MVSConnect 7.5

Technical User Guide

Documentation

Revision
3/24/2008

Contents

HOW TO USE THIS DOCUMENT

Introduction

This document contains information about how to use each module that is part of the MVSConnect family of products.

This document assumes you have installed Ascent Capture version 7 on your workstation. It also assumes you have a working knowledge of your computer hardware and operating system.

Items changed from the previous major version of the product are highlighted.

How This Document Is Organized

This document is divided into 2 Chapters and 6 Appendices, as described below:

Chapter 1 – BEFORE YOU BEGIN

This chapter provides a description of the main components necessary to run MVSConnect.

Chapter 2 – MVSCONNECT COMPONENTS, SETUP, AND USAGE

This chapter discusses each MVSConnect workstation component in detail.

Appendix A – REQUIRED ODBC ENTRIES

Explains the ODBC entries needed to run MVSConnect.

Appendix B – PLANNING WORKSHEET

Explains any other necessary events for running MVSConnect successfully.

Appendix C – MVSCONNET REJECT LOGIC

Explains how the Reject Logic in MVSValidation works.

Appendix D – SUMMARY OF IMAGE+ FIELDS USED IN MVSCONNECT

Gives a summary of all Image+ fields.

Appendix E – TROUBLESHOOTING

Gives examples of common errors and how to fix them.

Appendix F – TECHNICAL SUPPORT

Tells how to obtain Technical Support from Kofax and SYSCOM, Inc.

Reference Documents

Several sections of this document refer to the Getting Started Guide from Kofax's Ascent Capture. Having that document available will be helpful, as the document will reference it on several occasions.

CHAPTER 1 - BEFORE YOU BEGIN

Components of MVSConnect

MVSConnect has 2 main components:

MVSIndex

MVSIndex is the indexing component of MVSCONNECT, which connects to the FAF region through a TCP/IP communication protocol.

On the host, a CICS transaction (KFIP) is initiated, which then invokes the host module for indexing (depending on whether AIS+ or FWA is used).

All data and/or error messages passed from the workstation to the host can be viewed in a CICS temporary storage queue called TRAQKFX. TCP/IP error messages can be viewed in the TCP/IP transient data queue, which is part of the CICS job.

MVSStore

MVSStore is the store component of MVSConnect, which connects to the ODM region through a TCP/IP communication protocol. Once the data is passed by the workstation to the ODM region, all processing, logging of errors, etc. is done by ImagePlus ODM.

Note: Since ODM, which is a component of IBM ImagePlus, takes care of all store related activities, MVSConnect/MVSStore does not have the capability to log any errors that happen in the ODM region.

Troubleshooting

Checking the Kofax Trace Queue

The data exchanged between the MVSIndex client and host components can be viewed in the CICS temporary storage queue called TRAQKFX. In addition, when errors occur, they can be viewed in this queue also. To view this trace queue:

1. Go to the FAF CICS region and clear the screen.
2. Type CEBR TRAQKFX

This will display the MVSIndex trace queue.

Checking the CICS logs

Whenever data is sent up to the FAF region via TCP/IP protocol, this is logged in the CICS transient data queue, which is part of the CICS JOB. This is only available if the DCT entries have been set up for the CICS regions during the installation of CICS TCP/IP Sockets. Refer to **CICS TCP/IP Socket Interface Guide and Reference** for further information.

CHAPTER 2 - MVSConnect Components and Usage

Note: The components might differ slightly, depending on the release of each module installed.

MVSRelease

Overview

SYSCOM's MVSRelease for Ascent Capture is a "database" release script - that is, it uses a database to write data to before producing the required files for MVSIndex & MVSStore to store the image and its indexing data to the host. Running the MVSRelease installation package (Setup.exe) creates a new release script [**Ascent Capture ADO Database (MVS Modified)]** and install another SYSCOM program (MVSConverter.exe). These two programs will release the image and indexing data to be stored to the host.

Release Script Configuration

Each Batch Class created in Ascent Capture has a Release Script. For MVSConnect to store images properly, SYSCOM, Inc.'s Release script must be used. When creating the release script for a batch class, follow the steps below.

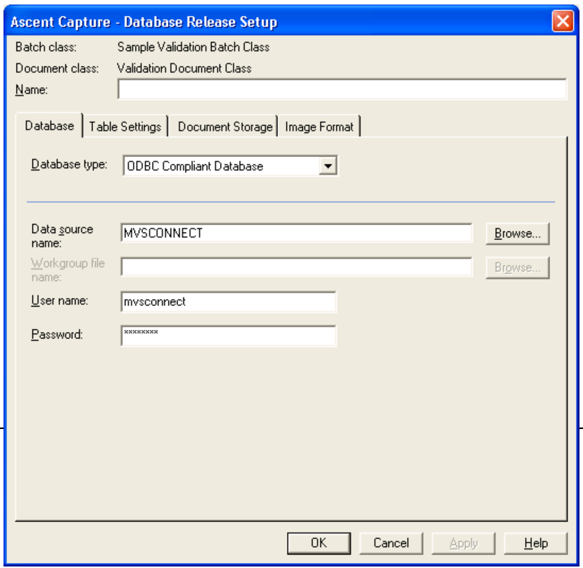
Note: Be sure that MVSRelease is installed on the PC that is being used to create the Batch Class.

The Microsoft Access Release is no longer available; please ensure you have an ODBC Connection defined. This connection may still point to an Access database.

The only purpose for the database definition is to define the fields for release purposes. The information is no longer written to the database.

1. Set up your document classes as described in the Ascent Capture documentation
2. For the Release Script, choose Ascent Capture Database Release.
3. On the Database tab, use the screen print below to enter the appropriate fields, with the following exception:

   3.1. For an ODBC Implementation. Select the database type **ODBC Compliant Database** and enter



Batch class: Sample Validation Batch Class
Document class: Validation Document Class
Name:
Database type: ODBC Compliant Database
Data source name: MVSCONNECT
Workgroup file name:
User name: mvsconnect
Password: ********

the Data Source Name set up in the ODBC Settings.

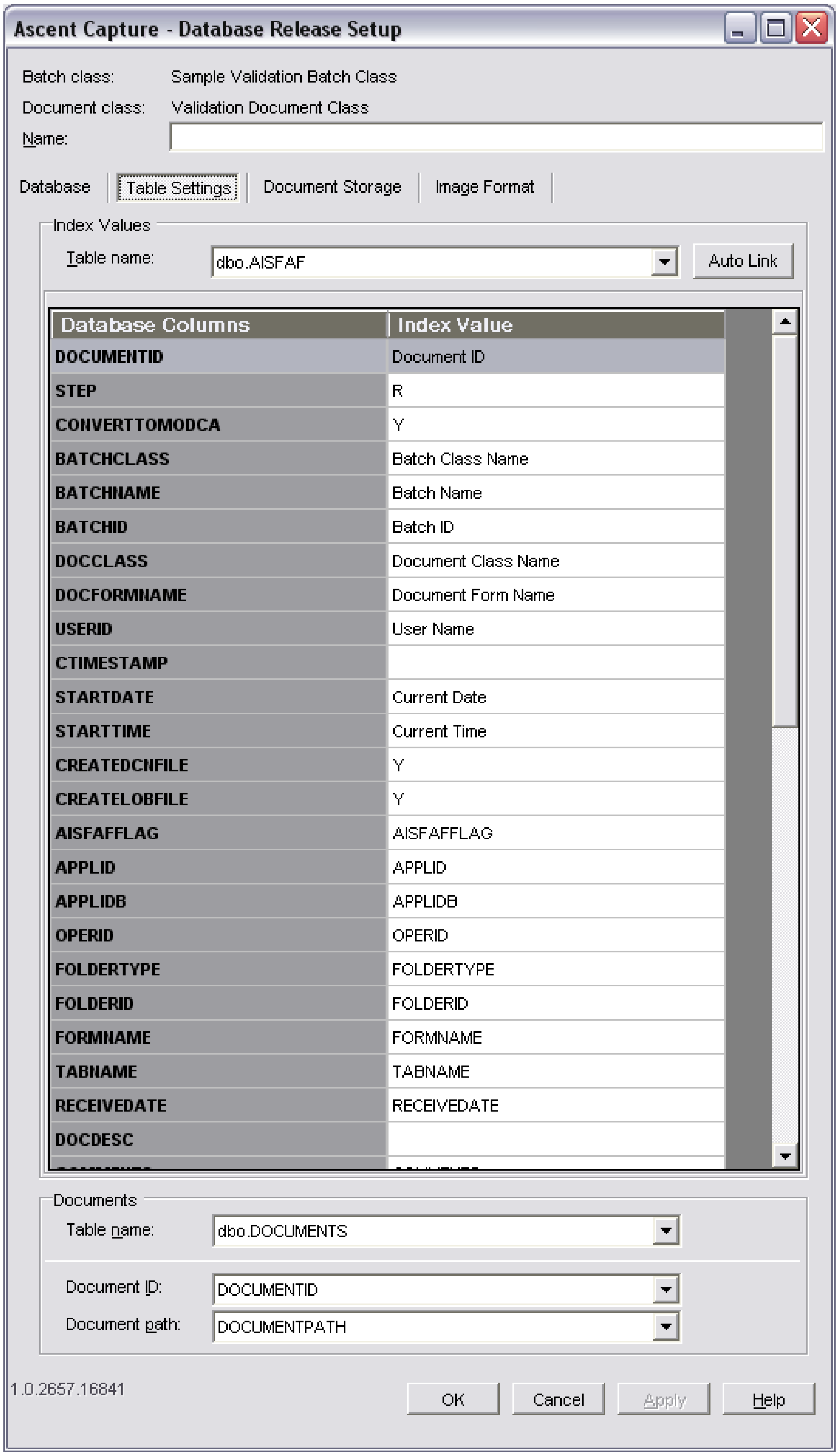
4. Next, choose the Table Settings tab, and enter the following data:

Index Values Section

Index Values, Table Name – choose "AISFAF"

Database Columns	Index Value
DOCUMENTID	Map to DOCUMENTID
STEP	Text Constant, type "R"
CONVERTTOMODCA	Text Constant, type "Y" to produce MODCA files, otherwise "N"
BATCHCLASS	Map Ascent Capture Values, Batch Class Name
BATCHNAME	Map Ascent Capture Values, Batch Name
BATCHID	Map Ascent Capture Values, Batch ID
DOCCLASS	Map Ascent Capture Values, Document Class Name
DOCFORMNAME	Map Ascent Capture Values, Document Form Name
USERID	Map Ascent Capture Values, User Name
CTIMESTAMP	(blank)
STARTDATE	Map Ascent Capture Values, Current Date
STARTTIME	Map Ascent Capture Values, Current Time
CREATEDCNFILE	Text Constant, type "Y" to produce a Document Class Name file, otherwise, set to "N"
CREATELOBFILE	Text Constant, type "Y" to produced an LOB (Line Of Business) file, otherwise, set to "N"

The remaining non-line-of-business fields are the standard set of AIS and FAF fields that data can be mapped to in Ascent Capture from defaults or entered data.



Batch class: Sample Validation Batch Class

Document class: Validation Document Class

Name:

Index Values

Table name: dbo.AISFAF

Database Columns	Index Value
DOCUMENTID	Document ID
STEP	R
CONVERTTOMODCA	Y
BATCHCLASS	Batch Class Name
BATCHNAME	Batch Name
BATCHID	Batch ID
DOCCLASS	Document Class Name
DOCFORMNAME	Document Form Name
USERID	User Name
CTIMESTAMP	
STARTDATE	Current Date
STARTTIME	Current Time
CREATEDCNFILE	Y
CREATELOBFILE	Y
AISFAFFLAG	AISFAFFLAG
APPLID	APPLID
APPLIDB	APPLIDB
OPERID	OPERID
FOLDERTYPE	FOLDERTYPE
FOLDERID	FOLDERID
FORMNAME	FORMNAME
TABNAME	TABNAME
RECEIVEDATE	RECEIVEDATE
DOCDESC	

Documents

Table name: dbo.DOCUMENTS

Document ID: DOCUMENTID

Document path: DOCUMENTPATH

1.0.2657.16841

Please note the change to the setup screen, especially the **Auto Link** button. Pressing this will automatically link exactly named fields together, thereby making setup easier.

Documents Section

Table Name – Choose “DOCUMENTS”
Document ID - Choose “DOCUMENTID”
Document Path – Choose “DOCUMENTPATH”

Next, choose the Document Storage tab. In the Release Directory field, enter or choose the directory where the documents will be released. The Release script has now been completed.

MVSConverter

MVSConverter.dll is the other file that is installed when running MVSRelease. This program will do any necessary image conversion as well as produce all necessary files needed to store the image to the host.

Converting Color Images to/from TIF to MOD:CA

The MVSConverter application detects whether a document is pure Black and White. If it is, it uses Accusoft™ to convert the black and white images to MOD:CA.

In the case where it is not a pure Black and White document, the application (using Accusoft™) saves the pages which are black and white to TIFF and the non-Black and White images to JPEG. The application then calls the IBM Images Services ™ Library to convert the TIFF and JPEG images to MOD:CA.

Running release as a service

Ascent Capture's Release module can run as a service under Windows 2000 or later. SYSCOM, Inc.'s Release modules can also be run as a service. To set services; please refer to the Installing Services section of the Ascent Capture Getting Started Guide. The only additional step needed to allow MVSRelease to run as a service is to enable the option "Allow Service to Interact with Desktop". Without this option, MVSRelease will not run properly as a service.

MVSDatabase

Overview

MVSConnect uses a database release script to release from Ascent Capture. The database is an Access97 database and is installed with SetupMDB.exe. The two Access 2000 databases are: MVSConct.MDB and MVSHist.MDB. MVSConct.MDB is the major database that is used to properly create the index data file that is stored to the host. It is also used to house any reconciliation data from each module if this feature is being used. MVSHist.MDB is a copy of the MVSConct.MDB database and is used as an archive of any reconciliation data from each module.

Additionally, a script is provided for a MVSConct SQL Server database. The SQL Server application is the recommended implementation for this version as MVS File Distribution Manager, MVS Index, and MVS Store can run as services.

Setup

The only setup required for the databases is to setup the ODBC configurations on each workstation. These ODBC Configurations are located in Appendix A of this document.

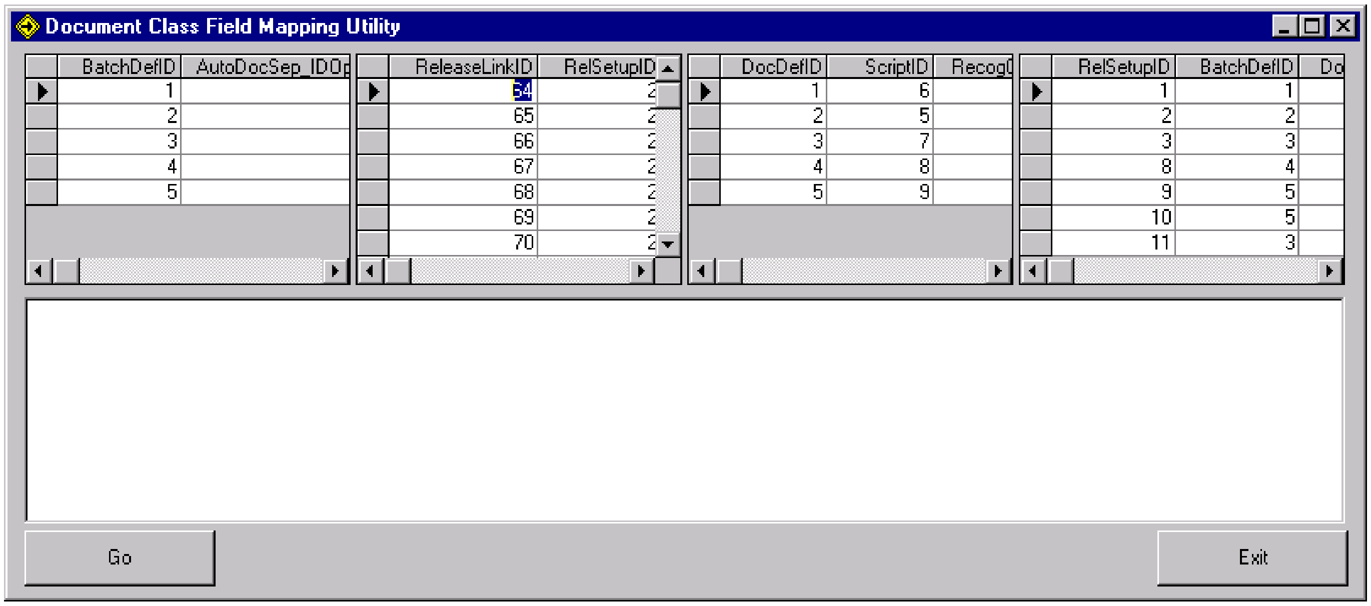
ACMapper Program (if MVSAdmin has been purchased)

Overview

The ACMapper module is a SYSCOM, Inc. module that takes your document classes from Ascent Capture's Administration database and consolidates necessary information into the DOCCLASSES table in the MVSConnect database. If you have purchased MVSAdmin, after a batch class has been created or edited, ACMapper must be run to update the MVSConnect database with the changes.

Operation

The ACMapper module looks like this:



BatchDefID	AutoDocSep_IDOp
1	
2	
3	
4	
5	

ReleaseLinkID	RelSetupID
64	2
65	2
66	2
67	2
68	2
69	2
70	2

DocDefID	ScriptID	Recog
1	6	
2	5	
3	7	
4	8	
5	9	

RelSetupID	BatchDefID	Do
1	1	
2	2	
3	3	
8	4	
9	5	
10	5	
11	3	

Click the GO button to begin the ACMapper utility. The program will ask you if you wish to purge old data. You should generally answer YES to this question.

You MUST NOT have any Ascent Validation (Indexing) activity occurring when you run this program.

Additionally, you should shut down any Ascent Index modules that are running prior to running this program, and it is recommended that you reboot any workstations that are running the Ascent Index module after you make doc class changes. This will ensure that your changes are utilized by those workstations.

MVSValidation

Overview

MVSValidation is a *SYSCOM, Inc.* module designed to save time and generate efficiency in the document imaging process. It incorporates newly enhanced mainframe functionality with workstation ease. An indexer can validate his or her work against mainframe DB2 tables directly at the time of input, and correct immediately, thus generating efficiency and saving time.

Batch Class setup for MVSValidation

FIELD TYPES FOR MVSVALIDATION

In Ascent Capture, field types are used to define index fields. For each field type, you specify the data type and any other information associated with that data type. For more on field types, see the Ascent Capture documentation.

For MVSValidation, there are several field types that must be created. All possible field types are listed in the table below, with the mandatory field types preceded by an asterisk.

Note: A sample batch class is located on the MVSConnect CD, titled "Sample Validation Batch Class.cab". Importing this file will create all necessary field types listed below. This is not necessary, but it can save time and eliminate the chance for errors creating the field types.

FIELD NAME	FIELD DESCRIPTION	LENGTH (VARCHAR DATA TYPE)
*AISFAFFlag	AISFAFFlag	1
*ApplID	Application ID	8
*ApplIDB	Application ID Binary	2
*OperatorID	Operator ID	8
*FolderType	Folder Type	8
*FolderID	Folder ID	26
TabName	Tab Name	16
ReceiveDate	Document Receive Date	8
DocumentDesc	Description of the Document	60
Comments	Comments	120
FolderDesc	Description of the Folder	60
SecInd1	SecIDX1	40
SecInd2	SecIDX2	40
SecInd3	SecIDX3	40
RouteFlag	Routing Flag	1
RLOB	RLOB	6
TranType	Transaction Type	6
RouteCode	Routing Code	6

FIELD NAME	FIELD DESCRIPTION	LENGTH (VARCHAR DATA TYPE)
UnitCode	Unit Code	4
PriorityInd	Priority Indicator	1
HoldDate	Hold Date	8
HoldTime	Hold Time	4
AssignUser	Assigned User	8
PaperKeptFlag	Paper Kept Flag	1
MaxPriority	Maxium Priority	3
ExtendedComments	Extended Comments	240
*FormName	Form Name	16

INDEX FIELDS

Index fields are setup in Ascent Capture as the fields that will be captured and sent to the host as indexing data for each image. After setting the field types, choose the necessary index fields into the batch class (see the Ascent Capture documentation for assistance entering index fields). As with field types, there are a few index fields that are mandatory for MVSValidation. They are:

- AISFAFFlag
- ApplID
- ApplIDB
- FolderType
- FolderID
- OperatorID
- FormName

Note: In order for MVSValidation to work properly, FormName must be the last index field in the list. Also, all index field names must be exactly as the field name is listed in the field types list above.

Any other index fields that will be used must also be entered prior to the FormName field, using the field names listed in the field types table above.

CREATING THE VALIDATION SCRIPT

Ascent Capture has validation scripts that are available for each batch class using the Validation module. MVSValidation has its own validation script that must be entered into the batch class to call the MVSValidation program. On the MVSConnect CD, there are two validation scripts:

- Validation Script no Recon.txt – to be used for MVSValidation without writing to the reconciliation database
- Validation Script.txt – to be used for MVSValidation when writing to the reconciliation database

The appropriate Validation script must be manually copied into the batch class being used for MVSValidation. To do this, follow the steps below:

1. Open the appropriate Validation script from the CD.
2. Choose Select All from the Edit menu.
3. Choose Copy from the Edit menu.
4. Create a new index\validation script in the Ascent Capture Administration module for the Batch Class that will be using MVSValidation.
5. Choose Select All from the Edit menu on the Validation script screen
6. Choose Paste from the Edit menu on the Validation to paste the new Validation script over the existing one.
7. Compile, save, and close the Index Script.
8. Publish the batch.

The Validation.txt Index Script has now been installed and is ready for MVSValidation. For more information on Validation Scripts, see the Ascent Capture documentation.

CREATING THE RELEASE SCRIPT

The release script for a Batch Class using MVSValidation is similar to any other release script. Each index field created in the steps above should be mapped to its corresponding Database Column in the Table Settings section of the Release Script. For more information on setting up the release script, please refer to the Ascent Capture manual.

MVSValidation Configuration File

The configuration file for MVSValidation is Validation.INI, located in the C:\WINNT\SYSTEM32 directory. Below is a list of each entry in the configuration file, along with its meaning and accepted values.

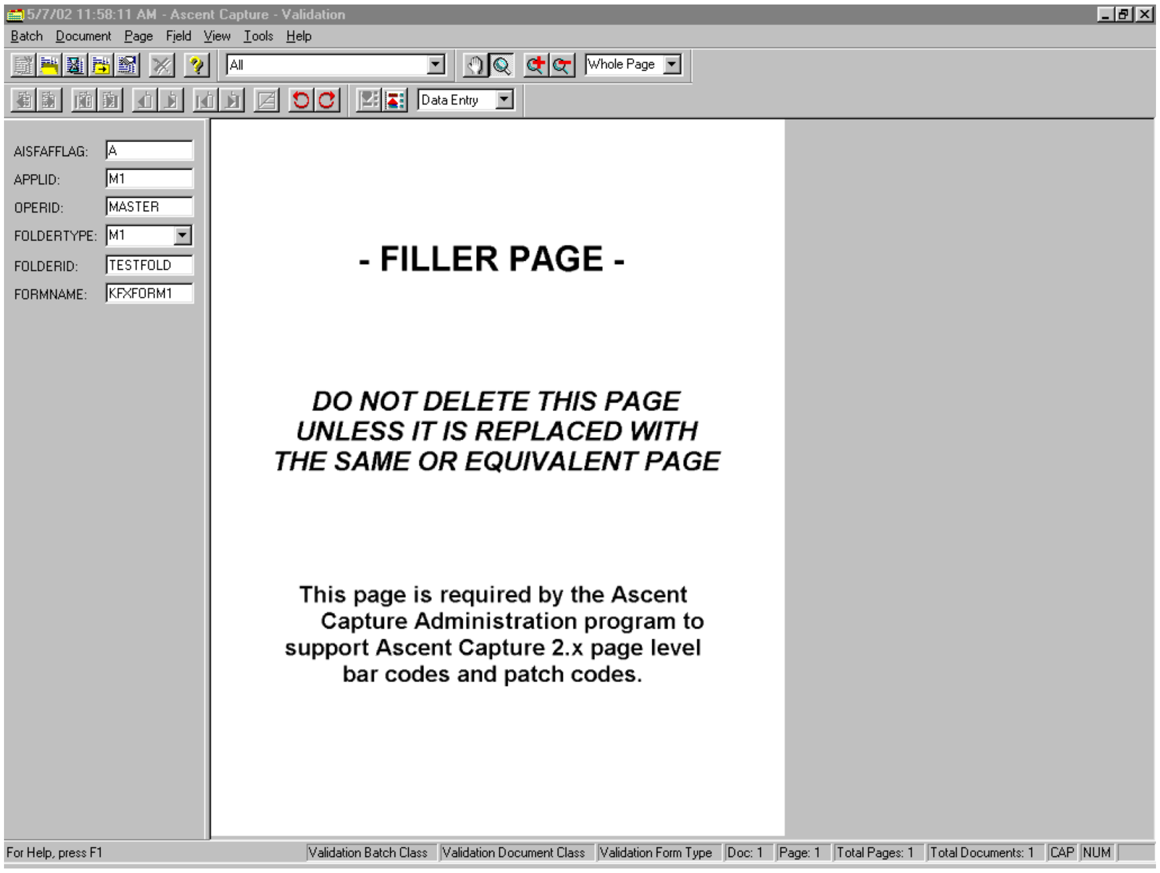
RUNOPTIONCODE	Run Option Code. This value tells the mainframe to only do validation. **MUST BE SET TO 01 FOR HOST VALIDATION.**
PAUSE	Pause time. This value sets the number of milliseconds for the processing of files to take place. **SHOULD BE SET TO ZERO UNLESS THERE IS A SPEED ISSUE GOING TO THE HOST.**
TRACEFILE	File used for tracing.
TRACEFILESIZE	Maximum size (in bytes) that the trace file can grow before creating a backup of the file.
FOLDERFLAG	Folder Flag. If set to "**Y**", the FolderType value is set to the first 2 characters of the FolderID. Otherwise, type "**N**", which requires that a FolderType value be entered at Validation time.
DEFAULT_APPLID	Default Application ID to be used for all documents processed. Can be left blank if no default ApplID is being used.
FILENAMETOSEND	This is the file written out by MVSValidation to be sent to the host for validation.
FILENAMETORECEIVE	This is the file returned from the host.

DEFAULT_TO_ERROR_FIELD	Set to “**Y”** to highlight the field in error after the host validates the data. Otherwise, set to “**N**”
TPNAME	Transaction name on the host (APPC connection only; likely set to **KFXM – no longer user**)
IMAGE_PLUS_IPADDRESS	The IP Address being used for Image Plus. If the AISFAFFlag=”C”, then use the Content Manager IP Address and Port. Otherwise, use the Image Plus IP Address and Port.
IMAGE_PLUS_PORT	Port being used for Image Plus (TCPIP connection only)
CONTENT_MANAGER_IPADDRESS	The IP Address being used for Content Manager. If the AISFAFFlag=”C”, then use the Content Manager IP Address and Port. Otherwise, use the Image Plus IP Address and Port.
CONTENT_MANAGER_PORT	Port being used for Content Manager (TCPIP connection only)
TCPTIMEOUT	Seconds the application should timeout if no connection to the host is made (TCPIP connection only)
FORM_LOCATION	Location of MVSValidation window on the screen. Enter **RIGHT**, **CENTER**, or **LEFT**.
USE_REJECT_LOGIC	Set to **NO** unless using MVSConnect Reject Logic, discussed in Appendix C
REJECT_CODE_FILE	Blank unless using MVSConnect Reject Logic, discussed in Appendix C
DISPLAY_USEREXITDATA	Set to **YES** to display the user exit data otherwise set to **NO**.
BUTTON1	Text to display on Button1 (Index Button) of the MVSValidation screen. If left blank, the default is **INDEX**.
BUTTON2	Text to display on Button2 (Store Button) of the MVSValidation screen. If left blank, the default is **STORE**.
BUTTON3	Text to display on Button3 (Cancel Button) of the MVSValidation screen. If left blank, the default is **CANCEL**.
BUTTON4	Blank unless using customized reject logic. Text to display on Button4 (Reject Button) of the MVSValidation screen. If left blank, the default is **REJECT**.

This completes the setup of a batch class using MVSValidation. Publish the batch class, and it is now ready to be used. Below are instructions on how to use MVSValidation.

Operation

A document that was scanned with a batch class set up for MVSValidation will enter into the Ascent Capture Validation queue as any other document would. The user will enter indexing data associated with the document, as seen in the screen print below:


5/7/02 11:58:11 AM - Ascent Capture - Validation
Batch Document Page Field View Tools Help
All
Whole Page
Data Entry

AISFAFFLAG: A
APPLID: M1
OPERID: MASTER
FOLDERTYPE: M1
FOLDERID: TESTFOLD
FORMNAME: KFXFORM1

- FILLER PAGE -

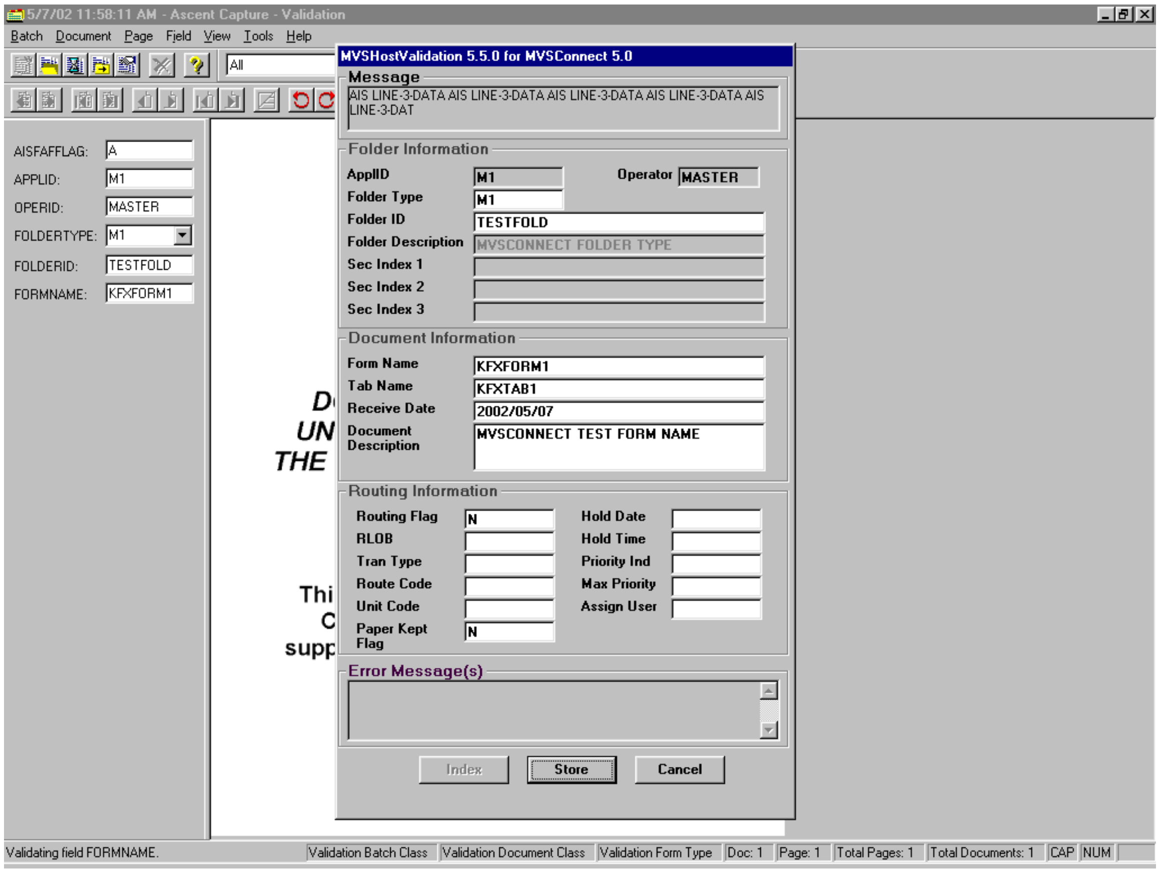
DO NOT DELETE THIS PAGE UNLESS IT IS REPLACED WITH THE SAME OR EQUIVALENT PAGE

This page is required by the Ascent Capture Administration program to support Ascent Capture 2.x page level bar codes and patch codes.

For Help, press F1 | Validation Batch Class | Validation Document Class | Validation Form Type | Doc: 1 | Page: 1 | Total Pages: 1 | Total Documents: 1 | CAP | NUM

Figure: Ascent Index Screen

When the user tabs out of the FormName field, the screen will then show the MVSValidation screen, as seen in the following screen print:


5/7/02 11:58:11 AM - Ascent Capture - Validation
Batch Document Page Field View Tools Help
All
D
UN
THE
Thi
C
supp
Index
Store
Cancel

AISFAFFLAG:	A
APPLID:	M1
OPERID:	MASTER
FOLDERTYPE:	M1
FOLDERID:	TESTFOLD
FORMNAME:	KFXFORM1

MVSHostValidation 5.5.0 for MVSConnect 5.0

Message

AIS LINE-3-DATA AIS LINE-3-DATA AIS LINE-3-DATA AIS LINE-3-DATA AIS LINE-3-DATA AIS LINE-3-DAT

Folder Information

ApplID	M1	Operator	MASTER
Folder Type	M1		
Folder ID	TESTFOLD		
Folder Description	MVSCONNECT FOLDER TYPE		
Sec Index 1			
Sec Index 2			
Sec Index 3			

Document Information

Form Name	KFXFORM1
Tab Name	KFXTAB1
Receive Date	2002/05/07
Document Description	MVSCONNECT TEST FORM NAME

Routing Information

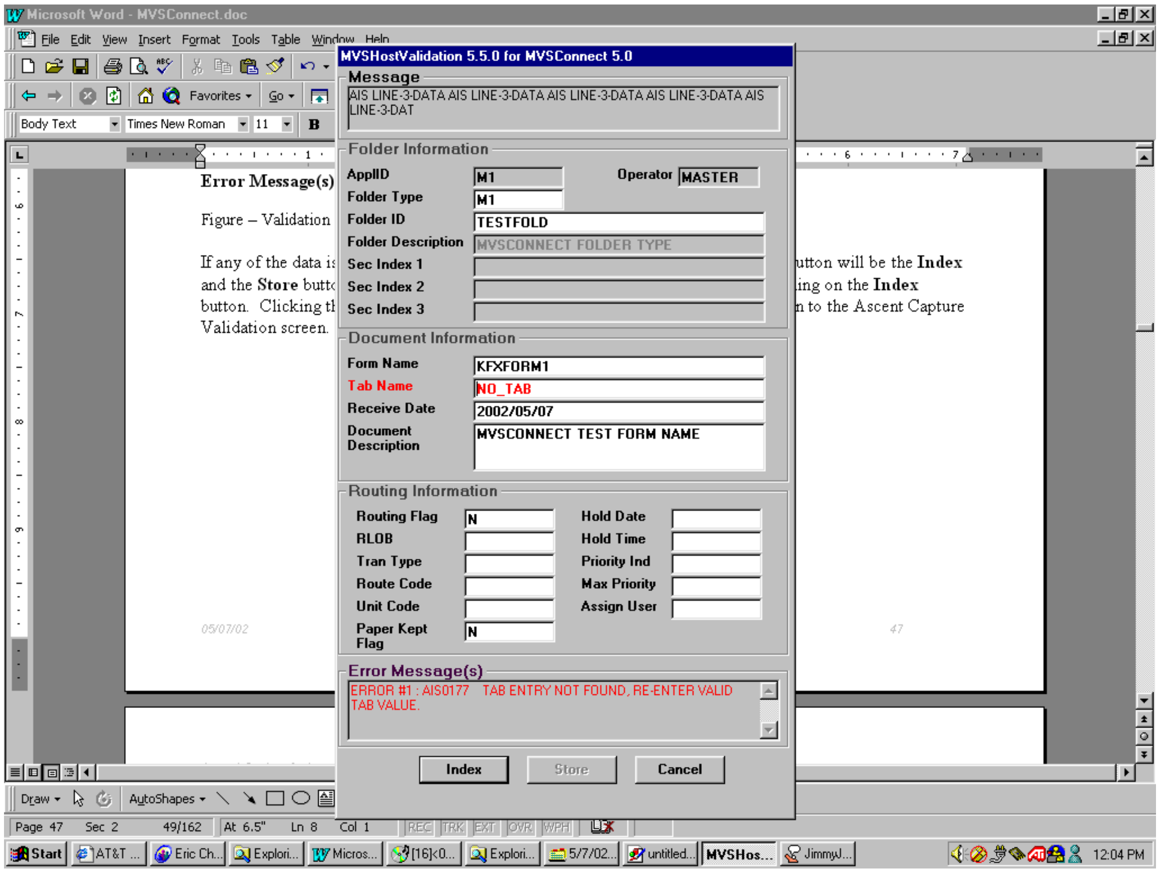
Routing Flag	N	Hold Date	
RLOB		Hold Time	
Tran Type		Priority Ind	
Route Code		Max Priority	
Unit Code		Assign User	
Paper Kept Flag	N		

Error Message(s)

Validating field FORMNAME. | Validation Batch Class | Validation Document Class | Validation Form Type | Doc: 1 | Page: 1 | Total Pages: 1 | Total Documents: 1 | CAP | NUM

Figure – Validation Screen

If the data entered is valid, the **Store** button will be enabled, while the **Index** button will be disabled. Clicking the **Store** button means all data is valid and this document is ready to be released and uploaded to the host. If there is any invalid data, the invalid field will turn **red**, and a message will display in the **Error Message(s)** box describing the error, as seen in the screen-print below:


MVSHostValidation 5.5.0 for MVSConnect 5.0

Message
AIS LINE-3-DATA AIS LINE-3-DATA AIS LINE-3-DATA AIS LINE-3-DATA AIS LINE-3-DAT

Folder Information
ApplID: M1
Operator: MASTER
Folder Type: M1
Folder ID: TESTFOLD
Folder Description: MVSCONNECT FOLDER TYPE
Sec Index 1:
Sec Index 2:
Sec Index 3:

Document Information
Form Name: KFXFORM1
Tab Name: NO_TAB
Receive Date: 2002/05/07
Document Description: MVSCONNECT TEST FORM NAME

Routing Information
Routing Flag: N
RLOB:
Tran Type:
Route Code:
Unit Code:
Paper Kept Flag: N
Hold Date:
Hold Time:
Priority Ind:
Max Priority:
Assign User:

Error Message(s)
ERROR #1 : AIS0177 TAB ENTRY NOT FOUND, RE-ENTER VALID TAB VALUE.

Figure – Validation Screen with an error

If any of the data is changed, or if there is an error validating the data, the default button will be the **Index** and the **Store** button gets disabled. The data can be edited and revalidated by clicking on the **Index** button. Clicking the **Cancel** button will close the MVSValidation screen and return to the Ascent Capture Validation screen.

MVS Error Viewer

Overview

When a document fails at MVSIndex time, messages are sent back to the workstation from the host indicating what the error is. The MVSErrorViewer application allows you to look at these errors along with the image to determine what the problem is, fix the necessary fields and accept the changes for subsequent index and storage back to the host. If there is a host error, simply resending the document after the host error has been resolved will work. But if it is bad data, this data can be fixed on the MVSErrorViewer screen and resent to the host. Also, any changes made to the index fields can be validated against the host in MVSErrorViewer. This feature will be discussed below.

MVSErrorViewer can be used in two different styles:

1. The user can select an error file to correct from a list of all documents in error (Select Method), or
2. The application can display the next error on the screen automatically for the user to update (Push Method).

Each of these styles will be described below.

Profiles

New in Version 7.5

Profiles were introduced in Version 7.5 as a way to minimize the 'clutter' on the desktop. In previous versions, to view different directories, the installer would use the package to install the initial instance and then copy what was installed into separate folders. Additionally, they would place a new icon on the desktop for the user to point to the new directory.

In this version (and all future versions), the concept of profiles exists. This allows the installer to install a single version of the application on the workstation and copy over the 'master' configuration file with the profiles installed.

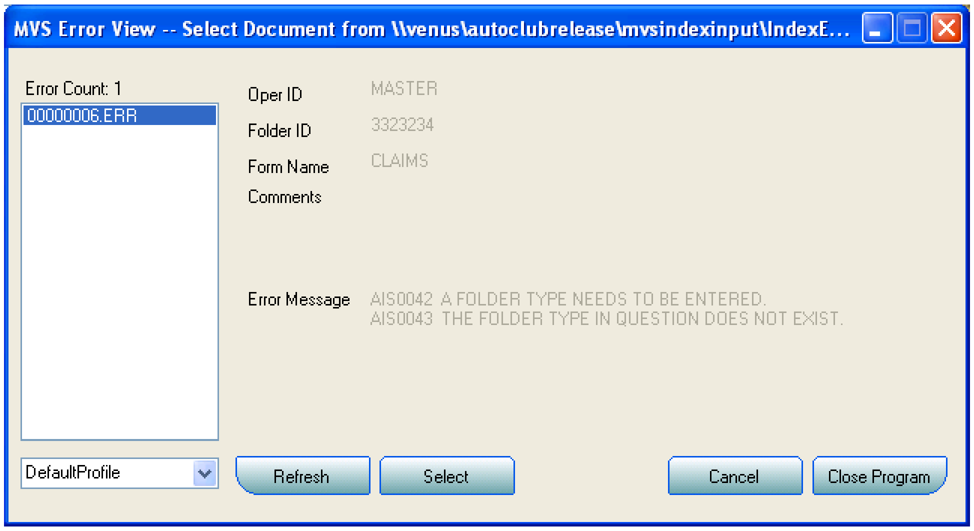
There are two places to change the profile being worked. The first is the Document Selection Screen. This option is only available if the current profile allows the user to select a document. The second method is via the tools menu, described later in this document.

Document Selection Screen

OVERVIEW

This window displays only when the user is in the *Select Method*. Users who are in a *Push Method* will go directly to the *Document Editing Window.* This screen allows the user to select a document to work, or to change the profile.

FUNCTIONALITY


MVS Error View -- Select Document from \\venus\autoclubrelease\mvsindexinput\IndexE...

Error Count: 1
00000006.ERR

Oper ID MASTER
Folder ID 3323234
Form Name CLAIMS
Comments

Error Message AIS0042 A FOLDER TYPE NEEDS TO BE ENTERED.
AIS0043 THE FOLDER TYPE IN QUESTION DOES NOT EXIST.

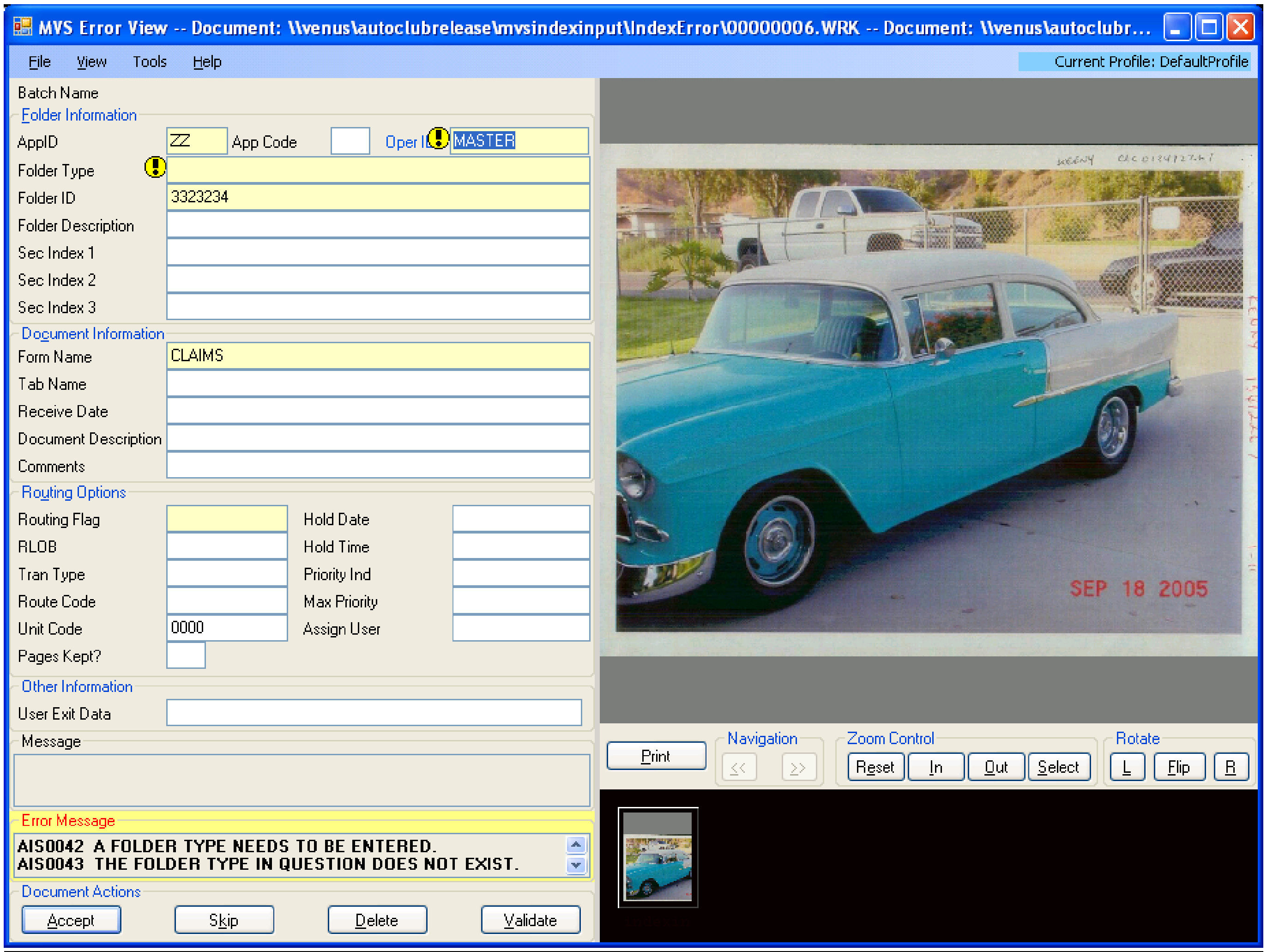
Working from the left side of the window:

- The error count relates to the error count for that directory and not to other error directories, if any.
- Underneath the ErrorCount are the files within the selected directory.
- Finally, the profile selection pull-down. This pull-down may not exist if there is only a single profile assigned to your environment.
- The title bar displays the error directory searched.
- The middle of the panel displays the index information for the selected document.
- The refresh button refreshes the error list.
- The select buton selects the document highlighted.
- The cancel button cancels this screen and returns the user to the document editing window.
- The close program button closes the application.

When you select a profile, the error list would change to match that profile. **NB:** If the profile selected is a 'push' profile, the user will be placed directly into the Document Editing window.

Document Editing Window

Once a document is selected, the following screen is displayed. From this screen the user is able to fix the errors that occurred during *MVSIndex* time. Below is a list of each different section of *MVSErorrViewer*. The current profile appears in the upper right hand corner (in this example, it is DefaultProfile).



MVS Error View -- Document: \\venus\autoclubrelease\mvsindexinput\IndexError\00000006.WRK -- Document: \\venus\autoclubr...

Current Profile: DefaultProfile

Batch Name	
AppID	ZZ
App Code	
Oper ID	MASTER
Folder Type	
Folder ID	3323234
Folder Description	
Sec Index 1	
Sec Index 2	
Sec Index 3	
Form Name	CLAIMS
Tab Name	
Receive Date	
Document Description	
Comments	
Routing Flag	
RLOB	
Tran Type	
Route Code	
Unit Code	0000
Pages Kept?	
Hold Date	
Hold Time	
Priority Ind	
Max Priority	
Assign User	
User Exit Data	

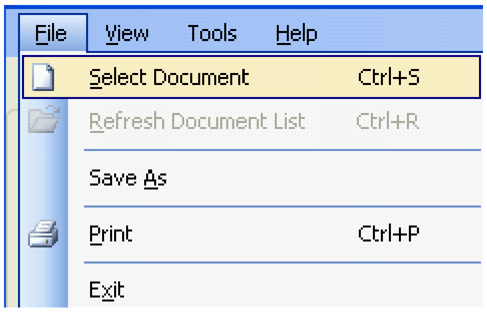
```
AIS0042  A FOLDER TYPE NEEDS TO BE ENTERED.
AIS0043  THE FOLDER TYPE IN QUESTION DOES NOT EXIST.
```

<u>SELECTED FIELD DESCRIPTIONS</u>

- **Error Messages -** This box displays the error message that corresponds to the selected Index Field. ImagePlus may send back multiple error messages and each message is shown in this box, with the corresponding index field has an exclamation mark next to it..
- **Accept Button -** This button is pressed once all errors have been corrected and the user wishes to re-index the current document. Performing this action takes the Index Field information and the document image and places them into the Release directory that was specified in the Path Setup screen. Either clicking it with the mouse or pressing the ALT-A key sequence can access this button.
- **Skip Button -** This button is pressed when the user wishes to skip this document and move onto the next document in error, without deleting the associated files. This allows the user to find the information necessary to fix the errors and re-index the document properly. This skipped document will be brought back into the MVSErrorView application the next time it is started. Either clicking it with the mouse or pressing the ALT-S key sequence can access this button.

- **Delete Button -** This button is pressed when the user wishes to remove this document from the system and move onto the next document in error. Performing this action deletes the Index Field file, the document image, and all other associated files from the directory. To get this document back into the system it must be rescanned and re-indexed. Either clicking it with the mouse or pressing the ALT-R key sequence can access this button.
- **Print Button** – This button is used to send the document to the printer to obtain a hard copy.
- **Validation Button** – This button is used to validate data in the index fields against the host (if this function is being used). When pressed, all index data is sent to the host for validity. The host will return any error messages to the **Error Messages** section of the screen, and highlight the invalid field. If no errors are returned, nothing will be displayed in the **Error Messages** field. More detail on validation is discussed later in this section.
- **Greater Than/Less Than Button -** These buttons allow the user to Page Up/Down within a multi-page image. The Less Than ('<<') button moves to the next page in the document, while the Greater Than ('>>') button moves to the preceding page.
- **In/Out Button** - These buttons allow the user to Zoom In/Out on the image in the window to make it easier to see the information. The **In** button increases the magnification while the **Out** button decreases the magnification.
- **Reset Button** – This button resets the zoom level to the default zoom level that was established on the configuration screen (shown below).
- **Select Button** – This button allows the user to 'draw' a rectangle around a selection and zooms into that rectangle. Please note: The rectangle will always be in proportion to the original image.
- **Rotate Buttons –** These buttons rotate the document clockwise (**R**ight), counter-clockwise (**L**eft), and upside down (**Flip**).
- The thumbnails exist underneath the image so that the user can quickly pick the document to view.

File Menu



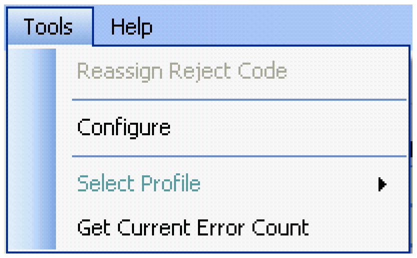
The **File Menu** has several entries.

- If the current profile allows the user to select a document, the **Select Document** option is available. This brings the user back to the Document Selection Screen.
- If the current profile does not allow the user to select a document then the **Select Document** option is not available, however the **Refresh Document List** is available. When the user selects this option, the application repopulates the internal list with the current directory contents.
- **Save As** allows the user to save a 'soft' copy of the document.
- **Print Document** sends a hardcopy of the document to the printer.
- **Exit Application** closes *MVSErrorViewer*.

View Menu

The view menu allows the user to do the same operations as they code by pressing the buttons, such as rotate, zoom, move between pages.

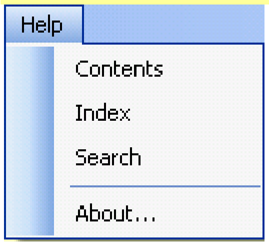
Tools Menu



The tools menu has several options

- Reassign Reject Codes. This feature is enabled only when reject files (or database entries) exist. Generally this option is disabled.
- Configure. The function allows the user to create and update profile information. **This function should only be executed by an administrator to avoid configuration problems.**
- Select Profile. When enabled, the user clicks on the right arrow which brings up a drop-down list of the profiles. The user then selects the desired one from the list. If there is an open document, then they are prompted to skip the current document.
- Get Current Error Count. This provides the current error count in the selected error directory.

Help Menu



The help menu has several options:

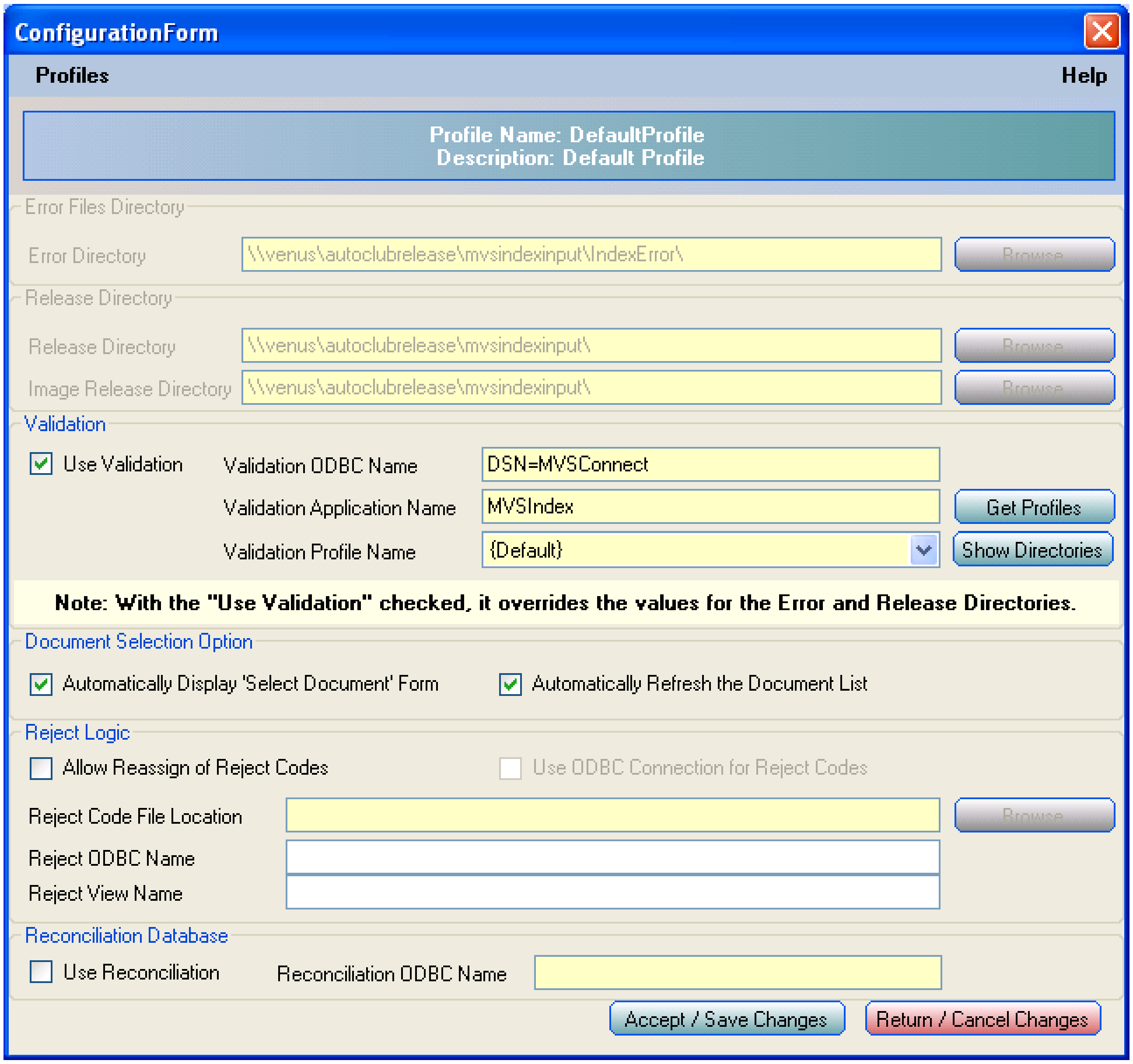
- Contents brings up the content view of the help file
- Index brings up the index view of the help file.
- Search brings up the search view of the help file.
- About displays the version information for the application.

Configuration Window

Overview

The configuration process has been completely revised for Version 7.5. The original model where all of the data exists in a configuration file on the workstation was also changed.

As version 7.5 includes services to run in the background for MVS File Distribution Manager, MVS Index, and MVS Store, the parameters for these applications exist in a database. The profiles on the workstations should hook into those parameters. In future versions, the functionality to read the directory structures from a local configuration file will be removed.



Profile Name: DefaultProfile
Description: Default Profile

Error Files Directory

Error Directory: \\venus\autoclubrelease\mvsindexinput\IndexError\

Release Directory

Release Directory: \\venus\autoclubrelease\mvsindexinput\

Image Release Directory: \\venus\autoclubrelease\mvsindexinput\

Validation

[✓] Use Validation

Validation ODBC Name: DSN=MVSConnect

Validation Application Name: MVSIndex

Validation Profile Name: {Default}

Note: With the "Use Validation" checked, it overrides the values for the Error and Release Directories.

Document Selection Option

[✓] Automatically Display 'Select Document' Form

[✓] Automatically Refresh the Document List

Reject Logic

[ ] Allow Reassign of Reject Codes

[ ] Use ODBC Connection for Reject Codes

Reject Code File Location:

Reject ODBC Name:

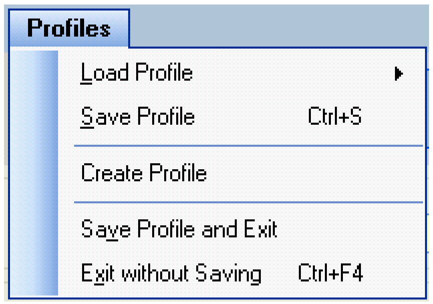
Reject View Name:

Reconciliation Database

[ ] Use Reconciliation

Reconciliation ODBC Name:

Profile Menu



The profile menu has several options:

- Load Profile. This allows a user to select an existing profile from a pull down. This functionality is the same as if the user did this in the *Select Profile* option on the Document Selection Window.
- Save Profile. This saves the current information to the local configuration file.
- Create Profile creates a new local profile.
- Save Profile and Exit saves the current information to the local configuration file and exits the window.
- Exit without saving closes the window without saving.

Configuring the Profile

Validation Section

The **Use Validation** check box indicates that the application should load the parameters from the database. The application then uses the information for the specified application to populate the Error, Release, and Image Release directories.

The **Validation ODBC Name** indicates the connection on the workstation used to get to the database.

The **Validation Application Name** is the application name used by the MVS Index applications.

The **Get Profiles** button will retrieve **the database** profiles associated with the Validation Application Name.

The **Validation Profile Name** pull-down is populated when the user presses the Get Profiles button.

The **Show Directories** button retrieves the directories and displays them above the Validation panel.

Document Selection Option

With the **Automatically Display 'Selected Document' Form** checked, when the user starts the application and when they close a document, the Document Selection screen displays.

To decrease the network impact of the application, the application gathers the available documents when it starts. The user works an internal list of files until it is exhausted, any files which are no longer on the network are ignored. If the user selects the **Automatically Refresh the Document List** option, whenever a document completes, the application gathers a list of the files again.

Reject Logic

Reject logic allows the user to use a standard set of reject responses for a document. This function is generally not used within Error View and is the purview of the Validation module.

Allow Reassign of Reject Codes, when checked, allows the user to reject a document. Generally this is done through a **reject Code File Location** located on the network. However, users could find this information in a database.

To use the database option, check the **Use ODBC Connection for Reject Codes** box and enter the ODBC Name and the View name within the database.

RECONCILIATION DATABASE

If the site has purchased the reconciliation license for error view, this option can be activated.

To record reconciliation information, check the **Use Reconciliation** box and enter the **Reconciliation ODBC Name**.

Validate Active-X DLL

Introduction

The Validate DLL is an ActiveX DLL that was written in Visual Basic. It is used by MVSErrorViewer to do any altering of an index field when the Validate button is clicked. The source code for the DLL is included and placed into the “Validate” directory. This source code is included to allow the user to custom fit the DLL to their environment.

DLL Use and Functionality

The purpose of the DLL is to allow the MVSErrorViewer application to verify and/or alter the data values that are entered into the field text boxes. The DLL is called each time the Validate button is clicked. When called, the DLL compares the value in the field with a set of values that are coded into the DLL and determines validity. Additionally, the DLL is used to cross check field values, or compare one field value against another field value. The source code for the DLL is included to allow the user open the source code in Visual Basic and code specific validation values.

The DLL function that is to be coded is called Validate_Field. This function allows the user to code values that are valid and invalid for each of the twenty-seven fields on the MVSErrorViewer application and also to write code to compare those field values when cross checking fields. The user is to write code to compare the string value from the MVSErrorViewer field with the values that they wish to validate against. When first installed the DLL blank, coded to allow any value in the field to be valid

Examples and notes are included in comments in the Validate Visual Basic Source Code that is included.

<u>Coding Rules and Constraints:</u>

- Only the Validate_Field function is to be changed.
- The function must return either a TRUE or FALSE value.
- An optional message may be returned when cross checking values. The formatting of this message or clarity is left up to the user.
- The UserData and UserComments fields are provided to allow the user to pass values and comments into the MVSErrorViewer application and to be written to the Reconciliation Database. The formatting and values of these variables is left up to the user and what ever information is contained in them when the document is processed is written to the database.
- The Compilation/Version Compatibility section below must be followed for MVSErrorViewer application to function properly.

Compilation/Version Compatibility

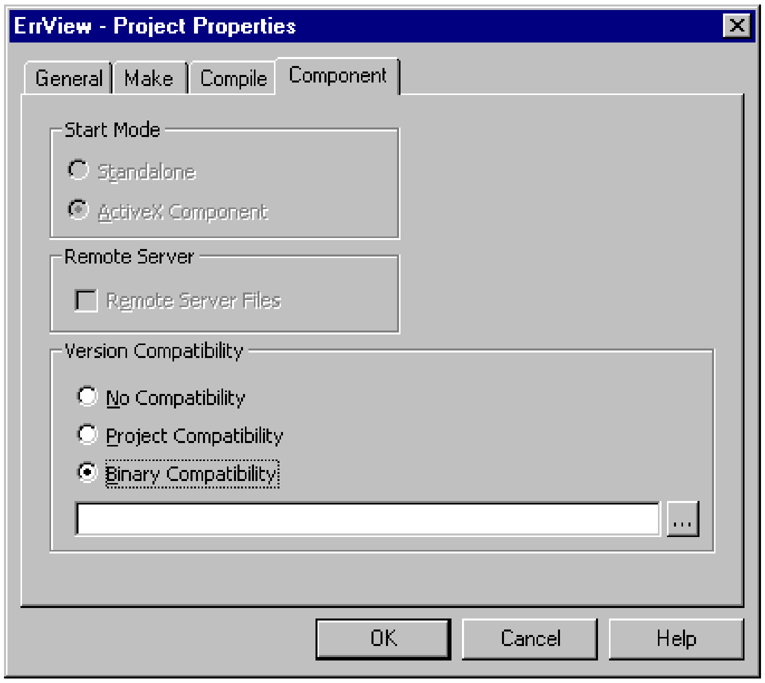
Before compiling the newly coded DLL you must make certain that the Version Compatibility setting is properly configured. This setting is within the Visual Basic development environment under the Project/Validate Properties menu. Once the properties screen is open click the Component Tab.

Within the Version Compatibility box the Binary Compatibility option must be selected and the location to the old Validate.DLL file must be entered in the text box. These settings must properly configured in order for the MVSErrorViewer application to use the newly compiled DLL.

File Location

The code for the Validate ActiveX DLL is located in the install directory in the Validate folder (the default install directory is c:\program files\ascent\bin\validate\).

After a new DLL is compiled it must be copied into the same directory as the old Validate.DLL file. Replace the old copy with the new one. The MVSErrorViewer application cannot be running when you try to copy the new file.



Start Mode
Standalone
[✓ ActiveX Component]

Remote Server
Remote Server Files

Version Compatibility
No Compatibility
Project Compatibility
[✓ Binary Compatibility]

MVSIndex

Overview

MVSIndex's function is to prepare documents to be stored to the host by sending the indexing data and returning a TempID. Below is a detailed description of MVSIndex and all its components.

Note: Some of the information provided on MVSIndex may not be applicable to your site, depending on what version of MVSConnect you are running.

MVSIndex.ini File

MVSIndex.ini is the configuration file associated with MVSIndex. It holds all the configuration settings that are entered on the MVSIndex GUI screens with the exception of the Connection State. MVSIndex is packaged as a TCPIP client and therefore ConnectionState=TCPIP.

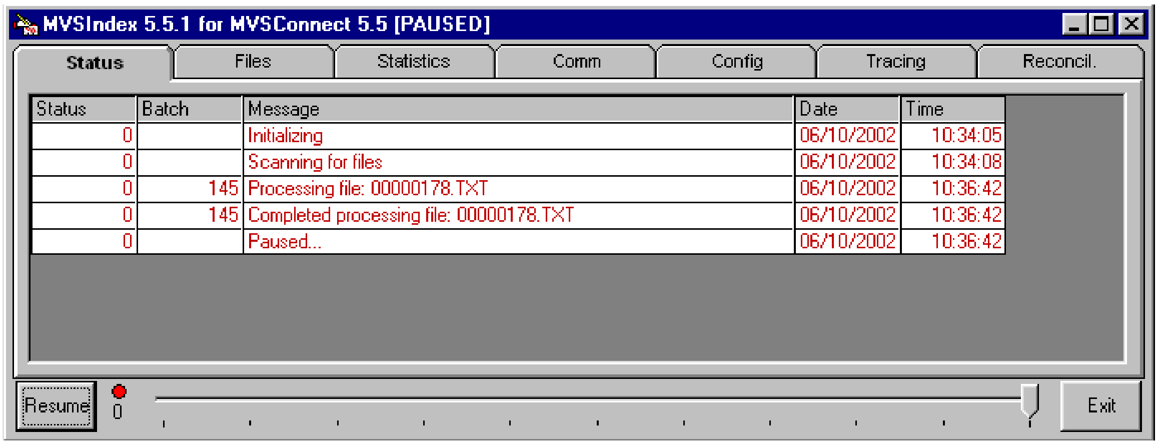
MVSIndex Component Tabs

MVSIndex contains six (6) tabs on the main window. (If you are running MVSConnect 5.5 or higher, there is a seventh tab on the main window). The functions of each of these tabs are as follows:

- **Status**: provides a running text log of program operations.
- **Files**: provides a window into the input, output, and error directories being utilized by the program.
- **Statistics**: produces a graph, which displays successes and errors at each hour of the day.
- **Comm**: communications parameters
- **Config**: operating directories and parameters.
- **Tracing**: tracing/logging parameters (MVSConnect 5.5 or higher)
- **Reconcil**: reconciliation database information/configuration (See section on MVSAdmin for information on the Reconciliation Database)

A detail of each tab is provided in this section.

STATUS TAB



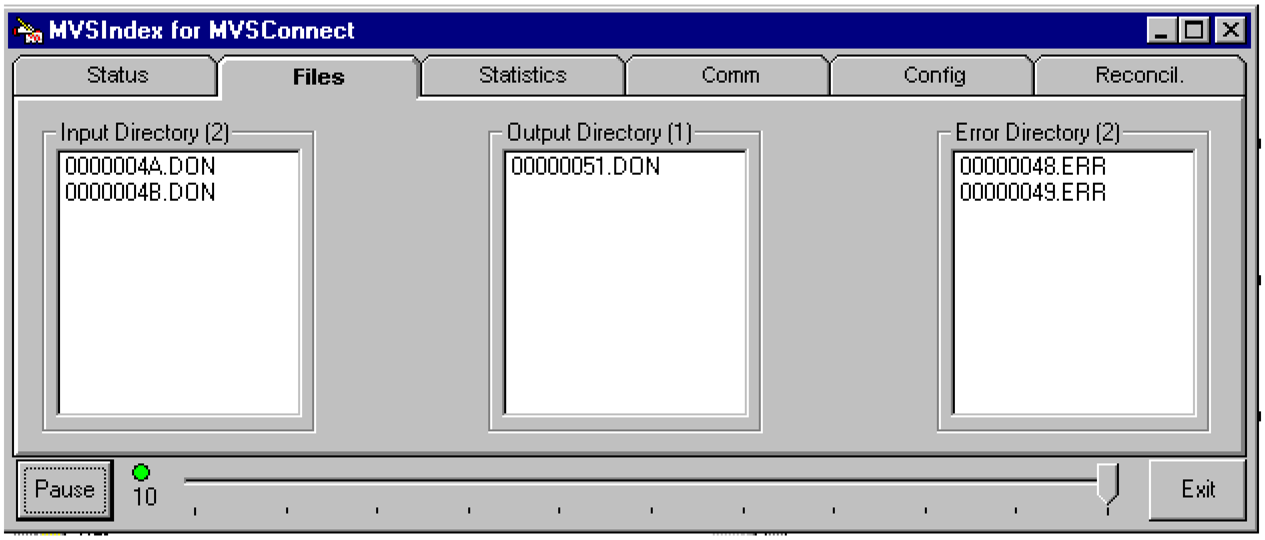
Status	Batch	Message	Date	Time
0		Initializing	06/10/2002	10:34:05
0		Scanning for files	06/10/2002	10:34:08
0	145	Processing file: 00000178.TXT	06/10/2002	10:36:42
0	145	Completed processing file: 00000178.TXT	06/10/2002	10:36:42
0		Paused...	06/10/2002	10:36:42

The Status tab provides a status of every document as it passes through MVSIndex. The messages on the screen are also being written to the log file. The components on the Status tab are:

- **Pause** - pauses program operations. No documents will be processed while the program is paused. This button's caption changes to **Resume** when the program is paused.
- **Slider** – the Slider at the bottom of the screen is there to control the speed at which MVSIndex operates. The colored speed indicator to the left of the **Slider** displays the speed percentage at which the program is currently running. The Slider would typically be left at 10 so that documents are indexed as quickly as possible, but dragging the arrow to the left or right provides a mechanism for slowing down the program for troubleshooting purposes, as well as for environments that run at slower speeds.
- **Exit -** terminates the program.

FILES TAB

The Files tab provides a window into the Input, Output and Error directories, as seen below.



Input Directory (2): 0000004A.DON, 0000004B.DON

Output Directory (1): 00000051.DON

Error Directory (2): 00000048.ERR, 00000049.ERR

Each section is described below:

Input Directory

The Input Directory shows all documents ready to be processed. Clicking on any file in this list will display the Input Viewer, which shows the indexing data in a small box as seen below.

The Input Viewer has 3 buttons:

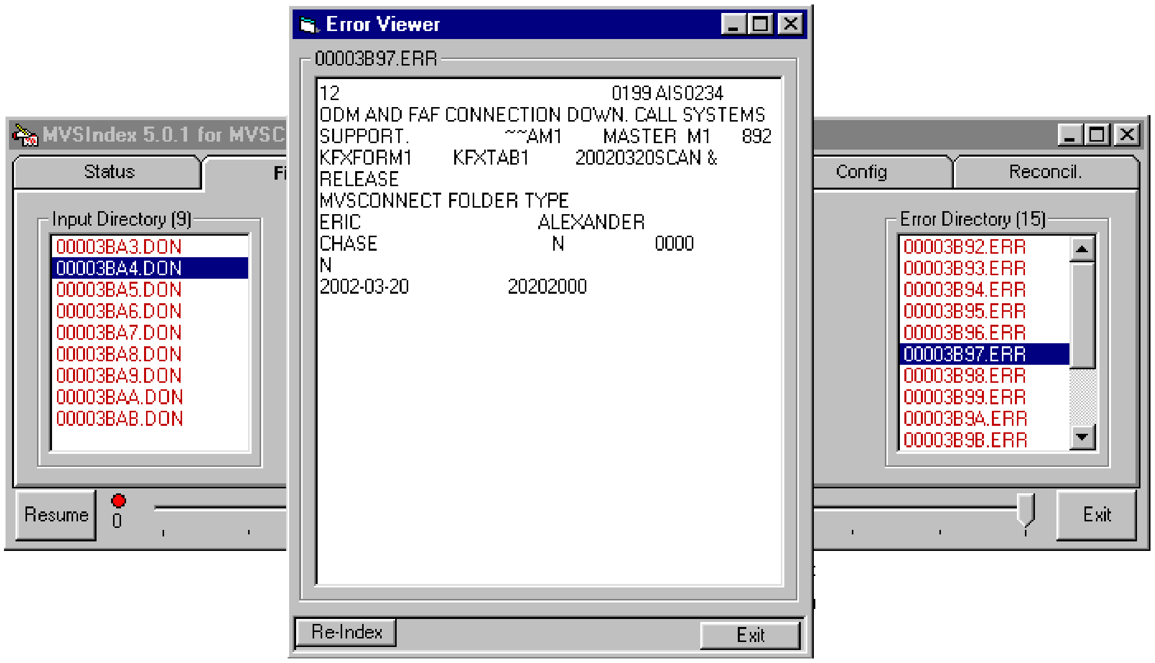
- **Delete** – deletes the document and all related files from MVSConnect without being indexed.
- **Corrupt** – moves the document and all related files to the corrupt directory, which is specified on the Config tab discussed below
- **Exit** – exits the Input Viewer and returns to the MVSIndex panel.

Output Directory

The Output Directory shows all documents successfully indexed and ready to be stored to the host.

Error Directory

The Error Directory shows documents that returned an error from the host during MVSIndex time. Clicking on any file in this list will display the Error Viewer, which shows the error returned from the host along with the indexing data that was sent in a small box as seen below.

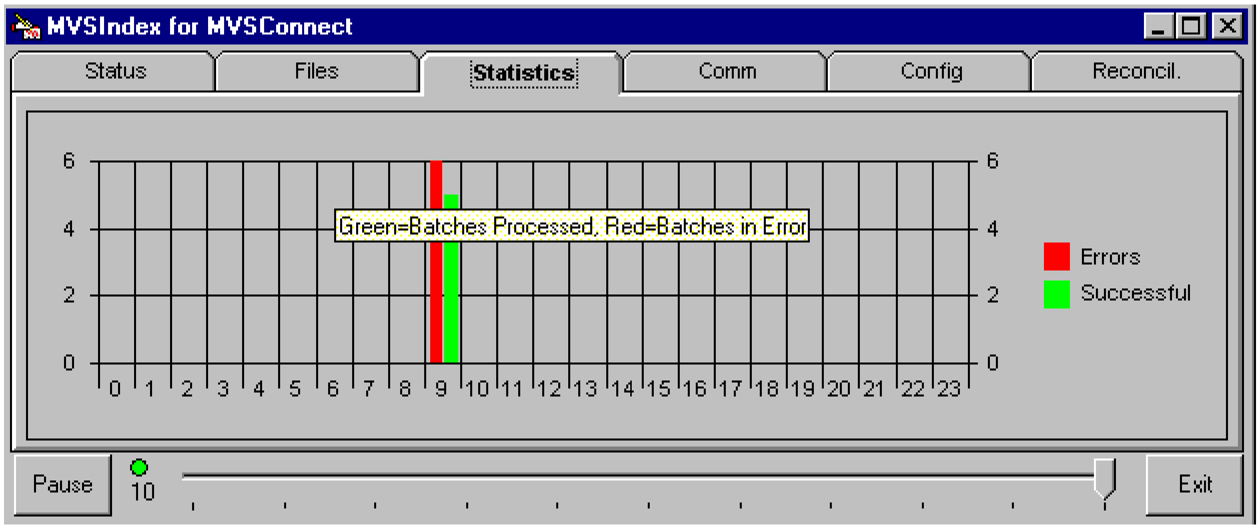


```
12                          0199 AIS0234
ODM AND FAF CONNECTION DOWN. CALL SYSTEMS
SUPPORT.            ~~AM1      MASTER  M1     892
KFXFORM1       KFXTAB1       20020320SCAN &
RELEASE
MVSCONNECT FOLDER TYPE
ERIC                        ALEXANDER
CHASE                          N           0000
N
2002-03-20              20202000
```

The Error Viewer has 2 buttons:

- **Re-Index** – sends the document back to the Index queue to be processed by MVSIndex. This works best when documents are sent to the error directory due to the host being down. Instead of going to MVSErrorViewer to send documents back to MVSIndex, it can be done here in MVSIndex.
- **Exit** – exits the viewer and returns to the MVSIndex panel.

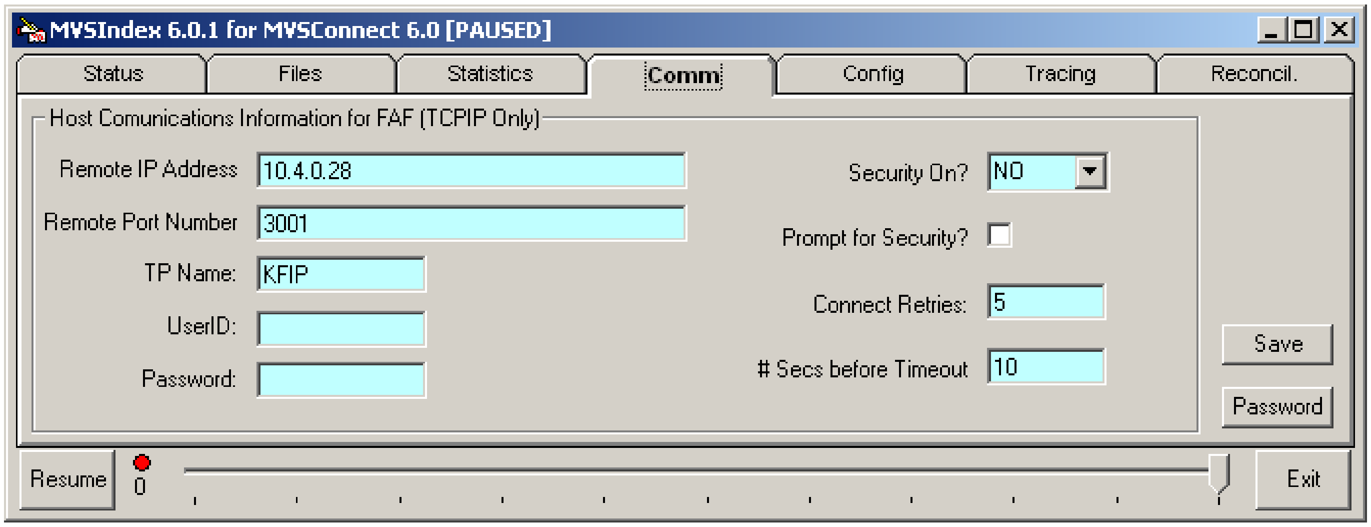
STATISTICS TAB



This tab shows a graphical count of successful and unsuccessful indexes. The graph is intended to show statistics over cumulative 24-hour periods from the time the program was started. When the program is stopped and restarted, the graph will be reinitialized, with the old data is not being preserved.

COMM TAB

The **Comm** tab provides communications and logging configuration options.

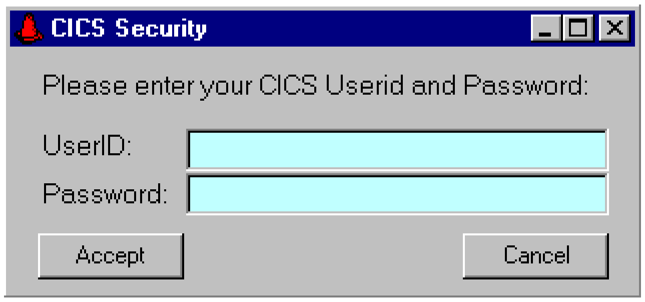


Host Comunications Information for FAF (TCPIP Only)

Remote IP Address	10.4.0.28
Remote Port Number	3001
TP Name:	KFIP
UserID:	
Password:	
Security On?	NO
Prompt for Security?	☐
Connect Retries:	5
# Secs before Timeout	10

The Comm Tab has several configurable fields, which are discussed below:

- **Remote IP Address** - The IP Address for the mainframe.
- **Remote Port Number** - Port number for TCPIP.
- **TP Name** - The Transaction Program name on the host that this program will talk to. This should be set to KFIP.
- **UserID** - If transaction security is to be used, enter the UserID to be used in the secure transaction.
- **Password** - If transaction security is to be used, enter the password to be used in the secure transaction. Note that this password is encrypted in the INI file.
- **Security On?** - If transaction security on the host is to be used, this entry should be set to YES. Otherwise set it to NO.
- **Prompt For Security?** - If this box is checked, the program will prompt for a CICS UserID and Password when it initially starts, and will use that CICS UserID and Password for host communications. An example of this security screen is shown below.


CICS Security

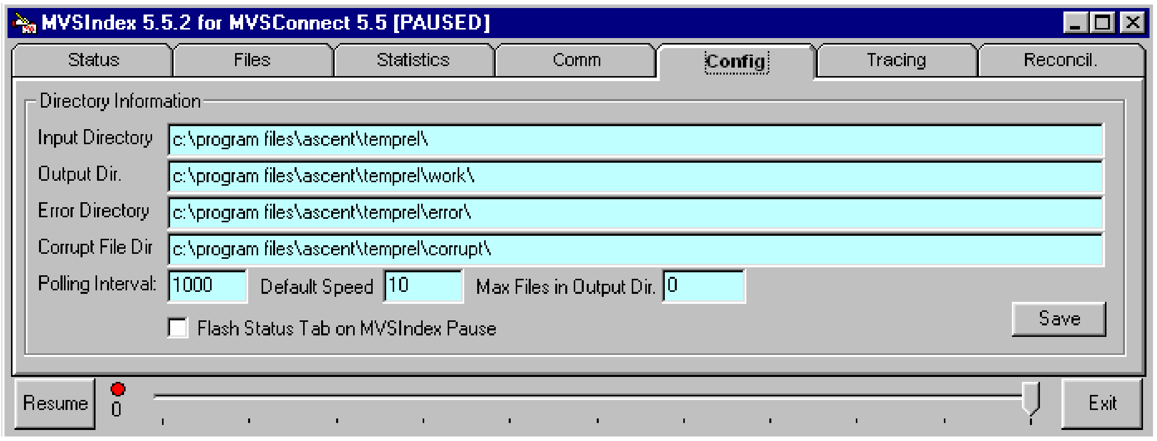
Please enter your CICS Userid and Password:

UserID:

Password:

- **Connect Retries** - The number of retries the application should attempt to make to get a host connection before timing out.
- **# Secs before timeout** - The number of seconds between each connection retry.
- The **Save** button saves any changes made to the configuration.

CONFIG TAB



Directory Information

Input Directory: c:\program files\ascent\temprel\

Output Dir.: c:\program files\ascent\temprel\work\

Error Directory: c:\program files\ascent\temprel\error\

Corrupt File Dir: c:\program files\ascent\temprel\corrupt\

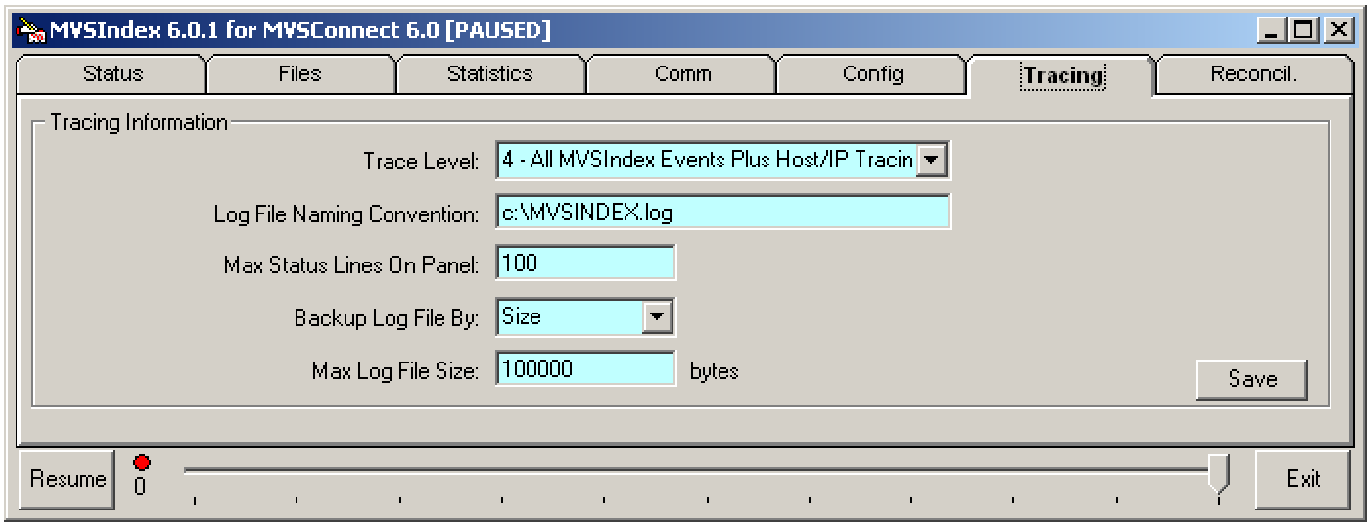
Polling Interval: 1000 Default Speed: 10 Max Files in Output Dir.: 0

☐ Flash Status Tab on MVSIndex Pause

This window allows various operating parameters of the program to modified.

- **Input Directory** - The input directory is the directory that the program will poll for .TXT files in. This path must exist for the program to operate properly.
- **Output Dir**. - The output directory is the directory where the program will move processed files for the MVSStore program to process.
- **Error Directory** - The error directory is the directory where files, which produce indexing errors, will be moved for processing by the MVSErrorViewer program.
- **Corrupt File Dir** - The corrupt file directory is the directory where unusable TXT (and associated) files will be moved. If a TXT file is found that has an invalid length, it and its associated files are moved here. Manual handling of such files is required.
- **Polling Interval** - This parameter specifies the number of milliseconds (1 millisecond = 1/1000th of a second) to pause between polls for files in the input directory. It is a good idea to pause at least a second between polls (i.e., to pause 1 second set this to 1000), perhaps more if the input directory is on a LAN. Frequent polling can cause data overruns on servers and network congestion, as well as unnecessary drive utilization.
- **Default Speed** - This is the default speed percentage at which the program will operate. Making this number less than 10 will reduce operating speed. Setting it to zero will pause the program at startup.
- **Max Files in Output Dir**. - This parameter states the maximum number of documents are to be in the output directory at one time. When the total documents in the Output Directory equals this number, MVSIndex will not process any more documents until the Output Directory has less than the number of files specified in this field. This is to avoid any slowdown of MVSStore due to a backlog of files in the Output Directory.
- **Flash Status Tab on MVSIndex Pause** – When checked, this parameter will cause the Status tab to flash red and white when MVSIndex is paused.
- The **Save** button saves any changes.

TRACING TAB



Tracing Information

Trace Level: 4 - All MVSIndex Events Plus Host/IP Tracin

Log File Naming Convention: c:\MVSINDEX.log

Max Status Lines On Panel: 100

Backup Log File By: Size

Max Log File Size: 100000 bytes

This panel contains all parameters for tracing activity with MVSIndex.

- **Trace Level** - This is the detail level for tracing. Trace levels are as follows:

  0=Tracing off

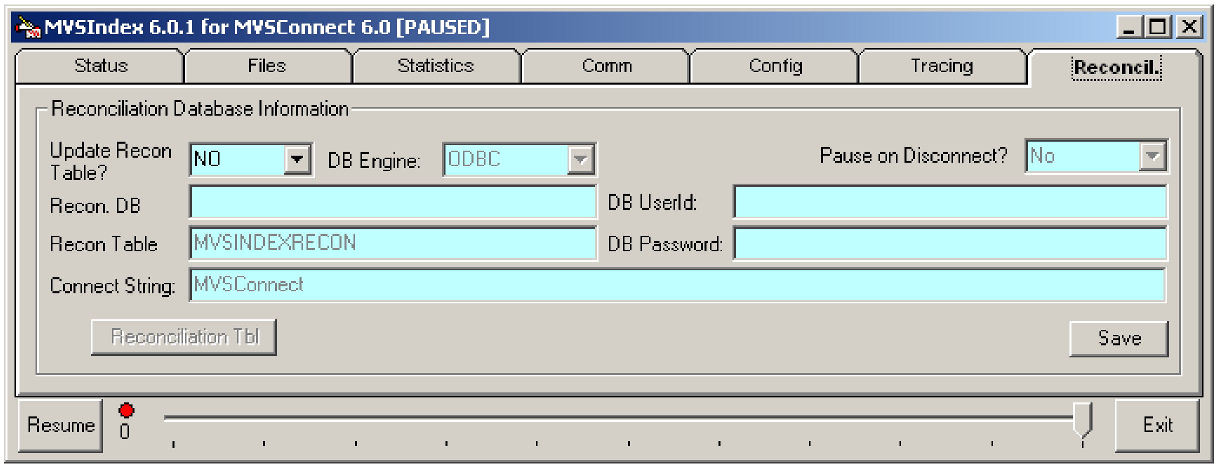
  1=Errors only

  2=Errors and Warnings

  3=All MVSIndex Events

  4=All MVSIndex Events Plus Host/IP Tracing
- **Log File Naming Convention** - The fully qualified path to the log file for trace data. Each log file created will be located in this directory with the initial name stated in this parameter.
- **Max Status Lines On Panel** - This specifies the maximum number of on-screen log messages to preserve (in the Status window). This parameter only affects the on-screen display, not the number of messages written to the log file.
- **Backup Log File By** – This is the way the log files are created.
  - Date – The log file will be created for each day, using the naming convention stated above, appended with MMDDYY.log.
  - Month – The log file will be created for each month, using the naming convention stated above, appended with MMYY.log.
  - Size – The log file will be created with the naming convention stated above, but after it grows to the size set in the Max Log Size parameter, the file will be renamed .BAK, and a new log file will be created.
- **Max Log File Size** - The maximum size (in bytes) that the log file will grow to before it is renamed with a .BAK extension. This option is only available when the Backup Log File By option is set to Size.

RECONCIL. TAB

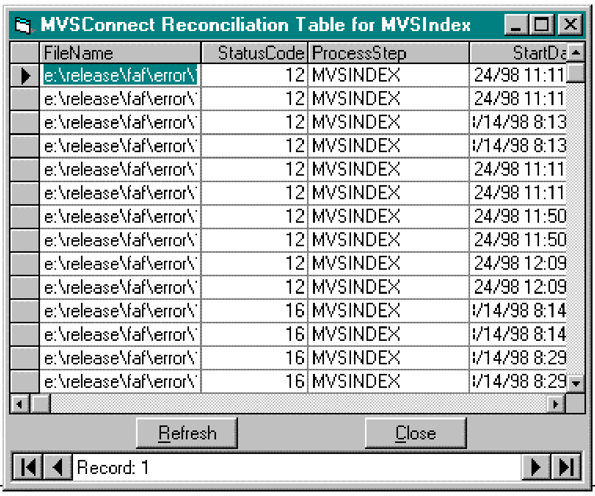

MVSIndex 6.0.1 for MVSConnect 6.0 [PAUSED]

Reconciliation Database Information

Update Recon Table?: NO
DB Engine: ODBC
Pause on Disconnect?: No
Recon. DB:
DB UserId:
Recon Table: MVSINDEXRECON
DB Password:
Connect String: MVSConnect

This panel allows for specifying reconciliation database parameters.

- **Update Recon Table?** - Set this to YES if updating the reconciliation table, NO if not. See section on MVSAdmin for info about the Reconciliation Table.
- **DB Engine** - Specify JET if using the Microsoft JET © Engine for updating a local Access © table. Specify ODBC if using an ODBC data source. Note that if you are using ODBC, you must have the proper ODBC driver installed and functioning and have proper DSN entries in your ODBC setup.
- **Pause on Disconnect** - Specify if MVSIndex should pause if the connection to the reconciliation table is lost.
- **Recon. DB** - If using the JET engine, specify the path to the reconciliation database here. This entry is not used for ODBC.
- **Recon Table** - This is the name of the reconciliation table to be updated. This is typically set to MVSINDEXRECON unless you have created your own database and table and are using ODBC.
- **Connect String** - This should be set to the name of the DSN specified in the ODBC setup portion of your system settings. Typically, the DSN should be set to MVSConnect, and an MVSConnect DSN entry made in the system ODBC settings.
- **Reconciliation Tbl** - The Reconciliation Tbl button brings up a grid displaying the reconciliation


MVSConnect Reconciliation Table for MVSIndex

FileName	StatusCode	ProcessStep	StartDa
e:\release\faf\error\	12	MVSINDEX	24/98 11:11
e:\release\faf\error\	12	MVSINDEX	24/98 11:11
e:\release\faf\error\	12	MVSINDEX	ı/14/98 8:13
e:\release\faf\error\	12	MVSINDEX	ı/14/98 8:13
e:\release\faf\error\	12	MVSINDEX	24/98 11:11
e:\release\faf\error\	12	MVSINDEX	24/98 11:11
e:\release\faf\error\	12	MVSINDEX	24/98 11:50
e:\release\faf\error\	12	MVSINDEX	24/98 11:50
e:\release\faf\error\	12	MVSINDEX	24/98 12:09
e:\release\faf\error\	12	MVSINDEX	24/98 12:09
e:\release\faf\error\	16	MVSINDEX	ı/14/98 8:14
e:\release\faf\error\	16	MVSINDEX	ı/14/98 8:14
e:\release\faf\error\	16	MVSINDEX	ı/14/98 8:29
e:\release\faf\error\	16	MVSINDEX	ı/14/98 8:29

table for MVSIndex, as seen in the screen print below.

- The **Save** button saves any changes.

MVSIndex Customization

Included with the MVSIndex module are two customizable DLL files: **MVSIndex.DLL** and **MVSIndexError.DLL**. Both files are located in the System or System32 subdirectory of the Windows directory on the C: drive. Included with these DLL files is the code for customization by the user.

MVSINDEX.DLL

The code for **MVSIndex.DLL** is located in the directory where MVSIndex was installed, in a separate directory called **UserExit\MVSIndex**. This is a Visual Basic 5.0 ActiveX DLL that contains two functions: **PreIndexExit()** and **PostIndexExit()**. The **PreIndexExit()** function is executed just after the .TXT file is found in the input directory of the MVSIndex program, and the **PostIndexExit()** function is executed just after the MVSIndex program has completed (successfully or not) indexing a document to the host and retrieving a TEMP ID. Instructions on coding and compiling this module are located in the code as comments.

MVSINDEXERROR.DLL

The code for **MVSIndexError.DLL** is located in the directory where MVSIndex was installed, in a separate directory called **UserExit\MVSIndexError**. This is a Visual Basic 5.0 ActiveX DLL that contains one function**: MVSIndexError()**. The **MVSIndexError()** function is executed when there is an error communicating to the host during the execution of MVSIndex. Any code entered here will be executed just after the error connecting to the Host occurs and the retries have been exhausted. Instructions on coding and compiling are located in the code as comments.

MVSINDEXRECONERROR.DLL

The code for **MVSIndexReconError.DLL** is located in the directory where MVSIndex was installed, in a separate directory called **UserExit\MVSIndexReconError**. This is a Visual Basic 5.0 ActiveX DLL that contains one function: **MVSIndexReconError()**. The **MVSIndexReconError()** function is executed when there is an error communicating to the reconciliation database during the execution of MVSIndex. Any code entered here will be executed just after the error connecting to the Host occurs and the retries have been exhausted. Instructions on coding and compiling are located in the code as comments.

MVSStore

Overview

MVSStore for MVSConnect is a part of the next generation of software included in SYSCOM's MVSConnect product line. Its function is to store documents to the host. Please note that some of the information provided on MVSIndex may not be applicable to your site, depending on what version of MVSConnect you are running.

MVSStore.ini File

MVSStore.ini is the configuration file associated with MVSStore. It holds all the configuration settings that are entered on the MVSStore GUI screens with the exception of the Connection State. MVSStore is packaged as a TCPIP client and therefore ConnectionState=TCPIP.

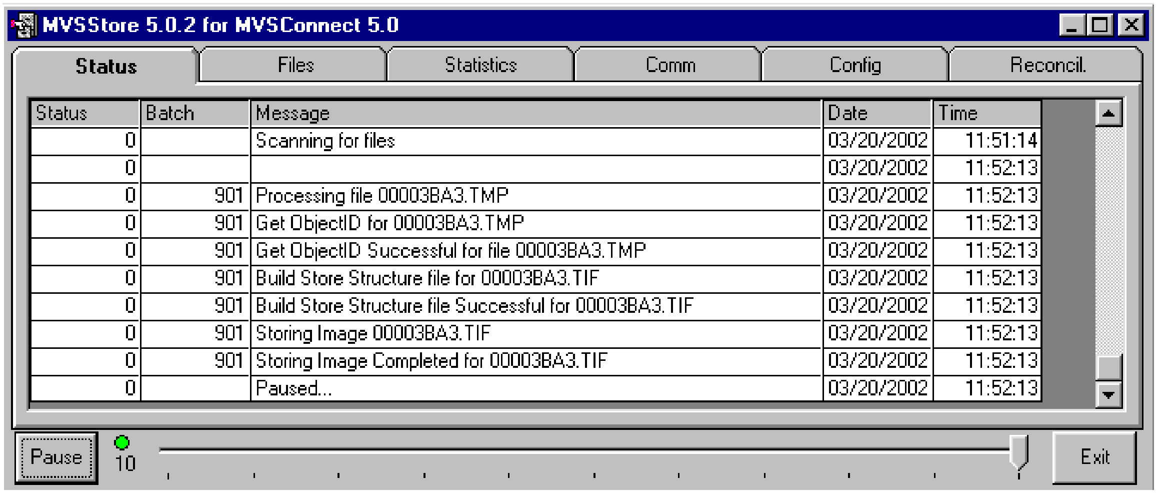
MVSStore Component Tabs

MVSStore contains six (6) tabs on the main window. (If you are running MVSConnect 5.5 or higher, there is a seventh tab on the main window). The functions of each of these tabs are as follows:

- **Status**: provides a running text log of program operations.
- **Files**: provides a window into the input, output, and error directories being utilized by the program.
- **Statistics**: produces a graph, which displays successes and errors at each hour of the day.
- **Comm**: communications parameters
- **Config**: operating directories and parameters.
- **Tracing:** tracing/logging parameters (MVSConnect 5.5 or higher)
- **Reconcil**: reconciliation database information/configuration (See section on MVSAdmin for information on the Reconciliation Database)

A detail of each tab is provided in this section.

STATUS TAB



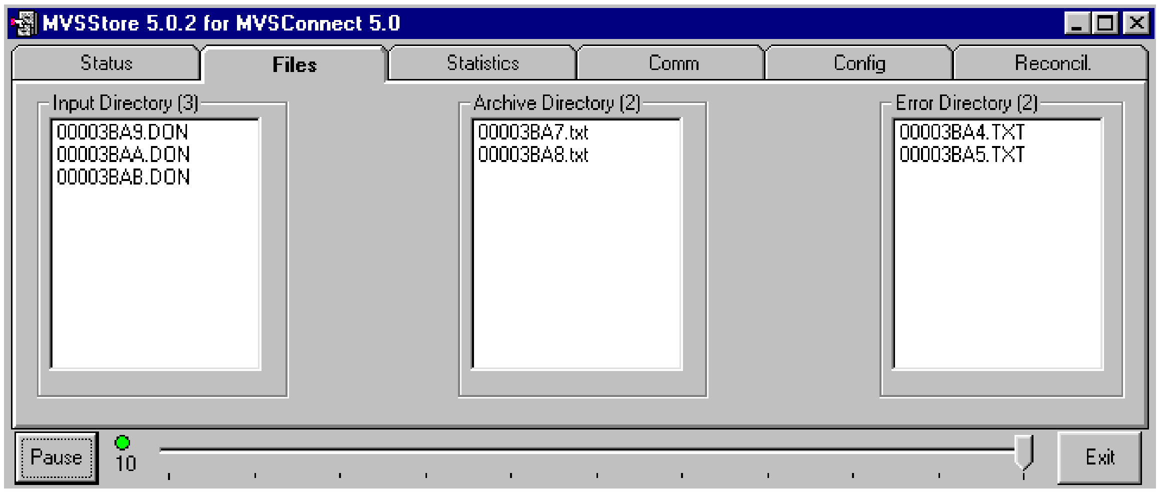
Status	Batch	Message	Date	Time
0		Scanning for files	03/20/2002	11:51:14
0			03/20/2002	11:52:13
0	901	Processing file 00003BA3.TMP	03/20/2002	11:52:13
0	901	Get ObjectID for 00003BA3.TMP	03/20/2002	11:52:13
0	901	Get ObjectID Successful for file 00003BA3.TMP	03/20/2002	11:52:13
0	901	Build Store Structure file for 00003BA3.TIF	03/20/2002	11:52:13
0	901	Build Store Structure file Successful for 00003BA3.TIF	03/20/2002	11:52:13
0	901	Storing Image 00003BA3.TIF	03/20/2002	11:52:13
0	901	Storing Image Completed for 00003BA3.TIF	03/20/2002	11:52:13
0		Paused...	03/20/2002	11:52:13

- **Pause -** pauses program operations. No documents will be processed while the program is paused. This button's caption changes to Resume when the program is paused.

- **Slider** – the Slider at the bottom of the screen is there to control the speed at which MVSStore operates. The colored speed indicator to the left of the **Slider** displays the speed percentage at which the program is currently running. The Slider would typically be left at 10 so that documents are indexed as quickly as possible, but dragging the arrow to the left or right provides a mechanism for slowing down the program for troubleshooting purposes, as well as for environments that run at slower speeds.
- **Exit -** terminates the program.

FILES TAB

The Files tab provides a window into the Input, Archive, and Error directories, as seen below.



Input Directory (3)
00003BA9.DON
00003BAA.DON
00003BAB.DON

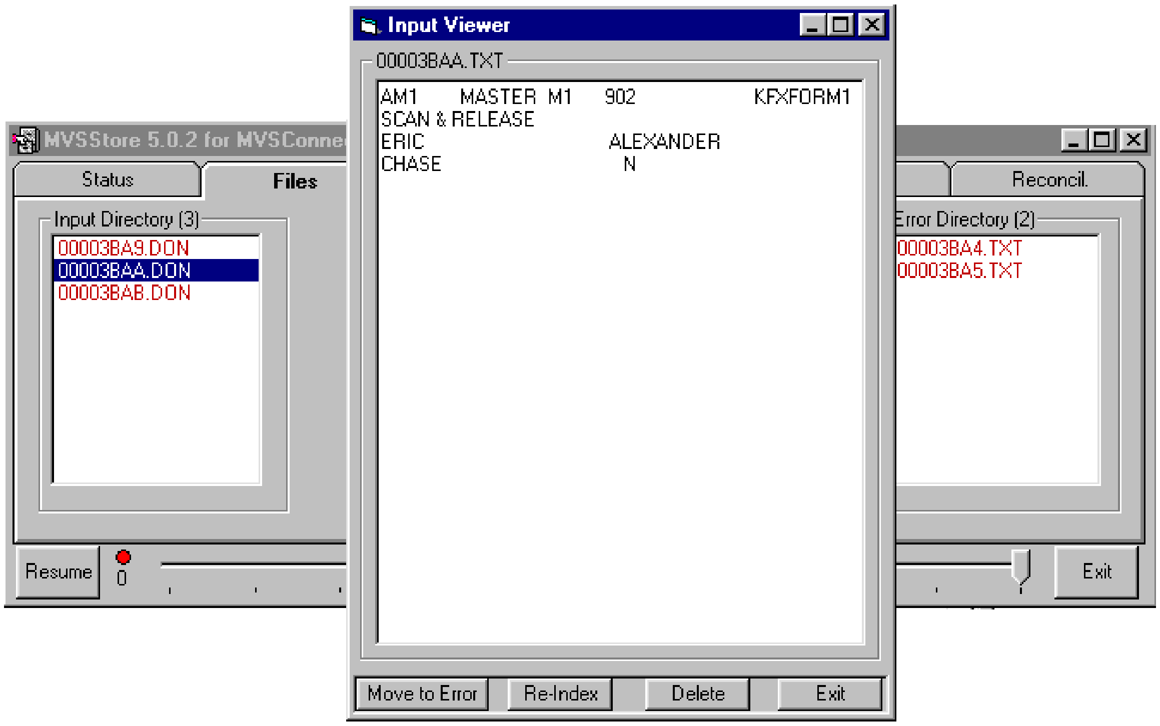
Archive Directory (2)
00003BA7.txt
00003BA8.txt

Error Directory (2)
00003BA4.TXT
00003BA5.TXT

Each section is described below:

Input Directory

The Input Directory shows all documents ready to be processed. Clicking on any file in this list will display the Input Viewer, which shows the indexing data in a small box as seen below.



00003BAA.TXT

```
AM1      MASTER  M1     902               KFXFORM1
SCAN & RELEASE
ERIC                     ALEXANDER
CHASE                      N
```

Input Directory (3): 00003BA9.DON, 00003BAA.DON, 00003BAB.DON

Error Directory (2): 00003BA4.TXT, 00003BA5.TXT

The Input Viewer has 4 buttons:

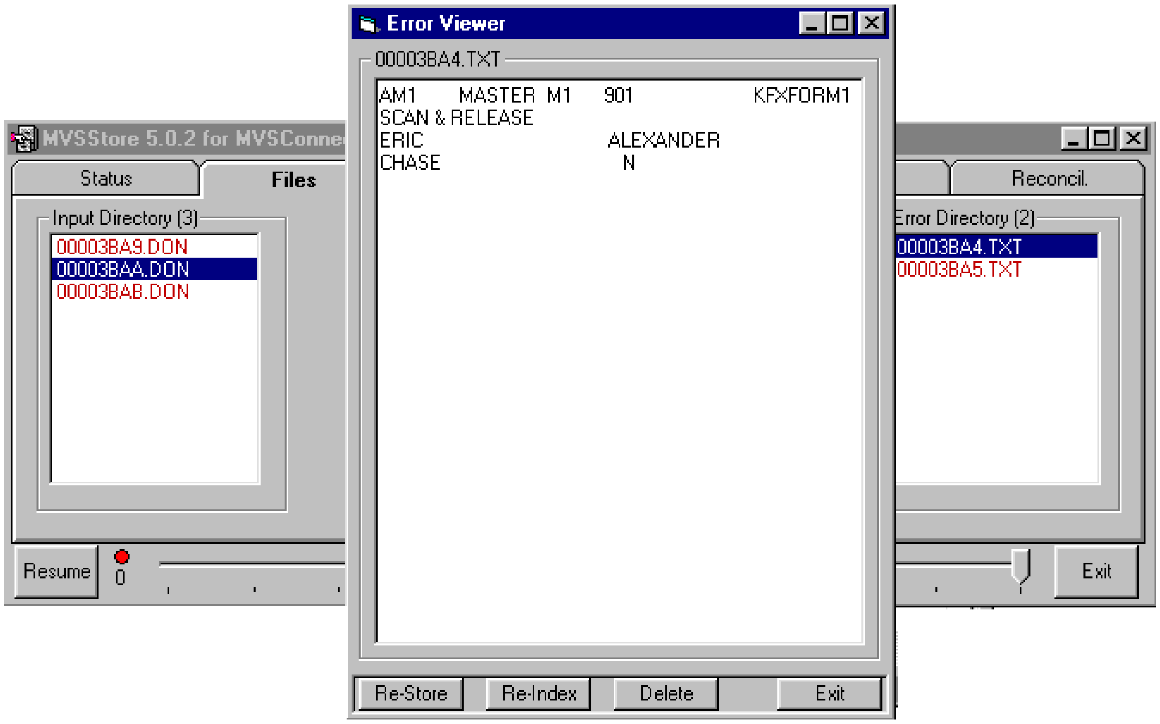
- **Move to Error** – moves a document ready to be stored to the error directory. The purpose of this is to move files that are having trouble storing.
- **Re-Index** – moves a document back to MVSIndex to be re-indexed. The purpose of this is to re-index files that may have a bad TempID from the host.
- **Delete** – deletes the document and all associated files from MVSConnect.
- **Exit** – exits the Input Viewer and returns to the MVSStore panel.

Archive Directory

The Archive Directory is a list of files that were archived after the document was stored. This option is set on the Config tab, discussed below.

Error Directory

The Error Directory shows documents that returned an error from the host during MVSStore time. Clicking on any file in this list will display the Error Viewer, which shows the indexing data that was sent in a small box as seen below.



00003BA4.TXT

```
AM1      MASTER  M1     901              KFXFORM1
SCAN & RELEASE
ERIC                       ALEXANDER
CHASE                        N
```

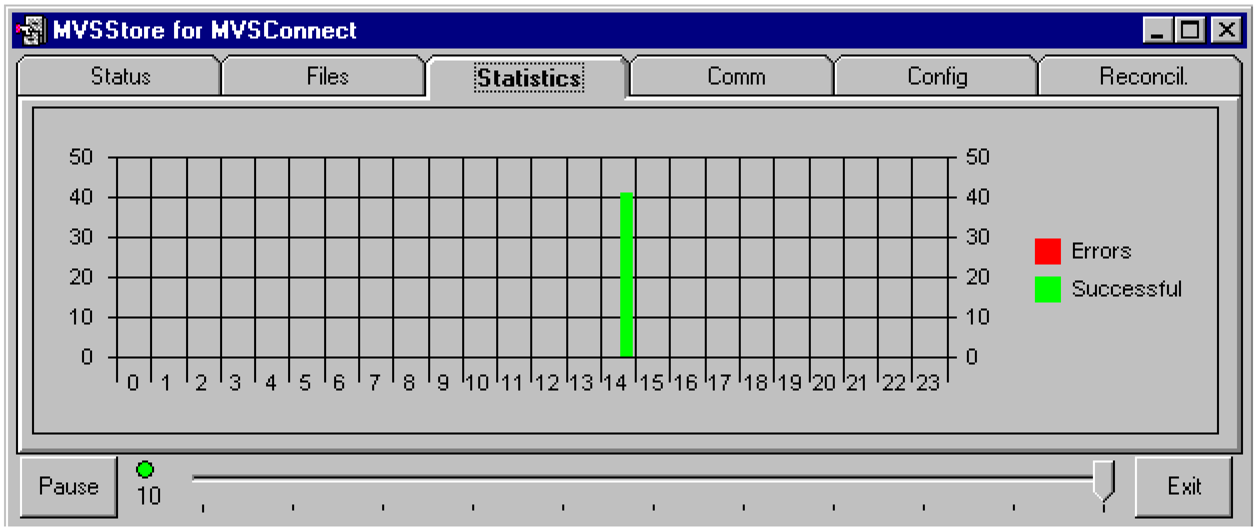
00003BA9.DON
00003BAA.DON
00003BAB.DON

00003BA4.TXT
00003BA5.TXT

The Error Viewer has 4 buttons:

- **Re-Store** – sends the document back to the MVSStore input directory to attempt to store it again. to be processed by MVSIndex. This works best when documents are sent to the error directory due to the host being down.
- **Re-Index** – sends the document back to the MVSIndex input directory to be re-indexed. This works best when the documents cannot store due to a bad TempID or a problem with the host.
- **Delete** – deletes the document and all associated files from MVSConnect.
- **Exit** – exits the viewer and returns to the MVSIndex panel.

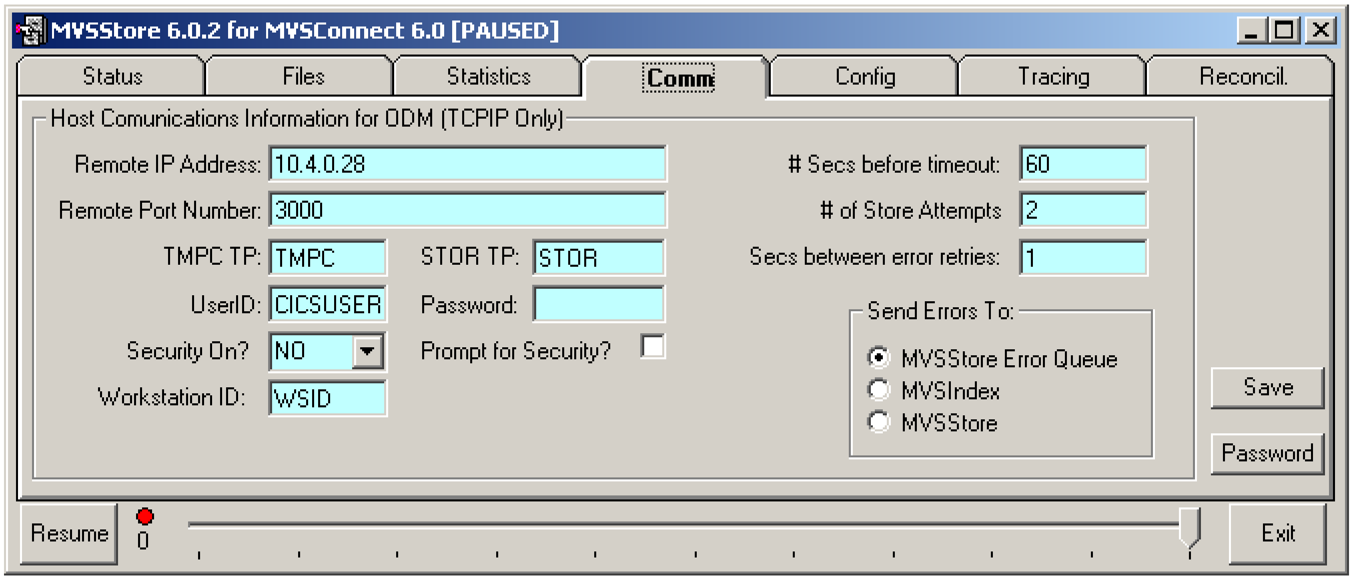
Statistics Tab



This tab shows a graphical count of successful and unsuccessful indexes. The graph is intended to show statistics over cumulative 24-hour periods from the time the program was started. When the program is stopped and restarted, the graph will be reinitialized, with the old data is not being preserved.

Comm Tab

The **Comm** tab provides communications and logging configuration options.



Host Comunications Information for ODM (TCPIP Only)

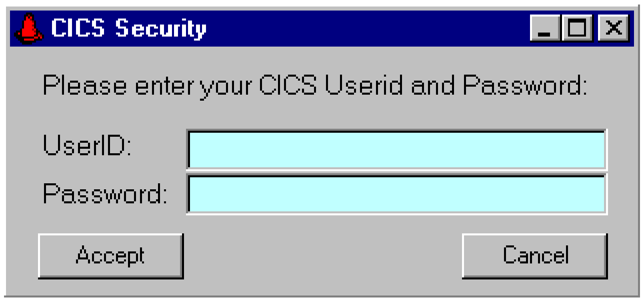
Remote IP Address:	10.4.0.28	# Secs before timeout:	60
Remote Port Number:	3000	# of Store Attempts	2
TMPC TP:	TMPC	STOR TP:	STOR
Secs between error retries:	1		
UserID:	CICSUSER	Password:	
Security On?	NO	Prompt for Security?	☐
Workstation ID:	WSID		

Send Errors To:
- [✓ MVSStore Error Queue]
- MVSIndex
- MVSStore

The Comm Tab has several configurable fields, which are discussed below:

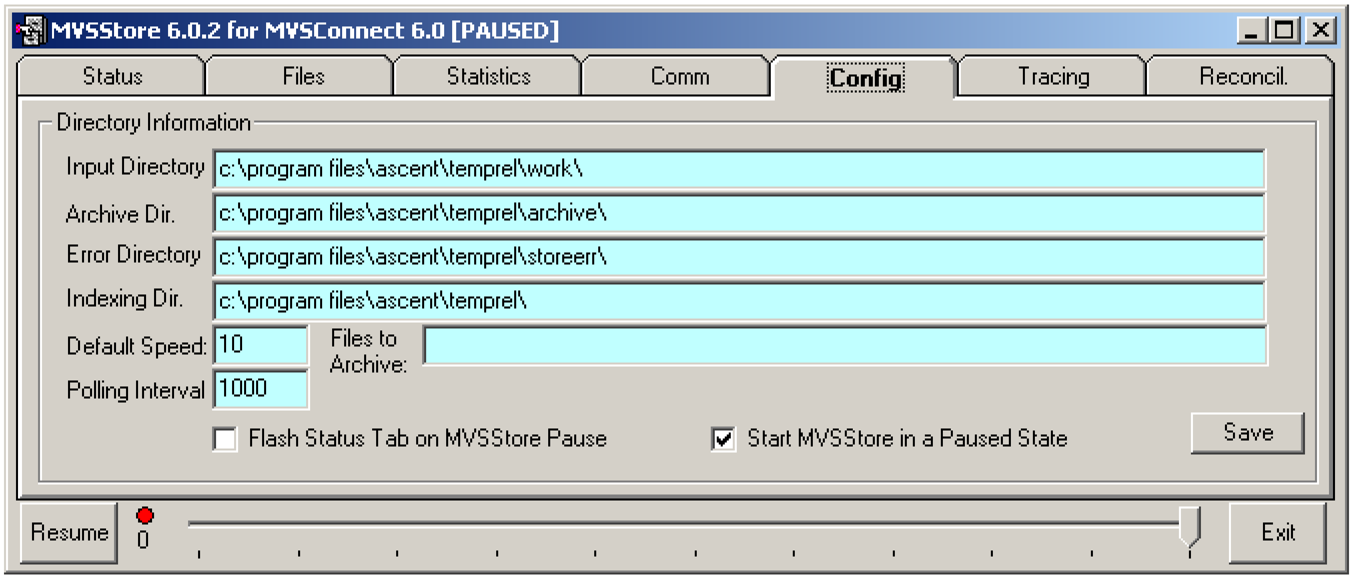
- **Remote IP Address** - The IP Address for the mainframe.
- **Remote Port Number** - Port number for TCPIP.
- **TMPC TP** – The transaction program name on the host that this program will communicate with to retrieve an Object ID. This should be set to TMPC.
- **STOR TP** – The transaction program name on the host that this program will communicate with to store the image file. This should be set to STOR.

- **UserID** - If transaction security is to be used, enter the UserID to be used in the secure transaction.
- **Password** - If transaction security is to be used, enter the password to be used in the secure transaction. Note that this password is encrypted in the INI file.
- **Security On?** - If transaction security on the host is to be used, this entry should be set to YES. Otherwise set it to NO.
- **Prompt For Security?** - If this box is checked, the program will prompt for a CICS UserID and Password when it initially starts, and will use that CICS UserID and Password for host communications. An example of this security screen is shown below.



- **Workstation ID** – The connection id for the workstation to the host. The default value in the ini file is “WSID”. This value is valid for the host connection. The value can be changed if so desired, but a value must exist; it cannot be blank.
- **# Secs before timeout** - The number of seconds between each connection retry.
- **# of Store Attempts** – The number of times to re-send a document that did not store before sending it to the error directory.
- **Secs Between Error Retries** – The number of seconds to pause before resending a document that did not store.
- **Send Errors To** – This section lets you decide where to send documents that fail in MVSStore. The choices are:
  - MVSStore Error Queue – All errors will go to the MVSStore Error Directory that is specified on the Config tab
  - MVSIndex – All errors will go back to MVSIndex to obtain another TempID
  - MVSStore – All errors will be retried in MVSStore using the same TempID already obtained from MVSIndex.
- The **Save** button saves any changes made to the configuration.

CONFIGUATION TAB



Directory Information

Input Directory c:\program files\ascent\temprel\work\

Archive Dir. c:\program files\ascent\temprel\archive\

Error Directory c:\program files\ascent\temprel\storeerr\

Indexing Dir. c:\program files\ascent\temprel\

Default Speed: 10

Files to Archive:

Polling Interval 1000

☐ Flash Status Tab on MVSStore Pause

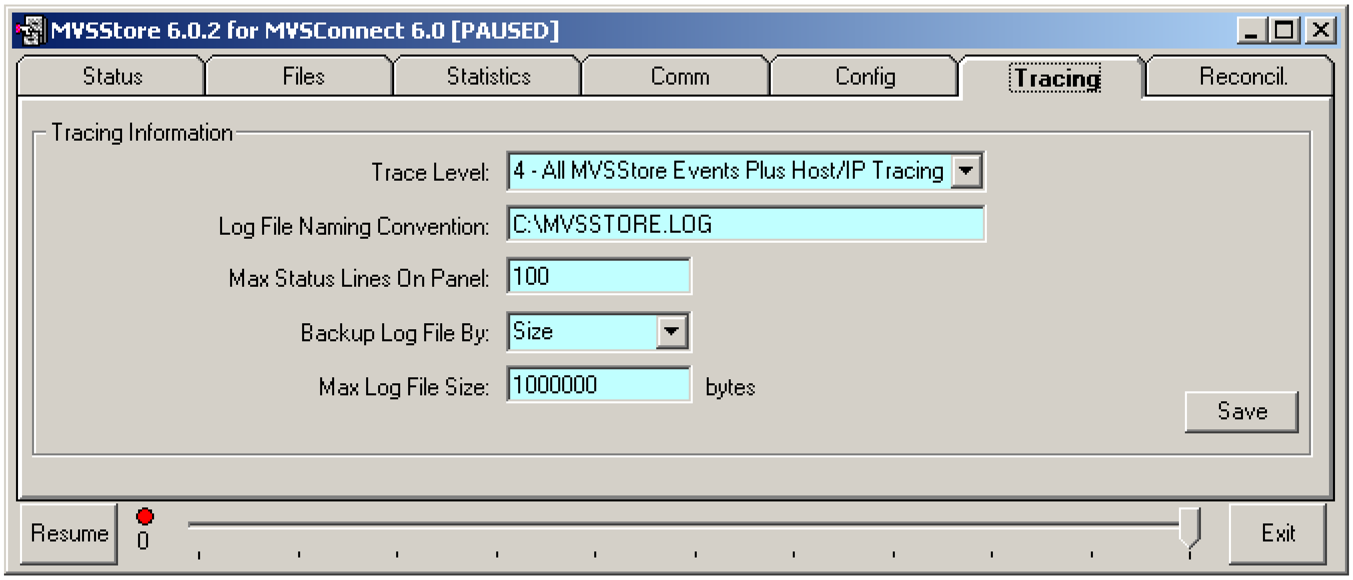
☑ Start MVSStore in a Paused State

0

This window allows various operating parameters of the program to be modified.

- **Input Directory** - The input directory is the directory that the program will poll for .TXT files in. This path must exist for the program to operate properly. This value must match the Output Directory of MVSIndex.
- **Archive Dir**. **–** The archive directory is the directory where the program will move processed files specified in the Files to Archive entry field after the MVSStore program processes them successfully. Files are only archived after the corresponding image file is successfully stored to the host.
- **Error Directory –** The error directory is the directory where files that produce store errors will be moved. This directory should be different than the MVSIndex Error Directory.
- **Indexing Directory –** Input Directory for the MVSIndex program. This value must match the Input Directory from MVSIndex.
- **Default Speed –** This is the default speed percentage at which the program will operate. Making this number less than 100 will reduce operating speed. Setting this parameter to zero will cause the program to pause at startup.
- **Files to Archive –** List the file extensions (comma delimited) to write to the Archive directory after a successful Store. The file extensions should be entered here, comma delimited, without spaces. For example: tif,txt,inf for archiving files with the extensions of .tif, .txt, and .inf. You will be responsible for cleaning up any files written to that directory. Those files will never be deleted by any MVSConnect program.
- **Polling Interval –** This parameter specifies the number of milliseconds (1 millisecond = 1/1000th of a second) to pause between polls for files in the input directory. It is a good idea to pause at least a second between polls (i.e., to pause 1 second set this to 1000), perhaps more if the input directory is on a LAN. Frequent polling can cause data overruns on servers and network congestion, as well as unnecessary drive utilization. This parameter also refreshes the input, archive, and error directory windows on the files tab.
- **Flash Status Tab on MVSStore Pause –** When checked, this parameter will cause the Status tab to flash red and white when MVSStore is paused.
- **Start MVSStore in a Paused State –** When checked, MVSStore will be paused when it is started.
- The **Save** button saves any changes.

TRACING TAB



Trace Level: 4 - All MVSStore Events Plus Host/IP Tracing

Log File Naming Convention: C:\MVSSTORE.LOG

Max Status Lines On Panel: 100

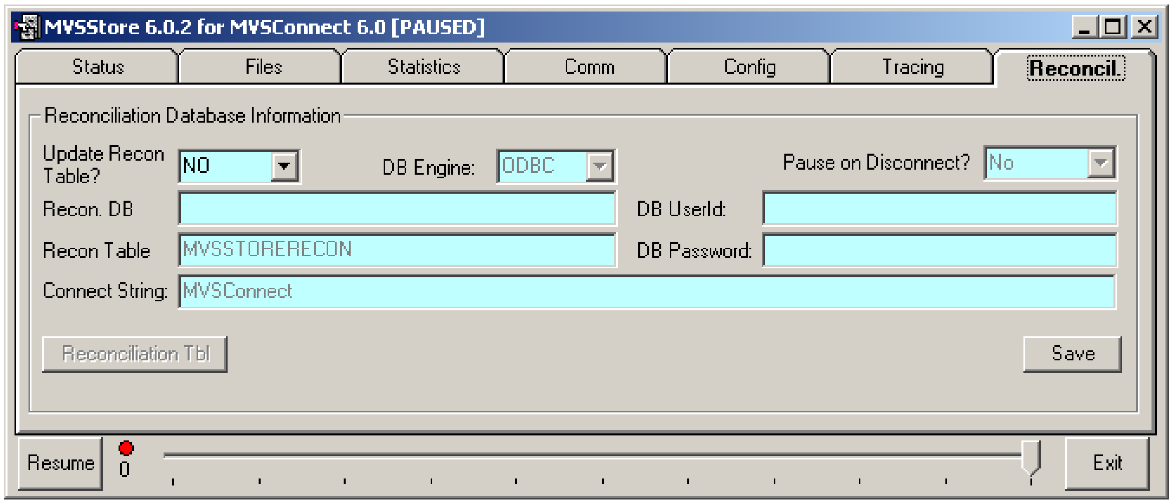
Backup Log File By: Size

Max Log File Size: 1000000 bytes

This panel contains all parameters for tracing activity with MVSStore.

- **Trace Level** - This is the detail level for tracing. Trace levels are as follows:
  - 0=Tracing off,
  - 1=Errors only,
  - 2=Errors and warnings,
  - 3=All MVSIndex Events,
  - 4=All MVSIndex Events Plus Host Tracing,
  - 5=All MVSIndex Events Plus TCPIP trace
- **Log File Naming Convention** - The fully qualified path to the log file for trace data. Each log file created will be located in this directory with the initial name stated in this parameter.
- **Max Status Lines On Panel** - This specifies the maximum number of on-screen log messages to preserve (in the Status window). This parameter only affects the on-screen display, not the number of messages written to the log file.
- **Backup Log File By** – This is the way the log files are created.
  - Date – The log file will be created for each day, using the naming convention stated above, appended with MMDDYY.log.
  - Month – The log file will be created for each month, using the naming convention stated above, appended with MMYY.log.
  - Size – The log file will be created with the naming convention stated above, but after it grows to the size set in the Max Log Size parameter, the file will be renamed .BAK, and a new log file will be created.
- **Max Log Size** - The maximum size (in bytes) that the log file will grow to before it is renamed with a .BAK extension. This option is only available when the Backup Log File By option is set to Size.

RECONCILIATION TAB

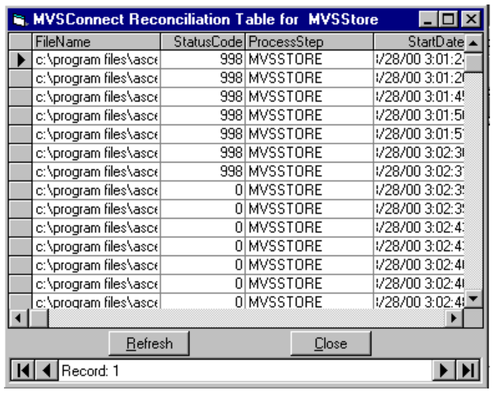

MVSStore 6.0.2 for MVSConnect 6.0 [PAUSED]

Reconciliation Database Information

Update Recon Table? NO
DB Engine: ODBC
Pause on Disconnect? No
Recon. DB
DB UserId:
Recon Table MVSSTORERECON
DB Password:
Connect String: MVSConnect

This panel allows for specifying reconciliation database parameters.

- **Update Recon Table**: Set this to YES if updating the reconciliation table, NO if not. See section on MVSAdmin for info about the Reconciliation Table.
- **DB Engine**: Specify JET if using the Microsoft JET © Engine for updating a local Access © table. Specify ODBC if using an ODBC data source. Note that if you are using ODBC, you must have the proper ODBC driver installed and functioning and have proper DSN entries in your ODBC setup portion of the Windows 95 or Windows NT system.
- **Pause on Disconnect**: Specify if MVSStore should pause if the connection to the reconciliation table is lost.
- **Recon. DB**: If using the JET engine, specify the path to the reconciliation database here. This entry is not used for ODBC.
- **Recon Table**: This is the name of the reconciliation table to be updated. This is typically set to MVSSTORERECON unless you have created your own database and table and are using ODBC.
- **Connect String**: This should be set to the name of the DSN specified in the ODBC setup portion of your system settings. Typically, the DSN should be set to MVSConnect, and an MVSConnect DSN entry made in the system ODBC settings.
- **Reconciliation Tbl**: The Reconciliation Tbl button brings up a grid displaying the reconciliation table for MVSStore, as seen in the screen print below.


MVSConnect Reconciliation Table for MVSStore

FileName	StatusCode	ProcessStep	StartDate
c:\program files\asce	998	MVSSTORE	i/28/00 3:01:2
c:\program files\asce	998	MVSSTORE	i/28/00 3:01:2
c:\program files\asce	998	MVSSTORE	i/28/00 3:01:4
c:\program files\asce	998	MVSSTORE	i/28/00 3:01:5
c:\program files\asce	998	MVSSTORE	i/28/00 3:01:5
c:\program files\asce	998	MVSSTORE	i/28/00 3:02:3
c:\program files\asce	998	MVSSTORE	i/28/00 3:02:3
c:\program files\asce	0	MVSSTORE	i/28/00 3:02:3
c:\program files\asce	0	MVSSTORE	i/28/00 3:02:3
c:\program files\asce	0	MVSSTORE	i/28/00 3:02:4
c:\program files\asce	0	MVSSTORE	i/28/00 3:02:4
c:\program files\asce	0	MVSSTORE	i/28/00 3:02:4
c:\program files\asce	0	MVSSTORE	i/28/00 3:02:4
c:\program files\asce	0	MVSSTORE	i/28/00 3:02:4

- The **Save** button saves any changes.

MVSStore Customization

Included with the MVSStore module are two customizable DLL files: MVSStore.DLL and MVSStoreError.DLL. Both files are located in the System or System32 subdirectory of the Windows directory on the C: drive. Included with these DLL files is the code for customization by the user.

MVSSTORE.DLL

The code for MVSStore.DLL is located in the directory where MVSStore was installed, in a separate directory called UserExit\MVSStore. This Visual Basic 6.0 ActiveX DLL contains six functions: **PreGetObjectIDExit**(), **PostGetObjectIDExit**(), **PreInsertObjectIDExit**(), **PostInsertObjectIDExit**(), **PreStoreExit**(), and **PostStoreExit**().

The **PreGetObjectIDExit**() function is executed prior to the MVSStore program establishing a conversation with the host. This function retrieve and inserts the object id for MODCA files only. The variables sent to the function is the filename of the document being processed .TMP.

The **PostGetObjectIDExit**() function is executed just after the Object ID is retrieved (or not retrieved) from the host. The variables returned are the file name and the return code. MVSStore.EXE is looking for a return code of “0” to proceed.

The **PreInsertObjectIDExit**() function is called just before the Object ID that was successfully returned from the host is inserted into the MODCA file (if TIFF files are being stored, this function is not called). The variable sent to the function is the filename being processed .TMP.

The **PostInsertObjectIDExit**() function is called just after the Object ID is inserted into the MODCA file (if the image is a TIFF, this function is not called). The variables returned are the file name and the return code. MVSStore.EXE is looking for a return code of “0” to proceed.

The **PreStoreExit**() function is called just before the image is stored to the host. The variable sent to the function is the filename of the image being processed .TMP.

The **PostStoreExit**() function is called just after the image is stored to the host. The variables returned are the file name and the return code. MVSStore.EXE is looking for a return code of “0” to proceed. Instructions on coding and compiling this module are located in the code as comments.

MVSSTOREERROR.DLL

The code for MVSStoreError.DLL is located in the directory where MVSStore was installed, in a separate directory called UserExit\MVSStoreError. This Visual Basic 6.0 ActiveX DLL contains one function: **MVSStoreError**(). The **MVSStoreError**() function is executed when there is an error communicating to the host during the execution of MVSStore. Any code entered here will be executed just after the error connecting to the Host occurs and the retries have been exhausted. Instructions on coding and compiling are located in the code as comments.

MVSSTORERECONERROR.DLL

The code for MVSStoreReconError.DLL is located in the directory where MVSStore was installed, in a separate directory called UserExit\MVSStoreReconError. This Visual Basic 6.0 ActiveX DLL contains a single function: **MVSStoreReconError**(). The **MVSStoreReconError**() function is executed when there is an error communicating to the reconciliation database during the execution of MVSStore. Any code entered here will be executed just after the error connecting to the Host occurs and the retries have been exhausted. Instructions on coding and compiling are located in the code as comments.

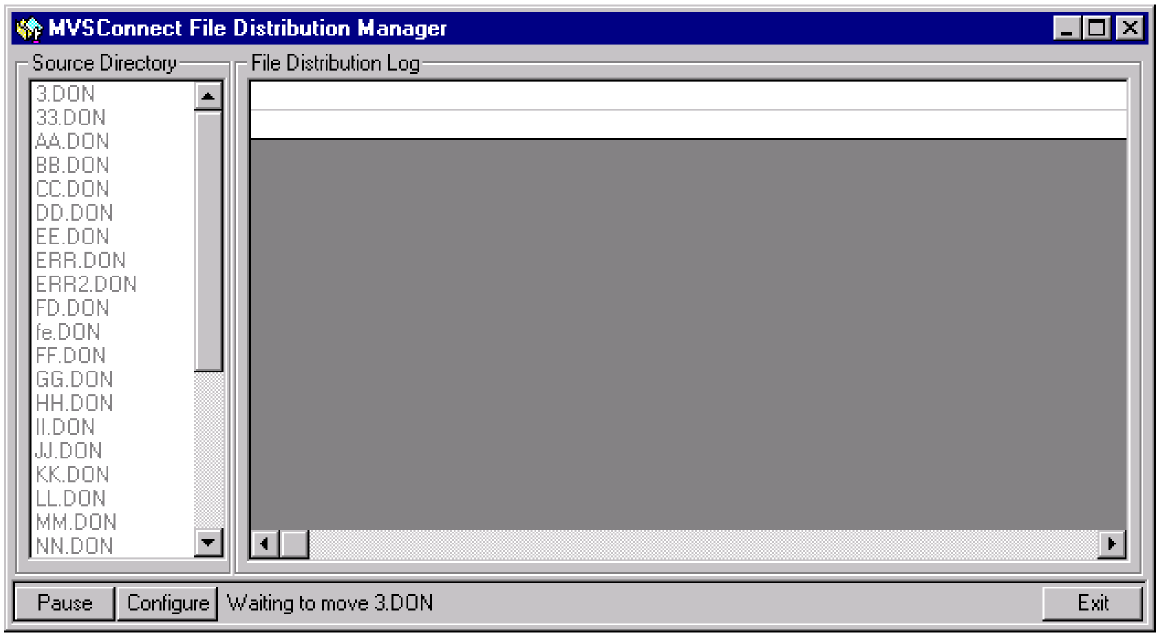
FileDistributionManager

Overview

The File Distribution Manager for MVSConnect provides a number of useful functions. First, files in a single directory can be distributed to any number of directories in a round-robin fashion. This is useful when the number of files being released to a single directory outpaces the MVSIndex and MVSStore programs.

Secondly, in LAN environments with very high file volumes, the File Distribution Manager can be used to manage the number of files "fed" to the MVSIndex and MVSStore programs so that their performance is not degraded. MVSIndex and MVSStore both build lists of files to be processed and displays these lists in their file windows. If there are thousands of files in their input directories, performance can be degraded when these windows are refreshed.

THE FILE DISTRIBUTION MANAGER MAIN WINDOW



Waiting to move 3.DON

This window shows the status of documents being managed by the FDM.

The "Source Directory" window on the left displays a list of all files to be moved.

The "File Distribution Log" on the right shows the status of any files that have been moved. In the above example, no files have been moved, so no messages are displayed.

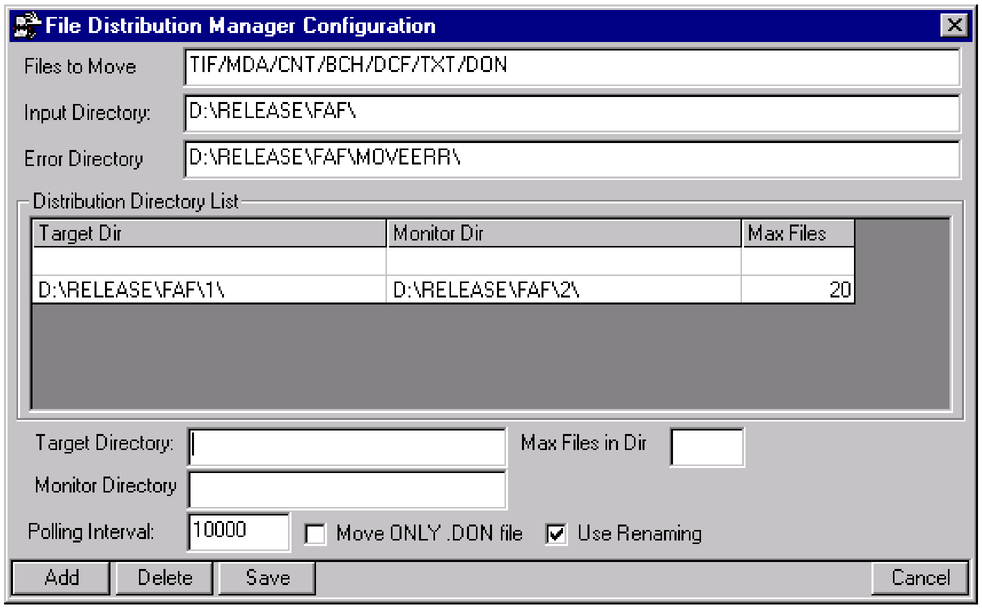
A status line at the bottom of the screen shows whether a document set is being moved or if the program is waiting to move the files.

Only the .DON file is displayed on this screen, but all associated files with the .DON are moved with it. The .DON file is always the LAST file to be moved, as it acts as a flag for other processes that all other files are present and can be worked.

- Clicking the "Pause" button will pause the process.
- Clicking the "Configure" button will bring up the configuration panel.
- Clicking the "Exit" button will terminate the process.

The File Distribution Manager Configuration Window

The **Configuration** Screen provides configuration options for FileDistributionManager.



File Distribution Manager Configuration

Files to Move: TIF/MDA/CNT/BCH/DCF/TXT/DON

Input Directory: D:\RELEASE\FAF\

Error Directory: D:\RELEASE\FAF\MOVEERR\

Distribution Directory List

Target Dir	Monitor Dir	Max Files
D:\RELEASE\FAF\1\	D:\RELEASE\FAF\2\	20

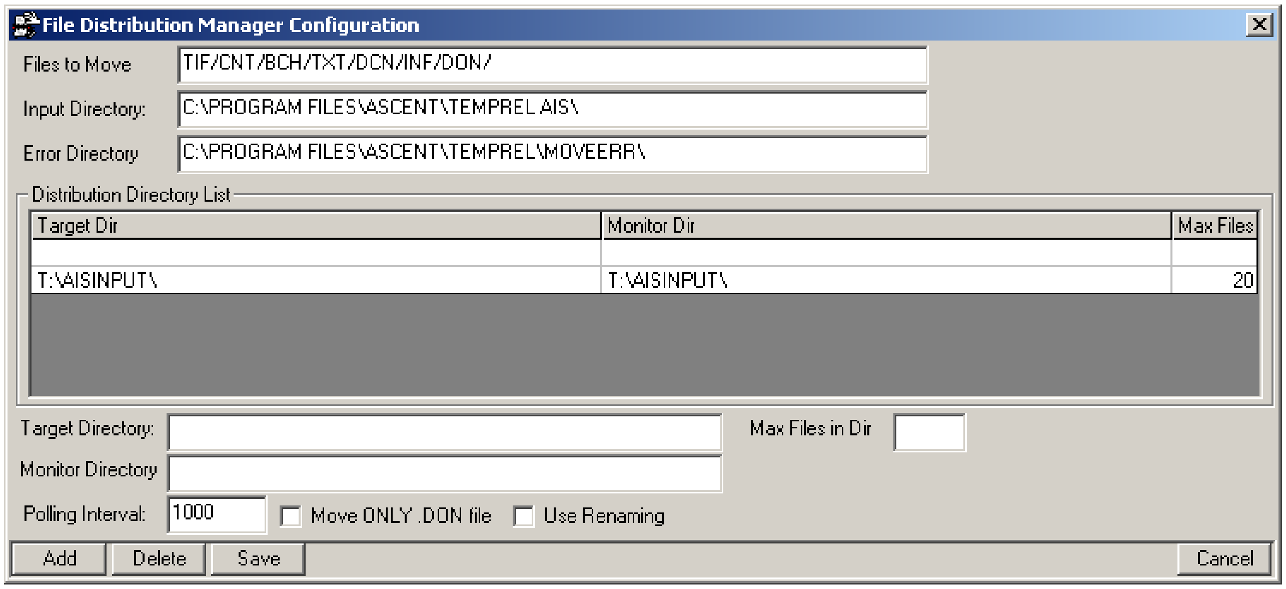
Target Directory:

Max Files in Dir:

Monitor Directory:

Polling Interval: 10000

[ ] Move ONLY .DON file [✓ Use Renaming]



File Distribution Manager Configuration

Files to Move: TIF/CNT/BCH/TXT/DCN/INF/DON/

Input Directory: C:\PROGRAM FILES\ASCENT\TEMPREL AIS\

Error Directory: C:\PROGRAM FILES\ASCENT\TEMPREL\MOVEERR\

Distribution Directory List

Target Dir	Monitor Dir	Max Files
T:\AISINPUT\	T:\AISINPUT\	20

Target Directory:

Max Files in Dir:

Monitor Directory:

Polling Interval: 1000

[ ] Move ONLY .DON file [ ] Use Renaming

- **Files to Move**: The list of file extensions to be moved must be listed here. Each document extension MUST be separated by a forward slash (“/”). The last file extension in the list MUST be “DON”. Files will be moved in the order they appear in this list. The last entry should NOT have a trailing slash.
- **Input Directory**: This is the directory that the FMD will poll for files to be distributed. You should include a trailing backslash.
- **Error Directory**: This should be the directory where files will be moved if they cannot be moved to the Target Directory that you specify. If using “Renaming” (described below) and a duplicate file error occurs (i.e., the file already exists in the target directory), then the file in the target directory will be moved to this error directory.

- **Distribution Directory List**: This displays the list of Target Directories, Monitor Directories and Maximum Files that the target and monitor directories may contain before receiving more files. Data is entered in the fields below the Distribution Directory List section (described below).
    - **Target Directory**: Enter the target directory to which files will be distributed.
    - **Monitor Directory**: Enter the directory that will be monitored for file volumes along with the volume in the Target Directory.
    - **Max Files in Dir**: Enter the maximum number of files that are allowed in the Target Directory. Note that both the Monitor Directory AND the Target directory are monitored for this volume of files. If EITHER directory exceeds this level, files will not be moved to that directory.
- **Polling Interval**: time (in milliseconds) to poll for new files to move.
- **Move ONLY .DON File**: By checking this checkbox, only the .DON file is actually moved to the target directory. All other files remain in the input directory.
- **Use Renaming**: By checking this checkbox, the program using file RENAMING rather than COPYING and DELETING. This is MUCH faster, but can ONLY be used if the Input drive and Target drive are the SAME. In other words, only use this option when you are moving files to different directories on the SAME drive. If you are moving files from one drive to another, DO NOT check this box.
- **Add Button**: When you wish to add another directory entry to the Distribution Directory List, enter the Target Directory, Monitor Directory and Max Files in Dir and click this Add button.
- **Delete Button**: To delete a line from the Distribution Directory List, highlight that line by clicking on it, and then click this Delete button.
- **Save Button**: To save your changes, click on this Save button. The screen will close.
- **Cancel Button**: Cancels changes without saving.

IndexOLEServer

Overview

The IndexOLEServer is used only if MVSAdmin has been purchased. Its function is to write indexing data to the reconciliation database. IndexOLEServer should be installed on any workstation that will be used to run Ascent Capture's Validation Module.

Setup

After installing IndexOLEServer on each workstation that has been identified as an Ascent Capture Validation workstation, Validation scripts using SYSCOM, Inc.'s Validation Scripts templates must be used in each batch class. On SYSCOM, Inc.'s MVSConnect CD, there is a "Softbridge Templates for Ascent" folder. Each template in this folder must be copied into the Ascent\AdminDB\Scripts directory on the server. Once the files are here, any new validation scripts that are created will contain the necessary code to write to the reconciliation database.

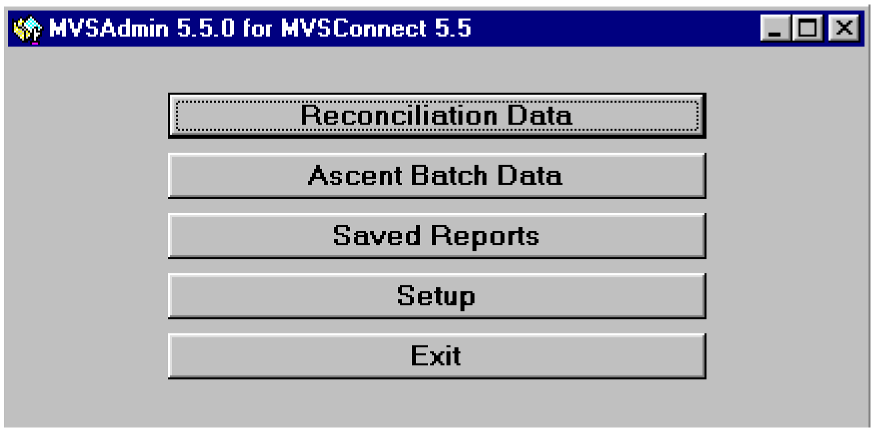
MVSAdmin

Overview

MVSAdmin for MVSConnect is to provide flexibility in viewing and reporting on Reconciliation data produced by Kofax's Ascent Capture © and SYSCOM's MVSConnect products.

Below, we will explore the different aspects of MVSAdmin, including viewing data, creating reports, purging data, migrating data to history, and importing Ascent Capture batch data.

MVSAdmin Screen Prints and Components



The screen shot below is the main window that appears when MVSAdmin is started.

Below is a description of each button

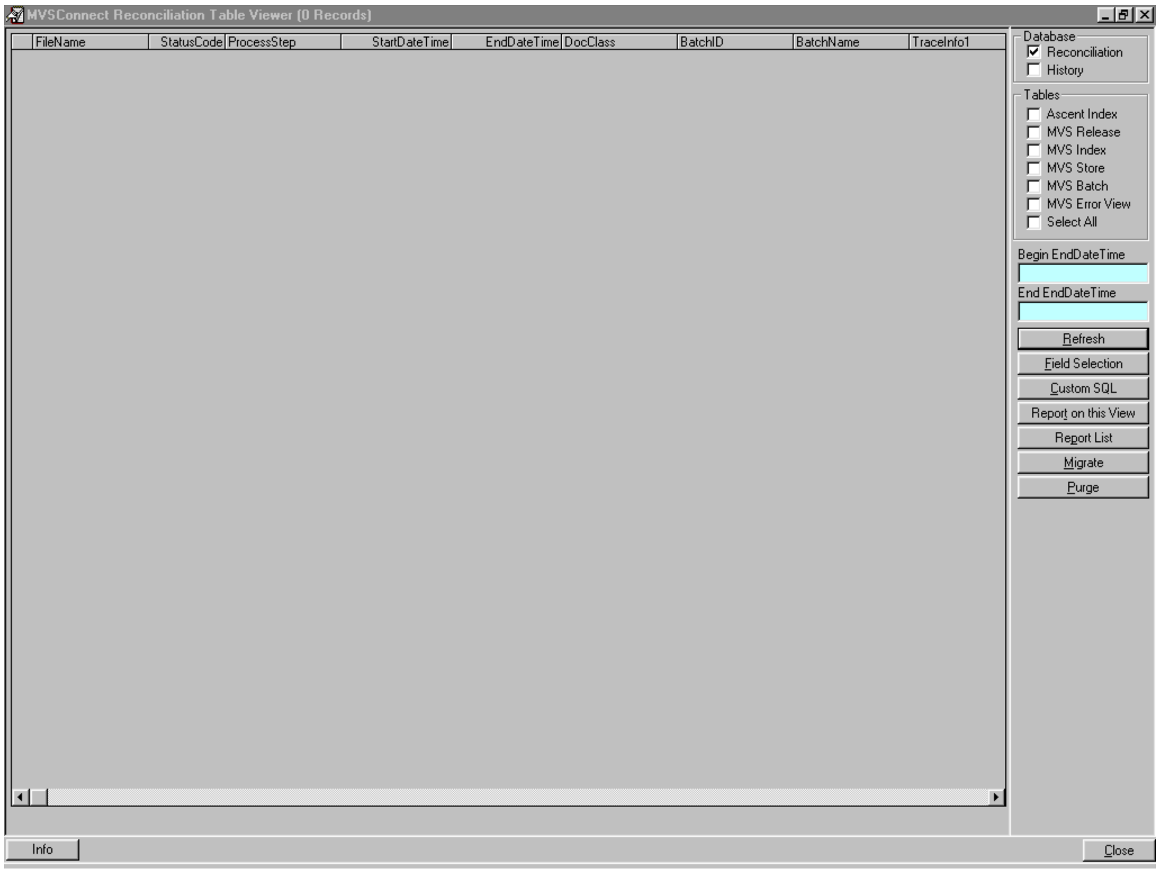
- **Reconciliation Data** – This button will display the MVSConnect Reconciliation data grid.
- **Ascent Batch Data** – This button will display the Ascent Capture Batch data grid.
- **Saved Reports –** This will display a screen that allows you to add, view, or remove saved reports. This will be discussed in detail in the Operation section of MVSAdmin
- **Setup** – MVSAdmin configuration variables.
- **Exit** - Terminate the program.

The function of each button is described in detail below.

RECONCILIATION DATA SCREEN

When the Reconciliation Data button is clicked, the Reconciliation Table Viewer screen will be displayed as seen below.

The large gray area is the data grid that will contain reconciliation data as tables are selected. On the right of the screen are several options for viewing data in the data grid. They are as follows:



<u>Databases</u>

There are two databases to choose from for viewing reconciliation data.

- **Reconciliation –** Displays data in the reconciliation database, which all MVSConnect components write directly to
- **History –** Displays data in the History database, which is data that has been migrated from the Reconciliation database.

Tables

There are 6 tables that can contain reconciliation data. The are:

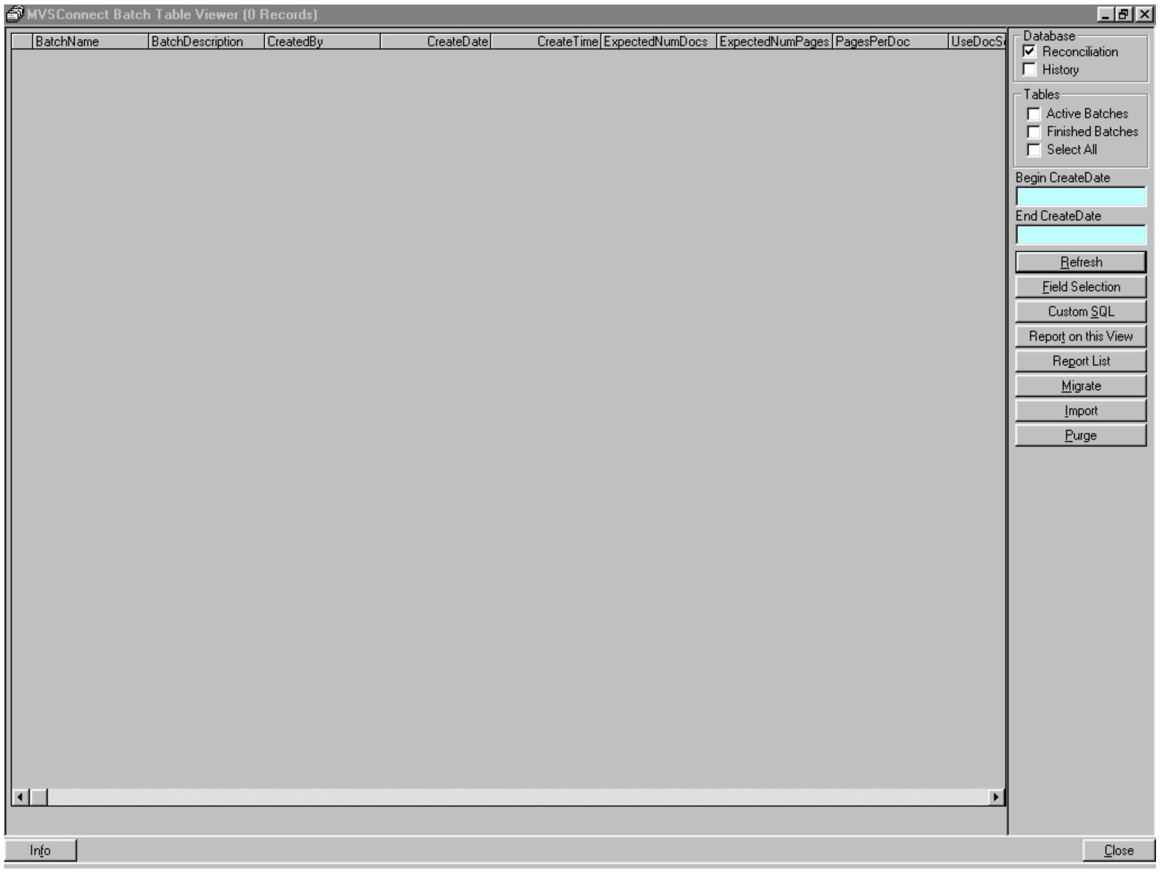
- Ascent Index – reconciliation data from Ascent Capture's Index/Validation module.
- MVSRelease – reconciliation data from the combined MVSConnect / Ascent Capture Release module.
- MVSIndex – reconciliation data from the MVSIndex module.
- MVSStore – reconciliation data from the MVSStore module.
- MVSBatch – reconciliation data from the MVSBatch module (only if running MVSBatch).
- MVSErrorView – reconciliation data from the MVSErrorViewer module.
- Select All – selects all tables for viewing reconciliation data.

The remaining options and buttons on the right of the screen are as follows:

- Begin EndDataTime – for running queries on a table, this is the earliest time in the EndDateTime field to display. If left blank, all entries will be displayed.
- End EndDateTime – for running queries on a table, this is the latest time in the EndDateTime field to display. If left blank, all entries will be displayed.
- Refresh Button – click to refresh the data grid after a change was made to the query options.
- Field Selection – click to choose specific fields to display on the data grid. Field selection files can be created to save for specific ways to view data.
- Custom SQL – click to create custom queries for viewing data in the data grid.
- Report on this View – creates a quick report on the data currently shown on the grid.
- Report List – list of all created reports to display
- Migrate – click to move data from the Reconciliation database to the History database (discussed in more detail in the Operation section)
- Purge – click to delete data currently displayed on the screen (discussed in more detail in the Operation section)

ASCENT BATCH DATA SCREEN

When the Ascent Batch Data button is clicked, the Batch Table Viewer screen will be displayed as seen below.



The large gray area is the data grid that will contain Ascent Capture Batch data as tables are selected. Ascent Batch Data consists of any existing batches in Ascent Capture, as well as batches that were completed and released from Ascent Capture. Retrieving this data is explained below.

On the right of the screen are several options for viewing data in the data grid. They are as follows:

Databases

There are two databases to choose from for viewing reconciliation data.

- Reconciliation – Displays data in the reconciliation database, which all MVSConnect components write directly to
- History – Displays data in the History database, which is data that has been migrated from the Reconciliation database.

Tables

There are 3 tables that can contain reconciliation data. The are:

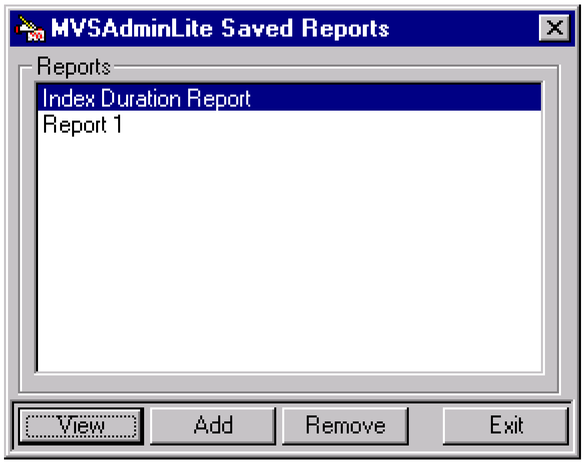
- Active Batches – reconciliation data of active batches currently in Ascent Capture.
- Finished Batches – reconciliation data of completed batches that have been released from Ascent Capture
- Select All – selects all tables for viewing reconciliation data.

The remaining options and buttons on the right of the screen are as follows:

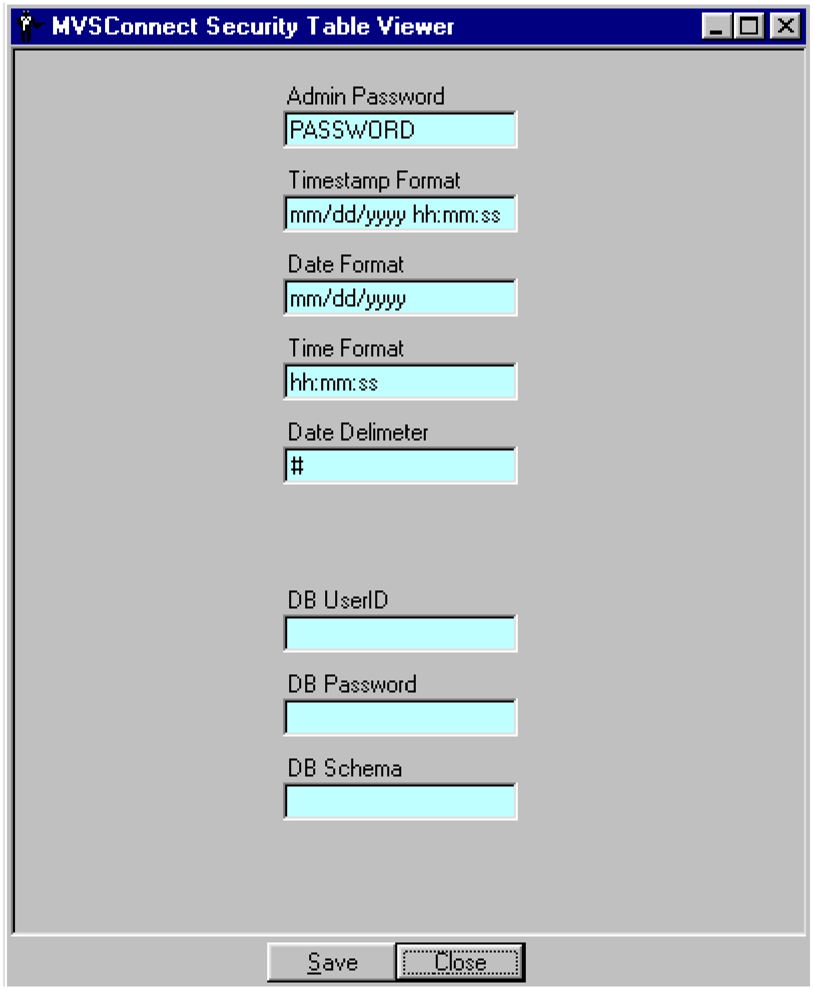
- Begin EndDataTime – for running queries on a table, this is the earliest time in the EndDateTime field to display. If left blank, all entries will be displayed.
- End EndDateTime – for running queries on a table, this is the latest time in the EndDateTime field to display. If left blank, all entries will be displayed.
- Refresh Button – click to refresh the data grid after a change was made to the query options.
- Field Selection – click to choose specific fields to display on the data grid. Field selection files can be created to save for specific ways to view data.
- Custom SQL – click to create custom queries for viewing data in the data grid.
- Report on this View – creates a quick report on the data currently shown on the grid.
- Report List – list of all created reports to display
- Migrate – click to move data from the Reconciliation database to the History database (discussed in more detail in the Operation section)
- Purge – click to delete data currently displayed on the screen (discussed in more detail in the Operation section)

SAVED REPORTS SCREEN

Clicking on *Saved Reports* will bring up a screen that allows you to add saved reports to a list for later retrieval and viewing. You may add and delete saved reports to this list. This list of reports can be reports that you have created in Crystal Reports, but which are not bound to the MVSAdmin reporting controls.



Clicking on the *View* button will display the selected report using Crystal Reports.



Admin Password: PASSWORD
Timestamp Format: mm/dd/yyyy hh:mm:ss
Date Format: mm/dd/yyyy
Time Format: hh:mm:ss
Date Delimeter: #
DB UserID:
DB Password:
DB Schema:

SETUP SCREEN

Setup is where all configuration information for MVSAdmin is located. The initial password for *Security* is "Password", but can be changes upon entering the *Setup* screen.

Each configuration field is described below:

- **Admin Password** : this is the password to get into this screen.
- **Timestamp Format**: Different databases may require different timestamp field formats for ODBC. Microsoft Access requires this field to be "mm/dd/yyyy hh:mm:ss", while DB/2 requires this field to be "yyyy-mm-dd hh:mm:ss.000000"
- **Date Format**: Enter "yyyy-mm-dd" for either DB/2 or Microsoft Access.
- **Time Format**: Enter "hh:mm:ss" for either DB/2 or Microsoft Access.
- **Date Delimiter**: Enter "#" for Microsoft Access and a single quote for DB/2.
- **DB UserID**: Enter any UserID that is required to connect to the MVSConnect database.
- **DB Password**: Enter any password that is required to connect t the MVSConnect database.
- **DB Schema**: Enter any schema that is required to read/write to MVSConnect tables. Microsoft Access typically does not require this. Other databases such as DB/2 do.

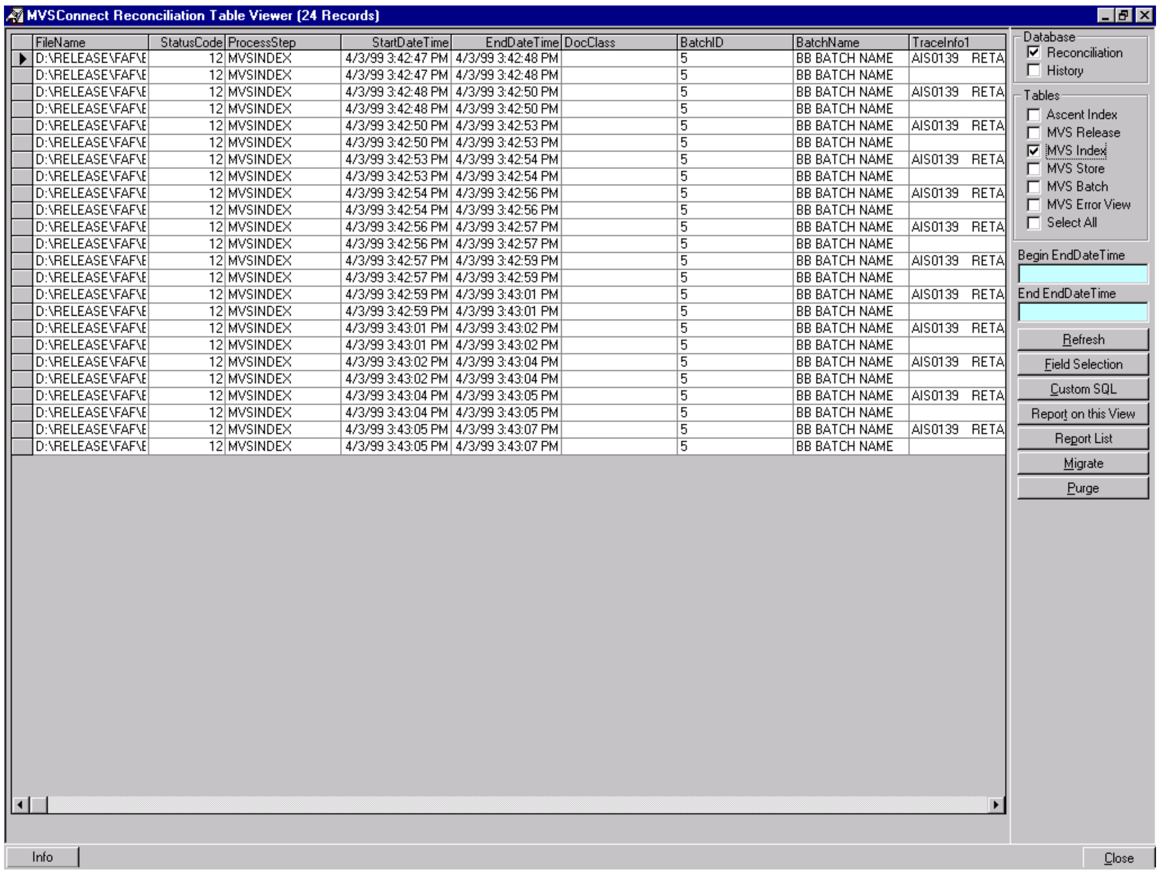
Operation

VIEWING DATA

Both Ascent Capture batch-level data and MVSConnect Reconciliation data may be viewed in data grids. Users may select either all fields in the database to view or a select number of fields. Fields may also be selected for viewing in any order. Sorting of the displayed data may also be done on any displayed field. Additionally, a date range may be selected for viewing, providing a method of limiting the amount of data being retrieved and viewed.

The screen displays that follow outline the features for both Reconciliation and Batch information, although only the Reconciliation data display will be shown.

SORTING DISPLAYED DATA



MVSConnect Reconciliation Table Viewer (24 Records)

FileName	StatusCode	ProcessStep	StartDateTime	EndDateTime	DocClass	BatchID	BatchName	TraceInfo1
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:47 PM	4/3/99 3:42:48 PM		5	BB BATCH NAME	AIS0139 RETA
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:47 PM	4/3/99 3:42:48 PM		5	BB BATCH NAME	
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:48 PM	4/3/99 3:42:50 PM		5	BB BATCH NAME	AIS0139 RETA
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:48 PM	4/3/99 3:42:50 PM		5	BB BATCH NAME	
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:50 PM	4/3/99 3:42:53 PM		5	BB BATCH NAME	AIS0139 RETA
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:50 PM	4/3/99 3:42:53 PM		5	BB BATCH NAME	
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:53 PM	4/3/99 3:42:54 PM		5	BB BATCH NAME	AIS0139 RETA
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:53 PM	4/3/99 3:42:54 PM		5	BB BATCH NAME	
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:54 PM	4/3/99 3:42:56 PM		5	BB BATCH NAME	AIS0139 RETA
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:54 PM	4/3/99 3:42:56 PM		5	BB BATCH NAME	
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:56 PM	4/3/99 3:42:57 PM		5	BB BATCH NAME	AIS0139 RETA
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:56 PM	4/3/99 3:42:57 PM		5	BB BATCH NAME	
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:57 PM	4/3/99 3:42:59 PM		5	BB BATCH NAME	AIS0139 RETA
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:57 PM	4/3/99 3:42:59 PM		5	BB BATCH NAME	
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:59 PM	4/3/99 3:43:01 PM		5	BB BATCH NAME	AIS0139 RETA
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:42:59 PM	4/3/99 3:43:01 PM		5	BB BATCH NAME	
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:43:01 PM	4/3/99 3:43:02 PM		5	BB BATCH NAME	AIS0139 RETA
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:43:01 PM	4/3/99 3:43:02 PM		5	BB BATCH NAME	
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:43:02 PM	4/3/99 3:43:04 PM		5	BB BATCH NAME	AIS0139 RETA
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:43:02 PM	4/3/99 3:43:04 PM		5	BB BATCH NAME	
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:43:04 PM	4/3/99 3:43:05 PM		5	BB BATCH NAME	AIS0139 RETA
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:43:04 PM	4/3/99 3:43:05 PM		5	BB BATCH NAME	
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:43:05 PM	4/3/99 3:43:07 PM		5	BB BATCH NAME	AIS0139 RETA
D:\RELEASE\FAF\E	12	MVSINDEX	4/3/99 3:43:05 PM	4/3/99 3:43:07 PM		5	BB BATCH NAME	

Database
- [✓] Reconciliation
- [ ] History

Tables
- [ ] Ascent Index
- [ ] MVS Release
- [✓] MVS Index
- [ ] MVS Store
- [ ] MVS Batch
- [ ] MVS Error View
- [ ] Select All

Begin EndDateTime

End EndDateTime

By clicking on a column title in the grid you may sort the contents displayed in the grid by that column.

Database Selection

If you wish to view the most recent data written to the database by the various components of MVSConnect, click on the *Reconciliation* checkbox. If you wish to view data that you have migrated to the history database, check the *History* checkbox.

Note that the *History* database can grow quite large over time and that retrieving all of the data in that database can be a bit time consuming. You may wish to restrict your view of these tables by entering a specific date/time range (note that the time is in 24-hr format) or viewing a single table rather than all of them at once. Or you may wish to save copies of the History Database (by month for example) to keep the size of it relatively small.

Tables Section

Here you can select which tables in the Reconciliation database you wish to view. Click on the specific tables you wish to view or the *Select All* button to view data in all tables.

Restricting the Data View by Date Range

The *Begin EndDateTime* and *End EndDateTime* entry fields allow you to restrict the data being viewed by a date range. Enter a valid date/time range and the click the *Refresh* button to see data that falls within your date range. The date/time range that you enter will use the *EndDateTime* field of the Reconciliation tables and the *CreateDate* field of the Batch tables to restrict the view. Note that for the Batch view, only a date may be entered.

Restricting the Fields to View

By clicking on the *Field Selection* button, the Field Selection screen is displayed. This screen (shown below) allows users to select which fields they wish to display. Field selections may be saved and later retrieved from this screen as well.

Important Note: You may find that when restricting the fields to display and you select multiple tables to view, that not all data in the multiple tables is displayed. Try viewing a single table at a time. This is an SQL issue with the Reconciliation databases. You can also try added a few other fields such as the *TraceInfo1*or *TraceInfo2* fields to your list of restricted fields.

The list on the left shows all fields in the tables. The fields displayed will differ for the Reconciliation tables and the Batch tables.

Fields can be either selected individually by highlighting a field and clicking the upper right arrow button or by double clicking on a field in the *All Fields* list. This adds the field to the *Selected Fields* list. Fields can be removed from the *Selected Fields* list by either highlighting the selected field and clicking the upper left-arrow button or by double clicking on the field in the *Selected Fields* list.

The entire list of fields can be selected by clicking on the lower right-arrow button (to add) or the lower left-hand button (to remove).

Clicking the *View* button will close this screen and use the selected fields for the display data.

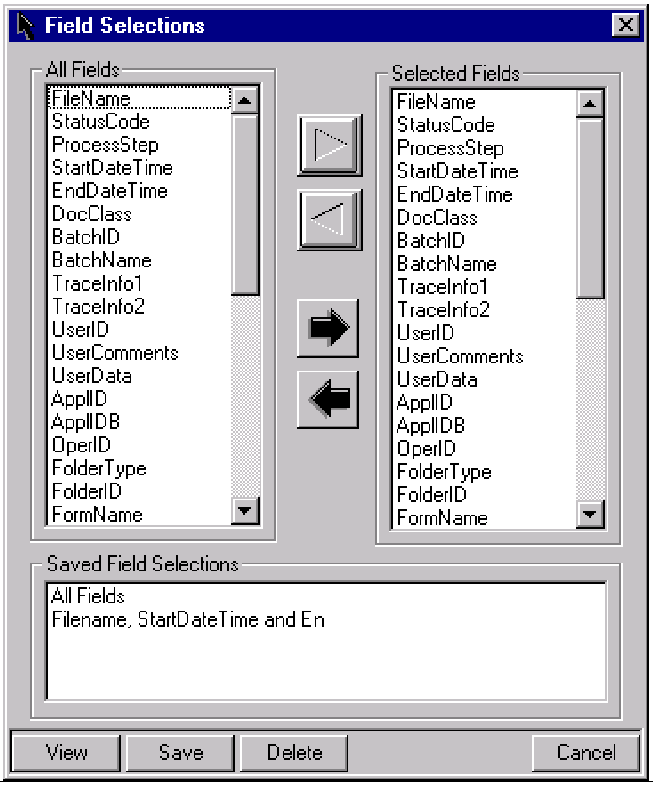
Clicking the *Save* button will prompt the user for a name under which to save the current selection. The saved selection will be added to the list of *Saved Field Selections* displayed at the bottom of the screen and may be retrieved by double-clicking the selection desired (which displays the fields to be viewed) and then clicking the *View* button.

Clicking the *Delete* button will delete the selected item from the list.

The *Cancel* button closes the window without making changes to the currently displayed data.

CUSTOM SQL STATEMENTS

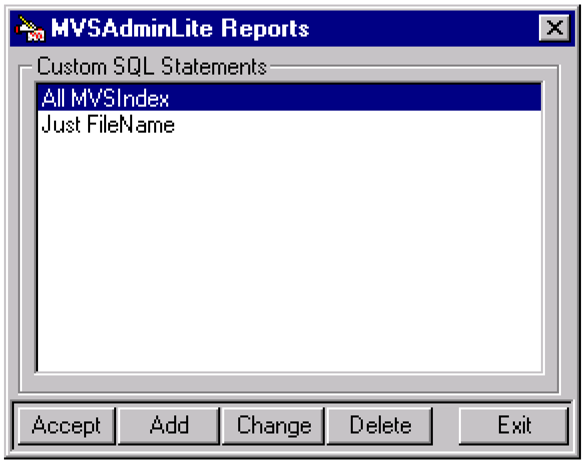
Clicking this button brings up a list of saved custom SQL statements, as shown below.



All Fields: FileName, StatusCode, ProcessStep, StartDateTime, EndDateTime, DocClass, BatchID, BatchName, TraceInfo1, TraceInfo2, UserID, UserComments, UserData, ApplID, ApplDB, OperID, FolderType, FolderID, FormName

Selected Fields: FileName, StatusCode, ProcessStep, StartDateTime, EndDateTime, DocClass, BatchID, BatchName, TraceInfo1, TraceInfo2, UserID, UserComments, UserData, ApplID, ApplDB, OperID, FolderType, FolderID, FormName

Saved Field Selections:
All Fields
Filename, StartDateTime and En



You may add new custom SQL statements to the list by clicking the *Add* button. You will be prompted for a name to identify the SQL statement (to be displayed in the list) and then prompted to enter the SQL statement itself. You may also *Change* and *Delete* entries from the list by clicking the appropriate button.

Clicking the *Accept* button uses the selected SQL entry as a basis for displaying data in the data grid.

CREATING REPORTS

There are a number of ways to create reports in MVSAdmin. Two buttons appear on the *Reconciliation Table Viewer* screen for this purpose – the *Report on this View* and *Report List* buttons.

The *Report on this View* button utilizes the current SQL statement being used to display data to build a *bound Crystal Reports © report*. (Note that you can click on the *Info* button at the bottom left of the screen to see exactly what SQL statement is being used).

From the Crystal Reports window, you can save the report to disk. Such reports are *bound* to the data control in MVSAdmin. You *cannot* view these reports from within the Crystal Reports application, but only from within MVSAdmin.

You *can*, however (and should) modify such saved reports within Crystal Reports to apply good formatting, column widths, etc., and then view them from within MVSAdmin.

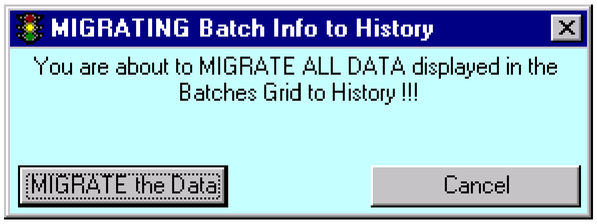
The *Report List* button displays a list of reports that you have saved from within MVSAdmin. You can add and delete reports to/from this list.

MIGRATING DATA TO HISTORY

MVSConnect comes with two databases – a Reconciliation database and a History database. All new data goes into the Reconciliation database.

You can migrate data from the Reconciliation database to the History database by clicking the *Migrate* button. This will migrate all *displayed* data in the grid.

Migrating data to History will help keep the Reconciliation database small. This can be important for performance considerations, especially if using Microsoft Access.

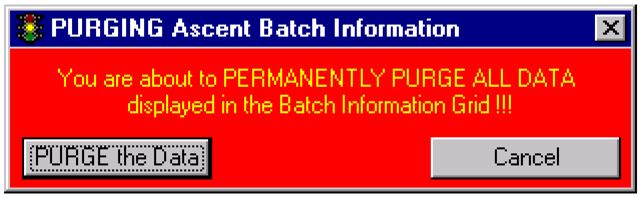


You are about to MIGRATE ALL DATA displayed in the Batches Grid to History !!!

The above screen gives you opportunity to continue and *MIGRATE the Data* or to *Cancel* the migration.

PURGING DATA

By clicking on the *Purge* button, you may delete (permanently) all data displayed in the display grid. You will be given several chances to cancel the purge. Be aware that once data is purged, it is *permanently deleted* .

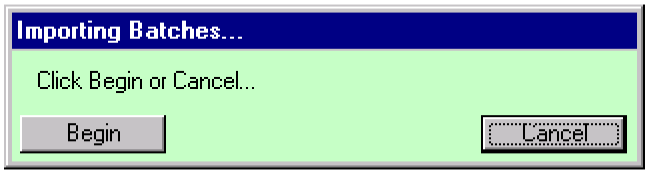

PURGING Ascent Batch Information

You are about to PERMANENTLY PURGE ALL DATA displayed in the Batch Information Grid !!!

The above window is displayed prior to purging. Click the *PURGE the Data* button to purge.

IMPORTING BATCH DATA

While Reconciliation data is automatically written to the Reconciliation tables as transactions occur, Ascent Capture Batch data must be first *exported* (in the case of Active Batch data) and then *imported* into the batch tables in the Reconciliation database.


Importing Batches...

Click Begin or Cancel...

The Batch data screen has an addition *Import* button whereby such exported data may be imported for reporting.

Batch level data is imported into the ActiveBatches and FinishedBatches tables within the MVSConnect database (or ActiveBatches3 and FinishedBatches3 in the case of Ascent Capture 3.0).

In prior releases of MVSAdmin, batch information was split into three tables – the BATCH table, the LOGFILE1 table and the LOGFILE2 table. The LOGFILE1 and LOGFILE2 tables have been consolidated into a single table. Now batch-level data is imported into just 2 tables (as mentioned above) – the ACTIVEBATCHES table for active batch information that has been exported from Ascent Capture and the FINISHEDBATCHES table for batches that have left Ascent Capture (for whatever reason).

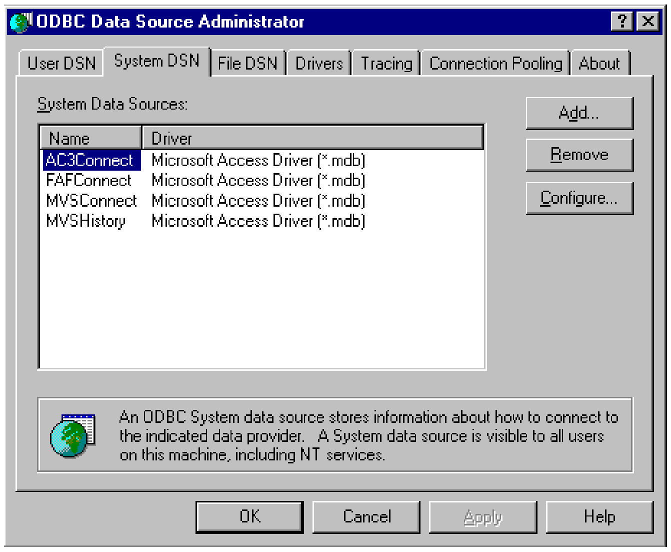
The import process can be quite time consuming, so it should be done on a regular basis.

NOTE: Once data is migrated to the History database, it is *PURGED* from the Reconciliation database. If you cancel the migration while it is in progress, data will NOT be purged. The next time you migrate the data, you may notice that the word “duplicate” appears next to the batch name being migrated. Duplicate data is NOT migrated to history.

The above screen gives the user opportunity to *Begin* importing batch data or to *Cancel* the import operation.

Appendix A – Required ODBC Entries

The following screen is an example of the ODBC Administration screen after configuration.



System Data Sources:

Name	Driver
AC3Connect	Microsoft Access Driver (*.mdb)
FAFConnect	Microsoft Access Driver (*.mdb)
MVSConnect	Microsoft Access Driver (*.mdb)
MVSHistory	Microsoft Access Driver (*.mdb)

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

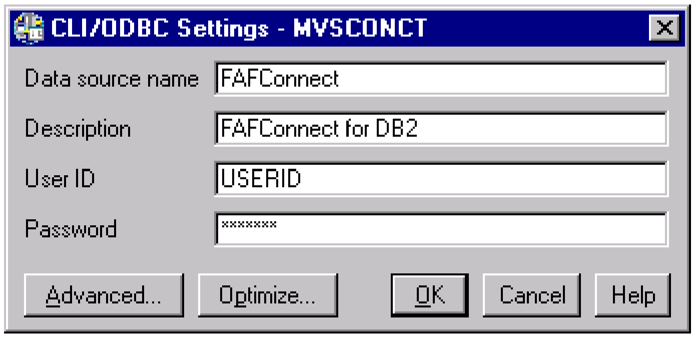
Be aware that the ODBC driver that you use will vary depending on the database used. SYSCOM has certified that IBM's Universal Database 5.2 for NT as well as Microsoft Access work with MVSConnect. Other databases may work, depending on their ODBC drivers and any peculiarities in their implementation of SQL.

Please set up the following Data Source Names exactly as you see them and select the supplied database that they should be pointing to. (Note: The paths to your database may differ from the illustration) The name of the database can be found on the diagram in the Database section.

MVSConnect DSN should point to the MVSCONCT.MDB file if using Microsoft Access, or to the same database as the MVSConnect DSN if using other ODBC data sources.

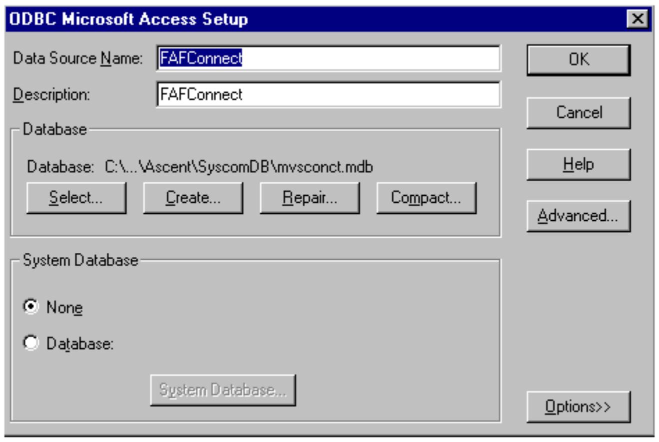
Below are sample screens from the DSN setup for both IBM's DB2 (UDB) and Microsoft Access.

DB2:



Data source name	FAFConnect
Description	FAFConnect for DB2
User ID	USERID
Password	*******

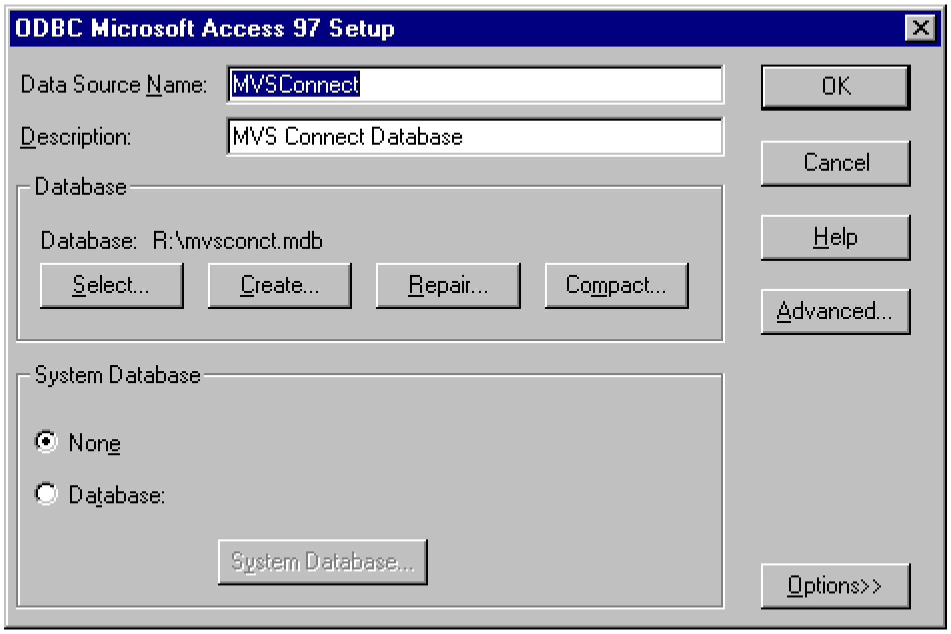
Microsoft Access:



Data Source Name:	FAFConnect
Description:	FAFConnect
Database:	C:\...\Ascent\SyscomDB\mvsconct.mdb

System Database: [✓ None] Database:

The MVSConnect DSN entry should be set up the same way.


ODBC Microsoft Access 97 Setup

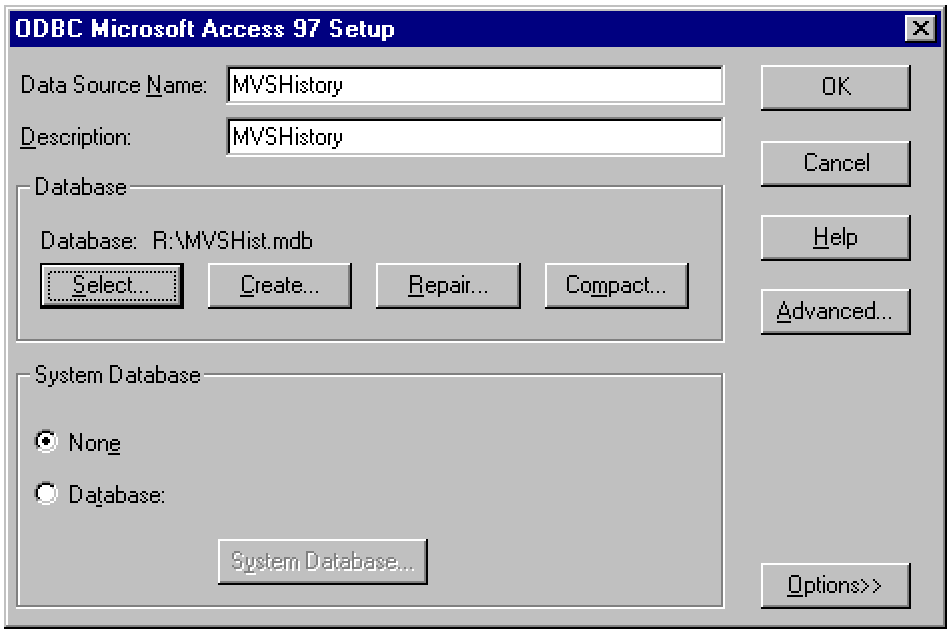
Data Source Name: MVSConnect

Description: MVS Connect Database

Database: R:\mvsconct.mdb

System Database: None

The MVSHistory DSN entry should point to the MVSHIST.MDB Access database or a history database set up on another ODBC data source.


ODBC Microsoft Access 97 Setup

Data Source Name: MVSHistory

Description: MVSHistory

Database: R:\MVSHist.mdb

System Database: None

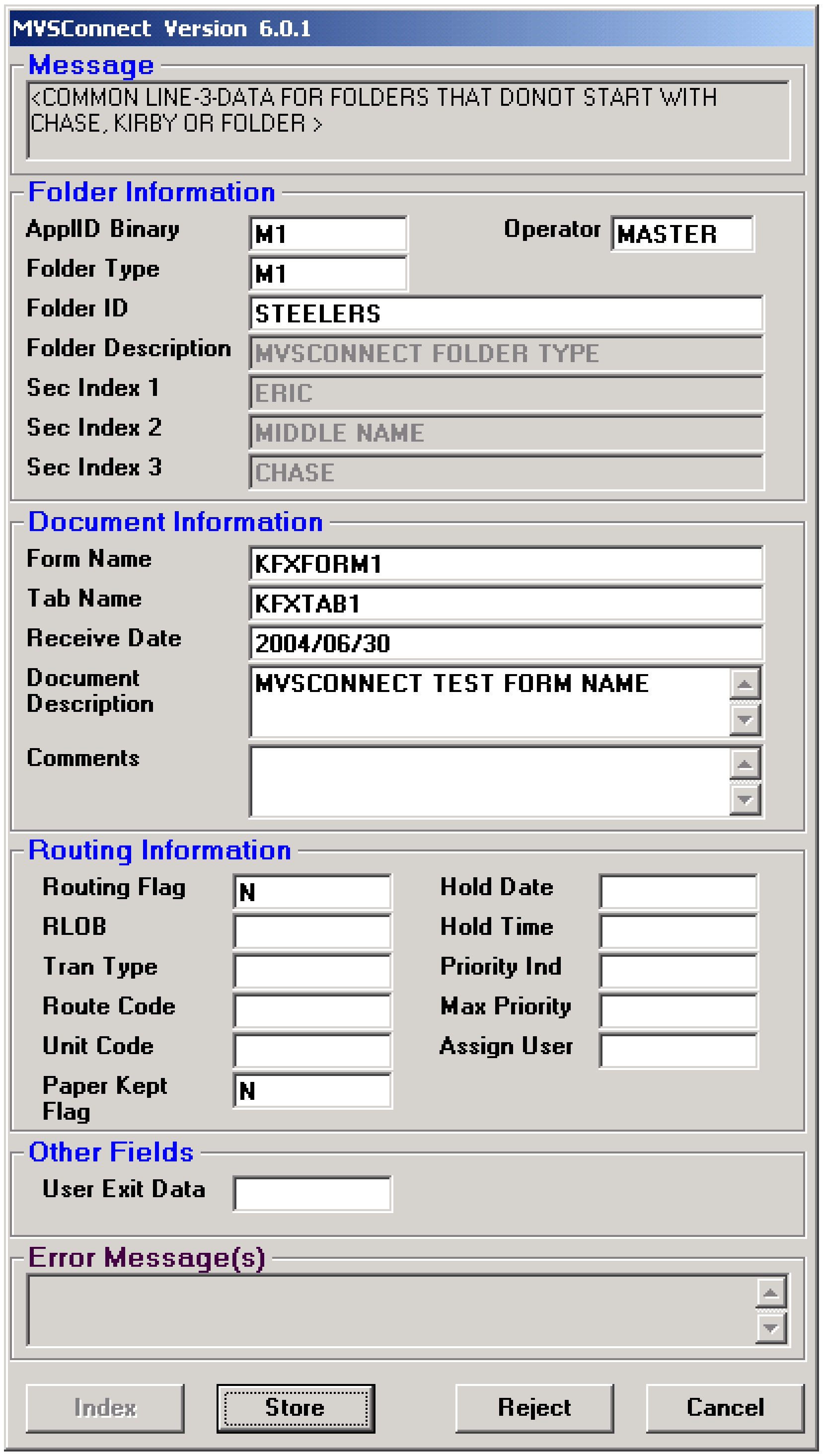
Appendix B – Planning Worksheet

The following directories are only suggestions for setting up your system. If you are installing on a **server** then the network drive used should be the one where **Ascent Capture Server** is installed, otherwise it should be your local drive. Create all directories in the Ascent Directory. These are the recommended directories that should be created:

- \temprel – Output Directory for MVSRelease, Input for MVSFileDistributionManager
- \temprel\input – Output Directory for MVSFileDistributionManager, Input Directory for MVSIndex and Indexing Directory for MVSStore
- \temprel\moveerr – Error (Output) Directory for MVSFileDistributionManager
- \temprel\work – Output Directory for MVSIndex , Input Directory for MVSStore
- \temprel\error – Error Directory for MVSIndex, Input Directory for MVSErrorView
- \temprel\archive – Archive (Output) Directory for MVSStore
- \temprel\storeerr – Error (Output) Directory for MVSStore
- \temprel\corrupt – Corrupt Error (Output) Directory in MVSIndex

Appendix C – MVSConnect Reject Logic

Reject Logic is a function to allow a document to be forced to *MVSErrorViewer* at *MVSValidation* time. When the **USE_REJECT_LOGIC** option in the **MVSValidation.ini** file is set to **YES**, the *MVSValidation* screen will have a **Reject** button as seen below.

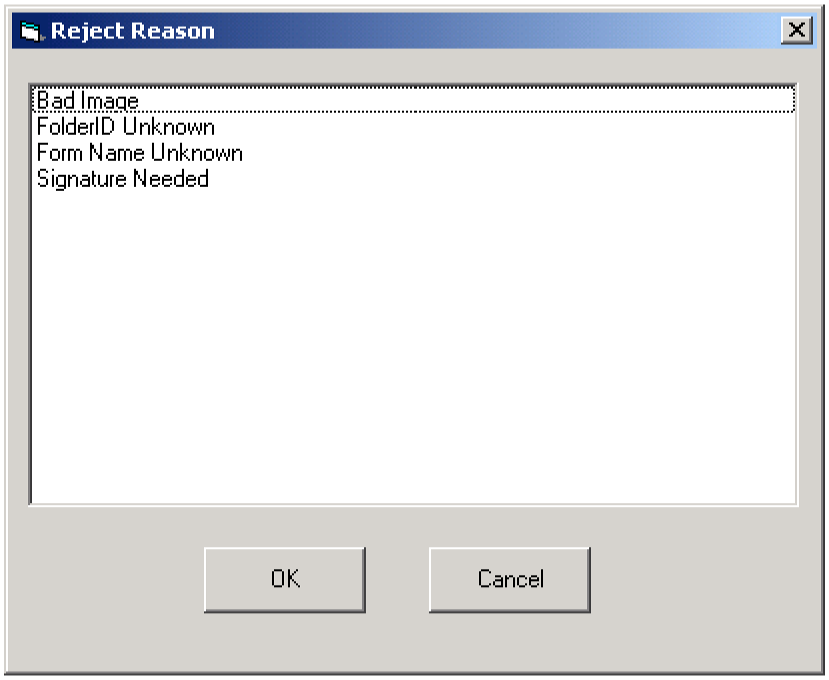


<COMMON LINE-3-DATA FOR FOLDERS THAT DONOT START WITH CHASE, KIRBY OR FOLDER >

ApplID Binary	M1
Operator	MASTER
Folder Type	M1
Folder ID	STEELERS
Folder Description	MVSCONNECT FOLDER TYPE
Sec Index 1	ERIC
Sec Index 2	MIDDLE NAME
Sec Index 3	CHASE
Form Name	KFXFORM1
Tab Name	KFXTAB1
Receive Date	2004/06/30
Document Description	MVSCONNECT TEST FORM NAME
Comments	

Routing Flag	N	Hold Date	
RLOB		Hold Time	
Tran Type		Priority Ind	
Route Code		Max Priority	
Unit Code		Assign User	
Paper Kept Flag	N		
User Exit Data			

Clicking the **Reject** button will prompt the user to choose a reject reason from a list of available reject reasons.



The **Reject Reasons** are to be entered into a text file by you the user. Once this text file of **Reject Reasons** has been created, it should be saved, the **Validaiton.ini** file will have an entry to point to that file to obtain the valid **Reject Reasons** for the user to choose from. A sample Reject Reasons file is located at c:\winnt\system32\rejects.txt.

Once a **Reject Reason** has been chosen, this document will be closed so the next document can be processed. The document will be rejected by *MVSIndex* and sent to *MVSErrorViewer*, where the appropriate action can be taken. To enable this **Reject Logic**, set the following fields in the **Validation.ini** file:

USE_REJECT_LOGIC=YES

REJECT_CODE_FILE=Path and filename where reject codes are located (ex: c:\winnt\system32\rejects.txt)

Appendix D –ImagePlus Fields Used In MVSConnect

Receive Data Structure

ImagePlus Fields passed into the MVSIndex step of MVSConnect

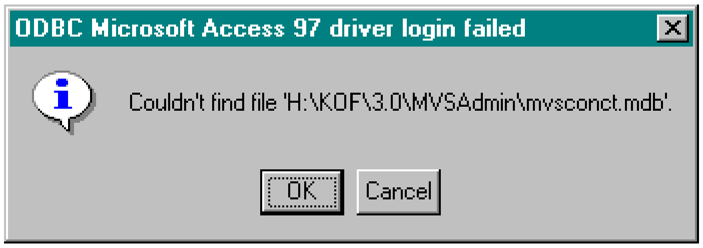
Field Name (Release Setup)	**Length**	**Host Representation AIS and EYPT tables**	**Notes**
ApplID Alpha	08	APPL_ID from AISAPPL	In FWA, up to 8 bytes are allowed. In AIS+, 2 bytes are allowed.
ApplID Binary	02 COMP Numeric	APPL_ID_CD from AISAPPL	Numeric 2 byte representation of the ApplID
OperatorID	08	Logged in the Event Table(EYPTEVNTxx)	User that indexed the image.
Folder Type	08	FLDR_FOLDTYPE from AISFLDR	There is a numeric equivalent in the EYPT tables of FOLDTYCD. AISFLDER = FLDR_FOLDTYCD.
FolderID	26	Primary Folder Index EYPTFOLDxx = FOLDID	There is a timestamp, FOLDTKN created as alternate way to reference this value after it is created.
Form Name	16	DCMT_FORMNAME from AISDCMT FORMCD in EYPT tables	
TabName	16	TABS_DESCRIPTION from AISTABS	There is a numeric equivalent in the EYPT tables of TABCD. AISTABS = TABS_TABCD.
RecieveDate	08	RECVDATE in EYPT tables	Defaults to current if not supplied.
DocDesc	60	DCMT_DESCRIPTION from AISDCMT OBJDESC in the EYPT tables	Defaults to the AISDCMT value if not supplied.
Comments	120	Logged in the EYPTEVNTxx table	
FolderDesc	60	FLDR_DESCRIPTION in AISFLDR FOLDDESC in the EYPT tables.	Defaults to the AISFLDR value if not supplied.
S1stIndex	40	Within the EYPTSNDXxx, Two parts: INDXTYPE=1,2, or 3 INDXVAL= Actual Index Value	Defined on the AISFLDR table.
S2ndIndex	40	" "	" "
S3rdIndex	40	" "	" "

Field Name (Release Setup)	Length	Host Representation AIS and EYPT tables	Notes
Routing Flag	01	N/A	Determines if the object should be placed into routing. Does not map to a table. The AISDCMT table has a default route information that will determine by default whether to route or not. That can be overridden with this flag. If set to "Y", you must populate the other route fields if the AISDCMT table does not have them already.
RoutingLineofBusiness	06	RLTT_RLOB in the AISRLTT table DCMT_RLOB in the AISDCMT table WORK_RLOB in the AISWORK table	Uses AISDCMT values if not supplied.
TransactionType	06	RLTT_TRAN_TYPE in the AISRLTT table DCMT_TRANTYPE in the AISDCMT table WORK_TRAN_TYPE in the AISWORK table	" "
RouteCode	06	UNRC_RT_CODE in the AISUNRC table. WORK_RT_CODE in the AISWORK table	
UnitCode	04 Numeric	UNRC_RT_UNIT in the AISUNRC table UNIT_RT_UNIT in the AISUNIT table WORK_RUNIT in the AISWORK table	
PriorityIndicator	01	WORK_PRIORITY from the AISWORK table	
HoldDate	08	WORK_HOLD_DATE from the AISWORK table	
HoldTime	04	WORK_HOLD_TIME from the AISWORK table	
AssignedUser	08	WORK_ASGN_EMP from the AISWORK table	
PaperKeptFlag	01	DCMT_PAPER_RET_FL from the AISDCMT table. ORIGKEPT from the EYPT tables	
MaxPriority	03	APPL-MAX-PRIORITY from AISAPPL	Allows you to override this value to make an ob
Extended Comments	240	Logged in the EYPTEVNTxx table	AIS+ only
LineOfBusiness1	10		
LineOfBusiness2	10		
LineOfBusiness3	10		
Line3Data	50		

Appendix E – Troubleshooting

ODBC Connection Problems

The program and/or the operating system will display various error windows if an ODBC connection cannot be made.



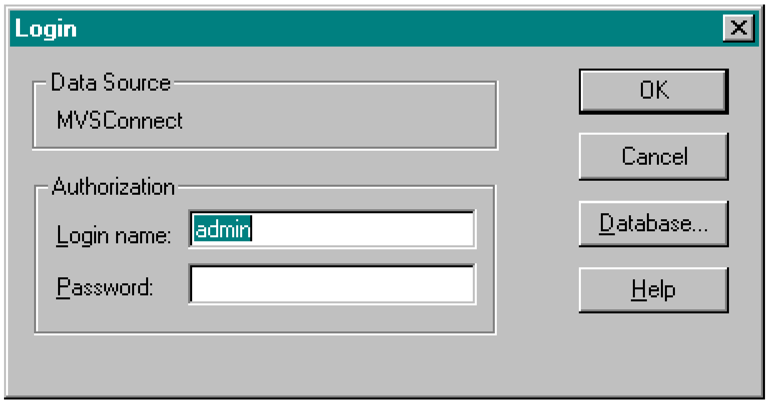
ODBC Microsoft Access 97 driver login failed

Couldn't find file 'H:\KOF\3.0\MVSAdmin\mvsconct.mdb'.

Below are some examples:

Occurs when the database specified in the ODBC DSN for MVSConnect cannot be found.

Clicking OK produces this:



Login

Data Source: MVSConnect

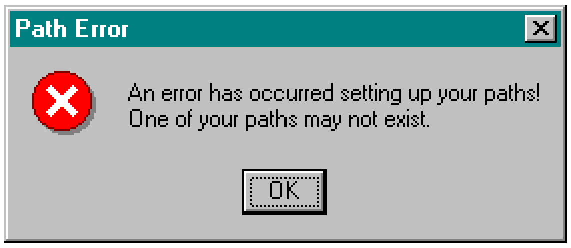
Authorization

Login name: admin

Password:

Path Problems

When the program starts and one of the paths in the config is bad, this screen is displayed:



Path Error

An error has occurred setting up your paths!
One of your paths may not exist.

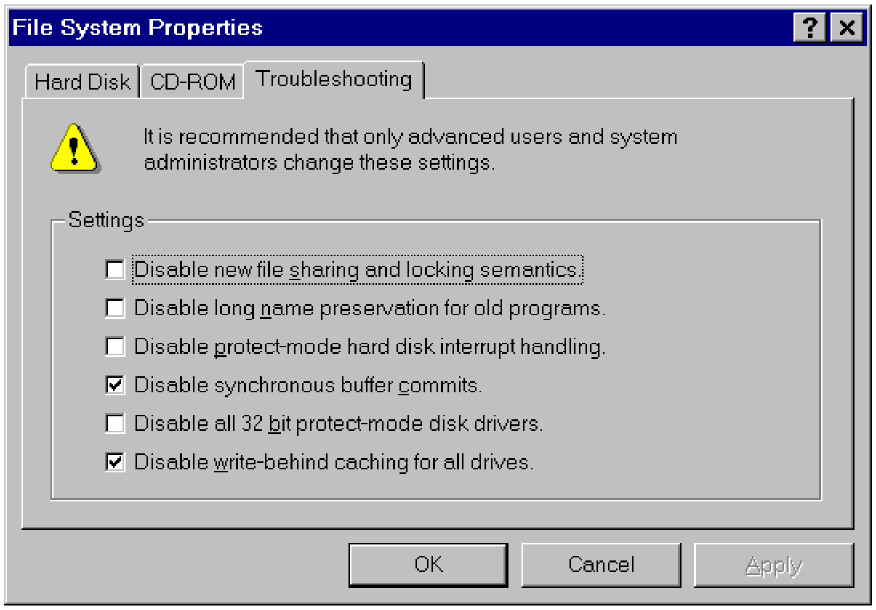
Changes to the Windows 95 Environment

The Index program “feeds” the Store program by generating files in a work directory. Windows 95 caching may cause synchronization problems between these programs, expressed as “file not found” errors. Should this occur, several changes could be made to the Windows 95 environment to resolve the problem.

To make these changes, do the following:

1. Bring up **Control Panel** from the Start/Settings menu.
2. Double click on the **System** icon.
3. Click on the **Performance** notebook tab.
4. Click on the **File System** button.
5. Click on the **Trouble Shooting** notebook tab.
6. Check the “Disable synchronous buffer commits” check box.
7. Check the “Disable write-behind caching for all drives” check box.

The picture below shows this screen:



It is recommended that only advanced users and system administrators change these settings.

Settings

- ☐ Disable new file sharing and locking semantics.
- ☐ Disable long name preservation for old programs.
- ☐ Disable protect-mode hard disk interrupt handling.
- ☑ Disable synchronous buffer commits.
- ☐ Disable all 32 bit protect-mode disk drivers.
- ☑ Disable write-behind caching for all drives.

Common MVSValidation Errors

Problem	Possible Cause
Return Code=53	Host communication configuration information incorrectly setup in the Validation.ini file.

Common MVSErrorViewer Errors

Problem	Possible Cause
Invalid Kofax Request Flag	Kofax Request flag is not valid. Should be ‘F’ or ‘A’. The batch class could be set up incorrectly, or there could be a host problem.

Common Index & Store Return Codes

When initially setting up the host and workstations, it is common that something in the configuration does not match between the two systems. Perhaps the Local LU names are different, or the Partner LU names. Perhaps the Symbolic Destination Name in the communications package does not match what was entered in the KOFCONFG program... or perhaps the workstation LU was not set up properly in VTAM on the host. Any number of configuration problems could cause the Index and Store programs to return error codes.

A few common codes are listed below with SOME of the possible causes.

Problem	Possible solution
rc = 24	CPIC side info in communications package not set up properly (when using SNA connection)
rc = 1	Session between host and workstation may not be allocated. This can occur under Windows 95, but usually not under NT. Could also be symptomatic of the workstation LU not configured properly on the host.
rc = 17	Host program has abended. Check CICS logs on the host
rc=22 or rc=27	ALTSENDF flag on the host in the IDWKCFTB table needs to be set to “Y”.
rc=992	It may be that the DB2 tables are not set to receive the image type that is attempting to store. For example, if you are trying to store a TIFF image, the DB2 tables must reflect this image type.

CPIC Return Codes

Other return codes are listed below:

Return Code	General Description
0	OK
1	Allocate failure - no retry
2	Allocate failure - retry
3	Conversation type mismatch
5	PIP not specified correctly
6	Security not valid
7	Sync Level Not supported sys
8	Sync Level not supported pgm
9	Transaction Program Name not recognized
10	Transaction Program not available - no retry
11	Transaction Program not available - retry
17	Deallocate Abend
18	Deallocate normal
19	Parameter Error
20	Product Specific Error
21	Program Error No Truncation
22	Program Error Purging
23	Program Error Truncation
24	Program parameter check
25	Program state check
26	Resource failure no retry
27	Resource failure retry
28	Unsuccessful
30	Deallocated Abend SVC
31	Deallocated Abend Timer
32	SVC Error No Trunc
33	SVC Error Purging
34	SVC Error Trunc
35	Operation Incomplete
36	System Event
37	Operation Not Accepted
38	Conversation Ending
39	Send-Recv mode not supported
40	Buffer Too Small
41	EXP Date not supported
42	Deallocate confirm rejected
43	Allocation Error
44	Retry Limit Exceeded
45	No secondary information
46	Security not supported
47	Security Mutual Failed
48	Call Not Supported

File Movement Logic in the Index and Store Programs

1. Upon release of batches from Ascent Capture into the input directory, the index program processes these files and creates a corresponding .DON file in the Output directory. *All files other than the .DON file stay in the input directory*. If a file produces an indexing error, an .ERR file is generated. This .ERR file and the corresponding .TIF file are moved to the Index program's Error directory. All other files remain in the Input directory. If a corrupt .TXT file is found (and thus cannot be processed), it and all associated files are moved to the "corrupt" directory.

2. The store program polls for .DON files in its Input directory (which *must* correspond to the Index program's Output directory). When a .DON file is found, the associated image file in the input directory is stored to the host. If this file is not successfully stored, it and all associated files are placed in the error directory. If the image file is successfully stored, all file types listed in the "to be archived" configuration entry are moved from the Indexing directory to the Archive directory.

3. It is important to note that the Indexing directory of the Store program is the same as the Input directory of the Index program. This means that the original files are not physically moved until they are either in error or successfully stored.