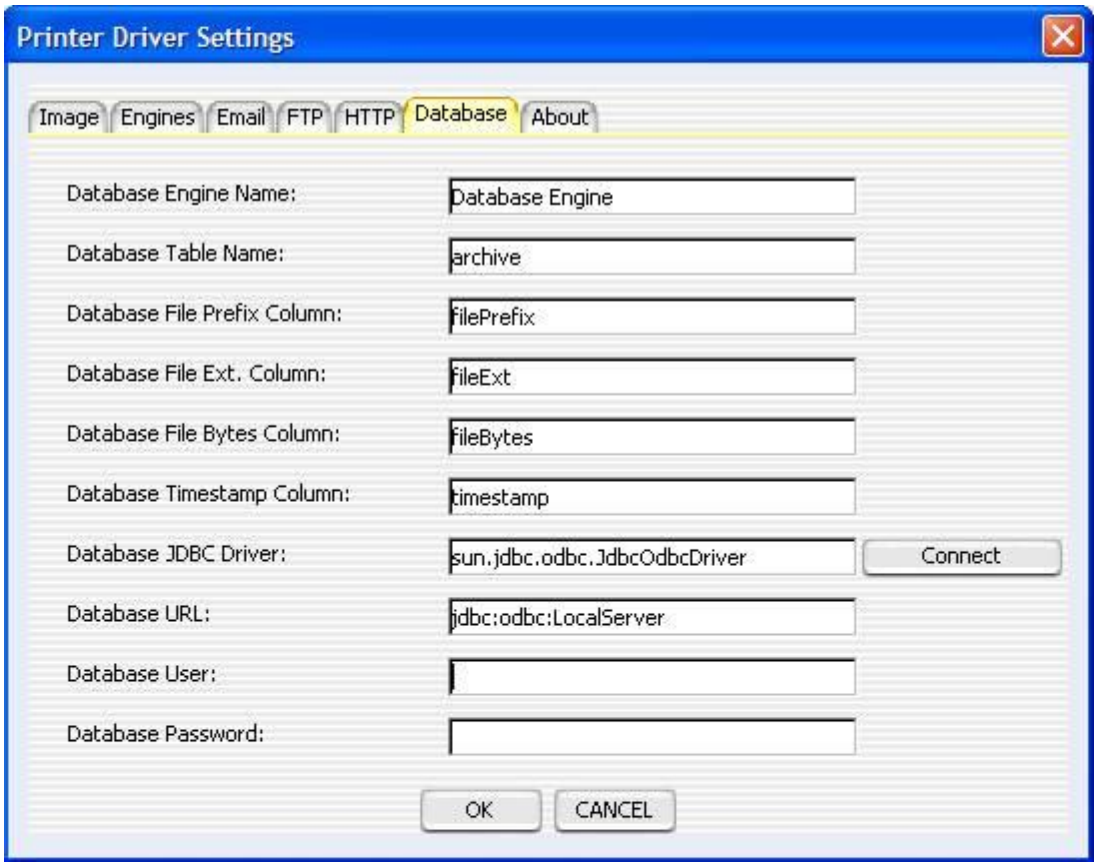
HTTP posts are also an option with the engine. The images that are produced can automatically be uploaded to a web server via HTTP or HTTPS on any TCP/IP network (Intranet, Internet, etc).



Server URL: http://www.myserver.com/ImageServlet

FIGURE 4

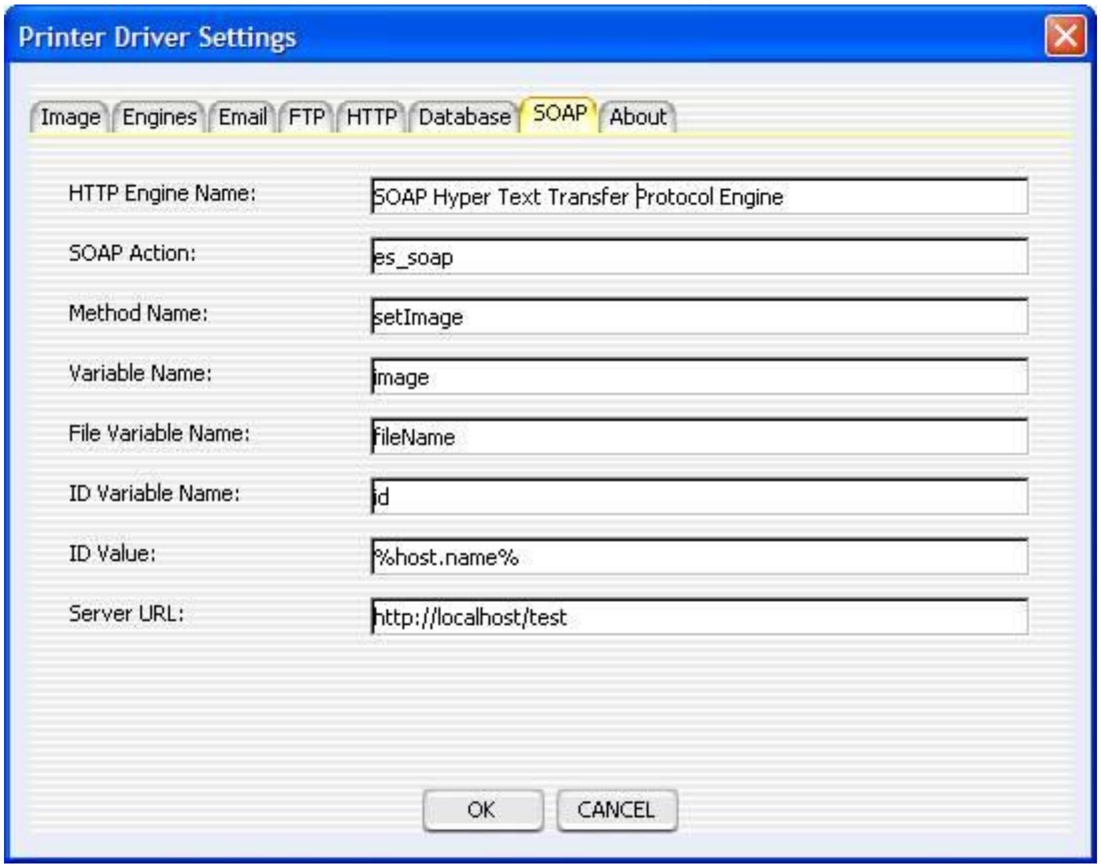
The engine has the capability to save the images into a database. This can be done with any standard JDBC or ODBC connection.



Database Engine Name:	Database Engine
Database Table Name:	archive
Database File Prefix Column:	filePrefix
Database File Ext. Column:	fileExt
Database File Bytes Column:	fileBytes
Database Timestamp Column:	timestamp
Database JDBC Driver:	sun.jdbc.odbc.JdbcOdbcDriver
Database URL:	jdbc:odbc:LocalServer
Database User:	
Database Password:	

FIGURE 5

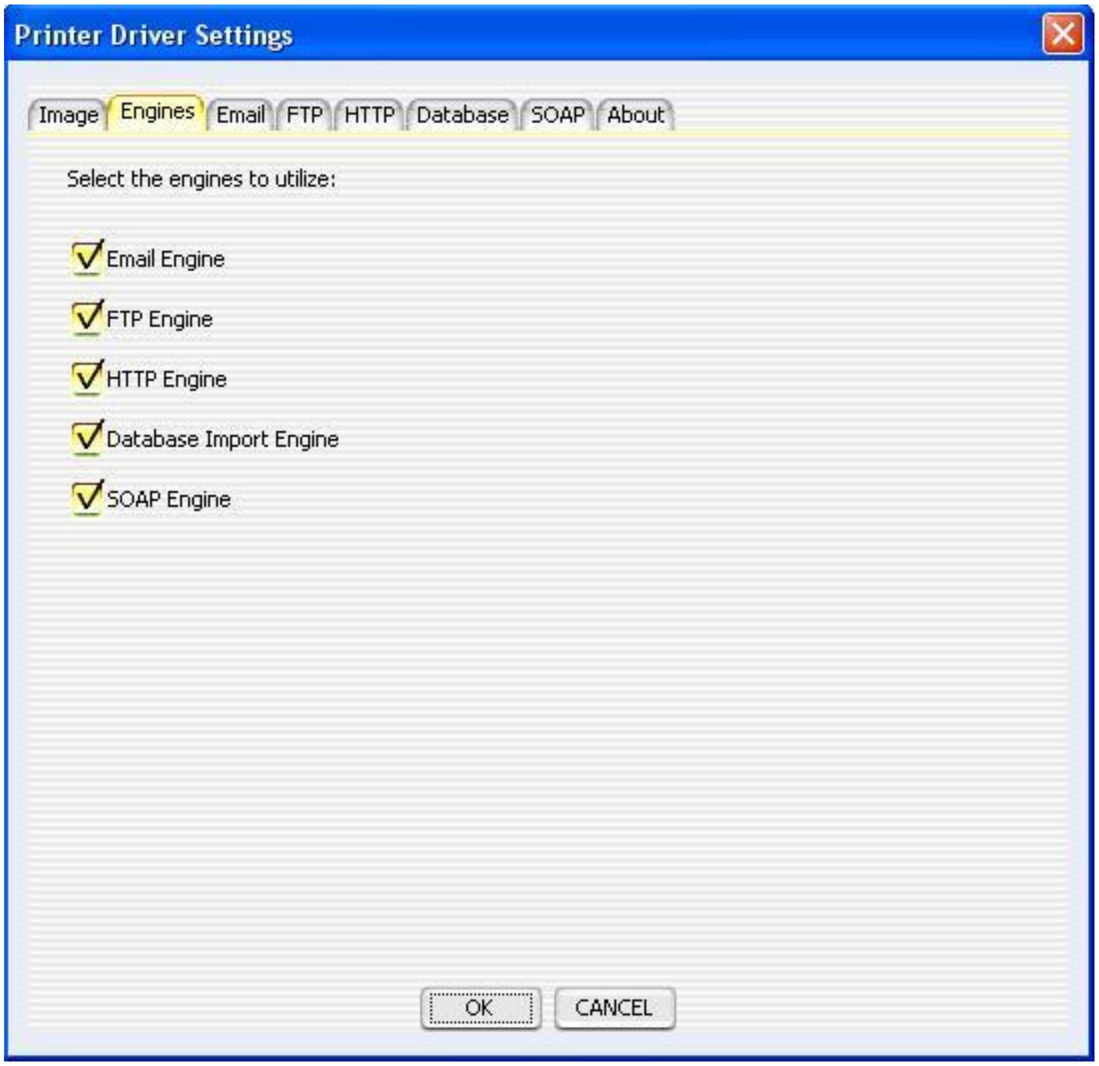
SOAP is becoming more and more popular in today's complex environments. It allows various languages, web services, environments, etc., to communicate with each other through a common programming interface. The engine has the ability to send a SOAP message to a Web Service, HTTP Server, etc. with the files base 64 encoded inside as parameters.



HTTP Engine Name:	SOAP Hyper Text Transfer Protocol Engine
SOAP Action:	es_soap
Method Name:	setImage
Variable Name:	image
File Variable Name:	fileName
ID Variable Name:	id
ID Value:	%host.name%
Server URL:	http://localhost/test

FIGURE 5b

The optional Email, FTP, HTTP, SOAP, and Database engines can easily be enabled/disabled from the Engines tab. This allows all the settings to be kept for those engines while preventing them from actually being utilized during the current capture.


Printer Driver Settings

Select the engines to utilize:

[✓] Email Engine

[✓] FTP Engine

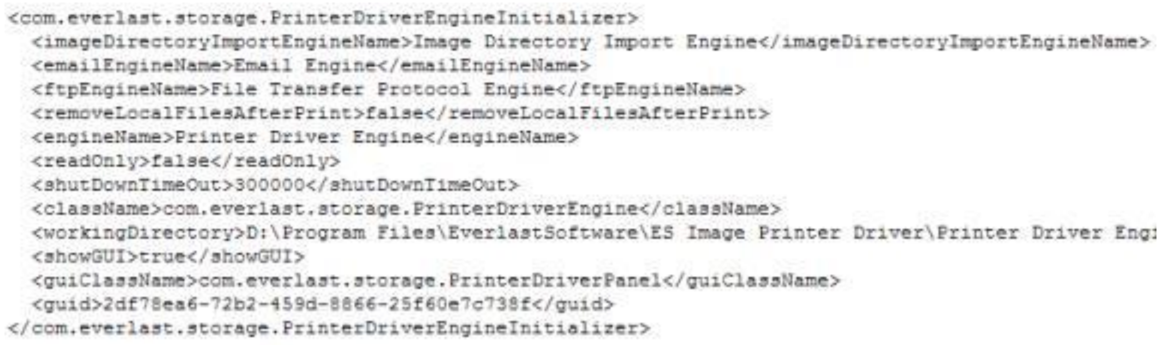
[✓] HTTP Engine

[✓] Database Import Engine

[✓] SOAP Engine

FIGURE 6

The engine is driven by XML files that can be administered for use in business environments or for mass capturing.



```xml
<com.everlast.storage.PrinterDriverEngineInitializer>
  <imageDirectoryImportEngineName>Image Directory Import Engine</imageDirectoryImportEngineName>
  <emailEngineName>Email Engine</emailEngineName>
  <ftpEngineName>File Transfer Protocol Engine</ftpEngineName>
  <removeLocalFilesAfterPrint>false</removeLocalFilesAfterPrint>
  <engineName>Printer Driver Engine</engineName>
  <readOnly>false</readOnly>
  <shutDownTimeOut>300000</shutDownTimeOut>
  <className>com.everlast.storage.PrinterDriverEngine</className>
  <workingDirectory>D:\Program Files\EverlastSoftware\ES Image Printer Driver\Printer Driver Eng
  <showGUI>true</showGUI>
  <guiClassName>com.everlast.storage.PrinterDriverPanel</guiClassName>
  <guid>2df78ea6-72b2-459d-8866-25f60e7c738f</guid>
</com.everlast.storage.PrinterDriverEngineInitializer>
```

FIGURE 6a

Purpose

All of these options make the Clipboard Monitor Engine an excellent tool for bug reporting, problem tracking, etc., for businesses wanting to maximize their business time. For home users, the engine is great for capturing web sites, images from other programs, emails, etc. It's also a quick way to share screenshots with friends and family.

All businesses (small, medium, and large) deal with and rely on many other computer programs in order to maximize their employees' time. They often utilize both, 3rd party software, and in-house software, developed specifically for their needs. Almost all businesses have had the experience of finding show-stopping bugs in various programs that prevent them from getting their job done. Often time, businesses have an entire department committed to helping the users either find work a rounds for the software, ways to fix the software, or submit incident reports to the developers of the software. This is a very time consuming task, not only for the computer support department, but also the employees that use the software who must communicate the problems to the support department in a clear manner so that the real problem can be understood. This is where the ES Clipboard Monitor Engine can be of great aid and save businesses thousands, if not millions, of dollars a year.

A picture can speak a thousand words, and that is exactly what the engine does. With the simple capture of any document, image, screenshot, etc., images are produced which can be instantly emailed to a support person, transferred to an FTP server, or simply saved to the local computer's storage with one simple click. All of this can happen in the background too, without requiring user interaction, thus saving MORE time for the employees, while maximizing the information for someone to help troubleshoot the problem.

Not only can the ES Clipboard Monitor Engine do what was mentioned above, it has been designed to easily develop and plug in other interfaces, so that businesses may tightly integrate with their own support systems, environment, etc. If a certain interface is important to your business, don't hesitate to ask about it, even if it is a completely custom interface. We are dedicated to helping our clients make their businesses more economical and to save them time. After all, time is money.

Clipboard Monitor/Printer Driver Settings

The engine has several standard options and settings in order to produce and save the images created. Figure 1 refers to these options.

1) Output Directory – This is the directory to save the produced images to. It can be any standard Windows drive, such as 'c:\myFiles', or it can be a UNC pathname such as '\\winserver\\C\uploadarea'. The optional button showing '…' next to the Output Directory displays a file dialog (Figure 1b below) where one can pick the directory with a graphical user interface.

2) File Prefix – The prefix to save each file with. This is always followed by an incremental number automatically by the engine. For example, if 'captured' was the prefix and 3 pages were printed, the actual file prefixes will be 'captured_1', 'captured_2', and 'captured_3'. The engine automatically detects filenames with the previous name if some already existed, and continues to number from there.

3) File Extension – The extension to save each file with. Typically this should be the same as the actual chosen image format, but some users may choose to make this something different. It has no impact on the actual image format produced and can be anything.

4) Image Format – This is the actual image format the files are saved as. The possible choices are PNG, BMP, JPEG, TIFF, Multi-Page TIFF, PNM, and PDF. Typically PNG files are the best choice for color prints because of the wide variety of color formats they support. Also, PNG files use lossless compression, which means the images produced are exact images of the data and don't change appearance from their compression. JPEG files can sometimes be smaller than PNG files, but the compression is lossy, so sometimes they can appear blocky or fuzzy. TIFF images are a very popular format used in many document imaging systems. They are probably the best choice for black and white printed documents because of the small file sizes, lossless Group 4 compression, and universal acceptance. However, if there are ever any doubts, PNG is probably the best bet. PDF and Multi-Page TIFF are the only formats that put all the images into a single file (multi-page).

5) Color Format – This is the color format the image will be saved in (if the selected Image Format allows such a color format). The possible choices are Default, Binary (black and white), 8 and 16 Bit Gray Scale, and 8, 16, 24, and 32 Bit color. If set to default, the image color format will be whatever is set in the native driver settings (Figure 8). This option will only have a true effect if the native driver prints in color. If the native driver is set to print in black and white, this option will convert the image to the appropriate color format but cannot produce real colors or gray scale. Printing in color with the native driver and then converting to black and white is much slower, which is why the native driver has the option for black and white (great for documents). If you want any color format besides black and white, make sure to choose 'Color' from the native driver settings (Figure 8).

6) Abort On Recoverable Error – Sometimes the driver can detect certain types of errors with individual files and ‘recover’, or basically skip, those files and continue on to the next ones. If this flag is checked, the driver will attempt to do this. If not, the first error will cause the driver to abort and not continue.

7) Launch Default Viewer – If this flag is checked, after all the images are produced by the driver, the default Windows program will be launched to view/handle the images.

8) Pass All Files to Default Viewer – This flag works in conjunction with the ‘Launch Default Viewer’ flag. It only applies when the ‘Launch Default Viewer’ flag is checked. If this flag is checked, all files are passed as multiple separate parameters to the Windows program. Some programs may be able to handle these multiple parameters causing all the images to be opened/viewed instead of just the 1st image.

9) Show Settings Dialog – If this flag is checked, the dialog displayed in Figures 1-6 will be displayed at print time to allow a user to control and change settings on the fly. Sometimes, especially in businesses, administrators may not want this to prevent the users from making a mistake and causing an error. If that is the case, this flag should not be checked so the settings are automatically used and no dialog is presented.

10) Store Files Locally – If this flag is checked, the images will be saved permanently in the specified location, even after all functions are performed (such as emailing, launching default viewer, HTTP post, etc). If it is not checked, the images will be removed after being processed through all the functions.

11) Auto Number Files – If checked, duplicate file names are detected and assigned an auto number (incremented by 1). If unchecked, the file name is reused and any existing file will be overwritten.

12) Force Image Size – If checked, all images that are larger than the specified max image size will be scaled down. This is useful for making thumbnails or small, consistent image sizes.

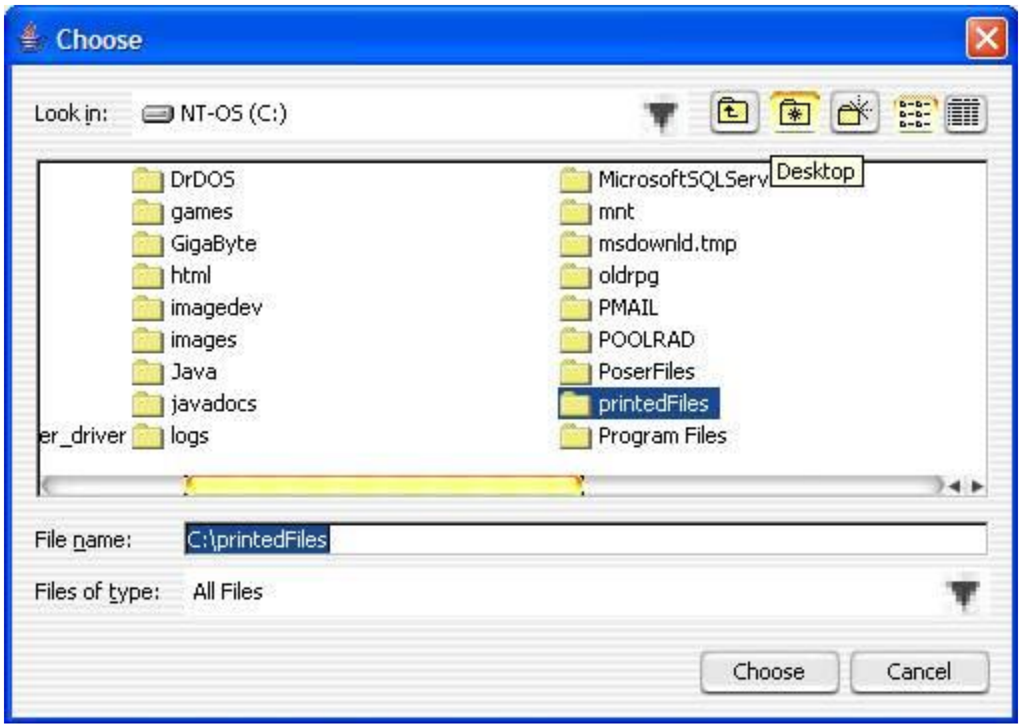
13) ZIP Files – If checked, all captured files will be batched up and compressed into a single ZIP file. This can be handy when emailing, archiving, or uploading many files and want to treat them as a single batch.

14) Max Image Size – This is the number of pixels to scale the images to if the 'Force Image Size' option is enabled. The engine will fit either the width or the height of the image to this value if the image is larger.

15) Console Program – An optional console program that may be executed and supplied the image file names as arguments. A window will be displayed that will show the standard output messages generated by the console program. The user can interact with the program by typing into standard input through the window if the program so requires. This is not to be confused with the default viewer program, which is a GUI based Windows program. Console programs are text based and usually legacy applications. Even though this is designed for console applications, a traditional Windowed application can be executed with this option as well (you can disable the extra console window from displaying via the XML file).

16) Ok – By clicking the 'Ok' button, the printer driver settings are saved. If the dialog is being displayed during a capture, 'Ok' also allows the capture to continue and produce images.

17) Cancel – By clicking the 'Cancel' button, the engine settings are discarded and stay as they were. If the dialog is being displayed during a capture, 'Cancel' causes the capture to abort.



File name: C:\printedFiles

FIGURE 1b

Email Settings

The engine has the optional ability to send an email to one or more people through SMTP. The captured files become attachments to the email. Figure 2 refers to these options.

1) SMTP Server – This is the IP Address or Domain Name of the SMTP server to use to send the email.

2) To Email Address – This is the list of email addresses to send the email to. The email addresses can be chained together with a comma as a delimiter. For example: 'info@everlastsoftware.com,sales@everlastsoftware.com'

3) From Email Address – This is the email address that the email appears to come from.

4) Subject – This is the subject/title of the email.

5) SMTP User Name – If the SMTP server requires authentication, this is the user name to utilize.

6) SMTP Password – If the SMTP server requires authentication, this is the password for the SMTP User Name supplied.

7) Message – This is the message, or body, of the email. Since the email is sent just like a normal email, this could be instructions, details about the captured images, questions, etc.

8) Notify After Send – If checked, a dialog will be displayed after an email has been sent to the SMTP server.

FTP Settings

One more mechanism in which the engine has the optional ability to send the captured files is by FTP (File Transfer Protocol). Figure 3 refers to these options.

1) FTP Server – This is the IP Address or Domain Name of the FTP server that will be receiving the files.

2) Port – The port in which the FTP server is listening on. This is almost always port 21.

3) Login Id – The login id to use to make the connection.

4) Password – The corresponding password for the login id.

5) Remote Directory – This is the location on the remote server where the files will be uploaded to.

6) Remote File Name – This is the filename prefix to be used for the files being uploaded. They will be assigned an incremental number in the same manner as the 'File Prefix' from Figure 1, only this time it will be on the remote server.

HTTP Settings

Besides being able to utilize FTP, the engine is also able to do HTTP (Hyper Text Transfer Protocol) posts. Figure 4 refers to these options.

1) Server URL – This is the full URL to a web server that can accept HTTP or HTTPS posts. The example in figure 4 is using HTTP. In order to do a secure upload, HTTPS may be used. To compare to figure 4, the URL would be 'https://www.myserver.com/ImageServlet' for secure transfer. The URL is parsed the standard way any HTTP URL is: protocol://domain_name:port/context.

Database Settings

The captured files can be automatically inserted into a database using JDBC or ODBC. Figure 5 refers to these options.

1) Database Engine Name – This is the name of the Database Engine to utilize. This allows for reuse of database specific settings (user, password, etc.) across multiple configurations if the same name is specified. If a different name is specified for multiple configurations, each will have it's own settings.

2) Database Table Name – The name of the table to insert into.

3) Database File Prefix Column (optional) – The name of the column to store the filename prefix.

4) Database File Extension Column (optional) – The name of the column to store the filename extension.

5) Database File Bytes Column – The name of the column to store the file's bytes.

6) Database Timestamp Column (optional) – The name of the column to store a timestamp of when the file was saved into the database.

7) Database JDBC Driver – This is the Java class name of the actual JDBC driver to use. On Windows, if ODBC is desired, always leave this set to 'sun.jdbc.odbc.JdbcOdbcDriver'. If a specific JDBC driver is wanted, simply plug in the class name and make sure the driver is in the classpath when the driver executes.

8) Database URL – The URL for the JDBC driver to connect to the database. On Windows, if ODBC is desired, the String must always start with 'jdbc:odbc:' and end with the data source name such as 'LocalServer'.

9) Database User – The user name to obtained connections to the database.

10)Database Password – The password for the user name to obtain connections to the database.

11)Connect – This button will automatically test the connection to the database using the supplied settings in the dialog. If any errors occur, they will be displayed in a dialog.

SOAP Settings

SOAP is becoming more and more popular in today's complex environments. The driver has the ability to send a SOAP message to a Web Service, HTTP Server, etc. with the files base 64 encoded inside as parameters. Figure 5b refers to these options.

1) HTTP Engine Name - This is the name of the HTTP Engine to utilize. This allows for reuse of server URL's across multiple configurations if the same name is specified. If a different name is specified for multiple configurations, each will have it's own URL

2) SOAP Action – The SOAP action for the message.

3) Method Name – The name of the method specified in the SOAP message.

4) Variable Name – The name of the attribute for the file bytes (base 64 encoded String).

5) File Variable Name – The optional name of the attribute for the file name (String).

6) ID Variable Name – This field is optional. It allows one to use the special tags in the 'ID Value' field in order to read environment variables and other settings to supply with the SOAP message. This is mainly used to uniquely identify a client sending the message.

7) ID Value – The actual value to send through the SOAP message. The tags should be utilized here if desired. The default is '%host.name%' which sends the host's name.

8) Server URL – The HTTP/HTTPS URL to the SOAP receiver. This will typically be a Web Service.

About

The about tab is simply for informational purposes about the product. It has no settings or options.

Displaying the Clipboard Monitor/Printer Driver Settings Dialog

As mentioned earlier, the printer driver settings dialog (again, this is the core to the clipboard monitor) will be displayed every time when a capture takes place if the 'Show Settings Dialog' checkbox from Figure 1 is checked. If it is not checked, you can still display the dialog by clicking 'Start->Programs->Everlast Software->ES Clipboard Monitor Engine Settings'. This will display the settings and even allow the re-checking of the 'Show Settings Dialog' checkbox so that it will be displayed, once again, at capture time.

Advanced Clipboard Monitor/Printer Driver Settings

There is an advanced way in which system administrators, power users, etc., may edit the settings without using the engine settings dialog. This is possible through modifying XML files. The clipboard monitor utilizes several engines (self-configuring code modules) which each have their own XML file for settings. They are all located in the installation directory area. More specifically, if the default installation directory was used: 'C:\Program Files\everlastsoftware\ES Clipboard Monitor Engine'.

The only files of importance (or usage) here are: ‘Clipboard Monitor Engine.xml’, ‘Printer Driver Engine.xml’, ‘Image Directory Import Engine.xml’, ‘File Transfer Protocol Engine.xml’, and ‘Email Engine.xml’.

Printer Driver Engine

This is the main engine utilized by the clipboard monitor engine.

1) ftpEngineName – The name of the FTP Engine to start. Whatever this is set to is what named XML file it will attempt to read from (minus the file extension).

2) emailEngineName – The name of the Email Engine to start. Again, determines which image settings XML file to use.

3) imageDirectoryImportEngineName – The name of the Image Directory Import Engine to start. Again, determines which image settings XML file to use.

4) readOnly – If set, the settings will never save, even if the print driver settings are changed and the user clicks the ‘Ok’ button. At capture time, the settings the user changes will take place for the capture, but will not be saved permanently in the XML file.

5) removeLocalFilesAfterPrint – Sometimes administrators don’t want the printed files to be saved locally on a user’s machine. Instead, the driver is more or less used for just transferring the images to someone else. If that is the case, this flag should be set to true.

6) showGUI – If true, the printer driver settings dialog will be displayed at capture time.

7) workingDirectory – This is the directory in which the Printer Driver Engine will be working. Temporary files, error log files, etc., will be dumped to this directory.

8) httpEngineName – The name of the HTTP Engine to start.

9) httpTransactionEventParameterName – The parameter name to be used for “before” and “after” events when sending a batch of images through a HTTP post. This helps the post handler to determine what files are in what batches in multi-threaded situations.

10) httpTransactionGUIDParameterName – The parameter name for the transaction id, again, to help determine a batch of images. A transaction id is always a unique string (a GUID) for every batch.

11) httpTransactionFileParameterName – The parameter name for the index number of the file being uploaded (a counter).

12) httpTransactionFilePathParameterName – The parameter name for the original file path and name being uploaded.

13) showHTTPTransactionResults – If this is set to true, a dialog will be displayed with any text returned from the HTTP post. If false, no dialog is displayed.

14) interfaceClass – The driver has the ability to call a Java class that is in the classpath if it implements a special interface. This allows developers to implement their own class to gain full control over the capture process. The class is called after the files have been produced by the driver. See the developer section for more details.

15) nativeLibraryInterface – A Windows developer also has the option to call a native DLL that has a special callback function. Just like the ‘interfaceClass’, the DLL will be called after the files have been produced by the driver.

16) useEmailEngine – If set to false, the email engine will not be used, even if it is configured.

17) useFTPEngine – If set to false, the ftp engine will not be used, even if it is configured.

18) useHTTPEngine - If set to false, the http engine will not be used, even if it is configured.

19) useDatabaseImportEngine - If set to false, the database import engine will not be used, even if it is configured.

20) useSOAPEngine – If set to false, the soap engine will not be used, even if it is configured.

21) ftpEngineTabVisible – If set to false, the ftp settings tab will be hidden when the settings dialog is displayed.

22) httpEngineTabVisible - If set to false, the http settings tab will be hidden when the settings dialog is displayed.

23) databaseImportEngineTabVisible - If set to false, the database settings tab will be hidden when the settings dialog is displayed.

24) soapEngineTabVisible - If set to false, the soap settings tab will be hidden when the settings dialog is displayed.

25) imageDirectoryImportEngineTabVisible - If set to false, the image settings tab will be hidden when the settings dialog is displayed.

26) engineSelectionTabVisible - If set to false, the engine selection settings tab will be hidden when the settings dialog is displayed.

27) showSystemTrayProgressIcon - If set to true, the engine will display a printer icon in the system tray during processing. If the mouse hovers over the icon, a brief tooltip message will be displayed describing what the engine is doing. This option only works if Java 6.0 or greater is installed.

28) debugMode - If set to true, the engine will log all messages produced instead of just errors.

Clipboard Monitor Engine

This is the engine that reads from the clipboard.

1) confirmClipboardCapture – If set to true (default), the user is presented with a dialog asking whether the image should be captured or not.

2) workingDirectory – This is the directory in which the Clipboard Monitor Engine will be working. Temporary files, error log files, etc., will be dumped to this directory.

Image Directory Import Engine

This is the engine that saves the actual images to the local computer and converts file formats (Figure 1).

1) abortOnError – This is the ‘Abort on Recoverable Error’ flag from Figure 1. If this is set to false, the engine will attempt to continue to the next file if a recoverable error occurs on an individual file.

2) autoRenameFileNameExtension – The extension to use for the files that are saved.

3) autoRenameFileNamePrefix – The prefix to use for the files that are saved.

4) encodeFormat – The actual image format to save the images as.

5) encodeImages – The native driver produces TIFF and BMP files. The Image Directory Import Engine then converts them to the specified format if they are not of those types. This can increase processing time by a small amount. If this flag is set to false, the encode format is ignored and the files are always left as their native format.

6) launchDefaultViewer – Once the images have been encoded, if this flag is set, the native Windows default program that handles the specified file extension will be launched.

7) outputDirectory – The directory to save the images to.

8) passAllFilesToDefaultViewer – If this flag is set to true, the filenames of all captured images will be passed to the default viewer instead of just the 1st filename. WARNING: This can cause the computer to run out of memory depending on how the program handles opening multiple files at once.

9) workingDirectory – This is the directory in which the Image Directory Import Engine will be working. Temporary files, error log files, etc., will be dumped to this directory.

10) storeFilesLocally - This flag indicates whether the files should be permanently stored at the specified location in figure 1 or just temporarily stored until all processing takes place (emails sent, FTP transfer, etc).

11) forceImageSize – If true, all images that are larger than the specified max image size will be scaled down. This is useful for making thumbnails or small, consistent image sizes.

12) maxImageSize – This is the number of pixels to scale the images to if the ‘forceImageSize’ flag is true. The engine will fit either the width or the height of the image to this value if the image is larger.

13) consoleProgram - An optional console program that may be executed and supplied the image file names as arguments. A window will be displayed that will show the standard output messages generated by the console program. The user can interact with the program by typing into standard input through the window if the program so requires. This is not to be confused with the default viewer program, which is a GUI based Windows program. Console programs are text based and usually legacy applications. Even though this is designed for console applications, a traditional Windowed application can be executed with this option as well (you can disable the extra console window from displaying via the 'showConsoleProgramOutput' setting).

14) showConsoleProgramOutput – If true, a window will be displayed as the console program executes, displaying the text being output and allowing user interaction. Set this flag to false if the program to be executed is a Window (GUI) application and not a console application.

15) binaryTIFFCompression – The compression type to be used inside binary TIFF files. The following options are available: 'CCITT T.6', 'CCITT T.4', 'CCITT RLE'. The default value is ‘CCITT T.6’.

16) colorTIFFCompression – The compression type to be used inside color TIFF files. The following options are available: ‘PackBits’, ‘LZW’, ‘JPEG’, ‘Deflate’, ‘ZLib’. The default value is ‘LZW’

17) jpeg2000Compression – The compression ratio to use for JPEG 2000 files. The value can range from 0.1 to 1.0. The lower the number the greater the compression (at the cost of quality). The default value is 0.5.

18) colorFormat – This is the color format the image will be saved in (if the selected Image Format allows such a color format). The possible choices are Default, Binary (black and white), 8 and 16 Bit Gray Scale, and 8, 16, 24, and 32 Bit color. If set to default, the image color format will be whatever was captured from the clipboard. This option will only have a true effect if the clipboard image is in color. If the clipboard had a black and white image, this option will convert the image to the appropriate color format but cannot produce real colors or gray scale. If the image was color to begin with, the actual colors will be converted.

19) packageInZipFile – If true, the captured file will be packaged into a single compressed ZIP file.

Email Engine

This is the engine that uses SMTP to send an email to one or more people with the captured images as attachments (Figure 2).

1) smtpServerName – This is the IP Address or Domain Name of the SMTP server.

2) smtpPort – The port that the SMTP server listens on.

3) defaultMessageBody – The message/body of the email.

4) defaultSendTo – The email addresses of the people to send the email to (recipients).

5) defaultSentFrom – The email address that the email should appear to come from.

6) defaultSubject – The subject of the email.

7) workingDirectory – This is the directory in which the Email Engine will be working. Temporary files, error log files, etc., will be dumped to this directory.

8) notifyAfterSend – If set to true, a dialog will be displayed after an email has been successfully sent.

8) smtpAuthentication – If set to true, the set user name and password will be used when calling the SMTP server.

9) smtpAuthenticationUserName – If the SMTP server requires authentication, this is the user name to utilize.

10) smtpAuthenticationPassword – If the SMTP server requires authentication, this is the password for the SMTP User Name supplied.

11) useTLS – If set to true, TLS will be used for encryption when sending emails.

File Transfer Protocol Engine

This is the engine that uploads the images to a FTP server (Figure 3).

1) loginId – The login id for the FTP server.

2) password – The password for the corresponding login id.

3) remotePort – The port in which the FTP server is listening on.

4) remoteDirectory - This is the location on the remote server where the files will be uploaded to. If the path contains subdirectories or fully qualified path, make sure to use the proper operating system path separator ('/' vs '\').

5) remoteFileName - This is the filename prefix to be used for the files being uploaded. They will be assigned an incremental number in the same manner as the 'File Prefix' from Figure 1, only this time it will be on the remote server.

6) remoteHostName - This is the IP Address or Domain Name of the FTP server that will be receiving the files.

7) workingDirectory – This is the directory in which the File Transfer Protocol Engine will be working. Temporary files, error log files, etc., will be dumped to this directory.

Hyper Text Transfer Protocol Engine

This is the engine that uploads the images to a web server via HTTP or HTTPS posts (Figure 4).

1) url – The full URL to the web server that will receive the post.

SOAP Engine

This is the engine that uploads the images to a web server via HTTP or HTTPS posts (Figure 5b).

1) httpEngineName - This is the name of the HTTP Engine to utilize. This allows for reuse of server URL's across multiple configurations if the same name is specified. If a different name is specified for multiple configurations, each will have it's own URL

2) soapAction – The SOAP action for the message.

3) methodName – The name of the method specified in the SOAP message.

4) variableName – The name of the attribute for the file bytes (base 64 encoded String).

5) fileVariableName – The optional name of the attribute for the file name (String).

6) idVariableName – This field is optional. It allows one to use the special tags in the 'ID Value' field in order to read environment variables and other settings to supply with the SOAP message. This is mainly used to uniquely identify a client sending the message.

7) idValue – The actual value to send through the SOAP message. The tags should be utilized here if desired. The default is '%host.name%' which sends the host's name.

Database Import Engine

This is the engine that saves the files into a database (Figure 5).

1) databaseEngineName – This is the name of the Database Engine to utilize. This allows for reuse of database specific settings (user, password, etc.) across multiple configurations if the same name is specified. If a different name is specified for multiple configurations, each will have it's own settings. This name must match the Database Engine XML file name (minus the '.xml').

2) databaseTableName - The name of the table to insert into.

3) databaseFilePrefixColumn – The name of the column to store the filename prefix.

4) databaseFileExtensionColumn – The name of the column to store the filename extension.

5) databaseFileBytesColumn – The name of the column to store the file's bytes.

6) databaseTimestampColumn – The name of the column to store a timestamp of when the file was saved into the database.

7) moveFiles – If set to true, the captured files will automatically be removed from the file system after being imported into the database.

Database Engine

This is the engine that stores user, password, etc., for the database (Figure 5). The 'databaseEngineName' in the Database Import Engine XML refer to one of these XML files.

1) databaseName – The name of the database to connect to. Not used by the driver, only for information.

2) databaseDriver – This is the Java class name of the actual JDBC driver to use. On Windows, if ODBC is desired, always leave this set to 'sun.jdbc.odbc.JdbcOdbcDriver'. If a specific JDBC driver is wanted, simply plug in the class name and make sure the driver is in the classpath when the driver executes.

3) databaseURL – The URL for the JDBC driver to connect to the database. On Windows, if ODBC is desired, the String must always start with 'jdbc:odbc:' and end with the data source name such as 'LocalServer'.

4) databaseUser – The user name to obtained connections to the database.

5) databasePassword – The password for the user name to obtain connections to the database.

6) maxConnectionPoolSize – The total maximum number of connections to utilize for connection pooling.

7) initialConnectionPoolSize – The initial number of connections to obtain from the database for use in connection pooling.

8) databaseShutdownStatement – If a special statement needs to be sent to shutdown a database after a capture, the SQL statement can be set here.

In order to use a custom JDBC database driver, the classes must be in the Java classpath, as mentioned in the 'databaseDriver' tag above. In order to do this, you will have to make sure to modify a few configuration files. Make sure to add the jar file to the 'JarFilePath' tag in the following files:

1) ES_CLIPBOARD_MONITOR.INI

2) ES_PRINTER_DRIVER_ENGINE_SETTINGS.INI

These files can be found in the installation directory.

Once the files have been updated, a reboot may be required in order to ensure the engine is unloaded from memory (if the Clipboard Monitor Engine is enabled) and picks up the new classpath.

Note: For all the mentioned XML files, only a subset of the information contained was mentioned. The other information is system settings that should not be changed. If the values are changed, the engines will most likely fail.

File and Output Directory Tags

The ES Clipboard Monitor Engine has the ability to specify special tags in the file prefixes, file extensions, and output directories (see Figure 1). This allows one to have great control over where files should go and what they should be named. These special tags are indicated by a value inside two '%' characters. For example, in order to have the file be named the current date, the following would be set in the file prefix field:

%month%-%day%-%year%

Those tags would create example filenames such as ’02-15-2005’, ’10-03-1995’, etc.

Regular characters can be mixed with tags as well. In order to have output such as ‘captured – 02-15-2005’ and ‘Captured – 10-03-1995’, the following prefix could be used:

Captured - %month%-%day%-%year%

One could do the following in order to create a subdirectory for each file, based on the current day, by placing the following in the file prefix field:

\%month%-%day%-%year%\captured_document

The following is a list of all the practical tags that may be used:

guid – Generates a random GUID (Globally Unique Identifier)
year – Four digit year
month – Two digit month
day – Two digit day
hour – Two digit hour
minute – Two digit minute
second – Two digit second
millisecond – Four digit millisecond
longtime – Single timestamp value to the millisecond
user.name – User name of the current user
host.name – The host name of the machine
host.address – The IP address of the host machine
user.home – The home directory for the current user
user.dir – The current user’s directory
java.io.tmpdir – The temp directory
launch.text – The text value selected by the user in the launcher
launch.engine – The engine name selected by the user in the launcher
xml.dir – The directory where the engine XML files are being utilized
log.dir – The directory where the main log XML files are being written