Kodak digital science™

Archive Writer Interface Software

Integrator's Guide

A-61057

1 Introduction

About this guide

Welcome to the *Kodak Digital Science* Document Archive Writer Interface Software (AWIS) Integrator's Guide. This guide is written for software developers who will be using the *Kodak Digital Science* Document Archive Writer in their applications. The Document Archive Writer takes electronic images in TIFF format and text files and writes them to 16 mm microfilm. This document discusses developing an application using the AWIS ActiveX control, and can be used as a reference for information about replacing the AWIS application with your own.

The AWIS software package that ships with the Archive Writer includes the following components:

- Archive Writer Interface Software
- ActiveX Control
- Archive Writer Object Layer (AWOL)
- Film Template server

AWIS communicates to the Archive Writer through the ActiveX control which in turn works through the AWOL to control the Archive Writer. The ActiveX control hides the details of using the AWOL, making the integration effort much easier. This guide's focus will be on the ActiveX control and the Film Template server.

Suggested prerequisites

Programmers who use the ActiveX control must be accomplished developers who have created applications for Microsoft Windows using Microsoft's development environments and ActiveX controls. An understanding of basic networking concepts, TCP/IP communication and utilities (FTP) is also necessary. The ActiveX control supports Microsoft Visual Basic 6.0 and Visual C++ 6.0.

Since the Document Archive Writer is a hybrid device we strongly recommend that all integrators be familiar with the use of microfilm, particularly the indexing schemes used for computer retrieval of microfilm images. For more information about microfilm and indexing, see the *Kodak Digital Science* Digital Document Archive System Installation Planning Guide, Version 2.0 (A-61055). Another good reference is the *Kodak Digital Science* Intelligent Microimage Scanner Software Image Quality and Training Package CD, Version 1.0 (A-61137).

Organization

In addition to this first chapter, this guide is organized as follows:

Chapter 2, *AWIS Application* — describes the AWIS software layers, functions, requirements, and the installation of all software layers.

Chapter 3, *AWIS ActiveX* — describes the functionality available through the ActiveX control and includes command information.

Chapter 4, *Film Template OLE Server* — includes a discussion of its purpose, interfaces, operation and database structure.

Chapter 5, *Examples* — discusses the proper sequence of events in an Archive Writer application. Includes a simple application and suggests methods to use in certain situations.

Chapter 6, *Status and Error Codes* — provides a complete listing of status and error codes for the ActiveX and the Archive Writer with explanations.

Appendix A, *Glossary* — provides terms and descriptions associated with AWIS.

Additional resources

The following publications are available:

- *Kodak Digital Science* Document Archive Writer User's Guide, Version 2.0 (A-61038)
- *Kodak Digital Science* Digital Document Archive System Installation Planning Guide, Version 2.0 (A-61055)
- *Kodak Digital Science* Archive Writer Interface Software User’s Guide, Version 2.0 (A-61056)

Getting technical support

If you experience problems with your system, contact the Kodak Response Center at one of the following numbers:

Within New York State:	1-800-462-6494
Outside New York State	1-800-822-1414
Alaska and Hawaii:	1-800-466-1414
Canada:	1-800-433-1414

For other countries, contact your local Kodak Service Support Center.

When contacting Technical Support, be prepared to supply the following information:

- The names and models of the computer and the Archive Writer you are using.
- Any memory-resident software installed, including memory managers, with version numbers.
- Any resident anti-virus software.
- Contents of the AWIS error log.
- Contents of the Archive Writer error log.
- Development tools used including version numbers.
- Any information about the error and action taken that can reproduce the problem you experienced. Include the exact text of any error messages you received.

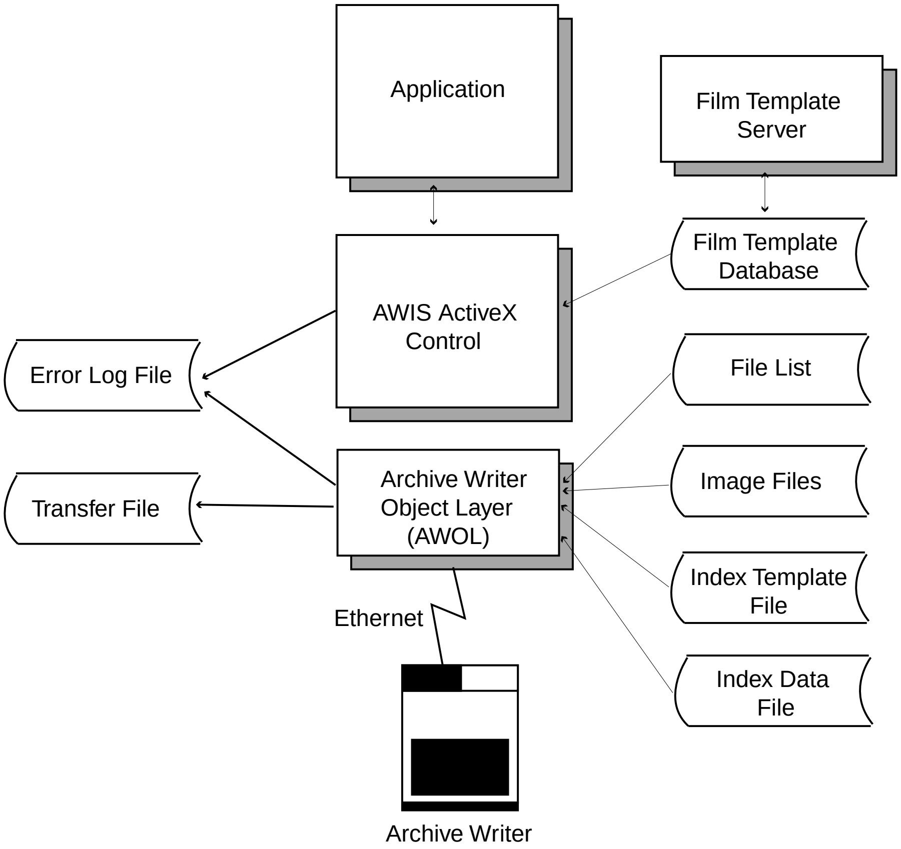
2 AWIS Application

Overview

The AWIS software that ships with the Digital Document Archive System consists of three software layers: the application layer, the OLE Custom Control layer (ActiveX), and the C++ Archive Writer Object Layer or AWOL. In an effort to ease further development, as much functionality as possible was implemented in the AWOL and the ActiveX layers.

Normally, the AWIS application provides user and administrative interfaces to the Archive Writer. The AWIS application uses the AWIS ActiveX control to communicate with the AWOL layer, which in turn communicates with the Archive Writer.

The AWOL receives commands from the ActiveX control and creates command files for the Archive Writer that controls its operation. These command files, along with the image files, are sent to the Archive Writer, which functions like a hard disk. The file system where the images and associated files reside can be either a local system connected to the PC running AWIS or a networked file system. We recommend keeping image files on a local disk to ensure seamless communication due to the vagaries of network congestion and design, which can cause problems with networked configurations. While we recognize that it is desirable to move files across a network to the Archive Writer, we suggest that you not perform this activity at the same time the images are being written. Any network problem can cause communication problems that can affect the job being run. The AWOL receives image log, status, and response information from the Archive Writer that can be stored in user-selected reporting and log files, or returned to the user interactively via the application.



Environment requirements

Development environment for using the ActiveX control — Visual Basic 6.0 and Visual C++ 6.0.

Run-time environment — Windows NT Version 4.0 Service Pack 4 or greater (workstation or server).

Hardware requirements

The minimum requirements for the AWIS PC:

- 133 MHz Pentium processor
- 32 MB RAM
- 1-2 gigabytes (GB) hard drive or enough space to hold the desired number of image files
- CD-ROM drive
- 28.8 baud modem (recommended)
- Monitor, keyboard, mouse
- Ethernet adapter for interfacing with the Archive Writer compatible with IEEE 802.3 Ethernet. Ethernet cards from *3 COM* (specifically the Etherlink 3) have been tested and are recommended.
- Network cabling from PC to Archive Writer

Communications

TCP/IP over Ethernet between the PC running the ActiveX and AWOL and the Archive Writer. The Archive Writer downloads its operating system or firmware from the AWIS host when it boots using FTP. After the Archive Writer boots, subsequent communication is done using TCP/IP sockets.

NOTE: Refer to the *Kodak Digital Science* Digital Document Archive System Installation Planning Guide, Version 2.0 (A-61055) for more details about the Archive Writer, the networking between AWIS and the Archive Writer and other system requirements.

Installation of ActiveX and AWOL

The AWIS setup program installs the OCX and other Kodak software components necessary for integrating the Archive Writer.

3 AWIS ActiveX

Overview

The ActiveX control is a software component that encapsulates the functions of the Archive Writer. The ActiveX uses two other software components from Kodak that are installed along with the ActiveX: The Film Template server and the AWOL.

The AWIS application uses the AWIS ActiveX control to communicate with the AWOL layer, which in turn communicates with the Archive Writer. AWIS provides user and administrative interfaces to the Archive Writer, separating these functions into two different sub applications.

As previously mentioned, the AWOL handles the actual communication with the Archive Writer. The Film Template server manipulates, stores and retrieves film templates, which contain the basic film settings used for common jobs, in a Microsoft Access database. By referring to a specific film template, the AWOL can communicate those settings to the Archive Writer.

There are two ways to interact with the ActiveX.

- You can set the properties programmatically, or
- you can set the properties via the user interface.

After familiarizing yourself with the ActiveX properties on the Property Pages in the user interface, you may want to set the properties programmatically.

The ActiveX communicates with the AWOL by calling the Archive Writer object. The parameter values of the ActiveX are updated by call back events. These events are sent back to the application through the OLE event mechanism.

The AWISOCX1_StatusEvent serves as the main notification method the AWOL uses to return information back to the ActiveX. Any problem encountered or a change in status at the AWOL level is communicated back to the application through this event. Each of the methods available in the ActiveX will typically return a certain set of events. The application should be equipped to handle and interpret the events raised.

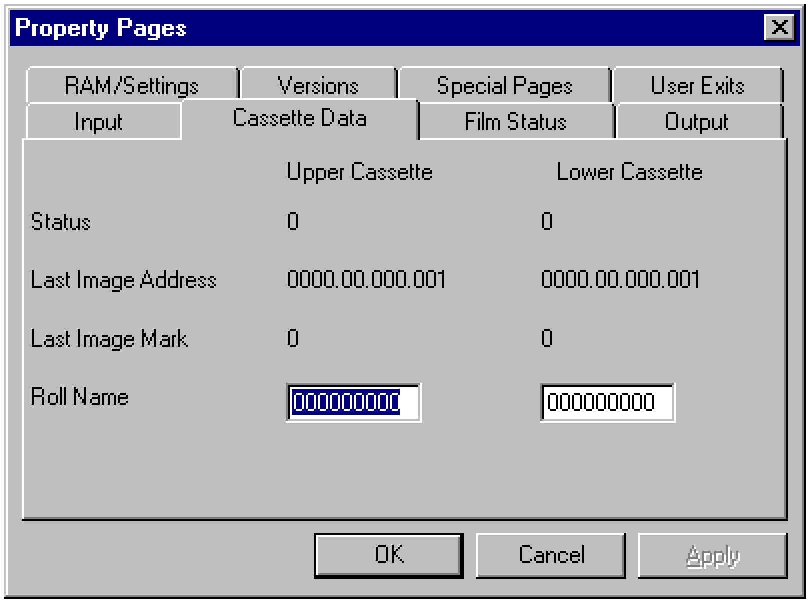
NOTE: You cannot run two instances of the ActiveX control successfully.

Property pages

The following dialog boxes appear in the design mode of the ActiveX. Property values are descriptions that appear on property pages. Read-only values have a gray background. Writable values have a border (ex. edit box).

Cassette Data tab

Following is an explanation of the fields on the Cassette Data tab.



	Upper Cassette	Lower Cassette
Status	0	0
Last Image Address	0000.00.000.001	0000.00.000.001
Last Image Mark	0	0
Roll Name	000000000	000000000

UpperCassetteStatus — an ASCII character from ‘0’ to ‘5’.

No Cassette Data	‘0’
New Cassette	‘1’
Cassette Out of Date	‘2’
New Roll of Film	‘3’
Previous Power Failure	‘4’
Cassette Data OK	‘5’
Default is No Cassette Data	‘0’

LowerCassetteStatus — same values and constraints as UpperCassetteStatus.

UpperLastImageAddress — a 15-byte string containing last image address written to film. The default is “0000.00.000.001”.

LowerLastImageAddress — same values and constraints as UpperLastImageAddress.

UpperLastImageMark — an ASCII character from ‘0’ to ‘3’.

Level Zero	‘0’
Level One	‘1’
Level Two	‘2’
Level Three	‘3’
Default is Level Zero	‘0’

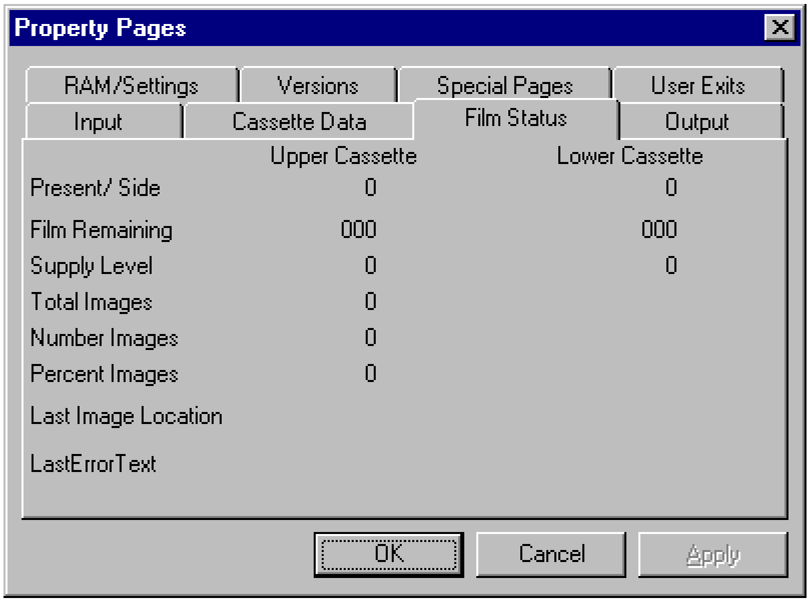
LowerLastImageMark — same values and constraints as UpperLastImageMark.

UpperRollName — an ASCII string (maximum of 8 bytes). This value must be numeric. The default is “00000000”.

LowerRollName — same values and constraints as UpperRollName.

Film Status tab

Following is an explanation of the fields on the Film Status tab.



	Upper Cassette	Lower Cassette
Present/ Side	0	0
Film Remaining	000	000
Supply Level	0	0
Total Images	0	
Number Images	0	
Percent Images	0	
Last Image Location		
LastErrorText		

UpperPresentSide — an ASCII character from ‘0’ - ‘5’.

Cassette Not Present	‘0’
Cassette on Side 1	‘1’
Cassette on Side 2	‘2’
Cassette Dead and Film Present	‘3’
Cassette Inserted Improperly	‘4’
Cassette Access Door is Open	‘5’

LowerPresentSide — same values and constraints as UpperPresentSide.

UpperFilmRemaining — a string (in inches or millimeters).

LowerFilmRemaining — same values as UpperFilmRemaining.

UpperSupplyLevel — a string from “0” to “10” (“0”: empty to “10”: full).

LowerSupplyLevel — same values as UpperSupplyLevel.

TotalImages — a long representing the total number of images in the job.

NumberImages — *see ImagesWritten*. The data it represents is contained in the property ImagesWritten.

ImagesWritten — a long representing the number of images written.

PercentImagesWritten — a short representing the percentage of images written.

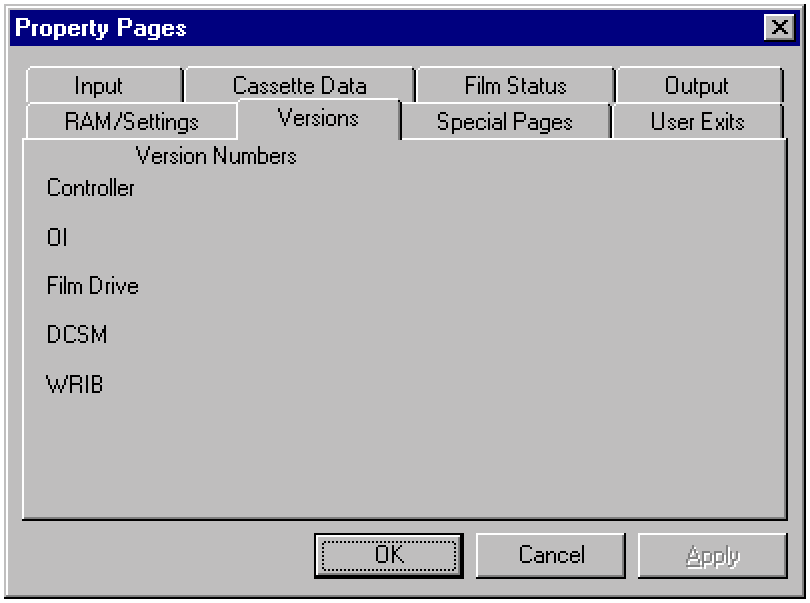
LastImageLocation — a string representing the name of the last file to be printed including its path.

LastErrorText — a string representing text that accompanies an error message.

Versions tab

Following is an explanation of the fields on the Versions tab.

The Versions tab displays the current Archive Writer parameter version numbers: Controller, OI (Operator Interface), Film Drive, DCSM and WRIB. These properties are informational only, they are not writable.



ControllerVersion — a string with a maximum of 11 bytes (abc.def.ghi).

OperatorInterfaceVersion — a string with a maximum of 11 bytes (abc.def.ghi).

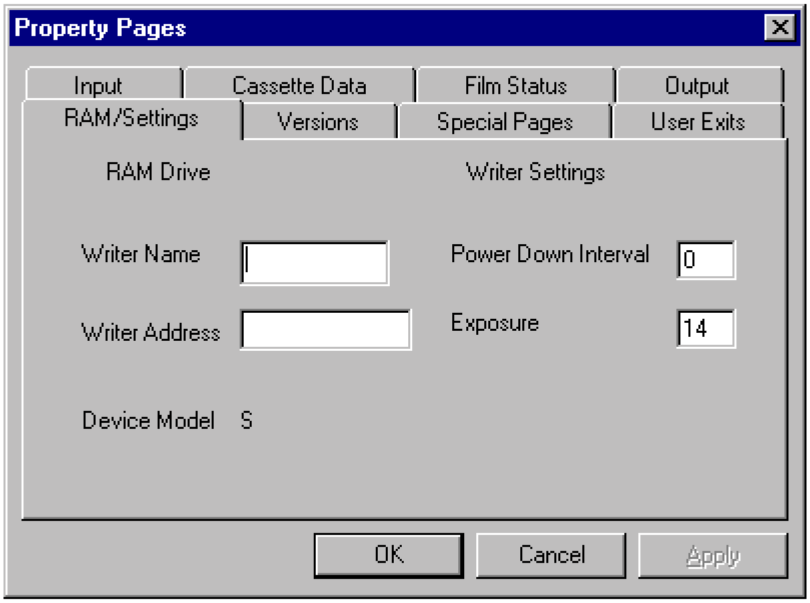
FilmDriveSoftwareVersion — a 4-digit ASCII string null terminated.

DCSMVersion — a 4-digit ASCII string null terminated.

WRIBVersion — a 4-digit ASCII string null terminated.

RAM/Settings tab

Following is an explanation of the fields on the RAM/Settings tab.



RAM Drive

Writer Name

Writer Address

Device Model S

Writer Settings

Power Down Interval 0

Exposure 14

WriterName — an ASCII string up to 18 characters in length.

WriterAddress — an ASCII string representing the IP address of the Archive Writer.

DeviceModel — an ASCII string representing the memory configuration of the Archive Writer. Either an **L** for Large or an **S** for Small is displayed. This is for information only and is not writable.

PowerDownInterval — an ASCII string representing the interval in minutes of inactivity after which the Archive Writer will automatically power itself down. To disable, enter 0 in this field.

Range: 0, 10 to 999

Default: 0

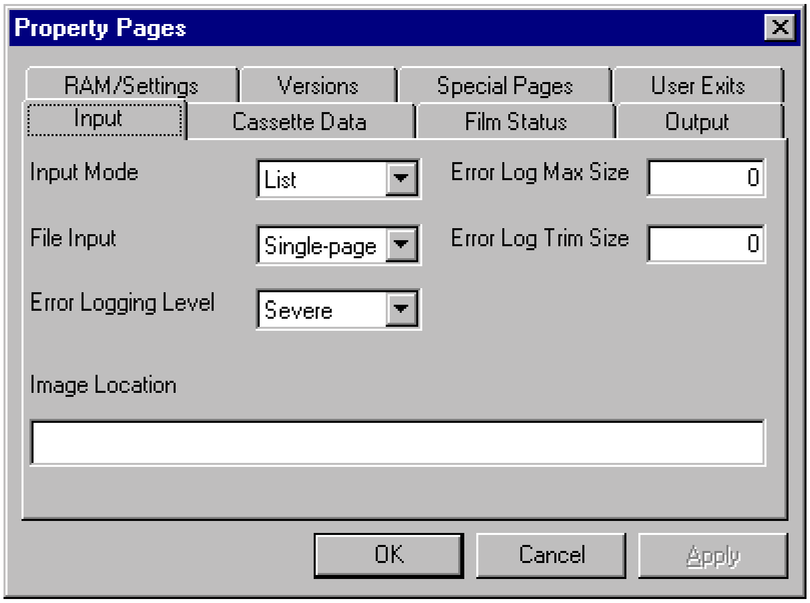
Exposure — an ASCII string that determines the print density of the microfilm. A larger value provides more density.

Range: 5 to 50

Default: 14

Input tab

Following is an explanation of the fields on the Input tab.



Input Mode: List

File Input: Single-page

Error Logging Level: Severe

Error Log Max Size: 0

Error Log Trim Size: 0

Image Location:

Types of Input — the basic input file format is single-page TIFF conforming to the TIFF V6.0 specification for bi-tonal images only. Within the file, the image data may be CCITT G3, G4, or uncompressed. Multi-page single-stripped TIFF files are supported. If the multi-page TIFF is too large to fit on the Archive Writer, the AWOL extracts all images from these files and creates single-page TIFF files to send to the Archive Writer.

The maximum size of a TIFF file sent to the Archive Writer is 1 megabyte. No image within the file can be larger than 1 megabyte.

Text files are converted to TIFF format in the AWOL and sent to the Archive Writer. Such a file can contain a maximum of 66 lines. Each line can contain a maximum of 80 characters and must be terminated with an additional carriage return and line feed (for a total of 82 characters). The file extension must be .txt.

InputMode — an ASCII character of ‘0’ or ‘1’.

List	‘0’
SortedBatch	‘1’
Default is List	‘0’

AWIS reads image files from a disk drive and sends them to the Archive Writer. There are two separate modes for feeding these files to AWIS: Batch mode and List mode.

- **Batch Mode** — in this mode, a path to a directory structure is specified. The control reads all image files in the directory and writes them to film. The resulting image address for each image is returned to the application. All files are sorted alphabetically by their path names.

  While a batch of files is processed, AWOL constructs the Transfer file, which contains the image file paths and corresponding image addresses. This file can be used by the host application to update its database or verify that intended image addresses were allocated correctly by AWOL.

  For multi-level indexing, the directory may contain subdirectories that indicate changes in indexing levels. For example, if two-level indexing with single-page files has been specified, the following directory structure would be used:

  <specified image directory path>\<subdir>\<filename>.TIF

- **List mode** — the *FileList* property contains the name of a file. This file contains a list of files to be written to film. Files are written to film in the same order as they appear in the list file. For multi-level indexing, the image address level is indicated by prefixing the filename with dashes. For example, a two-level list follows:

  ```
  c:\123000.TIF
  -c:\123010.TIF
  --c:\123011.TIF
  --c:\123012.TIF
  --c:\123013.TIF
  c:\124000.TIF
  ```

  In addition to the indicators in the examples above, specifying two dashes (--) results in an image being written with no image mark in the above example. This is known as a “Level Zero” image mark.

Both input modes can contain two-sided documents. For documents that contain fronts and backs, (meaning that duplex film is specified), the front of each document page is written to the A channel on the microfilm; the back side of each page is written to the B channel. If the application template specifies a film template for duplex film, image files will automatically be assigned to channel A and channel B as they are processed.

In the case of an uneven number of images, in single or multi-page, the AWOL software will send a *Print Remaining Image* command (number 39), which forces the Archive Writer to move back to the A channel.

The reader may notice the lack of coverage of a third input mode available in AWIS called “Poll Mode”. This feature allows for continuous polling of a directory for images. This feature is implemented in AWIS using the existing features of the OCX. It is not a feature implemented in the OCX itself. It is a feature implemented in AWIS using the existing features of the ActiveX control. For detailed information on Batch mode, List mode and list files see the *Kodak Digital Science* Digital Document Archive System Installation Planning Guide, Version 2.0 (A-61055).

FileInput — an ASCII character of ‘0’ or ‘1’.

Single-page	‘0’
Multi-page	‘1’
Default is Single-page	‘0’

ErrorLoggingLevel — an ASCII character from ‘0’ to ‘4’.

Event	‘0’
Severe	‘1’
Warning	‘2’
Informational	‘3’
Diagnostic	‘4’
Default is Event	‘0’

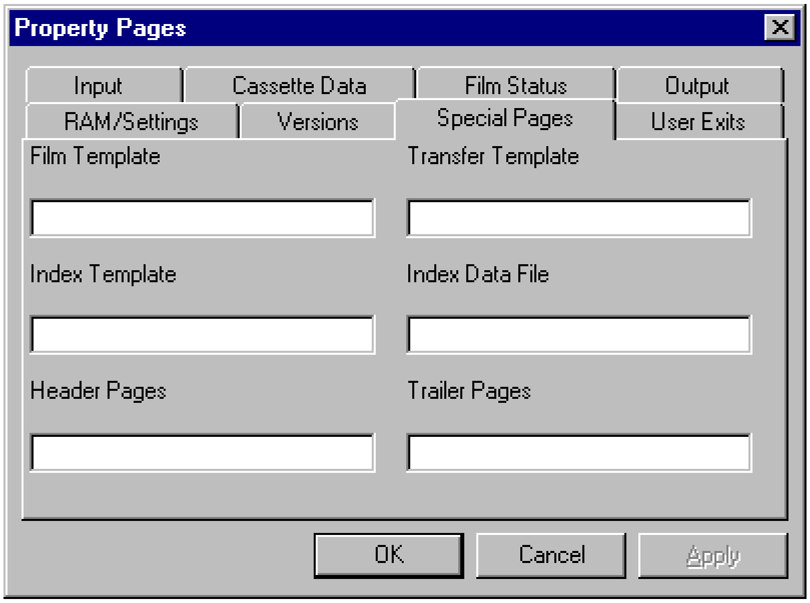
ImageLocation — a string representing the location of the directory/filelist containing the image files.

ErrorLogMaxSize — a long representing the maximum size of the error log.

ErrorLogTrimSize — a long representing the trim size of the error log. The default is 80% of the maximum size if no value is set.

Special Pages tab

Following is an explanation of the fields on the Special Pages tab.



FilmTemplate — a string representing the name of the film template. Commands are sent to the Archive Writer to set these values. The Writer must be connected when this property is set and the template must already exist.

The following errors are returned when this property is incorrectly set:

10000 - Film Template Dispatch Error
10001 - AWOL error sending the film template commands

IndexTemplate — this feature has not been implemented.

HeaderPages — a string representing the location (path) of the cover page files. Header and trailer pages are text or TIFF files written in alphabetical order to the beginning or end of the film at Image Address Level 0. When set programmatically, this property should be referenced as .coverpages.

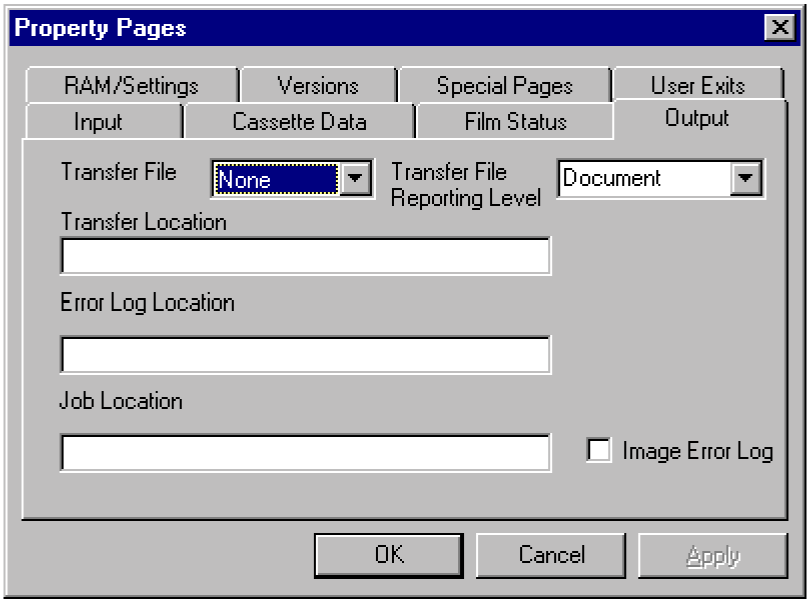
TransferTemplate — this feature has not been implemented.

IndexDataFile — a string representing location of the index data file used in creating the index pages or transfer file with appended index data.

TrailerPages — a string representing the location (path) of the trailer page files.

Output tab

Following is an explanation of the fields on the Output tab.



Transfer File: None

Transfer File Reporting Level: Document

Transfer Location:

Error Log Location:

Job Location:

□ Image Error Log

TransferFile — an ASCII character from ‘0’ to ‘2’.

None	‘0’
Standard	‘1’
Custom	‘2’ (enables the user exit)
Default is None	‘0’

The AWOL places transfer files and image error log files into the directory specified during installation of the AWIS application. AWIS creates subdirectories matching the chosen application template name through the application.

A transfer file can be generated by the AWOL while writing the images to film. The information in a transfer file (original file name, page number within the file, roll number, and image address delimited by tabs) can be used by external applications to update an external database. The creation of a transfer file is optional. The name of the transfer file matches the name of the roll as specified by the user. The file extension is “.xfr”.

TransferReportingLevel — an ASCII character of ‘0’ or ‘1’.

Document	‘0’
Document and Page	‘1’
Default is Document	‘0’

TransferLocation — a string representing the location of the transfer file including the path. Acceptable input includes: the full path name, the relative path name or no name at all. With a relative path name, the transfer file will be put in the working directory. With no name, the job directory ending in a backslash is passed to the user exit.

ErrorLogLocation — a string representing location of the error log.

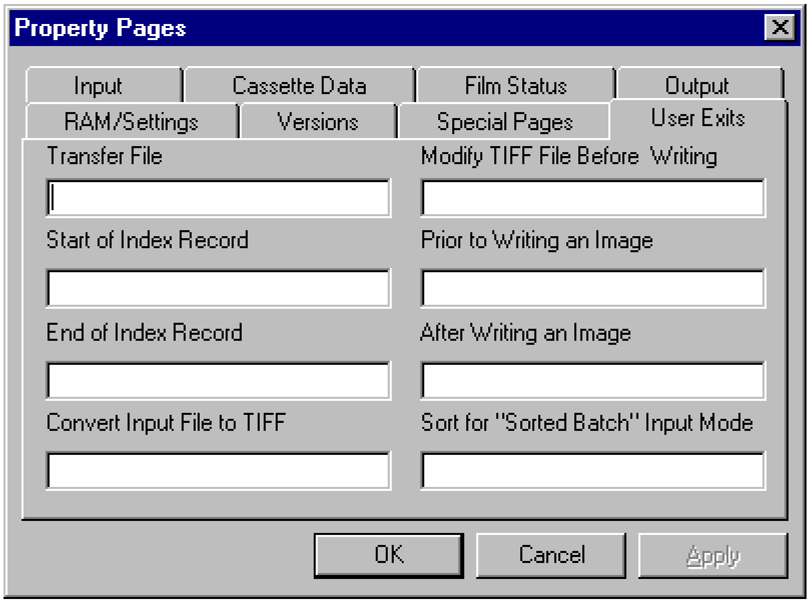
ImageErrorLog — this file contains a list of the images that failed to be written by the Archive Writer for a particular roll. Each line in this text file contains the image file name, a tab character, and error number. This file is converted to TIFF and written at the end of the film. AWOL determines which errors get logged to the image error log.

Boolean	On or Off
Default is	Off

JobLocation — an ASCII string representing a path to a location where AWOL can store temporary files.

User Exits tab

Following is an explanation of the fields on the User Exits tab.



TransferFile — a string representing the name of the function in AWOLEXIT.DLL. This function appends an index data file entry to a transfer file entry. The transfer file nExitNumber is 1.

StartIndexRecord (not implemented) — string representing name of DLL.

EndIndexRecord (not implemented) — string representing name of DLL.

ConvertToTIFF (not implemented) — string representing name of DLL.

ModifyTIFFFile (not implemented) — string representing name of DLL.

PriorImageWrite (not implemented) — string representing name of DLL.

AfterImageWrite (not implemented) — string representing name of DLL.

SortBatchExit (not implemented) — string representing name of DLL.

ActiveX methods

The following ActiveX methods are called by the AWIS application or another container. They return a 1 for failure or 0 for success.

- Initialize method
- SetupJob method
- RefreshProperties method
- BeginRoll method
- EndRoll method
- WriteImages method
- AdvanceFilm method
- ShutDown method

The examples given in this and other chapters of this guide are written in Visual Basic 6.0 and assume the existence of a containing application called "frmMain" which contains a single instance of the ActiveX control called "Awisocx1".

Initialize method

int Initialize(void)

Description

This method initializes logging for AWOL and calls the AWOL Initialize function, which sets the Writer name in AWOL and gets the parameters from the Archive Writer.

Parameters

None.

Properties Required

The following properties must be set to call this method:

- ErrorLog Max Size — size of Error Log.
- ErrorLog Trim Size — trim size of Error Log. Default is 80% of maximum.
- Error Logging Level — levels 1 through 4.

Severe	‘1’
Warning	‘2’
Informational	‘3’
Diagnostics	‘4’

- Error Log Location — location of Error Log.

Example

```
'first set the archive writer name and other properties
frmMain.Awisocx1.WriterName = cmbAWName.Text
frmMain.Awisocx1.ErrorLogMaxSize = ErrorLogMaxSize
frmMain.Awisocx1.ErrorLogTrimSize = ErrorLogTrimSize
frmMain.Awisocx1.ErrorLoggingLevel = ErrorLoggingLevel
 'then call the initialize method
Response = frmMain.Awisocx1.Initialize
```

Return Codes

0	Success
2011	Socket error
3014	Invalid writer drive
5057	Could not download language files

See Chapter 6, *Error and Status Codes* for all other errors.

SetupJob method

int SetupJob(void)

Description

Used to prepare a defined job for writing. This method calls the AWOL SetupJob function and sets the following properties on the Archive Writer.

Parameters

None.

Applicable Properties

The following properties should be set in the ActiveX before calling the function:

- ImageLocation — the full path to the image list file if the input mode specifies a list file. If batch mode is specified, contains the path to the directory structure containing the images. Do not include the trailing backslash.
- CoverPages — directory specification for the location of the cover page files (a.k.a. "Header pages"). These files are sorted by name and written to film during the BeginRoll operation. If this parameter is blank or the pointer is null, no header pages are written.
- TrailerPages — directory specification for the location of the trailer page files. These files are sorted by name and written to film during the EndRoll operation. If this parameter is blank or the pointer is null, no trailer pages are written.

- TransferFile:
  - If 0 is specified, no transfer file is created.
  - If 1 is specified, a transfer file is created where the file specification, roll ID, and image address are written to each line in the transfer file for each document written to film.
  - If 2 is specified, the data in the index data file is appended to the file specification, roll ID, and image address and written to the transfer file using a user exit. If a transfer template file is supplied in the pszXferTemplateFile parameter, the data from the index file is used along with the transfer template to format each record written to the transfer file.
- TransferLocation — directory specification for creation of transfer files. Transfer file names are constructed using the roll name specified in the call to the WriteImages function. If this parameter is blank or the pointer is null, no transfer file will be created.
- TransferTemplate — the file specification of the transfer template file. This file is used to construct entries in the transfer file. If this parameter is blank or the pointer is null, the ActiveX will not use an index data file and will not append external data to the transfer file.
- JobLocation — the path to a location where AWOL can store temporary files such as the temporary image error log pages for errors that occur during the writing operation.
- IndexDataFile — the file specification of the index data file. This file contains data supplied by the host application for creating transfer file entries. Each line in the file contains tab-delimited data for each document written to film. If this parameter is blank or the pointer is null, no index data file is used.
- FileInput
  - If set to 0, the Archive Writer will expect single-page TIFF files to be supplied by the host application.
  - If set to 1, the Archive Writer will expect multi-page TIFF files to be supplied. During the SetupJob operation, AWOL converts multi-page TIFF files to individual single-page TIFF files if they are too large to fit on the Writer. Each file has the same name as the original multi-page TIFF file, with an extension numbered from 000 upward. This setting allows more flexibility in dealing with the incoming stream of TIFFs.

- InputMode
  - If set to 0, indicates List file input mode.
  - If set to 1, indicates Batch input mode.

  As previously mentioned, Poll mode is not an input mode of the ActiveX control but a feature of AWIS.

- ImageErrorLog — sets the total images property on return.
  - If set to 0, indicates image error log pages are not generated during writing operations.
  - If set to 1, indicates image error log pages are generated and written to the roll directory under the appropriate application directory.

- TransferReportingLevel — sets the reporting level for the transfer file.
  - If set to 0, indicates document
  - If set to 1, indicates document and page

Example:

ReturnStatus = Awisocx1.SetupJob

Some Possible Return Codes

0	Success
5001	List file was not found
5007	No images in header page directory
5008	No images in trailer page directory
5009	No images in image directory
5013	No images in header or trailer page directories
5014	No images in header or image directories
5015	No images in trailer or image directories
5016	No images in trailer, header, or image directories
7002	BeginRoll operation in progress
7003	WriteImages operation in progress
7004	EndRoll operation in progress

See Chapter 6, *Error and Status Codes* for all other errors.

RefreshProperties method

int RefreshProperties(void)

Description

This method updates all the properties from the Archive Writer. It is often called after the ActiveX_StatusEvent returns a 0 (indicating there has been an update to the ActiveX' properties).

Parameters

None.

Example

Response = frmMain.Awisocx1.RefreshProperties

Return Codes

0	Success

See Chapter 6, *Error and Status Codes* for all other errors.

BeginRoll method

int BeginRoll(LPCTSTR RollName, LPCTSTR LastFileWritten, short RollNumber, LPCTSTR StartingIA, long lImagesWritten)

Description

This method is required for the first job on each new roll of film and should be used when IMC or header pages are enabled in the film template. This method sends the following parameters to the AWOL BeginRoll function, which is asynchronous. Since it spawns a thread in the ActiveX, BeginRoll returns control to the calling program immediately. The results of BeginRoll are reported through the ActiveX_StatusEvent.

Parameters

- RollName — the name of the roll. Must be numeric.
- LastFileWritten — the path to the last file written. This is blank or null if you have not written any files yet. Otherwise, it is the last value passed back.
- RollNumber — the number of the roll (starting with zero) within the current job. You must increment this for each new roll in the set of rolls that will contain all the images.
- StartingIA is the first image address to be used on the roll. It is needed to lay down the IMC code.
- LImagesWritten is the number (starting with -1) of images written successfully. This is -1 if you have not yet written any images.

- AWOL BeginRoll — sets the RollName on the cassette and in the IMC command, and sets the StartingIA in the IMC command. If IMC is selected, AWOL lays down the IMC code. If the LastFileWritten is null, AWOL starts at the beginning of the file list, otherwise it starts where it left off. LImagesWritten is used along with LastFileWritten to determine the last file written as file names can be duplicates.

Example

```
'Call Begin Roll Method
      ReturnStatus = frmMain.Awisocx1.BeginRoll(Roll_ID,
ImageFile, RollNumber, StartingImageAddress, -1)
```

Some Possible Return codes

0	Success
7002	BeginRoll operation already in progress
7003	WriteImages operation in progress
7004	EndRoll operation in progress

Values that can appear in the status event

0	Success. All values in the other data members are valid
3000	Failed to create command file on writer drive
3016	Timed out sending command to writer
3023	Cannot open transfer file
3030	Failed to get remaining film
3059	Failed to set the next transaction number
4001	Not enough film to write header pages
5000	End of writing images (normal operation)
5029	Failed to open image error log file
5030	Failed to write to image error log file
5031	Failed to write to transfer file
5034	Bad image
6002	Failed to rename the image log file

See Chapter 6, *Error and Status Codes* for all other errors.

EndRoll method

int EndRoll(LPCTSTR RollName, LPCTSTR LastFileWritten, short RollNumber, long lImagesWritten)

Description

This method is used to close a roll of film when there is no film or images left to write. It should also be used when trailer pages exist for a job. This method sends the following parameters to the AWOL EndRoll function, which is asynchronous. Since it spawns a thread in the ActiveX, EndRoll returns control to the calling program immediately. The results of EndRoll are reported through the ActiveX_StatusEvent.

Parameters

- RollName — the name of the roll. Must be numeric.
- LastFileWriten — the path to the last file written. This is blank or null if you have not written any files yet. Otherwise, it is the last value passed back.
- RollNumber — the number of the roll (starting with 1) within the current job. You must increment this for each new roll in the set of rolls that will contain all the images.
- LImagesWritten — the number (starting with -1) of the last image written successfully. This is -1 if you have not yet written any images.
- AWOL EndRoll — sets the RollName internally. If the LastFileWritten is null, AWOL starts at the beginning of the file list, otherwise it starts where it left off. LImagesWritten is used along with LastFileWritten to determine the last file written as file names can be duplicates.

Example

ReturnStatus = frmMain.Awisocx1.EndRoll(Roll_ID, ImageFile, RollNumber, -1)

Some Possible Return codes:

0	Success
7002	BeginRoll operation in progress
7003	WriteImages operation in progress
7004	EndRoll operation already in progress

Values that can appear in the status event

0	Success. All values in the other data members are valid.
3000	Failed to create command file on Writer drive
3016	Timed out sending command to Writer
3023	Cannot open transfer file
3030	Failed to get remaining film
3059	Failed to set the next transaction number
4001	Not enough film to write trailer pages
5000	End of writing images (normal operation)
5003	End of roll
5029	Failed to open image error log file
5030	Failed to write to image error log file
5031	Failed to write to transfer file
5034	Bad image
6002	Failed to rename the image log file

See Chapter 6, *Error and Status Codes* for all other errors.

WriteImages method

int WriteImages(LPCTSTR RollName, LPCTSTR LastFileWritten, short RollNumber, LPCTSTR Starting IA, long IimagesWritten, long IRestartMode)

Description

This method is used to write the main images of the job. It sends the following parameters to the AWOL WriteImages function, which is asynchronous. Since it spawns a thread in the ActiveX, WriteImages returns control to the calling program immediately. The results of WriteImages are reported through the ActiveX_StatusEvent.

Parameters

- RollName — the name of the roll. Must be numeric.
- LastFileWritten — the path to the last file written. This is blank or null if you have not written any files yet. Otherwise, it is the last value passed back.
- RollNumber — the number of the roll (starting with 1) within the current job. You must increment this for each new roll in the set of rolls that will contain all the images.
- StartingIA — the first IA to be used on the roll. It is needed to lay down the IMC code.
- LImagesWritten — the number (starting with zero) of the last image written successfully. This is zero if you have not yet written any images.

- LRestartMode — indicates how AWOL is to begin processing. Values are:

  1 New roll. If no image address is specified, it is calculated by the AWOL as the first address for the current film address setup specified in the film template. The number of images written is initialized to zero.

  2 A new roll has been started, but this is not the first roll in the roll set. The AWOL sets the image address to the first address based on the film template. The number of images written includes all images written to previous rolls.

  3 A new roll has been started, but this is not the first roll in the roll set. The AWOL calculates the image address as the next image address above the one supplied in pszImageAddress. The number of images written is reported as the total of all images, including those written to prior rolls.

  4 Restart writing using a new input source. The first image to write is determined from pszLastFileWritten and LImagesWritten. The address is determined from pszImageAddress. If no image address is supplied, the AWOL calculates a starting image based on the film template.

  5 Restart current job. The AWOL restarts the job based on current state. The starting image address is calculated as the next address after the one supplied in the pszImageAddress parameter. The AWOL picks up where it left off after the last WriteImages operation. This mode is useful for recovering from image writing errors.

This method starts the Archive Writer writing images and sets the RollName internally. It passes the name of the last file written. If the LastFileWritten is null, the AWOL starts at the beginning of the file list, otherwise it starts where it left off. ImagesWritten is used along with LastFileWritten to determine the last file written as file names can be duplicates.

Example

```
ReturnStatus = frmMain.Awisocx1.WriteImages(Roll_ID, _
ImageFile, RollNumber, StartingImageAddress, _ ImagesWritten,
RestartMode)
```

Return Codes

0	Success
7002	BeginRoll operation in progress
7003	WriteImages operation already in progress
7004	EndRoll operation in progress

Values that can appear in the status event

0	Success. All values in the other data members are valid
3000	Failed to create command file on Writer drive
3016	Timed out sending command to Writer
3023	Cannot open transfer file
3030	Failed to get remaining film
3059	Failed to set the next transaction number
5000	End of writing images (normal operation)
5003	End of roll
5029	Failed to open image error log file
5030	Failed to write to image error log file
5031	Failed to write to transfer file
5034	Bad image
6002	Failed to rename the image log file

See Chapter 6, *Error and Status Codes* for all other errors.

AdvanceFilm method **int AdvanceFilm(void)**

Description

This method advances the film by the default amount set in the film template or by the amount entered by the user.

Parameters

Example

ReturnValue = Awisocx1.AdvanceFilm(ValuePassedFromModal)

Return Codes

0	Success

Shutdown method **int Shutdown(shutDownLevel as Integer)**

Description

This method initiates a shutdown of the writing activity on the Archive Writer. It does not perform a power down.

Parameters

The only shutDownLevel that can be used is a 2.

Properties Required

None.

Example

ReturnStatus = AWISOCX1.shutdown(2)

Return Codes

0	Success

ActiveX events, errors and status reporting

As previously mentioned, the primary mechanism for receiving feedback from the AWOL is the ActiveX_StatusEvent. When this event returns a 0, it signifies that status is normal but property values may have changed. Performing a RefreshProperties will update the ActiveX to reflect the latest information. Some examples of the types of changes in information that may have caused the 0 event are as follows:

Remaining film — the "get remaining film" command is sent to the Archive Writer from the AWOL at intervals TBD. This information, in inches or millimeters and 10ths, is passed back to the ActiveX Film Status property page and a notification event is sent to the application.

Current image name — the current image name is sent back to the ActiveX from the AWOL. The ActiveX sends a notification event to the application.

Current image address — the current image address is sent back to the ActiveX Film Status property page from the AWOL and a notification event is sent to the application.

Percentage of input file collection written — the number of images in a collection is determined at the beginning of a session. As each file is written, a percentage of written files is determined in the AWOL and returned through the ActiveX to the application.

Number of images written — at the beginning of each writing session the image count is initialized to zero. Every time the Archive Writer halts for errors, user intervention, or for a new roll of film, the number of images written is returned from the AWOL through the ActiveX to the application.

Errors that occur on the Archive Writer will be returned to the AWOL, logged, and raised to the ActiveX. Archive Writer errors are logged to the location and filed in the ActiveX' properties with the error number, text, and level of error severity (0-none, 1-informational, 2-warning, 3-severe, 4-diagnostic). This logging function is at the AWOL level not at the ActiveX. To log ActiveX errors you need to trap the ActiveX' events and log those you recognize as errors.

See Chapter 5, *Sequence of Events and Code Examples,* for an example of error trapping using the ActiveX.StatusEvent.

For explanations of error and status codes you may encounter through the StatusEvent refer to Chapter 6 or the *Kodak Digital Science* Archive Writer Interface Software User's Guide, A-61056. Note that many of these particular states are possible using several methods. For example, the 5000 "error", indicates the Archive Writer finished writing images, can apply to the BeginRoll, WriteImages and EndRoll methods. Others, such as 3023, which indicate the transfer file cannot be opened, are specific to one method, in this case the WriteImages method.

General information about User Exits

The integrator can write custom code that is called at well-defined points from the AWOL module by supplying that code in a single, separate DLL. This allows the integrator to modify and extend the film writing capabilities implemented in the AWOL without modifying the AWOL.

The DLL should contain a set of functions that can be accessed by name. AWIS or other modules that use the AWOL DLL specify which function to call at appropriate points in the AWOL by sending the name of the function to be called to the AWOL during its' initialization. The name of the exit function DLL is AWOLEXIT.DLL. If this DLL is not on the system, user exit functionality will not be available. This DLL is placed in the WINNT system directory.

The AWOL *SetExit(long nExitNumber,char *pszFunctionName)* function is called when the property to specify the name of the user exit function is set. *nExitNumber* is an ordinal specifying which user exit name is to be set. These constants are defined in the AWOLEXIT.H file. This function can be called any time. It is usually called before any other functions in AWOL are called by the application.

When this function is called, the AWOL uses the GetProcAddress function to verify the specified function name exists in AWOLEXIT.DLL. If the function exists, the AWOL stores the procedure address in an internal array, which is used later to call the actual function in the exit DLL.

The maximum length of *pszFunctionName* is 32 characters, as defined by the MAXEXITNAMELENGTH constant in AWOLEXIT.H.

Return values from the *SetExit* function are:

0	Success
8000	No user exit DLL (AWOLEXIT.DLL)
8001	User exit name not found in AWOLEXIT.DLL

Transfer file User Exit

Currently this is the only exit function implemented in the ActiveX. It is called by the AWOL at three points in its execution:

- At the beginning of the WriteImages function in the AWOL.
- Each time the AWOL determines that an image has been successfully written to microfilm.
- At the end of the WriteImages function in the AWOL.

The prototype for the function (found in AWOLEXIT.H) is:

```
long ImageWritten(int nState,
        int nReason,
        char *pszXferFile,
        char *pszXferTemplateFile,
        char *pszIndexDataFile,
        char *pszImageFileName,
        long nImageFileNumber,
        long nAddressLevel,
        char *pszImageAddress,
        int nPageNumber,
        int lDocNumber,
        char *pszRollName,
        char *pszRemainingFilm,
        byte nReportingLevel,
        int nFilmingMode,
        int nIndexFormat,
        char *pszErrorMessage);
```

- **nState** — specifies the current state of the AWOL:

  1 – Starting up — the transfer file (pszXferFile) should be opened for appending and the file position should be set to end of file. The index data file (pszIndexDataFile) should be opened and the file position set using pszImageFileName in conjunction with nImageFileNumber to locate the correct starting position in the file. The number of the line in the file should be consistent with the file name. If a transfer template file is used, the exit function should open this file and read its contents at this point.

  The exit function can perform any other required initialization steps at this time. The length of time required to perform these operations is not critical.

  2 – Running — the transfer file template (if used), and create an entry in the transfer file.

  Operations performed by the exit function in this mode should execute as quickly and efficiently as possible. Lengthy operations reduce the overall performance of the AWOL.

  3 – Shutting down — the exit function should perform any cleanup operations.

- **nReason** — gives further information about the current state as indicated in the nState parameter.

  1 – Start of job — the first image written will be the first image on the first roll of film in the job or a new job has been started on the same roll with a new input source.

  2 – Continuing job — the job was stopped and is being restarted. The first image written is determined from the *pszImageFileName* and *nImageFile* parameters.

  3 — shutting down due to image writing failure.

  4 — shutting down because end of roll has been reached.

  5 — pause job. User pressed the STOP button.

  6 — all images in job have been written.

- **pszXferFile** — the full path to the transfer file. If no transfer file exists, the parameter is null or points to a zero-length string.
- **pszXferTemplateFile** — path to the transfer template file. The format of this file is defined by the user exit function. This file can be used by the exit function to combine the file data with index data to create entries in the transfer file. If no template file exists, this parameter is null or points to a zero-length string.
- **pszIndexDataFile** — full path to the index data file. If no index data file is to be used, this parameter is null or points to a zero-length string.
- **pszImageFileName** — the name of the last image file written. This can be used by the exit function to verify the correct index data item in the index data file fetched. When starting a new job, this variable has the first image in the job.
- **nImageFileNumber** — the number of the image file within the entire list of images that has just been written to film. This number starts at 1 for the first file in the image list. The image list does not include header or trailer pages; only the images supplied in the main image list. When starting a new job, this number will be 1.
- **NAddressLevel** — the address level of the current image.
- **pszImageAddress** — the image address of the last image written.
- **nPageNumber** — the page number of the image that has just been written. This number is always 1 for single level jobs. For two level jobs, this number is the number of the file in the document.
- **pszImageFileName** — is the same for all images in the file and nPageNumber is the number of the image within the file.

- **lDocNumber** — the number of the last written document level image. In single level jobs each page is considered a document.
- **pszRollName** — the name of the current roll.
- **pszRemainingFilm** — the amount of remaining film, in the format n.nu where u is ‘F’ for feet and ‘M’ for meters. The number n.n may or may not contain a decimal point. The remaining film can be interpreted by removing the unit indicator from the end of the string and converting the remaining string to a decimal number.
- **nReportingLevel** — what level of entries should appear in the transfer file. This is a byte with a bit set for each level to be recorded. These values can be logically OR’ed:

0x01	Record level 0 images
0x02	Record level 1 images
0x04	Record level 2 images
0x08	Record level 3 images

- **int nFilmingMode** — simplex or duplex filming mode.

1	simplex
2	duplex

- **nIndexFormat** — level of film.

0	zero level
1	single level
2	two level

- **pszErrorMessage** — a string of up to 1024 characters of message text describing an error condition.

Return values
These values should be returned to AWOL by the exit function and are defined in AWOLEXIT.H:

0 Success. The AWOL continues processing.
1 Warning. The AWOL logs message and continues processing.
2 Error. The AWOL closes all files and stops writing.

The AWOL logs errors to the AWIS Error Log file. The AWOL formats message text for each of the errors listed and includes the parameters passed to the exit function listed in the error message. The pszErrorMessage text returned to the AWOL is included in the error message. If a value of 2 is returned to the AWOL, it stops writing images and calls the exit with the state set to *shutting down*.

4 Film Template OLE Server

Overview

The Film Template Server is a 32-bit OLE Custom Control that can be used with Microsoft Visual Studio 6 on Windows NT 4.0. This component provides an interface to a Film Template database that can be used for both film writing and image retrieval applications. A film template assigns a name to a set of film writing or film retrieval parameters. The Film Template Server allows for easy storage and retrieval of film templates. It is an in-process server that is usually installed in the Program Files\Common Files\OleSrv directory as filmtemp.dll.

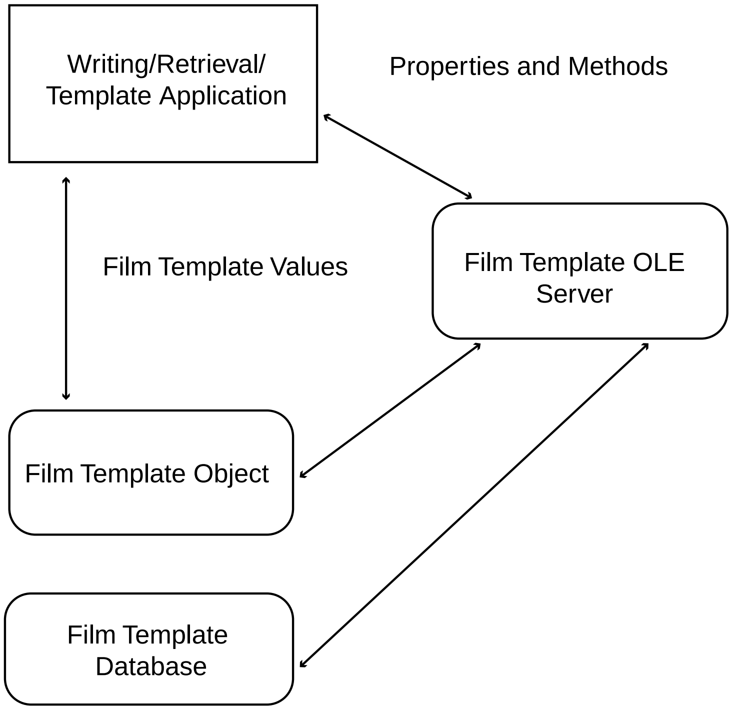
The Film Template OLE Server is accessed and manipulated with the Film Template Object, the C++ Class that contains all the properties that apply to the OLE Server as well as the functionality that is represented in the OLE Server methods.

The Film Template OLE Server performs the following tasks:

- Maintains film writing parameter templates in a Microsoft Access database for access by an application.
- Provides a GUI interface that allows users to create new templates and modify templates for film writing or reading.

The Film Template OLE Server can be used by retrieving or saving templates using Film Template methods at run-time or by invoking the Property Page dialog box of the Film Template OLE Server. In either case, you can examine or change the individual properties of the templates. The properties always represent the settings for the template named in the Template Name field.

After the Film Template Server retrieves the property data from the database, the data in that particular film template is available to the application via the Film Template Object’s properties. At run-time, the operations of the film template are accomplished through the OLE Server methods.



Film Template properties

The property pages of the Film Template OLE Server provide an interface to set the following film writing/retrieving parameters that are stored in the Film Template database. Descriptions of the database fields that represent these properties can be found later in this chapter.

- **FilmTemplateDatabasePath (read-only)** — displays the path to the Film Template database.
- **FilmTemplateName** — contains the name of the current film template.
- **CassetteUsage** — the minimum number of cassettes required by the Archive Writer to perform filming. If two cassettes are required, two originals will be written for each source image.
- **FilmMeasurementUnits** — determines the measurement system: English (inches) or Metric (millimeters). The default is English.
- **FilmLeaderLength** — contains the desired length of the film leader (from 36 to 120 inches / 914 to 3048 millimeters). The default is 36 inches/914 millimeters.
- **FilmAdvance** — contains the desired length of the film advance (from 1 to 99 inches / 26 to 2515 millimeters). The default is 1 inch / 26 millimeters.
- **Filmmode** — select either Simplex or Duplex.

- **EnableIMC** — determines the settings for the Image Management Code on different Kodak devices: Disable IMC, Enable for IL70 code, Enable for RIM2000 code, or Random Batch, which prints random batch for non-sequential addresses when IMC is disabled. The default is Disable IMC.
- **SearchProgramNumber** — contains the number of the search program. The range is 0 to 31; the default is 18.
- **SpliceDefinition** — contains the splice definition: Ignore Splices (default), Count Splice as Level 1 image mark, Count Splice as Level 2 image mark, or Count Splice as Level 3 image mark.
- **DuplexFrontChannel** — defines the duplex front location: select either Channel A (default) or Channel B.
- **ImageMarkAuthor** — defines the image mark author. Select either Kodak (default) or Other image marks.
- **ImagePolarityReductionRatio** — use a predefined option (24X, 40X, 50X) or Custom. If you select Custom, the Scaling Factor option becomes enabled and you can enter a Scaling Factor value (0-99).
- **ScalingFactor** — represents the scaling factor as a number (i.e., 32 would be a 32X reduction ratio). Used for Custom reduction ratios.
- **OffsetAddressing** — available with Level 2 indexing only. When enabled, page level addressing is ignored.
- **IndexFormat** — three options are available: No Indexing, Single Level and Two Level. For more information on index formats, refer to the *Kodak Digital Science* Digital Document Archive System Installation Planning Guide, (A-61055).

  NOTE: Three Level indexing is not currently supported.
- **EnableImageBorder** — when this option is enabled, a thin border is placed around all images. This helps identify the boundaries on positive images.
- **EnableFrameAnnotation** — allows annotation information to be written to film. This is currently not supported.
- **LevelRule** — rules that determine the next image mark or indexing level to go to from the specified level.
- **ImageMarks** — determines whether small, medium, or large image marks will appear on the film.

- **ImageAddressFieldWidths** — determines the overall length of the image address. The total length of an image address cannot exceed 12 characters. The total of each level field cannot exceed 9 characters. The field width is enabled or disabled depending on the Index Format you selected. If you select Single Level Index format, only Level 1 field width is enabled. If you select Two Level Index format, Level 1 and Level 2, are enabled.
- **ImageOrientation** — select either **Portrait** (image orientation is the shape of a conventional portrait, where height is greater than width) or **Landscape** (image orientation is the shape of a conventional landscape painting, where width is greater than height). Portrait is the default.
- **ImagePolarity** — select either **Positive** or **Negative**.
  - **Positive** — if the original image is a clear background with black text, selecting Positive will produce a clear background with black text.
  - **Negative** — if the original image is a clear background with black text, selecting Negative will produce a black background with clear text.
- **ImageScaling** — select either **No scaling** or **Automatic scaling**, which is applied to an image before it is written to film:
  - **No scaling**: the reduction ratio and scaling factor is ignored.
  - **Automatic scaling**: the destination image on film is reduced by the factor selected in the Reduction Ratio or Scaling Factor controls, regardless of resolution.
- **Inter-Document Gap** — a value between 0.6 mm and 5.0 mm to set the required document gap.

Methods

You can use the following methods to create, get, save and view film templates.

CreateTemplate method

int CreateTemplate(String TemplateName, String BaseTemplateName)

Description

Creates a new template based on an existing or default template. If the BaseTemplateName is null, the Property Page dialog box appears with the default template values already loaded.

Parameters

TemplateName — a name for the new template.

BaseTemplateName — an existing template the new one will be based on.

Example

```
Response =
FilmOBJ.CreateTemplate(txtNewFilmTemplateName.Text,_
txtBaseFilmTemplateName.Text)
```

GetTemplate method

int GetTemplate(String TemplateName)

Description

Retrieves the named template from the template database. If the template does not exist, a dialog box appears, saying that the template cannot be found. Select a template name by clicking on a name in the Film Template Name drop-down list.

Paramters

TemplateName — the name of an existing film template.

Example

```
Response = FilmOBJ.GetTemplate(txtFilmTemplateName.Text)
```

ViewTemplate method

int ViewTemplate(String TemplateName, String ViewOnly)

Description

Presents the Property Page dialog box so you can view or edit the properties. Once in View mode, you can retrieve, create, or edit additional templates.

Parameters

ViewOnly = True (properties cannot be changed)

ViewOnly = False (properties can be changed)

Properties Required

Example:

RetVal = FilmOBJ.ViewTemplate(txtFilmTemplateName.Text, False)

Events

Not implemented. As a result, the Film Template OLE Server does not communicate back to the calling application.

Errors result in a nil object exception. Common examples are when the database is corrupted or non-existent. Look for an error statement in between creating the Film Template object and setting the default template.

Film Template database design

The Film Template database resides in the \application directory\db directory as filmtemp.mdb. The following table shows the fields AWIS uses and their constraints:

Field Name	**Type**	**Description**
FilmTemplateName	Text	Name of Template
FilmModes	Num	0 = Simplex, 1 = Duplex
Scaling	Num	0 = No scaling, 1 = Auto, 2 = Fixed
ReductionRatio	Num	0 = Custom, 1 = 24X, 2 = 40X, 3 = 50X
ImagePolarity	Num	0 =Positive, 1 = Negative
FilmMeasurementUnits	Num	0 = English, 1 = Metric
FilmLeaderLength	Text	36 to 120
ScalingFactor1	Text	
FilmAdvance	Text	001 to 099
IndexFormat	Num	0 = none, 1 = single level, 2 = two level, 3 = three level
LevelRule1	Num	0 = level 0, 1 = level 1, 2 = level 2, 3 = level 3
LevelRule2	Num	0 = level 0, 1 = level 1, 2 = level 2, 3 = level 3
LevelRule3	Num	0 = level 0, 1 = level 1, 2 = level 2, 3 = level 3
LevelRule0	Num	0 = level 0, 1 = level 1, 2 = level 2, 3 = level 3
FieldWidth1	Text	see rule
FieldWidth2	Text	see rule
FieldWidth3	Text	see rule
FieldWidthFixed	Text	see rule
EnableWriteAnnotation	Yes/No	0 = disable, 1 = enable
FrameAnnotationOrientation	Text	0 = comic, 1 = cine, 2 = comic-180, 3 = cine-180
FrameAnnotationPosition	Text	
EnableIMC	Num	0 = disable, 1 = enable IL70, 2 = enable RIM2000, 3 = Random Batch
SearchProgramNumber	Text	0 to 31
Splicedefinition	Num	0 = ignore, 1 = count level one, 2 = count level two, 3 = count level three
DuplexFrontChanel	Num	0 = channel A, 1 = channel B
ImageMarkSmall	Yes/No	
ImageMarkMedium	Yes/No	
ImageMarkLarge	Yes/No	
ImageMarkAuthor	Num	0 = Kodak, 1 = other
ImageOrientation	Num	0 = Portrait, 1 = Landscape
RetrievalReductionRatio	Text	0 to 99
OffsetAddressing	Yes/No Oinary	
NumberCassettes	Text	0 = write one original, 1 = write two originals
DocumentGap	Text	0.6 to 5.0 Default is 0.7
ImageBorder	Num	0 = No, 1 = Yes

5 Sequence of Events and Code Examples

The first priority of your development effort will be to effectively control the ActiveX and write images. The order in which the various methods are called is important. The calling order of the ActiveX methods is described below.

Event sequence

The following shows the basic sequence of events that you should follow to create a simple application.

1. Fill in the ActiveX property values that are appropriate for your application except for the Film Template name. Make sure to include the required properties covered in Chapter 4 for the Initialize method. Create the film template object and set it to the default using the FilmTemplate.GetTemplate method. If this method is called for a template that no longer exists or is invalid, a run-time error will occur. Although obvious, it is worth stating that the program should catch such errors with an On Error statement and handle them smoothly.
2. Set the ActiveX WriterName property to the name of the Writer and call the Initialize method to create a connection with the Writer. The ActiveX verifies that the Writer name is associated with a valid Archive Writer. The return value indicates whether the connection to the Archive Writer was successful or not. The ActiveX sends a command to the Archive Writer to request information about its current setup and the status of the film cassettes. The Archive Writer passes the setup information back to the ActiveX and the setup and status information is available to the application through the properties of the ActiveX.
3. Set the Film template property in the ActiveX to the name of the film template to be used to configure the Archive Writer. Set any other remaining ActiveX properties at this time. Using the Film Template name, the ActiveX requests the Film Template OLE Server to read the corresponding record from the Film Template database. The film template information is then available to the ActiveX through the properties of the Film Template object. For informational purposes, the diagram also shows that the Film Template properties can be accessed directly by the application through the Film Template OLE Server if necessary.
4. Initiate the ActiveX.SetupJob method. All of the images to be written to film including the images in the cover and trailer pages directories, if specified, are checked to make sure they can be found, read and are not larger than the maximum size for an image file. The return value indicates whether the method was successful or if an error occurred.

It is important to note that SetupJob will not return control to the calling program until all images have been scanned. The amount of time it takes to scan images depends on the number, size, type (single- or multi-page) and location (local or networked) of the images.

In normal operations, the SetupJob method is only performed once during the writing of a set of images to film. However, it is possible to design an application that can terminate the writing process (using the ActiveX.Shutdown method) before all of the images have been written and restart it at a later time, starting at the point where it was terminated. In this scenario, if the application is not terminated, it is not necessary to perform the SetupJob method again before restarting the image writing process. However, if the application is terminated, it is necessary to perform the SetupJob method again prior to restarting the writing process. In the case where a job is stopped and restarted, if any of the images are removed or renamed, an error will result.

5. Initiate the BeginRoll method if header pages or Image Management Code have been enabled in the selected film template. If the ActiveX COVERPAGES property contains a header pages directory pathname, the images in that directory are sorted by their filename and written to film. Immediately after each image is written, the callback event, ActiveX_StatusEvent, occurs returning a value in the retStatus parameter. If it contains a zero, the image was successfully written. Otherwise, the value must be examined to determine what action to take. When the BeginRoll method completes, ‘5000’ is returned in the retStatus parameter.

   NOTE: The BeginRoll method must be initiated for the first job on a roll even if no header pages or Image Management Code are to be written to film. This must be done so the roll name is copied to the non-volatile memory of the film cassette(s).

6. Initiate the ActiveX WriteImages method. Each image identified through the ActiveX IMAGELOCATION property is copied to the Archive Writer and written to film. Immediately after each image is written, the callback event, ActiveX_StatusEvent, occurs returning a value in the retStatus parameter. If it contains a zero, the image was successfully written. Otherwise, the value returned must be examined to determine what action to take. After all images have been written, the ActiveX_StatusEvent returns ‘5000’ in the retStatus parameter.

7. Initiate the ActiveX EndRoll method. If the ActiveX TRAILERPAGES property contains a trailer pages directory pathname, the images in that directory are sorted by their filename and written to film. Immediately after each image is written, the callback event, ActiveX_StatusEvent, occurs returning a value in the retStatus parameter. If it contains a zero, the image was successfully written. Otherwise, the value must be examined to determine what action to take. After all of the trailer pages have been written, the image error log is (optionally) written to film. After the end roll process is complete, the ActiveX_StatusEvent returns ‘5000’ in the retStatus parameter.

   If neither trailer pages nor the image error log file are to be written to film, the EndRoll method does not have to be performed.

   NOTE: Notice the value returned in the callback event to indicate the end of each of the BeginRoll, WriteImages, and EndRoll methods are the same. The application must therefore be able to determine which process is being performed so the next sequential process can be initiated from the callback event.

Sample application

The following sample application should be used for illustrative purposes only. This is sample code only and is not intended to be an example of robust development ready for implementation.

Module Level Declarations

```
'some awol error codes – see Chapter 6 for others

Public Const NoError_Update As Integer = 0
Public Const EndOfFilm As Integer = 4002
Public Const EndOfImages As Integer = 5000
Public Const NoListFile As Integer = 5001
Public Const NoRoomLeftOnRoll As Integer = 5003
Public Const SmoothShutdown As Integer = 5004
Public Const NoImagesToWrite As Integer = 5009
Public Const NoHeaderPages As Integer = 5007
Public Const NoTrailerPages As Integer = 5008
Public Const Begin As Integer = 1
Public Const Write As Integer = 2
Public Const End As Integer = 3
Public Const Shutdown As Integer = 4
```

Module or Form Level Code

```
'Filmtemplate OLE Server
Public FilmOBJ As Object

Dim RollID, LastImageFile, StartingImageAddress, Operation As String
Dim RollNumber, ReturnStatus As Integer
Dim RestartMode, ImagesWritten As Long
```

```
Private Sub Awisocx1_StatusEvent(ByVal Status As Integer, ShutDown
        As Integer)

 txtStatus.Text = "The StatusEvent is" +  Str(Status)

Select Case Status

    Case EndOfImages
        GoTo Operation

    Case NoListFile
        ‘Take appropriate action

    Case NoRoomLeftOnRoll
        ‘Take appropriate action

    Case NoErrorUpdate
        ‘refreshproperties

    Case Else
        ‘unhandled error

End Select
Exit Sub

Operation:

Select Case Operation

    Case "Begin"
       If Status <> EndOfImages Then
        MsgBox Operation + Str(Status), vbOKOnly
        Else
              PerformWriteImages
        End If

    Case "Write"

    If Status <> EndOfImages Then
      If Status = 0 Then
        ‘read properties & update form for progress
      Else
         MsgBox Operation + Str(Status), vbOKOnly
      End If
    Else
      PerformEndRoll
    End If

  Case "End"

    If Status <> EndOfImages Then
      MsgBox Operation + Str(Status), vbOKOnly
    Else
      ‘take appropriate action
```

```
        End If

    Case "Shutdown"

        If Status <> SmoothShutdown Then
            MsgBox Operation + Str(Status), vbOKOnly
        Else
            MsgBox "Success" + Operation + Str(Status), vbOKOnly
        End If

    Case Else

        MsgBox Operation + Str(Status), vbOKOnly

   End Select

End Sub

Private Sub cmd1_Click()

  InitialSettings

End Sub

Public Sub InitialSettings()

Operation = "Init"

RollID = "1"
LastImageFile = ""
RollNumber = 1
StartingImageAddress = "00001.0000001"
RestartMode = 1
ImagesWritten = -1

With Awisocx1

        .WriterName = "writer"
        .WriterAddress = "100.100.100.100"
        .InputMode = 1 'batch mode
        .FileInput = 0 'single page tiff
        .ImageLocation = "c:\images\"
        .ErrorLoggingLevel = 2 'warning level
        .ErrorLogMaxSize = 256 'convert this to 256k
        .ErrorLogTrimSize = 256 ' convert this to 250k
        .ErrorLogLocation = "c:\images\error\"
        .TransferFile = 1 'Standard file
        .TransferReportingLevel = 0 'Document level only
        .TransferLocation = "c:\images\xfer\"
        .JobLocation = "c:\images\job\"
         ‘Header Page Location
        .CoverPages = "c:\images\header\"
         ‘Trailer Page Location
        .TrailerPages = "c:\images\trailer\"
```

```
End With  'Awisocx1

'Create and initialize the Filmtemplate server
Set FilmOBJ =
         CreateObject("FilmTemplateServer.FilmTemplateObject")
FilmOBJ.GetTemplate "Default"

MsgBox "Properties set and film template created", vbOKOnly +
         vbApplicationModal
PerformEstablishWriterConnection
End Sub

Public Sub PerformEstablishWriterConnection()
'Step 2

Operation = "Connect"

On Error GoTo WriterInitializeError
ReturnStatus = Awisocx1.Initialize

If Response <> NoError_Update Then
   'The attempt to connect failed so a call to
   'check writer status & error codes and
   'take appropriate action is made.
   MsgBox "WriteInitFailed" + Str(Response), vbOKOnly +
         vbApplicationModal
Else
   'The attempt to connect succeeded.

   'Step 3 - put the film template name in the awisocx property
   MsgBox "Writer Intialized", vbOKOnly + vbApplicationModal

   'On Error GoTo CheckTemplateError
   Awisocx1.FilmTemplate = "OneLevelSimplex"
   MsgBox "film template set", vbOKOnly + vbApplicationModal

   PerformSetupJob

End If

WriterInitializeError:
  'take appropriate action

CheckTemplateError:
   MsgBox "Film Template Error" + Str(Err.Number), vbOKOnly
End Sub
```

```
Public Sub PerformSetupJob()

‘Step 4

    Operation = "Setup"

'On Error GoTo SetUpJobErrorRoutine
    ReturnStatus = Awisocx1.SetupJob

'Check for error in setup job
    If ReturnStatus > NoError_Update Then

      Select Case ReturnStatus

        Case Is = NoHeaderPages
          'No header pages but they wanted them
          MsgBox "NoHeader", vbOKOnly

        Case Is = NoTrailerPages
          'No trailer pages but they wanted them
          MsgBox "NoTrailer", vbOKOnly

        Case Is = NoListFile
          MsgBox "NoListFile", vbOKOnly

        Case Is = NoImagesToWrite
          MsgBox "NoImagesToWrite", vbOKOnly

        Case Else
          MsgBox "SetupJobFailed" +  Str(ReturnStatus), vbOKOnly

      End Select
      Exit Sub
    Else
      MsgBox "SetupJob complete", vbOKOnly + vbApplicationModal
      PerformBeginRoll
    End If

'SetUpJobErrorRoutine:
  ‘Take appropriate action


End Sub
```

```
Public Sub PerformBeginRoll()

‘Step 5

'Call Begin Roll Method

    Operation = Begin

    ReturnStatus = Awisocx1.BeginRoll(RollID, LastImageFile,
        RollNumber, StartingImageAddress, -1) 'ImagesWritten)

    'Check for errors
    If ReturnStatus <> NoError_Update Then
      MsgBox "BeginRollFailed" + Str(ReturnStatus), vbOKOnly +
        vbApplicationModal
    Else
      MsgBox "BeginRollReturnStatus=0", vbOKOnly +
        vbApplicationModal
    End If

End Sub

Public Sub PerformWriteImages()

‘Step 6

    Operation = Write

    ReturnStatus = Awisocx1.WriteImages(RollID, LastImageFile,
        RollNumber, StartingImageAddress, ImagesWritten,
        RestartMode)

    'Check for errors and handle if present
    If ReturnStatus <> NoError_Update Then
      MsgBox "WriteImagesFailed" + Str(ReturnStatus), vbOKOnly
    Else
      MsgBox "WriteImagesReturnStatus=0", vbOKOnly +
        vbApplicationModal
    End If

End Sub
```

```
Public Sub PerformEndRoll()

‘Step 7

    Operation = End

    ReturnStatus = Awisocx1.EndRoll(RollID, LastImageFile,
       RollNumber, ImagesWritten)

    If ReturnStatus <> NoError_Update Then
      MsgBox "EndRollFailed" + Str(ReturnStatus), vbOKOnly
    Else
      MsgBox "EndRollReturnStatus=0", vbOKOnly +
       vbApplicationModal
    End If

End Sub

Public Sub PerformShutdown()

    Operation = Shutdown

    ReturnStatus = Awisocx1.ShutDown(2)

    If ReturnStatus <> NoError_Update Then
      MsgBox "ShutdownFailed" + Str(ReturnStatus), vbOKOnly
    Else
      MsgBox "ShutdownReturnStatus=0", vbOKOnly
    End If

End Sub
```

6 Status and Error Codes

ActiveX error log

The chart below defines error log numbers, corresponding messages, and an explanation of each error. In certain cases, corrective actions are provided. These errors mostly occur at the AWOL and are subsequently raised to the ActiveX_StatusEvent.

NOTE: %s Actual message contains character designator or file name(s).

%d Actual message contains numeric value.

#	Message	Description
0	Success	This is the normal return status for all operations. The error log will not usually contain any of these messages.
1000	Error opening file: %s.	The specified file cannot be opened. **Probable cause**: • File may not exist which may be caused by abnormal operation of the Archive Writer (such as powering the Writer down or setting it to Offline while image-writing is in process). • May also be caused by manually deleting files AWIS expects to find, such as the index data file and template files.
1001	Error copying file: %s to %s.	The specified file cannot be copied. **Probable cause:** • Invalid source or destination file paths or by insufficient disk space on the destination drive. • Incorrect security settings on the files. The source file must be readable, the destination file must be writable, and both files must not be in use or locked by another program.
1002	Error reading file: %s.	The specified file cannot be read. **Probable cause:** • The file may not exist, may be locked by another program, or may be corrupted. • Low memory conditions, where there is not enough memory to allocate a buffer for reading the file contents.
1004	No index data file specified	*Informational message*. The application did not provide the name of an index data file.

#	Message	Description
1007	Failed to write to file %s.	The file was successfully created, but a failure occurred in writing to the file. **Probable cause:** Disk failures or running out of disk space. If the file is being written to the Archive Writer drive, this error may be caused by the Archive Writer being shut down or by an error occurring in the Writer that removes or corrupts the remote drive connected to the Writer.
1008	Failed to rename file %s to %s.	An attempt to rename a file has failed. **Probable cause:** Failure of the disk the file resides on or the destination file may already exist.
1009	Cannot write file %s: no data to write to it...	An internal program error has caused an attempt to write a zero-length data buffer to the file. This error should never occur.
1010	Error creating file %s.	Failed to create the specified file. **Probable cause:** the disk the file is to be created on is corrupted, missing, or out of space.
1011	Error closing file %s.	Failed to close the specified file. **Probable cause:** the disk is corrupted.
1012	Error code %d returned during attempt to create file %s.	Failed to create the specified file. Probable cause: the disk the file is to be created on is corrupted, missing, or out of space. The error code is documented in the Microsoft Visual C++ documentation. Refer to the section on error codes returned from the file IO functions.
1013	Error code %d returned during attempt to read file %s.	Failed to read the specified file. **Probable cause:** the disk the file is on is corrupted or missing. The error code is documented in the Microsoft Visual C++ documentation. Refer to the section on error codes returned from the file IO functions.
1014	Error code %d returned during attempt to write to file %s.	Failed to write to the specified file. **Probable cause:** the disk the file is on is corrupted or missing. The error code is documented in the Microsoft Visual C++ documentation. Refer to the section on error codes returned from the file IO functions.
1015	Error code %d returned during attempt to close file %s.	Failed to close the specified file. **Probable cause:** the disk the file is on is corrupted or missing. The error code is documented in the Microsoft Visual C++ documentation. Refer to the section on error codes returned from the file IO functions.

#	Message	Description
1016	Error during attempt to create directory %s.	Failed to create temporary directory to place single-page tiff files from multi-page tiff files.
1017	Couldn’t send an input file specification to the writer for file %s.	Socket error.
1018	Couldn’t send an output file specification to the writer for file %s.	Socket error.
1019	Couldn’t get an input file specification acknowledgment from the writer for file %s.	Socket error.
1020	An input file specification contained an invalid filename: %s.	Socket error.
1021	An input file specification acknowledgment contained and invalid value: %c.	Socket error.
1022	And input file specification acknowledgment contained an invalid value: %c.	Socket error.
1023	Couldn’t get a file content acknowledge from the writer for file %s.	Socket error.
1024	Failure writing contents of %s to writer.	Socket error.
1025	Couldn’t send the file content acknowledgment to the writer for file %s.	Socket error.
1026	Couldn’t write file %s to the writer.	Socket error.
1027	Couldn’t read file %s from the writer.	Socket error.
1028	Couldn’t get an output file specification acknowledgment from the writer for file %s.	Socket error.
2000	File open error	Could not open file.
2001	Timeout waiting for semaphore file %s to be deleted.	The Writer has been given a command to execute. The command was not executed within the timeout period allowed (usually 90 seconds). **Probable cause:** the Writer may be offline or inoperable. Be sure the Writer is online and the Writer disk is available. It may be necessary to reboot the Writer and make sure the NFS connection to the Writer is working properly.

#	Message	Description
2010	Socket Error %d	
2011	Socket Error %d	Could not connect to Writer. Socket problem.
2012	Socket Error %d. Host sockets have been reset.	The Writer has reset the socket connection.
2013	Socket Error %d. No socket data found. Check Services file.	Port and protocol information could not be found. The Services file may not have entries for the socket connections.
2014	Socket Error %d. Host not found. Check Writer name.	The Writer name could not be found either in the host file or on the network.
2016	Can’t find winsock.dll.	The winsock.dll is not on the system.
2017	Socket Error. Timeout error.	The socket has timed out.
3000	Error writing command %s.	Failed to create a command file containing the specified command. **Probable cause:** the Writer may be offline or inoperable. Be sure the Writer is online and the Writer disk is available. It may be necessary to reboot the Writer and make sure the NFS connection to the Writer is working properly.
3001	Error sending command.	Failed to create a command file containing the specified command. **Probable cause:** the Writer may be offline or inoperable. Be sure the Writer is online and the Writer disk is available. It may be necessary to reboot the Writer and make sure the NFS connection to the Writer is working properly.
3002	Error creating semaphore.	Failed to create a semaphore file. **Probable cause:** the Writer may be offline or inoperable. Be sure the Writer is online and the Writer disk is available. It may be necessary to reboot the Writer and make sure the NFS connection to the Writer is working properly.
3003	Error creating thread.	Failed to create a thread. This should never occur. It indicates a problem in the Windows NT operating system. Close the application and/or reboot the workstation.
3004	Error in Astring parse.	An internal programming error has occurred in the AWOL parsing routine. This error should never occur.

#	Message	Description
3005	Response file for command %d, no response value in file.	An empty response file was created by the Writer. **Probable cause:** there may be an internal fault in the Writer. Close the application and restart the Archive Writer.
3006	Error creating semaphore.	Failed to create a semaphore file. **Probable cause:** the Writer may be offline or inoperable. Be sure the Writer is online and the Writer disk is available. It may be necessary to reboot the Writer and make sure the NFS connection to the Writer is working properly.
3009	Error parsing response parameter number.	An internal programming error has occurred in the AWOL parsing routine in attempting to parse the Response parameter number from the string. This error should never occur.
3010	Error parsing Archive Writer error number.	An internal programming error has occurred in the AWOL parsing routine in attempting to parse the error number from the response file. This error should never occur.
3012	Parameter number out of range.	The parameter number contained in the response file is not valid. This error should never occur.
3013	Image buffer full.	The internal image buffer is full. This is normal operation. This error should never appear in the error log file.
3014	Invalid writer drive.	The application has supplied a drive letter that is invalid. Make sure the specified drive letter is correct.
3016	Command timed out. Command: %s.	The Archive Writer has not responded to the command within the allowed period of time (usually 90 seconds). Be sure the Archive Writer is online and the drive assigned to the Writer is available. It may be necessary to restart the Archive Writer.
3018	No more files.	This is a normal internal status indicating that no more files are available for processing.
3021	File mismatch. Expected: %s, got: %s, original file: %s.	The response file returned by the Writer contains the wrong file specification. **Probable cause:** indicates a communication or other internal failure in the Writer. Close the application and restart the Writer.

#	Message	Description
3022	No image address in %s (Image log file).	The image log file returned by the Writer does not contain a valid image address where one was expected in the file. **Probable cause**: indicates a communication or operational failure in the Write. Close the application and restart the Writer.
3023	Can't open transfer file.	The transfer file cannot be opened. **Probable cause:** may be a corrupted hard disk or an invalid path is specified for the location of the transfer files.
3024	Can't close transfer file.	The transfer file cannot be closed. **Probable cause:** a corrupted hard disk.
3027	Rollname too long.	More than 8 characters have been specified for the roll name.
3030	Failed to get remaining film.	The command to get the remaining film in the Writer cassettes has failed. **Probable cause:** indicates a failure of the communication link with the Writer or an internal failure in the Writer. Close the application and restart the Archive Writer.
3031	Command buffer empty.	An internal programming error in the AWOL DLL has occurred.
3032	No more disk space.	There is not enough disk space on the Writer drive to create a file. This error should never appear in the error log.
3033	Can't overwrite command.	This is an internal program status and will never appear in the error log file.
3034	Failed to copy file to writer.	A file cannot be copied to the Writer. Probable cause: indicates a failure of the communication link with the Writer or an internal failure in the writer. Close the application and restart the Archive Writer.
3035	Status file %s was returned for this command string %s.	**Probable cause:** an internal error condition in the Writer. Status files are normal for certain operations and conditions in the Archive Writer. The value in the status file contains the severity code for the error. These errors are interpreted by AWIS and converted to more meaningful error messages which are written to the error log file.

#	Message	Description
3036	Could not find response file, %s.	A request has been made for information from the Archive Writer, the command was executed, but no response file was returned. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3037	Error creating the film leader.	An error was returned from the Archive Writer while attempting to create a film leader. **Probable cause:** indicates a film cassette failure, not enough film in the cassette, or by an internal error in the Archive Writer. Be sure the film cassette contains enough film and that the Archive Writer communication link is working properly.
3038	Error in image address on cassette.	The film cassette contains an invalid image address.
3040	Error sending IMC command.	The IMC command could not be successfully sent to the Archive Writer. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3041	Error sending Image Frame command.	The command to set the next image frame could not be successfully sent to the Archive Writer. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3042	Error sending Leader Length command.	The command to set the leader length could not be successfully sent to the Archive Writer. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3043	Error sending System command to writer.	The System command could not be successfully sent to the Archive Writer. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.

#	Message	Description
3044	Error sending command to get remaining film.	The command to get the remaining film could not be successfully sent to the Archive Writer. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3045	Error sending command to get the film status.	The command to get the film status could not be successfully sent to the Archive Writer. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3046	Error sending command to get the film cassette data.	The command to get the cassette data could not be successfully sent to the Archive Writer. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3047	Error sending command to set the diagnostic port.	The command to set the diagnostic port could not be successfully sent to the Archive Writer. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer..
3048	Error sending command to set the status monitor port.	The command to set the status monitor port could not be successfully sent to the Archive Writer. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3049	Error sending command to set the image writing parameters.	The command to set the image writing parameters could not be successfully sent to the Archive Writer. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3050	Error sending command to set the disk emulation.	The command to set disk emulation could not be successfully sent to the Archive Writer. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.

#	Message	Description
3051	Error sending command to set the date and time in the writer.	The command to set the internal date and time could not be successfully sent to the Archive Writer. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3052	Error sending command to set the frame annotation.	The command to set the frame annotation could not be successfully sent to the Archive Writer. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3053	Error getting command 42, get frame annotation.	The response file for the command to get the frame annotation does not contain a valid response. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3054	Error getting command 19, get system command.	The response file for the command to get the Archive Writer system parameters does not contain a valid response. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3055	Error getting command 4, get leader length.	The response file for the command to get the leader length does not contain a valid response. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3056	Error getting command 11, get image frame.	The response file for the command to get the next image frame does not contain a valid response. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3057	Error getting command 38, get IMC.	The response file for the command to get the IMC settings does not contain a valid response. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.

#	Message	Description
3059	Error code returned when trying to issue a SetNextTransaction Number command.	The command to set the next transaction number failed. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3060	Timeout waiting to rename log file %s to get image address for file %s (original: %s).	The command to create an image log file for the last images written has timed out. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer. Close the application and restart the Archive Writer.
3061	Removed status file:%s.	This is an informational error only.
3062	Can't remove status file:%s.	Unable to delete the specified status file from the writer. **Probable cause:** indicates a failure of the communication link with the Writer or by an internal failure in the Writer or by an internal failure or synchronization problem between the writer and the AWOL software layer. Close the application and restart the Archive Writer.
3063	Multipage tif sent to writer as a single-page tiff.	This is an internal diagnostic message.
3064	Expected to get response for page %d of file %s; instead got page %d.	This is an internal diagnostic message.
3065	File preceding %s is bad.	The image file preceding the specified file is bad. Replace the bad file with a good one and restart the operation.
3068	Error getting command 57, get power down interval.	Error using command 57, get power down interval. Reboot writer.
3069	Error getting command 58, set power down interval.	Error using command 58, set power down interval. Reboot writer.
3070	Error getting command 60, get interdocument gap.	Error using command 60, get interdocument gap.
3071	Error getting command 59, set interdocument gap.	Error using command 59, set interdocument gap.
3072	Error getting command 20, get version numbers.	Error using command 20, get version number.
3073	Error sending command 40, get online status.	Error using command 40, get online status.
3074	Error sending command 1, advance film.	Error sending command 1, advance film.

#	Message	Description
3075	Could not successfully read status file, %s.	Failed reading status file. Reboot the Writer.
3078	Could not convert text file %s to a TIFF file.	Problem converting text file to TIFF.
3079	Using the 'NumCommands' value of %d found in registry for total # of simultaneous AWIS print commands.	Informational message. AWIS is using a value in the system registry that has been entered by a system administrator or Kodak representative for purposes of performance enhancement. This value sets the number of simultaneous print commands that can be active.
3080	The 'NumCommands' value found in registry (%d) is either zero or exceeds system maximum of %d. Setting to system default of %d.	Warning message to indicate to system administrator or Kodak personnel that an invalid value has been entered into the system registry. This value is intended to enhance performance, but will not be used because it is outside of the allowable limits. System will use a default value instead of the illegal value found in the registry.
3081	Using the 'NumFiles' value of %d found in registry for max number of image files per AWIS print command.	*Informational message*. AWIS is using a value in the system registry that has been entered by a system administrator or Kodak representative for purposes of performance enhancement. This value sets the maximum number of image files that can be sent in a single print command.
3082	The 'NumFiles' value found in registry (%d) is either zero or exceeds system maximum of %d. Setting to system default of %d.	Warning message to indicate to system administrator or Kodak personnel that an invalid value has been entered into the system registry. This value is intended to enhance performance, but will not be used because it is outside of the allowable limits. System will use a default value instead of the illegal value found in the registry.
3083	The 'NumCommands' value (%d) multiplied by the 'NumFiles' value (%d) exceeds the system limit of %d. Using default values of %d and %d.	*Warning message*. Indicates to system administrator or Kodak personnel that although valid NumCommands and/or NumFiles value(s) have been entered into the system registry, the two values together produce an invalid result. These values are intended to enhance performance, but will not be used because their product is outside of the allowable system limit. System will use default values for both fields.
3084	Setting number of simultaneous print commands to %d; max number of image files per command to %d.	*Informational message*. Indicates what ‘total print commands’ and ‘max image files per command’ values AWIS will use.
3085	Unable to retrieve AWIS installation directory from registry.	AWIS installation directory is not in the registry.

#	Message	Description
3087	While forming a command, had to insert a print remaining command after file %s.	Moved Writer back to the A channel when uneven amounts of images are present for duplex film.
3090	The ‘FileWaitTime’ value found in registry (%ul) is invalid. Setting to system default of %ul.	FileWaitTime in registry is invalid. Using defaults.
3091	Using the ‘FileWaitTime’ value of %ul found in the registry.	Using FileWaitTime registry value.
4001	Not enough film on roll.	There is not enough film on the roll to write the images. Increase the amount of unexposed film in the cassette.
4008	Can't flush writer.	Need to reboot to proceed. A communication failure has occurred which prevents Command 58 from deleting files from the Archive Writer directories. Close the application and restart the Archive Writer.
5000	End of Image Writing. All images supplied by the application have been written.	This is a normal, successful condition.
5001	List file not found: %s.	The list file supplied by the application cannot be found. Specify a valid list file.
5002	No more images in list.	*Informational message*.
5003	End of Roll.	The End of Roll operation has completed.
5004	Smooth Shutdown	The shutdown operation has completed.
5006	No images found in image list or image directory.	The application has specified a directory (in batch mode) or a list file (in list mode) that contains no images to be written.
5007	No images found in Cover Page directory.	The application has specified a directory for cover (“header”) pages to be written which contains no images.
5008	No images found in Trailer Page directory.	The application has specified a directory for trailer pages to be written which contains no images.
5009	No images found in Image List File or in Image Directory.	The application has specified a directory (in batch mode) or a list file (in list mode) that contains no images to be written.
5011	No files found in image log directory.	*Information message.* Indicates no pages were created containing image error messages to be written at the end of film.
5012	Inconsistency between Image Address Field Widths and/or Offset Addressing and/or selected filming Level.	

#	Message	Description
5013	%s | %s	No images in specified cover directory or specified trailer directory.
5014	%s | %s	No images in specified cover or specified image directory.
5015	%s | %s	No images in specified image or specified trailer directory.
5016	%s | %s | %s	No images in specified cover or specified trailer or specified image directory.
5018	SetCurrentFile, file at %d position in file list is %s; file expected at this position was %s.	*Information message.*
5019	File %s exceeds max file size for Archive Writer in simplex mode of %d bytes; its size is %d bytes.	File too large for Archive Writer in simplex mode.
5020	File %s exceeds max file size for Archive Writer in duplex mode of %d bytes; its size is %d bytes.	File too large for Archive Writer in duplex mode.
5021	Memory reallocation error.	There is not enough memory to perform the current operation. Add more memory to the computer, by specifying a larger page file size in Windows setup, or by reducing the size of the image list file or by reducing the number of images in batch directories.
5022	File not found: %s.	A file specified in the image list file was not found. Verify that the file specified in the list file exists.
5023	Error resetting file list.	The file specified by the application cannot be found in the list file (in list mode) or in the image directory (in batch mode). This problem is corrected by verifying that the list file or batch input directories contain the file specified by the application.
5027	No index page directory specified.	The application did not supply a path in the JobPath parameter for creating index pages. This is an internal programming error.
5028	No writer path specified.	The application did not supply a valid Archive Writer path. This is an internal programming error.
5029	Cannot open image log file: %s.	The image log file cannot be opened. **Probable cause:** Insufficient hard disk space.
5030	Cannot write to image log file: %s.	The image log file cannot be written to. **Probable cause:** Insufficient hard disk space.

#	Message	Description
5031	Cannot write to transfer file: %s.	The transfer file cannot be written to. **Probable cause:** Insufficient hard disk space.
5032	Cannot allocate memory.	There is not enough memory to perform the current operation. Add more memory to the computer, by specifying a larger page file size in Windows setup, or by reducing the size of the image list file or by reducing the number of images in batch directories.
5034	Error found in response file for file %s (page %d). Got Writer Error number %d.	An error has been detected in the response file for the filename specified. **Probable cause:** There may be an internal error in the Archive Writer. The interpretation of the error code should appear in the error log following this error.
5035	%s is not a valid Digital Archive Writer path - the %s directory is missing.	The specified Archive Writer path does not appear to be valid because the specified directory is missing. Close the application and restart the Archive Writer.
5036	Cannot communicate with writer (failed to set next transaction number).	Close the application and restart the Archive Writer.
5038	Not enough space on your hard disk.	You need at least 1MB of free space. Your hard disk does not have enough space for AWIS to perform its normal operations. Increase your free disk space.
5039	Archive Writer cannot be reset to continue processing.	The command to reset the Archive Writer has failed. Close the application and restart the Archive Writer.
5040	Film template requires two cassettes in writer.	The film template indicates that two cassettes are required for this job. Be sure two cassettes are inserted in the Archive Writer.
5041	A cassette needs to be loaded into the Archive Writer.	There are no cassettes in the Archive Writer. If you do have a cassette in the Writer, a mechanical or electrical problem in the Writer has failed to detect the presence of the cassette.
5042	Cannot proceed until the Archive Writer is put in Online mode.	The Archive Writer is currently set to Offline. Use the front panel buttons on the Archive Writer to set the mode to Online.
5043	Writer in recoverable error state. Fix error above and restart.	*Informational message*. Indicates the Archive Writer may be restarted from the current error condition.
5045	Writer error has occurred.	Check previous error messages in error log file.

#	Message	Description
5046	The image address provided must have a “0” value for the Level 1 part of the image address.	Check addressing setup in film template.
5048	Line %d of list file %s contains %d dashes preceding the filename; maximum number of dashes allowed is %d.	AWIS encountered a line in the list file on which there was more than the allowed number of dash characters preceding the file name. Each dash on a line indicates that the file is to be indexed one level lower than the index level specified for the job. For a 2-level job, the maximum number of dashes on a line is two; for a 1-level job, it is one. For example, in a 2-level job, two dashes in a list file line entry would cause the file to be indexed at level zero (two level minus two dashes equals zero). Note that only the first line on which the violation was found is flagged; check the entire list file to ensure that all lines are correct. Change the list file to meet the file specification requirements.
5049	Film template requires the upper cassette to be loaded into the Archive Writer.	Put upper cassette in the Archive Writer.
5050	Film template requires the lower cassette to be loaded into the Archive Writer.	Put lower cassette in the Archive Writer.
5051	Upper cassette battery needs to be replaced.	Replace upper cassette battery.
5052	Lower cassette battery needs to be replaced.	Replace lower cassette battery.
5053	Upper cassette improperly inserted.	Upper cassette improperly installed. Remove cassette and reinstall.
5054	Lower cassette improperly inserted.	Lower cassette improperly installed. Remove cassette and reinstall.
5055	Cassette access door is open.	Cassette access door is open.
5056	Cassettes are on different sides. Please insert matching cassettes.	Cassettes are on different sides. Insert matching cassettes.
5057	Unable to load language files on writer.	Failed to load language files on the writer. Reboot writer.
5058	Unable to access registry.	Failed to access registry. Check registry entry.
5059	The controller version number on the Archive Writer is %s. A version of %s or greater is needed to run with the AWIS application software.	Check version numbers for AWIS and the Writer. Report differences to service.

#	Message	Description
5060	Bad ASCII character in text file %s.	Check ASCII file. Character not in character set 0-255.
5061	Line too long in text file %s.	Check ASCII file. Line greater than 80 characters.
5062	Too many lines in text file %s.	Check ASCII file. More than 66 lines.
5065	The first line in list file %s must be specified to be indexed either at level %d or at level zero.	The first file specified in a list file must be indexed either at the job level or as a level zero image. In the list file specified in the error message, the first line contained a single dash, indicating that the first file in a two-level job was to be indexed at level 1. This is not legal. If the list file is a continuation of a previous job, the list file must be broken apart on document boundaries. Fix the list file such that the first line in it has no dashes.
5066	Unable to open or read file %s to interrogate TIFF tags. Please check that file is present and is a valid TIFF file.	
5067	An Archive Writer is not mapped to drive %s.	Returned from Initialize if FindFirstFile call returns nonexistent path for the ArchiveWriterPath provided. This differentiates from 5035 where the drive provided is not an Archive Writer, but it exists.
5068	Standalone AWOLArchiverWriter class method called with bad listtype value: %d.	
5200	End of Roll warning.	The end of the film has been reached.
5201	Ten feet warning.	10 feet left on roll.
5202	Fifteen feet warning.	15 feet left on roll.
6000	Sending command to get remaining film...	This is an internal diagnostic message.
6001	Application told us to terminate.	This is an internal diagnostic message.
6004	Opening transfer file and image data file.	This is an internal diagnostic message.
6005	Sending a group of images.	This is an internal diagnostic message.
6008	Write Images Method: %s.	This is an internal diagnostic message.
6009	Exiting WriteImages with return code %d (%s).	This is an internal diagnostic message.
7000	Writer is idle.	This is an internal diagnostic message.
7010	Invalid restart mode.	Not a valid restart mode.
8000	No user exit DLL found “AWOLEXIT.DLL”.	No user exit named AWOLEXIT.DLL found in the Window’s system directory.

#	Message	Description
8001	User exit %s not found.	User exit name not found in AWOLEXIT.DLL.
8002	Error in user exit %s.	Error in user exit. Check text returned from user exit.

Archive Writer errors

The chart below defines Archive Writer error numbers, the corresponding error messages, and an explanation of what each error signifies. These usually appear on the Archive Writer display panel and in the Archive Writer's error log.

#	Message	Explanation
103	FD Cassette Access Door Closed	The cassette access door was closed.
100	FD End Of Roll Warning	The film remaining in the upper cassette is equal to or less than 3 feet.
201	FD Ten Foot Warning	The film remaining in the upper cassette is equal to or less than 10 feet.
202	FD Fifteen Foot Warning	The film remaining in the upper cassette is equal to or less than 15 feet.
203	FD Film Breakage Upper Cassette	The film drive servo reported a loss of motion on the upper cassettes film encoder. Either the cassette is out of film or it is a hardware error.
204	FD Film Breakage Lower Cassette	The film drive servo reported a loss of motion on the lower cassette's film encoder. Either the cassette is out of film or it is a hardware error.
205	FD Out Of Film Upper Cassette	The film remaining in the upper cassette is equal to or less than 5 feet.
206	FD Out Of Film Lower Cassette	The film remaining in the lower cassette is equal to or less than 5 feet.
207	FD Low Battery Upper Cassette	The battery in the upper cassette is low.
208	FD Low Battery Lower Cassette	The battery in the lower cassette is low.
209	FD Loss Of Comm Upper Cassette	The battery in the upper cassette is dead or there is a hardware problem.
210	FD Loss Of Comm Lower Cassette	The battery in the lower cassette is dead or there is a hardware problem.
211	FD Cassette Access Door Open	The cassette access door was opened.
212	FD Upper Cassette Inserted Improperly	The upper cassette was inserted improperly.
213	FD Lower Cassette Inserted Improperly	The lower cassette was inserted improperly.
214	FD Invalid Num Cassettes	Configuring the number of cassettes required with an invalid number.
215	FD CP Invalid Fixed Length	Configuring the fixed length advance with an invalid length.
216	CD CP Invalid Message Data	Command Processor: error in message data invalid data was passed with a command.

#	Message	Explanation
217	CD CP Unknown Request	Command Processor: invalid command received.
218	FD CP Proc Illegal FDS Cmd	Unknown film drive request.
219	FD CP Invalid Leader Length	Advancing film or setting the leader length with an invalid amount.
220	FD CP Invalid Tension Length	Setting the tension length with an invalid amount.
221	FD Cassette Filmlengths Differ	The film amounts between the upper and lower cassettes differ by more than 6 feet.
222	FD Cassette Image Address Differ	The image addresses in the upper and lower cassettes differ.
223	FD Cassette Roll Numbers Differ	The roll numbers in the upper and lower cassettes differ.
224	FD Cassette Sides Differ U1 L2	The upper cassette is on Side 1 and lower cassette is on Side 2.
225	FD Cassette Sides Differ U2 L1	The upper cassette is on Side 2 and lower cassette is on Side 1.
226	FD End Of Roll Warning Duo Side A	The end of a roll for duo Side A has been reached.
227	FD End Of Roll Warning Duo Side B	The end of a roll for duo Side B has been reached.
228	FD Upper Cassette Not Present	The system is configured for two cassettes and the upper cassette is not present or the system is configured for one cassette and both cassettes are not present.
229	FD Lower Cassette Not Present	The system is configured for two cassettes and the lower cassette is not present or the system is configured for one cassette and both cassettes are not present.
230	EH WRIB Kvalue Too Big	WRIB error: PM2 K value too big.
231	EH WRIB Cannot Process Tiled Image	WRIB error: Cannot process tiled images.
232	EH WRIB Cannot Process G4 Multistrip Image	WRIB error: Cannot process Group 4 multi-strip images.
233	FD Invalid Roll Number	An invalid roll number was entered. Either it was non-numeric or it was too long.
234	FD Invalid Job Number	An invalid job number was entered. Either it was non-numeric or it was too long.
236	CC II File Opening Error	Writer cannot open file on RAM disk.

#	Message	Explanation
237	CC II File Read Error	
238	EH IC Invalid Image Origin	Image Composition invalid image X, Y origin.
239	EH IC Invalid Image Scaling	Image Composition invalid image scaling.
240	EH IC Image Too Long	Image Composition scaled image too long.
241	EH IC Image Too Wide	Image Composition scaled image is too wide for frame.
242	EH IC Invalid Image Limits	Image Composition invalid image frame limits.
243	EH IC Invalid Image Orientation	Image Composition invalid image orientation.
244	EH IC Invalid Image Polarity	Image Composition invalid image polarity.
245	EH IC Invalid Image Resolution	Image Composition invalid image resolution.
246	EH IC Invalid Scaling Factor	Image Composition invalid image scaling factor.
248	EH DOS Disk Init Transaction Failed	Could not instantiate a command given a semaphore file's contents.
250	EH DOS Disk Command File Open Failed	Cannot open the host application interface manager's command file that was written.
251	EH DOS Disk Invalid Command Id	Command ID for a command is not one of the support command IDs.
252	EH DOS Disk Command Parameters Invalid	Parameters not valid for a command (e.g. wrong parameter IDs).
253	EH DOS Disk No Command Id	No command ID contained in the command file.
255	IA Invalid Data	Invalid image address data.
256	IA Field Overflow	Image address field overflow.
257	IA Format Width	Image address field too wide.
258	IA Format Length	Image address length too long.
259	IA Format Definition	Incompatible image address format.
260	IA Format Number	Invalid number of address fields.
261	IA Nonsequential Address	
262	Ls Language File Read Error	An error occurred while loading a language file.
263	PD Invalid Powerdown Interval	Invalid power down interval was entered.
264	ST Invalid Time Format	The time parameter entered was in an incompatible format.

#	Message	Explanation
265	ST Invalid Date Format	The date parameter entered was in an incompatible format.
266	CD CP Command Not Allowed Off Line	A command was received from the host that is not valid in the off-line mode.
267	CD CP Command Not Allowed With Critical Error Condition	A command was received after the Archive Writer reported a critical error but before the warning was corrected.
268	CD CP Command Not Allowed With Recoverable Error Condition	A command was received after the reported correctable error but, before the warning was corrected.
269	CD CP Command Not Allowed With Warning Error Condition	A command was received after the Archive Writer reported a warning but before the warning was corrected.
270	IC No File Specified	No image file was specified in the print image command. Specify an image file with the print image command.
271	EH IO Image Log Size Exceeded	The image log file is at its maximum size. Rename the log file, remove the old log file.
272	EH NET Cant Add Host	Could not add a host to the Host table that exists. Check the command file being used for Set Network Specification and make sure it contains a valid host name and host IP address.
273	FD Verify Film Upper Cassette	A cassette is inserted in upper film bay of the Archive Writer that has not been previously used in an Archive Writer. It is possible that the cassette is an *Imagelink* 30 or 70 cassette. Verify the cassette inserted in the upper film bay is the proper cassette.
274	FD Verify Film Lower Cassette	A cassette is inserted in lower film bay of the Archive Writer that has not been previously used in an Archive Writer. It is possible that the cassette is an *Imagelink* 30 or 70 cassette. Verify the cassette inserted in the lower film bay is the proper cassette.
276	CC II IMC Required Before Printing	The host is attempting to print a job with IMC enabled before sending the Archive Writer the IMC parameters and 'make IMC' command. Command the Archive Writer to write IMC via the host application.

#	Message	Explanation
277	EH IC Type Invalid	An invalid film format was received from the host. Simplex or duplex should be specified.
278	EH IC Scaling Invalid	An invalid image scaling value was received from the host.
279	EH IC Origin Invalid	An invalid image origin was received from the host.
280	IA Invalid Level	An attempt to set the image level to an invalid value occurred. The value must be between 0 and x. Where x is the number of image levels used.
290	EH FT Transfer Timed Out	Failed to receive data packet before watchdog timed out.
298	FD 5 Meter Warning	The film remaining in the upper cassette is equal to or less than 5 meters.
299	FD 3 Meter Warning	The film remaining in the upper cassette is equal to or less than 3 meters.
300	CD CP Command Substitution Error	Command Decoder: command being over-written.
301	CD CP Initialization Error	Command Processor: initialization error.
302	CD CP MsgQ Receive Error	Command Processor: error getting message from queue.
303	CD CP MsgQ Time Out Error	Timeout occurred waiting for message from queue.
304	CD CP Invalid MsgQ Return Code	Unknown return code received for messageGet.
305	CD CP Executing Uninitialized Command	Command Processor: command function not initialized.
306	CD CP Error Checking Unitialized	Command Processor: error checking function not initialized.
307	CD CP Command Element Init Invalid Index	Index is not within the list of functions.
308	CD CP Invalid Param Components Length	Command parameter length exceeds CD message size.
309	FD Msg Init Data Length Error	Initializing a film drive message with data longer than the buffersize.
310	FD Cassette Data Reservation Failed	Cassette data semaphore. Take failed.

#	Message	Explanation
311	FD Cassette Data Release Failed	Cassette data semaphore. Give failed.
312	FD Cassette Data Illegal Cassette Id	Cassette index is out of range.
313	FD Cassette Status Reservation Failed	Cassette status semaphore. Take failed.
314	FD Cassette Status Release Failed	Cassette status semaphore. Give failed.
315	FD Cassette Status Illegal Cassette Id	Cassette index out of range.
316	FD Message Send Error	Error sending to message queue.
317	FD CP Initialization Error	Film Drive Command Processor Task initialization error.
318	FD CP MsgQ Receive Error	Film Drive Command Processor error getting message from queue.
319	FD CP Cmd Time Out Error	Command response timer timed-out.
320	FD CP Illegal FD State Error	Film Drive Command Processor is in an illegal state.
322	FD CP Invalid MsgQ Return Code	Application Message Queue message. Get returned an invalid status.
323	FD CP Cmd Check Array Full	No room in the queue for the current command's response parameters (command ID; requesters ID; response required; and response timeout).
324	Received Invalid Message	From FDS. The command terminator received does not match one of the valid commands in the film drive.
325	FD RH Receive Buffer Overrun	The film drive servo sent a message longer than the embedded controller's receive buffer.
326	Film Drive Serial Port Read Error	Embedded Controller error reading a character from the serial port.
327	FD RH MsgQ Send Error	Film Drive Response Handler task reported an error sending a message to a message queue.
328	EH WRIB DOS Reserve Failed	WRIB error: DOS reserve failed.
329	EH WRIB DOS Release Failed	WRIB error: DOS release failed.
330	EH WRIB File Read Error	WRIB error: DOS file read error.
331	EH WRIB Unknown Decompress Unblock	WRIB error: Unknown semaphore unblock reason

#	Message	Explanation
332	EH WRIB Memory Reservation Failed	WRIB error: memory semaphore take.
333	EH WRIB Memory Release Failed	WRIB error: memory semaphore give.
334	EH WRIB Ram Frame Malloc Failure	WRIB error: RAM memory malloc failure.
335	EH WRIB Max Images Per Frame Exceeded	WRIB frame failure: maximum images per frame exceeded.
336	EH WRIB Frame Malloc Failure	WRIB error: memory malloc failure.
337	EH WRIB Invalid Image Index	WRIB error: image index invalid
338	EH WRIB Frame Not Found In WRIB Memory	WRIB memory warning: frame to delete not found.
339	EH IO MsgQ Length Too Small	Image Output message queue length too small.
341	EH IO Image Log Open Error	Image Output log open failure, file could not be opened.
342	EH IO Image Log Close Error	Image Output log close failure, file could not be closed.
343	EH IO Image Log Write Error	Image Output log write failure, write error.
344	EH IO Initialization Error	Image Output initialization error.
345	EH IO MsgQ Send Error	Image Output message queue send error.
346	EH IO MsgQ Receive Error	Image Output message queue receive error.
347	EH IO MsgQ Time Out Error	Image Output message queue timeout error.
348	EH IO Invalid MsgQ Return Code	Image Output invalid message queue return value.
353	EH IO IA String Too Long	Image Output frame image address string is too long.
354	EH IO Filename Too Long	Image Output frame image filename string is too long.
355	EH IO Too Many Image Files	Image Output frame contains too many image files.
356	EH IO Framemessage Retrieve Index	Image Output invalid frame message index.
357	CC II Memory Allocation Error	Image Input: memory allocation failure.
358	CC II Disk Reserve Error	Disk reserve semaphore take error.
359	EH IC MsgQ Send Error	Image Composition message queue send error.
360	EH IC Initialization Error	Image Composition initialization failure.

#	Message	Explanation
361	EH IC MsgQ Receive Error	Image Composition message queue receive error.
362	EH IC MsgQ Time Out Error	Image Composition message queue time-out error.
363	EH IC Invalid MsgQ Return Code	Image Composition invalid message queue return code.
364	EH IC Unknown Message Command	Image Composition unknown message command.
365	EH IC MsgQ Length Too Small	Image Composition message queue length too small.
366	EH IC Unknown Scaling Type	Image Composition unknown image scaling type.
367	EHIC Overlapping Frame Images	Image Composition overlapping images in frame.
368	EH IC DOS Reserve Failed	Image Composition DOS reserve failed.
369	EH IC DOS Release Failed	Image Composition DOS release failed.
370	EH IC File Delete Error	Image Composition image file delete error.
372	EH IC Too Many Images In Frame	Image Composition has too many images in the frame.
373	EH IC Trying To Insert Too Many Images Into Frame	Image Composition trying to insert too many images into frame.
374	EH IC Invalid Blip Request	Image Composition invalid blip type.
375	EH IC Memory Allocation Error	Image Composition memory allocation error.
376	EH IC Invalid Blip Level	Image Composition invalid blip index level.
379	EH DOS Disk Message Received No Transaction Active	Host application interface manager received a response from the Archive Writer when a command was not being processed.
380	EH DOS Disk Error In Retrieving Message	Host application interface manager error is receiving message from queue.
381	EH DOS Disk Response Not Received In Time	Host application interface manager error is not receiving response from system to a command within a specified period of time.
382	EH DOS Disk Invalid MsgQ Return Code	Host application interface manager unknown status in receiving from message queue.
383	EH DOS Disk Failed To Create Directories	Cannot create the DOS file system directories.
384	EH DOS Disk Failed To Create Disk	Cannot reserve the DOS file system.

#	Message	Explanation
386	EH DOS Disk Reservation Failed	Cannot reserve the DOS file system; cannot take semaphore protecting the file system.
387	EH DOS Disk Release Failed	Cannot release the DOS file system; cannot give semaphore protecting the file system.
388	EH SCSI Invalid Script Inst Int	Invalid instruction reported from SCSI I/O controller.
389	EH SCSI Invalid Script Entry Calculated	Internal programming error; could not determine what instruction the SCSI I/O controller should execute next.
390	EH SCSI Message Receive Error	SCSI target driver error in receiving message from queue.
391	EH SCSI Invalid Script Entry Requested	Internal programming error; determined an instruction for the SCSI I/O controller to execute that is not a valid instruction.
392	EH SCSI Invalid Script Inst Int Read	Invalid SCSI I/O processor instruction reported from the I/O processor.
394	EH Error Index Out Of Range	Error handler index out of range.
396	EH Negative Index	Cannot process negative index.
397	IA Sem Take Failed	Error entering IA critical region.
398	IA Sem Give Failed	Error exiting IA critical region.
399	IA Mem Alloc Failed	Unable to allocate memory for IA field.
403	CD MsgQ Send Error	Error sending to message queue.
404	CD Msg Data Length Error	Messaged data is longer than the message buffer size.
406	SCSI Error	
414	CC IMC Annotate Fail	
415	OI MsgQ Send Error	Error sending message to queue.
417	OI Window Invalid Index	Window index is larger than the number of declared windows.
418	OI State Invalid State	Key state received is larger than the number of declared states.
419	OI State Invalid Key	Key index received is larger that the number of declared keys.
420	OI MP Initialization Error	OI message processor initialization error.
421	OI MP MsgQ Receive Error	OI message processor error getting message from queue.

#	Message	Explanation
422	OI MP MsgQ Time Out Error	OI message processor timeout occurred waiting for message from queue.
423	OI MP Invalid MsgQ Return Code	
424	UI MsgQ Receive Error	UI send: error getting message from queue.
425	UI MsgQ Time Out Error	UI send: timeout occurred waiting for message from queue.
426	UI Invalid MsgQ Return Code	UI send: unknown return code received from messageGet.
427	UI Send Time Out Error	UI send: timeout waiting for command response from the UI.
428	UI MsgQ Send Error	Error sending message to queue.
429	UI Receive Initialization Error	Receive task initialization error.
430	UI Serial Port Read Error	Embedded controller error reading the UI serial port.
431	UI Receive Invalid Key Code	Receive unknown key code from the user interface.
432	CC OI Param Data1 Not Found	Parameter data not found in the message received.
433	CC OI Param Data2 Not Found	Parameter data not found in the message received.
434	CC OI Param Data3 Not Found	Parameter data not found in the message received.
435	CC OI Param Data4 Not Found	Parameter data not found in the message received.
436	CC OI Param Data5 Not Found	Parameter data not found in the message received.
437	CC OI Param Data6 Not Found	Parameter data not found in the message received.
438	EH Memory Malloc Error	Error log memory malloc failed.
440	EH IO No File To Append Image Log Entry	
441	EH System Startup Failed	
442	EH Sys Config File Release Failed	
443	EH Sys Config File Reserve Failed	

#	Message	Explanation
444	EH Set Environment Failed	
445	EH WRIB Invalid Operating Parameters	
449	IC Memory Malloc Error Struct	The image parameter data structure could not be created.
450	IC Memory Malloc Error Field	The image parameter data field could not be created. Call service.
451	PD Sem Give Failure	Unable to give semaphore to indicate system activity.
452	EH NET Cant Add Gateway	Could not add a gateway to the Gateway table that exist in the gateway destination list. Call service.
453	EH NET Cant Create Nfsdaemon	Failure of nfsdInit within NetworkInitializationManager. Call service.
454	EH NET Cant Export Filesystem	Failure of nfsExport within NetworkInitializationManager. Call service.
455	ST Unable To Read System Clock	The embedded firmware could not read/set the system clock. An error was returned from one of the system time/date routines. Call service.
456	ST Unable To Set System Clock	The embedded firmware could not read/set the system clock. An error was returned from one of the system time/date routines. Call service.
457	ST Unable To Set System Time	The embedded firmware could not read/set the system clock. An error was returned from one of the system time/date routines. Call service.
458	ST Unable To Set System Date	The embedded firmware could not read/set the system clock. An error was returned from one of the system time/date routines. Call service.
459	Diagnostic Serial Port Read Error	Diagnostic serial port read error.
460	LT MP Initialization Error	LT MP Initialization error.
461	LT MP MsgQ Receive Error	LT MP MSGQ receive error.
462	LT MP Invalid MsgQ Return Code	MP invalid MSGQ return error.
463	LT App Data Recv Error	LT APP data receive error.
464	LT App Data Recv Invalid MsgQ Return Code	LT APP data invalid MSGQ return code.
465	LT MP Unknown Scc Request	LT MP unknown SCC request.
466	LT MsgQ Send Error	LT MP MSGQ send error.

#	Message	Explanation
467	LT MP Cmd Time Out	LT MP CMD timeout.
468	LT MP Unknown Scc Msg 55 Prog Num	LT MP unknown MSG 55 prog num.
469	LT MP Unknown Scc Msg 52 Data Type	LT MP unknown MSG 52 data type.
470	EH Aim Reservation Failed	Unable to take AIM semaphore.
471	EH Aim Release Failed	Unable to release AIM semaphore.
472	EH DOS Disk Expected Priority Transaction	Writer is expecting a priority transaction and host sent normal transaction.
473	EH Transaction Already Active	The host tried to initiate a command using a transaction number of a currently active transaction.
474	EH Aimend Couldnt Signify Transaction Completion	Writer lost communication with host.
475	EH NET Cant Initialize Gateways	Failure to add gateway to Gateway table. Call service.
476	EH NET Cant Create Socket	Failure to create socket. Call service.
477	EH NET Cant Accept Transactions	Failure to accept connection on the input or output socket. Call service.
478	EH NET Cant Read In Socket	Failure to read from the input socket. Call service.
479	EH NET Cant Write Out Socket	Failure to write to the output socket. Call service.
480	EH II Too Many Image FDS	
481	EH FT Cant Transfer File	
700	EH WRIB Unexpected Pm2 Status Change	WRIB PM2 unexpected status.
701	EH WRIB Unexpected Image Decompression Done	WRIB unexpected Image Decompression Done.
702	EH WRIB Unknown WRIB Interrupt	WRIB unknown interrupt.
703	EH WRIB Unexpected Image Print Error	WRIB unexpected Image Print Error.
704	EH WRIB Unexpected Input Fifo Half Full	WRIB unexpected input FIFO half full.
705	EH WRIB No Images In Frame	WRIB download found no images in frame.
706:	EH WRIB Memory Full Too Long memory full	WRIB download failure.

#	Message	Explanation
707	EH WRIB Frame Too Large	WRIB download failure: frame dimensions outside limits.
708	EH WRIB Unknown Memory Full Exception	WRIB error: invalid memory result.
709	EH WRIB Download Unknown Status Change	WRIB error: unknown download status change.
710	EH WRIB Image Download Timeout	WRIB download failure: image download timeout.
711	EH WRIB Image Decompression Timeout	WRIB download failure: image decompression timeout.
712	EH WRIB Image Decompression Failure	WRIB download failure: image decompression failure.
713	EH WRIB Fifo Full Timeout	WRIB download failure: FIFO full timeout.
714	EH WRIB Pm2 Status Error	WRIB download failure: PM2 status error.
715	EH WRIB Unknown Decompress Exception	WRIB error: decompress exception.
716	EH WRIB Unknown Fifo Exception	WRIB error: FIFO exception.
717	EH WRIB Unknown Retry Error	WRIB error: retry error.
718	EH WRIB Output Unknown Status Change	WRIB error: frame output unknown status change.
719	EH IO Frame Print Timeout	Image Output failure: frame printing timeout.
720	EH IO WRIB Print Error	Image Output failure: WRIB print error.
721	Id Invalid Film Remaining Flag	
722	EH IC Annotation String Truncated	
901	FD Motor Failure	1700 board hardware motor failure. The error is reported if any of the motor voltage; current; or speed signals.
902	FD Invalid Command Received	Unknown or unimplemented command received by film drive servo from the embedded controller.
903	FD Counter Oscillator Error	1700 board hardware failure.
904	FD Aper Encoder Error	1700 board hardware failure.
905	FD Freq Miltiplier Error	1700 board hardware failure.
906	FD Servo Feedback Encdr Err	1700 board hardware failure.
907	FD Servo Amplifier Error	1700 board hardware failure.
908	FD Micro Controller Error	1700 board hardware failure.

#	Message	Explanation
909	FD Aper Source Switch Error	1700 board hardware failure.
910	FD Stop After Source Switch	1700 board hardware failure.
911	FD Diag Switches On	The 1700 PCB's dip switch pack S2 switch 1 has been moved to the On position.
912	FD Upper Supply Ecndr Error	1700 board hardware failure.
913	FD Lower Supply Ecndr Error	1700 board hardware failure.
914	FD Freq Multiplier Error	1700 board hardware failure.
915	FD Shutter Failed To Open	The shutter did not open within the specified time interval.
916	FD Shutter Failed To Close	The shutter did not close within the specified time interval.
917	FD Async Shutter Change	Unexpected change in the shutter position.
918	FD Invalid M Value	An invalid motor speed setting was sent to the film drive servo.
919	FD Cassette Data Update Error	A cassette reported an error with the data sent by the film drive servo. The problem could be caused by a weak battery in the cassette or a dirty IR transmitter or receiver. Replace the cassette battery and if the problem persists call service.
920	FD Servo Reset	Either the reset button was pressed or the film drive servo reset due to loss of power or a hardware error.
921	Received Unknown Async Error From FDS	1700 board hardware/software error.
922	EH DOS Disk Command File Close Failed	Cannot close command file.
923	EH DOS Disk Status File Write Failed	Cannot write status file.
924	EH DOS Disk Response File Write Failed	Cannot write response to a file.
925	EH DOS Disk Command File Delete Failed	Cannot delete command file.
927	EH SCSI Invalid Dma Status	Invalid status reported from SCSI I/O controller.
928	EH SCSI Unknown Dma Status	Unknown status reported from SCSI I/O controller.
929	EH SCSI Invalid SCSI Status	Invalid status reported from SCSI I/O controller.

#	Message	Explanation
930	EH Aimstart MsgQ Retrieve Error	
931	EH Aimstart MsgQ Send Error	Could not add a host to the Host table that exists on the boot line.
933	EH Aimwd Cannot Start Watchdog	
934	EH Aimwd Cannot Stop Watchdog	
935	EH Aimwd Reservation Failed	
936	EH Aimwd Release Failed	
937	EH Aimwd Cannot Give Timeout Semaphore	
938	FTP Buffer Malloc Failed	
939	FTP Connection Failed	
940	FTP Data Malloc Failed	
941	FTP Transfer Failed	
942	FTP Local File Create Failed	
943	EH Cant Create FT Daemon	
944	EH Cant Delete FT Daemon	
945	EH Cant Create FT Watchdog	
946	EH Cant Delete FT Watchdog	
947	EH FT Cant Close File	
948	EH FT Cant Delete File	
949	EH FT Cant Start Transfer Timer	
950	EH FT Cant Stop Transfer Timer	

7 Glossary

ActiveX
OLE 2.0 Custom Control.

Archive Writer
The internal name of the hardware that writes image files to microfilm. The official name is *Kodak Digital Science* Document Archive Writer, Model 4800.

AWIS
Archive Writer Interface Software. The official name is *Kodak Digital Science* Archive Writer Interface Software.

AWIS ActiveX
The OLE Custom Control that is used by the AWIS application to communicate with the AWOL software component.

AWIS Application
The Visual Basic application that uses the AWIS ActiveX to control the Archive Writer.

AWOL
Archive Writer Object Layer. This is a platform independent C++ object library that contains most of the functions that control the Archive Writer.

Digital Workstation
Microfilm retrieval device.

DLL
DLL is the acronym for dynamic link library, which is a generic software design concept intended to eliminate duplication of coding effort and increase productivity in the software industry. DLLs are based on the concept of most applications using standard routines. These common routines are stored on disk in one place – the dynamic link library, which conserves a computer's hard disk space and saves the effort of incorporating the routines into the various applications.

Film Template

A list of settings for a roll of microfilm. These settings consist of a number of index levels, channel offsets, polarity, orientation, etc. AWIS maintains a database of film templates that can be chosen from for writing film.

Header Page

Contains any kind of information supplied by the application. Written at the beginning of the film.

IMC

Image Management Code. Code written at the beginning of a roll of film. It automatically sets up the Digital Workstation with roll number, A and B channel positions, recovery code, zoom lens magnification, and image orientation.

Index Levels

The number of image levels on the film. Usually referred to as "Book, Chapter, Page," or "Block, Batch, Item." Image levels are indicated on the film by large, medium and small black image marks next to the images.

Index Template

A text file supplied by the host application, which is used by AWIS AWOL to create index information pages and write them to film.

List File

A text file containing a list of file specifications for image files. The image files are written to microfilm in the order they appear in the list file.

Property Page

Logical groupings of properties or variables that affect the operation of the control. Properties can be parameters that control how certain operations work, they can contain information that is a result of some condition or action, and cause actions to take place when they are set or changed.

Resolution Target
Images written to film for image quality use.

Session
The time during which the Archive Writer is actually running.

TIFF
Tagged Image File Format. A standard for storing image data in a file. Image data is stored as CCITT G3, G4.

Trailer Page
Contains any kind of information supplied by the application. Written at the end of the film.

Transfer File
A file created by the AWOL while writing the images to film. The file contains the original file name, page number within the file, roll number, and image address and is delimited by tabs. The information in the file can be used by external applications to update an external database. The creation of a transfer file is optional. The name of the transfer file matches the name of the roll as specified by the user.

Transfer File Template
A file containing text and tags supplied by the host application. AWIS AWOL substitutes values for the tags and appends the resulting information to the transfer file.

VAR
Value Added Reseller.

EASTMAN KODAK COMPANY
Document Imaging
Rochester, New York 14650

Printed on recycled paper.

DOCUMENT
IMAGING

Kodak

A-61057 2/2000

Printed in U.S.A.