# Large Mailing Operations
# Standards Specification

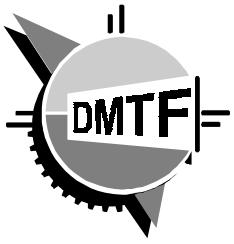
*Version 1.2 FINAL*

*February 28, 1996*

This document defines a standard set of interfaces, objects, and management protocols for managing large mailing and print-on-demand document finishing workflow applications involving multiple hardware and software processes of different types. It is for programmers and hardware developers who need such information to develop or adapt a product to interoperate with other components in an automated document finishing environment. It is a reference, not a tutorial, for readers who design such products and their management applications.

This specification is a product of the IBM Sponsored Desktop Management Task Force Finisher Work Group:

| | |
|---|---|
| Bell + Howell | Kern |
| BESTE Bunch | LASERMAX (Wallace) |
| Böwe Systec | Moore |
| C.O.P.E. | MJB Systems |
| Delphax | Pitney Bowes |
| DevTech Associates | Roll Systems Inc. |
| Gunther International | RR Donnelly & Sons |
| Hunkeler | Siemens-Nixdorf |
| IBM | Standard Register |
| International Data Post | Strålfors |
| J.Q. Technologies (Johnson & Quin) | Xerox |



Technical inquiries and editorial comments should be directed in writing to:

Desktop Management Taskforce(DMTF
M/S JF2-51 Finisher Work Group
2111 N.E. 25th Avenue
Hillsboro, OR 97124
FAX: (503) 696-9027

email: wg-software@dmtf.org

Additional copies of this specification may be obtained free of charge
by contacting:

Desktop Management Task Force
2111 NE 25th Avenue
M/S JF2-51
Hillsboro, OR 97124
PHONE: (503) 696-9300
FAX: (503) 696-9027

**IMPORTANT INFORMATION AND DISCLAIMERS**

1. THIS SPECIFICATION (WHICH SHALL INCORPORATE ANY REVISIONS, UPDATES, AND MODIFICATIONS HERETO) IS FURNISHED FOR INFORMATIONAL PURPOSES ONLY. INTEL CORPORATION, MICROSOFT CORPORATION, DIGITAL EQUIPMENT CORPORATION, HEWLETT-PACKARD, INTERNATIONAL BUSINESS MACHINES CORPORATION, NOVELL INC., SUNCONNECT (A SUN MICROSYSTEMS COMPANY), AND SYNOPTICS COMMUNICATIONS, INC. MAKE NO WARRANTIES WITH REGARD THERETO, AND IN PARTICULAR DO NOT WARRANT OR REPRESENT THAT THIS SPECIFICATION OR ANY PRODUCTS MADE IN CONFORMANCE WITH IT WILL WORK IN THE INTENDED MANNER OR BE COMPATIBLE WITH OTHER PRODUCTS IN NETWORK SYSTEMS. NOR DO THEY ASSUME RESPONSIBILITY FOR ANY ERRORS THAT THE SPECIFICATION MAY CONTAIN OR HAVE ANY LIABILITIES OR OBLIGATIONS FOR DAMAGES INCLUDING, BUT NOT LIMITED TO, SPECIAL, INCIDENTAL, INDIRECT, PUNITIVE, OR CONSEQUENTIAL DAMAGES WHETHER ARISING FROM OR IN CONNECTION WITH THE USE OF THIS SPECIFICATION IN ANY WAY.
2. NO REPRESENTATIONS OR WARRANTIES ARE MADE THAT ANY PRODUCT BASED IN WHOLE OR IN PART ON THE ABOVE SPECIFICATION WILL BE FREE FROM DEFECTS OR SAFE FOR USE FOR ITS INTENDED PURPOSE. ANY PERSON MAKING, USING OR SELLING SUCH PRODUCT DOES SO AT HIS OWN RISK.
3. THE USER OF THIS SPECIFICATION HEREBY EXPRESSLY ACKNOWLEDGES THAT THE SPECIFICATION IS PROVIDED AS IS, AND THAT THE DMTF MEMBERS, NEITHER INDIVIDUALLY NOR COLLECTIVELY, MAKE ANY REPRESENTATIONS, EXTEND ANY WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, ORAL OR WRITTEN, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTY OR REPRESENTATION THAT THE SPECIFICATION OR ANY PRODUCT OR TECHNOLOGY UTILIZING ANY ASPECT OF THE SPECIFICATION WILL BE FREE FROM ANY CLAIMS OF INFRINGEMENT OFINTELLECTUAL PROPERTY, INCLUDING PATENTS, COPYRIGHT AND TRADE SECRETS OF ANY THIRD PARTY, OR ASSUMES ANY OTHER RESPONSIBILITIES WHATSOEVER WITH RESPECT TO THE SPECIFICATION OR SUCH PRODUCTS. IN NO EVENT WILL THE DMTF BE LIABLE FOR ANY LOSSES, DAMAGES INCLUDING, WITHOUT LIMITATION, THOSE DAMAGES DESCRIBED IN SECTION 1 ABOVE, COSTS, JUDGMENTS, OR EXPENSES ARISING FROM THE USE OR LICENSING OF THE SPECIFICATION HEREUNDER.
4. A LICENSE IS HEREBY GRANTED TO COPY AND REPRODUCE THIS SPECIFICATION FOR INTERNAL USE ONLY. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY OTHER INTELLECTUAL PROPERTY RIGHTS IS GRANTED HEREIN.

# Preface

The Large Mailing Operations Standards work group has been meeting since September 1992 under the sponsorship of IBM to define the interfaces, objects and processes for managing the large mailing environment. In January 1994, the work group selected the Desktop Management Interface (DMI) technology as its preferred method for defining and delivering its open system solutions to the field. Detailed standards were developed using the DMI architecture base.

In September of 1994, the working group formally joined the Desktop Management Task Force as the Finisher Work Group. The goal of the Finisher Work Group is to use the infrastructure of the DMTF to make its standards more widely available and broaden their usage throughout the industry. In October of 1994, invitations were extended to all major vendors to join the Finisher Work Group in this effort.

The standard specifications in this document are produced and maintained by the DMTF Finisher Work Group. If you or your company would like to join the working group you may email your request to dmtf-info@dmtf.org.

## DMTF Finisher Work Group

| | | | Fax Number |
|---|---|---|---|
| Tony Stuart Chairman | IBM | 303-924-6081 | 303-924 5935 |
| Ralph Anderson | C.O.P.E. | 602-746-3241 | 602-647-6949 |
| Scott Bannor | Johnson&Quin | 708-647-6900 | 708-647-6949 |
| Mark Bolin | Standard Register | 610-435-8811 | 610-435-8965 |
| Linda Bonanno | Bell & Howell | 919-405-3253 | 919-544-0077 |
| Ron Blake | Wallace Lasermax | 708-734-0000 | 708-734-0070 |
| Ernie Bunch | Beste Bunch | 602-997-6452 | 602-997-7266 |
| Ken Chernabrow | Delphax | 905-238-2961 | 905-238-6265 |
| Don Danner | Johnson&Quin | 708-647-6900 | 708-647-6949 |
| John Davis | Pitney Bowes | 708-505-5909 | 708-505-5948 |
| Warren Denby | Delphax | 617-828-9917 | 617-828-9962 |
| Heinz Dettwiler | Kern | 41-31-791-09-11 | 41-31-791-09-22 |
| David Dye | Bell & Howell | 919-941-8588 | 919-941-8511 |
| Rudolf Eichler | Böwe Systec | 49-821-5702-335 | 59-821-5702-610 |
| Goran Eiler | Siemans-Nixdorf | 49-8121-72-4346 | 49-8121-72-3173 |
| Robert Ellis | Siemans-Nixdorf | 407-994-7140 | 407-998-9160 |
| Majid Esfahani | Johnson&Quin | 610-435-8811 | 610-435-8965 |
| Mike Fegeley | Bell & Howell | 919-941-8606 | 919-941-8511 |
| Michael Field | Pitney Bowes | 203-790-3054 | 203-790-3935 |
| Authur Ford | Xerox | 310-333-7470 | 310-333-7640 |
| Jon Freeman | Böwe Systec | 516-822-7575 | 516-822-1666 |
| Bill Hart | Pitney Bowes | 203-790-3158 | 203-790-3699 |
| Joel Hirschman | JHIRSCHMAN Resources | 520 -577-3222 | Call First |
| Jesper Rømer Holm | International Data Post | 45-33-25-20-00 | 45-33-25-33-09 |
| Richard Jenkins | Standard Register | 708-916-6620 | 708-916-6588 |
| Tom Kolniak | Moore | 716-773-0580 | 716-773-2470 |
| Ted Kennedy | Bell & Howell | 919-405-3200 | 919-554-0077 |
| Jim Lee | Delphax | 905-238-2961 | 905-238-6265 |
| Robert Lee | Moore Research Center | 716-744-0776 | 716-744-0770 |
| Joachim Lindow | Strålfors | 46-372-850-00 | 46-372-839-26 |
| Rober Liu | Delphax | 905-238-2961 | 905-238-6365 |
| Antonio Lustres | Kern | 41-31-791-09-11 | 41-31-791-08-22 |
| Harry Lewis | IBM | 303-924-5337 | 303-924-5840 |
| Jim Marlin | JMarlin Resources | 303-530-4202 | |
| Stan McConnell | Xerox | | |
| Paul Mitchell | Pitney Bowes | 203-790-3760 | 203-790-3802 |
| Grant Miller | RRDonnelly&Sons | 708-719-6646 | 708-719-6718 |
| Jim Reville | Moore Research Center | 716-773-0221 | 716-773-0210 |
| Gene Rozea | Roll Systems Inc. | 617-229-2266 | 617-229-0486 |
| Tom Ruehle | IBM | 303-924-7488 | 303-924-5935 |
| Bob Rushby | Delphax | 905-238-2961 | 905-238-8227 |
| Ed Schneiderhan | MJB Systems | 708-724-7466 | 708-724-3643 |
| Karl Schumacher | Pitney Bowes | 203-790-3701 | 203-790-3935 |
| Eric Schuyler | Moore Research Center | 716-773-0565 | 716-773-0210 |
| Virgil Sealy | Bell & Howell | 919-405-3234 | 919-544-0077 |
| Rufus Smith | Gunther International | 860-823-1427 | 860-886-0135 |

| | | | |
|---|---|---|---|
| Bob Spencer | C.O.P.E. | 617-828-9917 | 617-828-9962 |
| Kevin Tomberlin | IBM | 303-924-6924 | 303-924-4049 |
| Peter Turchyn | Delphax Systems | 905-238-2961 | 905-238-6265 |
| Gunther Waffler | microwi electronic | 49-831-7720-200 | 49-831-7720-070 |
| Jim Walsh | Böwe Systec | 516-822-7575 | 516-822-1666 |
| Ed Warnol | Standard Register | 909-885-6821 | 909-884-6996 |
| Dieter Wiesenbart | microwi electronic | 49-821-72020-31 | 49-821-72020-70 |
| Andreas Wüthrich | Hunkeler | 41-62-745-61-61 | 41-62-745-62-76 |
| Gary Zimmerman | Devtech Associates | 708-335-4404 x134 | 708-335-2944 |
| Joe Zuech | C.O.P.E. | 520-746-3241 | 520-741-9234 |

# Contents


**INTRODUCTION** 13
- APPLICATION MODEL 13
- SYSTEM ENVIRONMENT 15
- MAILING OPERATION MANAGER MODEL 16

**DMI OVERVIEW: A BRIEF TECHNICAL SUMMARY** 17
- DESKTOP MANAGEMENT ARCHITECTURE BASE 17
- MOTIVATION FOR SELECTING THE DMI 17
- BASIC TERMINOLOGY 17
- OVERVIEW 18
- DMI DOCUMENTATION 19
- LMO DATABASE INSTRUMENTED COMPONENT 19

**OBJECT OVERVIEW** 21

**PRODUCT CLASS OBJECTS** 21
- PRODUCT COMPONENTID GROUP 21
- PRODUCT TABLE 21
- PRODUCT HISTORY TABLE 21
- PROCESS PATH TABLE 22
- PROCESS HISTORY TABLE 22
- PRINT HISTORY TABLE 22
- INSERT HISTORY TABLE 23

**MAIL JOB CLASS OBJECTS** 23
- MAIL JOB COMPONENTID GROUP 23
- PRINT JOB TABLE 23
- INSERT JOB TABLE 24
- BILL OF MATERIALS TABLE 24
- MAILPIECE XREFERENCE TABLE 25
- MAILPIECE HISTORY TABLE 25

**PRINTER CLASS OBJECTS** 26
- PRINTER COMPONENTID GROUP 26
- PRINTER GENERAL GROUP 26
- PRINTER LOCALIZATION TABLE 26
- INPUT DEVICE TABLE 27
- INPUT DEVICE -REQUIRED 27
- MARKER DEVICE TABLE 27
- MARKER SUPPLY TABLE 28
- MARKER COLORANT TABLE 28
- OUTPUT DEVICE TABLE 29
- OUTPUT DEVICE -REQUIRED 29
- FINISHER DEVICE TABLE 29
- ALERT TABLE 30

**INSERTER CLASS OBJECTS** 30

INSERTER COMPONENTID GROUP 30
INSERTER GENERAL GROUP 31
INSERTER LOCALIZATION TABLE 31
FINISHER DEVICE TABLE 31
FINISHER INPUT TABLE 31
FINISHING FEATURES TABLE 32
FINISHER OUTPUT TABLE 33
FINISHER SUPPLY TABLE 34
FINISHER COLORANT TABLE 34
INSERT FEEDER DEVICE TABLE 35
INSERT FEEDER DEVICE -REQUIRED 35
ENVELOPE FEEDER DEVICE TABLE 35
INSERTER OUTPUT DEVICE TABLE 36
INSERTER OUTPUT DEVICE -REQUIRED 36
INSERTER OUTPUT SUPPLY TABLE 37
OUTPUT DEVICE COLORANT TABLE 37
ALERT TABLE 37
CURRENT JOB GROUP 38

**FINISHER CLASS OBJECTS -STANDALONE DEVICES 38**

**PROCESS CLASS OBJECTS 39**

**RECONCILIATION AGENT CLASS OBJECTS 39**
RECONCILIATION COMPONENTID GROUP 39
RECONCILE PRODUCT GROUP 39
RECONCILE PRINT JOB GROUP 40
RECONCILE JOB CONTROL GROUP 40
RECONCILE REPRINT CONTROL GROUP 40
RECONCILE PRODUCT CONTROL GROUP 41

**LMO UTILITY CLASS OBJECTS 41**
LMO UTILITY COMPONENTID GROUP 41
INPORT EXPORT GROUP 41
QUERY ROWS GROUP 42
INDICATE CONTROL GROUP 42
WATCH CONTROL TABLE 42
WATCH TABLE 43
HISTORY CONTROL TABLE 43

## OBJECT FORMAT 45
OBJECT SYNTAX 45
DATA TYPES 45
GLOBAL ENUMERATIONS 46

## PRODUCT CLASS OBJECTS 49
PRODUCT COMPONENTID GROUP- MANDATORY 49
LMO PRODUCT TABLE 50
PRODUCT HISTORY TABLE 51
PROCESS PATH TABLE 52
PROCESS HISTORY TABLE 53
PRINT HISTORY TABLE 53

INSERT HISTORY TABLE 54

**MAIL JOB CLASS OBJECTS 55**
MAIL JOB COMPONENTID GROUP- MANDATORY 55
PRINT JOB TABLE 56
INSERT JOB TABLE 57
BILL OF MATERIALS TABLE 59
MAILPIECE XREFERENCE TABLE 60
MAILPIECE HISTORY TABLE 61

**PRINTER CLASS OBJECTS 63**
PRINTER MODEL 63
PRINTER COMPONENTID GROUP- MANDATORY 63
PRINTER GENERAL GROUP -MANDATORY 64
PRINTER LOCALIZATION TABLE 66
PRINTER INPUT TABLE -MANDATORY 66
PRINTER INPUT TYPE ATTRIBUTES EXTENDED -OPTIONAL 69
PRINTER INPUT MEDIA ATTRIBUTES EXTENDED -OPTIONAL 69
MARKER DEVICE TABLE 70
MARKER SUPPLY TABLE 73
MARKER COLORANT TABLE 74
PRINTER OUTPUT TABLE -MANDATORY 75
PRINTER OUTPUT TYPE ATTRIBUTES EXTENDED -OPTIONAL 76
PRINTER OUTPUT FEATURE ATTRIBUTES EXTENDED -OPTIONAL 78
FINISHER DEVICE TABLE -OPTIONAL 79
PRINTER ALERT TABLE 79

**INSERTER CLASS OBJECTS 81**
INSERTER MODEL 81
INSERTER COMPONENTID GROUP -MANDATORY 82
INSERTER GENERAL GROUP: 82
INSERTER LOCALIZATION TABLE 84
FINISHER DEVICE TABLE -MANDATORY 84
FINISHER INPUT TABLE -MANDATORY 85
FINISHER INPUT TYPE ATTRIBUTES EXTENDED -OPTIONAL 87
FINISHER INPUT MEDIA ATTRIBUTES EXTENDED -OPTIONAL 87
FINISHING FEATURES TABLE 88
FINISHER OUTPUT DEVICE TABLE -MANDATORY 92
FINISHER OUTPUT TYPE ATTRIBUTE EXTENDED -OPTIONAL 94
FINISHER OUTPUT DIMENSION ATTRIBUTES EXTENDED -OPTIONAL 94
FINISHER OUTPUT FEATURES ATTRIBUTES EXTENDED -OPTIONAL 95
FINISHER SUPPLY TABLE 96
FINISHER COLORANT TABLE 98
INSERT FEEDER TABLE -MANDATORY 99
INSERT FEEDER TYPE ATTRIBUTE EXTENDED -OPTIONAL 101
INSERT MEDIA ATTRIBUTE EXTENDED -OPTIONAL 101
ENVELOPE FEEDER TABLE -MANDATORY 102
ENVELOPE FEEDER TYPE ATTRIBUTE EXTENDED -OPTIONAL 104
ENVELOPE MEDIA ATTRIBUTE EXTENDED -OPTIONAL 105
INSERTER OUTPUT DEVICE TABLE -OPTIONAL 105
INSERTER OUTPUT TYPE ATTRIBUTE EXTENDED -OPTIONAL 106

INSERTER OUTPUT DEVICE SUPPLY TABLE 107
OUTPUT DEVICE COLORANT TABLE 108
INSERTER ALERT TABLE 109
CURRENT JOB GROUP 110

**RECONCILIATION AGENT CLASS OBJECTS 111**
RECONCILIATION COMPONENTID GROUP- MANDATORY 111
RECONCILE PRODUCT GROUP 112
RECONCILE PRINT JOB GROUP 112
RECONCILE JOB CONTROL GROUP 113
RECONCILE REPRINT CONTROL GROUP 114
RECONCILE PRODUCT CONTROL GROUP 114

**UTILITY AGENT CLASS OBJECTS 117**
UTILITY COMPONENTID GROUP- MANDATORY 117
IMPORT EXPORT GROUP 118
QUERY ROWS GROUP 119
INDICATE CONTROL GROUP 119
WATCH CONTROL GROUP 120
WATCH TABLE 121
HISTORY CONTROL TABLE 122

**APPENDIX A. ALERTS 125**
OVERVIEW 125
INSTRUMENTATION 125
ALERT TABLES 125
ALERT TABLE MANAGEMENT 126
ALERT TABLE FORMAT 126
COMPONENT ALERT TABLE 127

**APPENDIX B. DMI CLIENT NETWORK INTERFACE 131**
DMI NETWORK INTERFACE EXTENSIONS 131
DMI CLIENT-SERVER COMMUNICATION SEQUENCES 136

**APPENDIX C. INSTRUMENTED COMPONENT AGENTS 142**

**RECONCILE AGENT COMPONENT 143**
RECONCILE AGENT CONTROL TABLES 143
RECONCILE AGENT COMMUNICATION SEQUENCES 144
SYSTEM COMMANDS 145
RECONCILE EXAMPLES 146

**WATCH AGENT COMPONENT 149**
WATCH CONTROL TABLE 149
WATCH CONTROL TABLE USAGE 150
WATCH TABLE 151
WATCH TABLE USAGE 152
WATCH EXAMPLE 152

**HISTORY AGENT COMPONENT 154**

HISTORY CONTROL TABLE 154
HISTORY AGENT USAGE 155
HISTORY AGENT EXAMPLE 156
COMMUNICATION SEQUENCE 157

**UTILITY AGENT GROUPS 159**
IMPORT EXPORT GROUP 159
IMPORT EXPORT EXAMPLES 161
QUERY ROWS GROUP 162

**APPENDIX D. DMISET TAG LOGICAL ELEMENTS 163**
OVERVIEW 163
RULES FOR USING DMISET TLES 163
IMPLEMENTATION NOTES 164

**GLOSSARY 165**

# INTRODUCTION

High volume customer mailing has become integral to the operations of many companies. As this trend continues more and more demand is being made of vendors to provide products that allow process control techniques to be applied to managing the overall mailing operations from printing and finishing of documents to their enclosure in envelopes and delivery to customers.

The benefits to customers from such products can readily be seen in terms of increased efficiencies, optimum machine usage, savings in time and labor, and improved service to their clients. The difficulty in achieving these benefits comes from the complexity of the problem and the number of different disciplines and vendor products that go into a solution. This is compounded by the existence of many proprietary solutions and lack of industry standard interfaces to enable integration.

In 1992 an Advisory Council of customers and vendors was formed to identify the key requirements of a Large Mailing Operations system and to explore the possibility of defining an open systems architecture standard for meeting them. This work resulted in demonstrating the capabilities of an integrated LMO system at XPLOR and a commitment by the vendors to carry out a cooperative effort to establish industry standards for managing the LMO environment.

The companies participating in the cooperative development of the LMO standards defined in this document include Beste Bunche, Bell & Howell, BOWE, C.O.P.E., Delphax, DevTech Associates, Gunther International, Hunkeler, IBM, International Data Post, JQ Technologies, Kern, LASERMAX (Wallace), Moore Business Forms, MJB Systems, Pitney Bowes, Roll Systems, RRDonnelly & Sons, Siemens-Nixdorf, Standard Register, and Xerox.

An object oriented approach has been selected for the development of the standard based on the DeskTop Management Interface (DMI) specification produced by the DeskTop Management TaskForce DMTF.

## APPLICATION MODEL

An abstract model of a Large Mailing Operations system is shown in figure 1.

Work begins in the data processing portion of the system with applications that generate print data to be inserted into envelopes for mailing. In many instances, these are existing legacy applications that produce billing statements for customers. In today's modern environment, there are many tools available to assist in generating customized print output.

As shown in the figure, the print output from customer applications is further processed by Value Add (VA) applications and Advanced Function Presentation (AFP) functions that condition the data for mailing and build control objects for down stream operations.

Examples of Value Add functions are programs which provide address verification, pre-sorting of statements by their postal characteristics, building insertion instructions based on information contained in demographic and marketing databases and segmenting print output into manageable units of work.

Examples of Advanced Function Presentation functions, are services that convert line data into page data, build document index objects for locating individual groups of pages (statements) in print files for re-print and viewing, and archiving services for storing and retrieving the manageable printing units of work for mailing.

Most VA programs and AFP services are application independent in that they do not require changes in the customers print producing application in order to perform. their function

Once VA and AFP processes complete, the print files are scheduled for printing by the Print Service component and the output delivered to the inserter. Control information for the insertion process is separately sent to the Finishing Server when the finishing hardware is not inline in the print path.

Barcode or Optical Recognition marks on the paper are used by the Finishing Server to correlate the finishing instructions for a print job with individual mailpieces to be assembled and trayed for postal delivery.

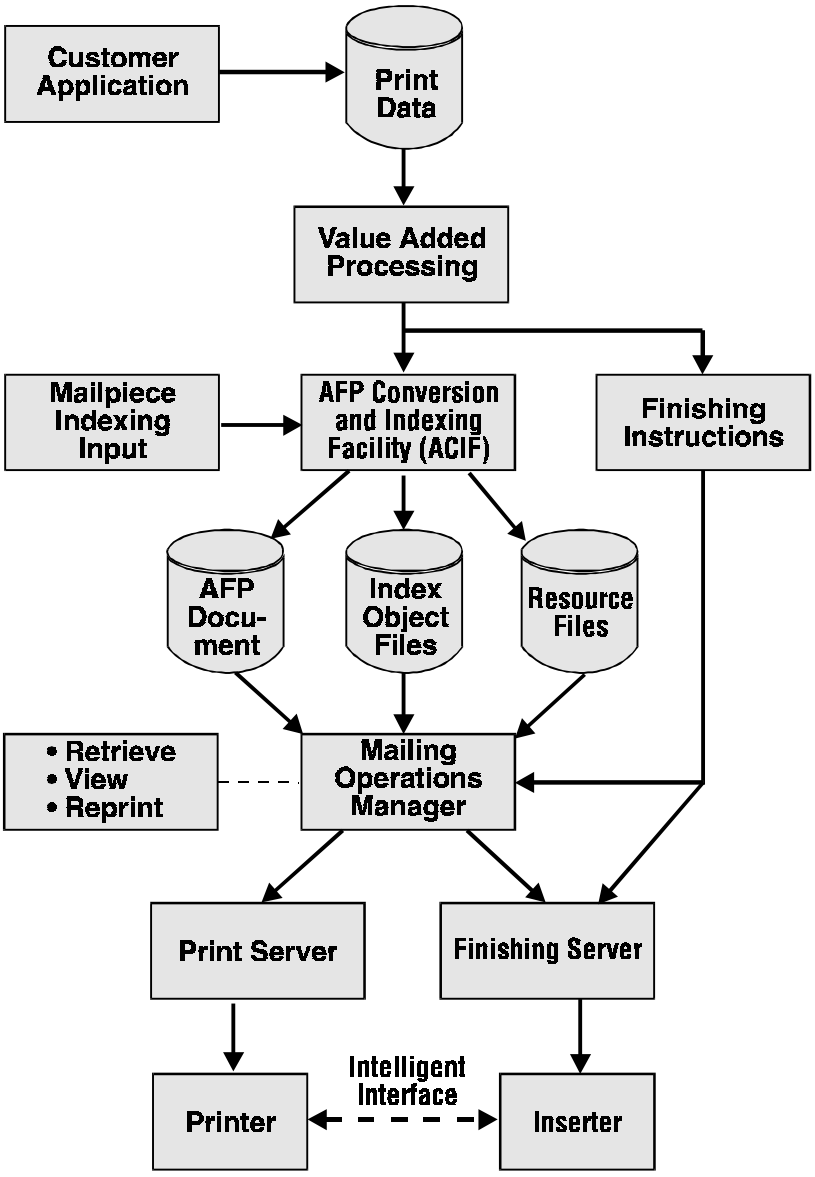



Figure 1. Abstract Model of an LMO System. A Large Mailing Operation System is a coordinated set of hardware and software components and interfaces that work together to automate the output processes associated with high volume printing, finishing, and delivery of individual mailpieces to recipients, most often via postal services. The generic interface model for an LMO System is shown in this diagram.

## SYSTEM ENVIRONMENT

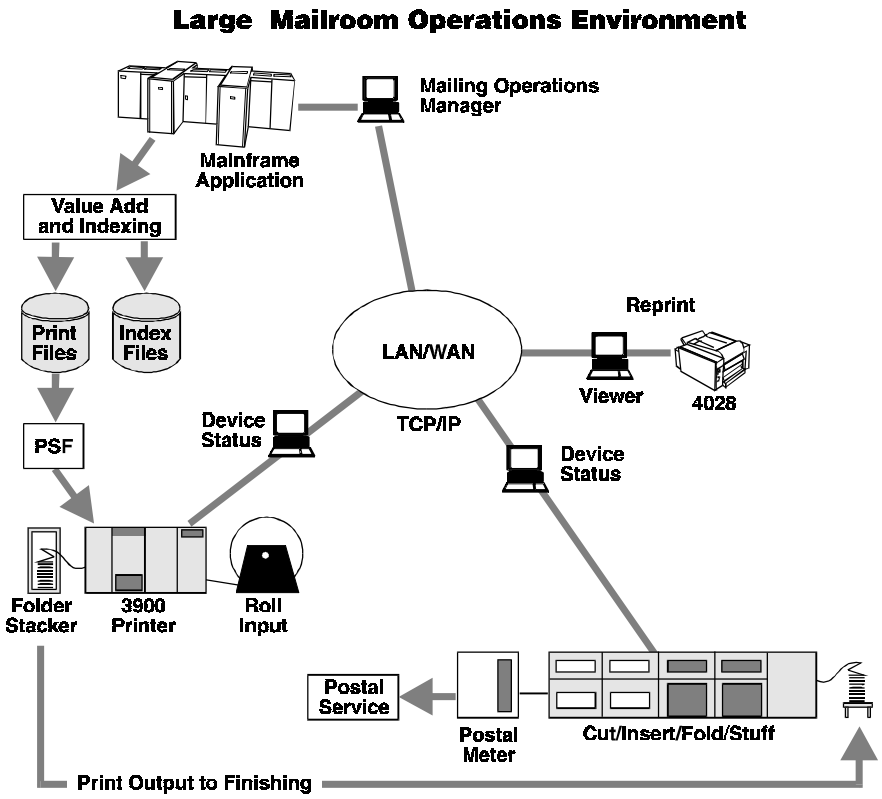



Figure 2. LMO System Environment. A generic system configuration of an LMO System is depicted in this diagram. In today's environment customer applications, value add programs, indexing, and batch printing functions are typically done on large mainframe processors while the inserting operations and the overall monitoring, tracking and control functions are performed on workstation devices connected via networks. Reprint of mutilated mailpieces is done on demand on smaller remote print stations attached to the network.

## MAILING OPERATION MANAGER MODEL

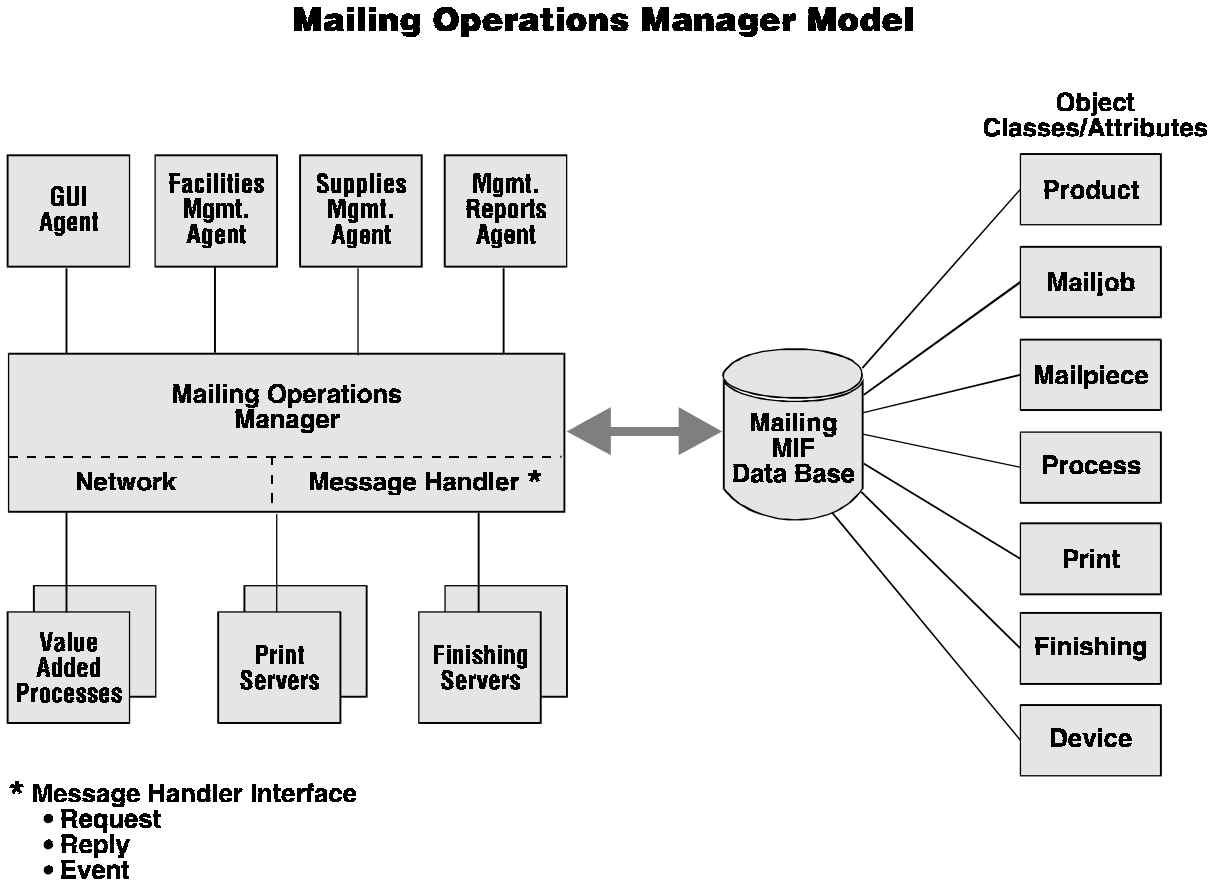



Figure 3. Mailing Operations Manager Model. A generic model of a Mailing Operations Manager is shown in this figure. It provides a message handler interface that is used by the various hardware and software processes to define themselves to the system and report changes in status. It maintains information about these processes in a common data basein a standard Management Information Format. Lastly, it provides Management Interfaces for use by application programs, program agents and Graphical User Interface to query, set and get attributes and access meta information about product, mail jobs, and mailpieces that are active in the system.

# DMI OVERVIEW: A BRIEF TECHNICAL SUMMARY

## DESKTOP MANAGEMENT ARCHITECTURE BASE

The demand for standards is accompanied by the need for selecting an architecture base that is widely accepted, easy to implement, and extendible to future requirements.

Customers and vendors alike are reluctant to invest in automating the LMO environment unless they feel comfortable that their solutions and products are built on interfaces that are durable over time and can take advantage of the emerging technologies that provide easy-to-understand graphical interfaces common at the desktop.

In looking at the open system standards available today for modeling the functions required in the LMO environment, the standard base that most closely meets these requirements is the DeskTop Management Interface, or DMI.

The DMI standard is managed by a group of companies calling themselves the DeskTop Management Task Force (DMTF). The charter members of this group include Digital Corp. (DEC), Hewlett-Packard, IBM Corp., Microsoft Corp., Novell Inc., SunConnect(a Sun Microsystems Inc. business), and SynOptics Communications, Inc. To date more than 150 companies have joined the DMTF as participating members representing virtually all aspects of the computing industry.

## MOTIVATION FOR SELECTING THE DMI

The objective of the LMO standards is to define interfaces, objects and management protocols specific to large mailing operations that allow the various processes in the system to interoperate with one another and allow installation managers to determine the status of the overall operation, as well as each individual component in the system.

The DMI provides an architecture base for achieving these objectives by specifying an interface that is:

- independent of a specific computer or operating system
- independent of a specific management protocol
- easy for vendors and customers to adopt and install
- usable locally—no network required
- mappable to existing management protocols (e.g., CMIP, SNMP)

Implementations of the DMI are available today or committed in OS2, Workplace OS, and DOS. Other platforms are soon to follow. By building LMO objects and management protocols on the DMI, LMO standards may be established now in a uniform manner across all of these environments with actual implementations beginning this year, not years in the future.

## BASIC TERMINOLOGY

In this document a system refers to an LMO System. **Components** are physical or logical entities on a system, such as hardware, software or firmware. Components may come with the system or may be added to it. The code that carries out management actions for a particular component is known as the **component instrumentation**.

A **management application** is a program that initiates management requests. A management application uses the DMI to perform management operations. A management application can be a program such as an application with a graphical user interface or an application program agent, or it may be a network management protocol agent that translates requests from a standard network management protocol (such as SNMP or CMIP) to the DMI and back again.

The **service layer** coordinates access to component instrumentation and component-provided data.

A functional diagram of DMI system is shown in figure 4. The natural relationship of this model to the model of the LMO Mailing Operations Manager is readily apparent.

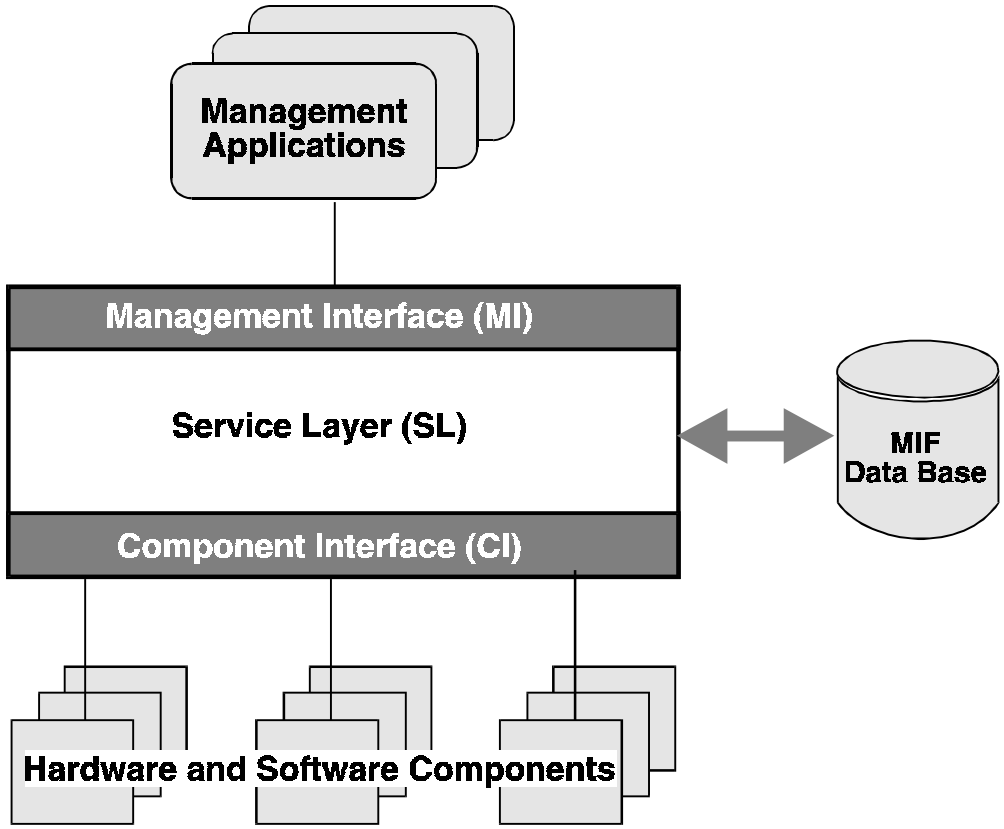



Figure 4. Desktop Management Interface. A generic model of the DMI is shown in this figure.

## OVERVIEW

The DMI has four distinct elements:

1. a format for describing information
2. a format for data transfer
3. a mechanism for data transfer
4. a set of services for facilitating communication

Component descriptions are defined in a language called the **Management Information Format**, or **MIF**. Each component has a **MIF file** to describe its manageable characteristics. When a component is initially installed into the system, the MIF is added to the **MIF database** for use by the service layer.

The **Component Interface** or **CI** is used by component providers to describe access to management information and to enable a component to be managed. The CI and the MIF shield vendors from the complexity of encoding styles and management registration information. They do not need to learn the details of the popular and emerging management protocols.

The **Management Interface**, or **MI**, is used by applications that wish to manage components. The MI shields management application vendors from the different mechanisms used to obtain management information from elements within the system.

The CI and MI are data interfaces, as opposed to procedural interfaces. Data blocks form the format for data transfer, not parameters to a function call. The behavioral mechanics of the CI and MI make up the data transfer mechanism.

The **Service Layer,** or **SL**, is an active, resident piece of code running on a computer system that mediates between the MI and CI.

In the LMO system environment, the graphical user interface is a management application provided with the system. Other management applications such as tracking and re-print agents are provided by customers and vendors alike to meet the requirements of the system. Some vendors may choose to provide both MI and CI function to achieve maximum competitive advantage.

## DMI DOCUMENTATION

Documentation for the DMI is provided in a publication titled "**Desktop Management Interface Specification**. " This document defines a set of rules for describing and obtaining management information for the hardware and software components of the system. It is for readers who will build the specified interfaces and services. It is not a tutorial for reader who define management information or design management applications.

The "Desktop Management Interface Specification" document is a product of the Desktop Management Task Force.

## LMO DATABASE INSTRUMENTED COMPONENT

The LMO database instrumented component provides general purpose support for dynamic table management. With this instrumented component, rows may be added, deleted, or queried using wildcard keys. A management application can: 1) create a new row by invoking a DmiSetAttribute request with keys that do not refer to a row that exists in the table; 2) update an existing row by invoking a DmiSetAttribute request with keys that refer to a row that already exists in the table; 3) delete an existing row by issuing a DmiSetAttribute request on the Delete Row attribute with a value of True; 4) query existing rows using wildcard keys by issuing a DmiGetAttribute request to the Query attribute - the request returns a comma/newline delimited list of keys that match the wildcard keys supplied in the request.

Note that devices may provide alternative table management semantics for dynamic tables (such as Alert Tables). For these tables, the Delete Row and Query attributes are optional.

# OBJECT OVERVIEW

# PRODUCT CLASS OBJECTS

## PRODUCT COMPONENTID GROUP

This group makes general information provided by the installer of this component available to management applications. It is a required group in all Management Information Format (MIF) component definitions.

Manufacturer

Product

Version

Serial Number

Installation Date

Verify

## PRODUCT TABLE

A Large Mailing Operation may have many application products active at the same time. The Product Table contains an entry for each product. Each entry is an ordered set of attributes describing the general characteristics of the product and it's projected and current size.

Product Name

Product Instance Qualifier

Description

Owner

Date Created

Input Origin

Segment Size Unit

Average Segment Size

Estimated Number of Segments

Current Number of Segments

Final Number of Segments

Accounting Data Format

Accounting Data

Current Status

Delete Row

Query

## PRODUCT HISTORY TABLE

This table provides a history of products as they proceed through the system.

Product Name

Product Instance Qualifier

Time Stamp

Current Status

Delete Row

Query

## PROCESS PATH TABLE

The sequence of steps that a product goes through to produce print data for mailing may be automated by some installations. The entries in this table are indexed by Product Name, Product Instance Qualifier and Process Name. Only those products that are to be managed in this fashion have entries in this table. Each entry has the following attributes.

Product Name

Product Instance Qualifier

Process Name

Process Order

Process Accounting Data

Current Status

Delete Row

Query

## PROCESS HISTORY TABLE

The history of status changes for each process step can be recorded in this table.

Product Name

Product Instance Qualifier

Process Name

Time Stamp

Current Status

Delete Row

Query

## PRINT HISTORY TABLE

The print output of a product may be segmented into manageable units of work. Each unit of work is a unique print job. The history of status changes for each print job can be recorded in this table.

Product Name

Product Instance Qualifier

Print Job Name

Time Stamp

JES Assigned ID

Current Device

Current Status

Delete Row

Query

### INSERT HISTORY TABLE

The print output from one or more products may be input into an insert job. The manageable unit of work for the inserter is the insert job. The history of status changes for an insert job can be recorded in this table.

Product Name

Product Instance Qualifier

Insert Job Name

Time Stamp

Current Device

Current Status

Delete Row

Query

## MAIL JOB CLASS OBJECTS

### MAIL JOB COMPONENTID GROUP

This group makes general information provided by the installer of this component available to management applications. It is a required group in all Management Information Format (MIF) component definitions.

Manufacturer

Product

Version

Serial Number

Installation Date

Verify

### PRINT JOB TABLE

The Print Job Table contains an entry for each active print job in the system. Product names, print job names, and print file names are the keys used by management applications to access entries in the table. Each entry contains the following group of attributes.

Product Name

Product Instance Qualifier

Print Job Name

Description

Rquired Media Name

Associated FormDef Name

Associated PageDef Name

Associated Job Ticket Name

Associated Document Index Name

Print File Name

Print File Format (e.g., Line Data, MODCA-P, Mixed)

JES Assigned ID

Date Created

Mail Job Size

Print Job Size Unit

Estimated Print Job Size

Actual Print Job Size

Current Device

Current Status

First Mailpiece Identifier

Last Mailpiece Identifier

Delete Row

Query

## INSERT JOB TABLE

The Insert Job table contains an entry for each active insert job in the LMO System. Insert jobs control finishing and inserting of physical materials into envelopes or containers for mailing and delivery. There may be one or more insert job definitions associated with a given product. For example, machine setups, bill of materials, and job ticket instructions can vary within a product for different postal classes of service or special account mailings e. g. platinum card holder billing. Each entry in the table is a unique insert job specification.

Product Name

Product Instance Qualifier

Insert Job Name

Description

Associated Job Ticket Name

Date Created

Insert Job Size Unit

Estimated Insert Job Size

Actual Insert Job Size

Current Device

Current Status

Delete Row

Query

## BILL OF MATERIALS TABLE

This table lists the bill of materials associated with insert jobs. Each entry in the table identifies a specific type of insert such as print output from a print job, call inserts setup by operators, or named media stock supplied from inventory. Multiple bill of material entries of the same or different types may be associated with a given Insert

Job. Bill of materials for multiple insert jobs may be specified in the table. Each entry has the following attributes.

Product Name

Product Instance Qualifier

Insert Job Name

Material Name

Type (printjob|call insert|named media)

Description

Number Required

Number Used

Delete Row

Query

## MAILPIECE XREFERENCE TABLE

This table provides a cross reference index to used by tracking, archive, and reprint agents to correlate mailpiece identifers to print jobs and print files.

Product Name

Product Instance Qualifier

Mailpiece Identifier

Process Assigned Identifier

Print Job Name

Insert JobName

Mailpiece Size Unit

Mailpiece Size

Mailpiece Postage Unit

Mailpiece Postage

Current Status

Delete Row

Query

## MAILPIECE HISTORY TABLE

This table provides a history of Mailpieces as they proceed through the system.

Product Name

Product Instance Qualifier

Mailpiece Identifier

Time Stamp

Current Status

Delete Row

Query

# PRINTER CLASS OBJECTS

## PRINTER COMPONENTID GROUP

This group makes general information provided by the installer of this component available to management applications. It is a required group in all Management Information Format (MIF) component definitions.

Manufacturer

Product

Version

Serial Number

Installation Date

Verify

## PRINTER GENERAL GROUP

The Printer General Group specifies attributes that apply to the printer component and its status.

Config Changes

Current Localization

Reset

Input Default Index

Output Default Index

Marker Default Index

Media Path Default Index

Console Localization

Console Number of Display Lines

Console Disable

Current Operator

Service Person

Status

## PRINTER LOCALIZATION TABLE

This table identifies the natural language, country and character set in which character sets are expressed. One or more localizations may be supported by a printer. Each entry has the following group of attributes.

Language (ISO639 language code)

Country (ISO3166 country code)

Localization Character Set

## INPUT DEVICE TABLE

A printer may have multiple input-devices associated with it. For example, a printer can have two input source trays. The input device objects associated with a printer are grouped together in an Input Device Table. Each entry in the table is an input device object definition with an ordered set of attributes. Only those attributes applicable for the input device need be specified.

## INPUT DEVICE -REQUIRED

Index

Type (bin, tray, roll. . )

Dim Unit

Declared Media Dim Feed Dir

Declared Media Dim XFeed Dir

Chosen Media Dim Feed Dir

Chosen Media Dim XFeed Dir

Capacity Units

Max Capacity

Current Level

Status

Media Name

### Input Type Extended -Optional

Name

Vendor Name

Model

Version

Serial Number

Description

Security

### Input Media Extended -Optional

Media Weight

Media Type

Media Color

Media Form Parts

## MARKER DEVICE TABLE

A marker is the mechanism that produces marks on the print media. A printer can contain one or more marking mechanisms. Some examples are: a printer with separate markers for normal or magnetic ink, or an image setter that can output to both a proofing device and final film. Each marker is represented by an entry in the Marker Table. Each device can have its own characteristics associated with it such as marking technology and resolution.

Index

Mark Tech

Counter Unit

Power On Count (accumulative count since power on)

Life Count (accumulative count over life of marker)

Process Colorants

Spot Colorants

Addressability Unit

Addressability Feed Dir

Addressability XFeedDir

North Margin

South Margin

West Margin

East Margin

Status

## MARKER SUPPLY TABLE

A marker can have one or more supplies associated with it. For example, a marker may use both toner and fuser oil supplies. A marker may also have more then one source for a given type of supply e.g., multiple supply sources of toner. Each unique source of supply for a marker device is an entry in the marker supply table. Each supply entry can have its own characteristics associated with it such as colorant and current supply level.

Index

Marker Index

Colorant Index

Class (container or receptacle)

Type

Description

Supply Unit

Max Capacity

Current Level

## MARKER COLORANT TABLE

A marker can have one or more colorants associated with it. Each colorant supported by the printer is an entry in this table.

Index

Marker Index

Role (process or spot)

Value

Tonality

## OUTPUT DEVICE TABLE

A printer may have multiple output devices associated with it. For example, a printer can have multiple output bins. The output device objects associated with a printer are grouped together in an Output Device Table. Each entry in the table is an output device object definition with an ordered set of attributes. Only those attributes applicable for the output device need be specified.

## OUTPUT DEVICE -REQUIRED

Index

Type (bin, tray, continuous roll, mailbox,. . )

Capacity Unit

Max Capacity

Remaining Capacity

Status

## Output Type Extended -Optional

Name

Vendor Name

Model

Version

Serial Number

Description

Security

## Output Dimensions Extended -Optional

Dim Unit

Max Dim Feed Dir

Max Dim XFeed Dir

Min Dim Feed Dir

Min Dim XFeed Dir

## Output Device Features -Optional

Stacking Order

Page Delivery Orientation

Bursting

Decollating

Page Collated

Offset Stacking

## FINISHER DEVICE TABLE

A printer may have multiple finisher devices associated with it in the production path of the output medeia. For example, one finisher may trim and cut the media to size and another device may stitch the resulting pages. Each

entry in the table is an finisher device definition with an ordered set of attributes. Only those attributes applicable for the finisher device need be specified.

Index

Name

Vendor Name

Model

Serial Number

Description

Status

### ALERT TABLE

The alert table lists all the critical and non-critical alerts currently active in the printer. A critical alert is one that stops the printer from printing immediately and can not continue until the critical alert condition is eliminated. Non-critical alerts are those items that do not stop printing but may do so at some future time. The table contains information on severity, component, detailed location within the component, alert code and description for each critical alert that is currently active in the printer.

Index

Severity Level

Training Level

Group

Group Index

Location

Alert Code

Description

Time Stamp

Delete Row

Query

## INSERTER CLASS OBJECTS

### INSERTER COMPONENTID GROUP

Manufacturer

Product

Version

Serial Number

Installation Date

Verify

## INSERTER GENERAL GROUP

Config Changes

Current Localization

Current Operator

Service Person

Status

## INSERTER LOCALIZATION TABLE

Language (ISO639 language code)

Country (ISO3166 country code)

Localization Character Set

## FINISHER DEVICE TABLE

An inserter component may have multiple document finishing devices associated with it. For example, one device may cut and sequence sheets into pages and another may fold the pages for insertion into envelopes. Each entry in the table is a document Finishing device definition with an ordered set of attributes. Only those attributes applicable for the feeder device need be specified.

Index

Name

Vendor Name

Model

Serial Number

Description

Status

## FINISHER INPUT TABLE

Each Document Finishing Device may have one or more inputs associated with it. Each entry in the table defines an input for a specific Document Finisher. Thus the table contains input definitions for one or more Document Finishers each of which may have one or more input entries.

### Finisher Input Device-Required

Index

Finisher Index

Type

Feed Control Mode

Dim Unit

Media Dim Feed Dir

Media Dim XFeed Dir

Capacity Unit

Max Capacity

Current Level

Status

Media Name

### Input Type Extended -Optional

Name

Vendor Name

Model

Version

Serial Number

Description

Security

### Input Media Extended -Optional

Media Weight

Media Thickness

Media Type

Media Color

Media Forms Part

## FINISHING FEATURES TABLE

A Document Finisher device may apply one or more finishing functions to each of the input media sources associated with it. There is an entry in the Finishing Features Table for each input source associated with each Document Finisher Device. Each entry, in turn, specifies a list of finishing functions and identifies whether or not they are present, and if so, their current status and characteristics.

Finisher Index

Input Source Index

Stitching

Stitching Type

Max Pages Stitched

Folding

Folding Type

Max Pages Folded

Binding

Binding Type

Max Pages Bound

Binding Color

Trimming

Trimming Type

Max Pages Trimmed

Die Cutting

Die Cutting Type
Max Pages Die Cut
Punching
Punch Type
Max Pages Punched
Perforating
Perforation Type
Slitting
Slitting Type
Separation Cut
Separation Cut Type
Imprinting
Imprinting Type
Wrapping
Wrapping Type
Max Pages Wrapped
Banding
Banding Type
Max Pages Banded

## FINISHER OUTPUT TABLE

A finisher may have multiple output devices associated with it. For example, a finisher may have two output trays. The output devices associated with a finisher are identified in this table. Each entry in the table is an output definition with an ordered set of attributes describing the output characteristics and indicating the current status or compacity. Only those attributes applicable to the output device must be specified.

Index
Finisher Index
Type
Capacity Unit
Max Capacity
Remaining Capacity
Status

### Output Type Extended - Optional

Name
Vendor Name
Model
Version
Serial Number
Description
Security

### Output Dimensions Extended -Optional

Dim Unit

Max Dim Feed Dir

Max Dim XFeed Dir

Min Dim Feed Dir

Min Dim XFeed Dir

### Output Features -Optional

Stacking Order

Page Delivery Orientation

Bursting

Decollating

Page Collated

Offset Stacking

## FINISHER SUPPLY TABLE

A finisher can have one or more supplies associated with it. For example, a finisher may use both binding tape and stitching wire supplies. A finisher may also have more then one source for a given type of supply e.g., multiple supply sources of ink for imprinters. Each unique source of supply for a finisher device is an entry in the finishing supply table. Each supply entry can have its own characteristics associated with it such as colorant and current supply level.

Index

Finisher Index

Colorant Index

Class (container or receptacle)

Type

Description

Supply Unit

Max Capacity

Current Level

## FINISHER COLORANT TABLE

A finisher can have one or more colorants associated with it. Each colorant supported by a finisher is an entry in this table.

Index

Finisher Index

Role (process or spot)

Value

Tonality

## INSERT FEEDER DEVICE TABLE

An inserter component may have multiple insert feeder devices associated with it. For example, a given mailpiece mail can call for selective insertion of one or more forms into the same envelope. Each entry in the table is a insert feeder device object definition with an ordered set of attributes. Only those attributes applicable for the insert feeder device need be specified.

## INSERT FEEDER DEVICE -REQUIRED

Index

Type

Feed Control Mode

Dim Unit

Media Dim Feed Dir

Media Dim XFeed Dir

Capacity Unit

Max Capacity

Current Level

Status

Media Name

Action On Misses

Action On Doubles

### Device Type Extended - Optional

Name

Vendor Name

Model

Version

Serial Number

Description

Security

### Input Media Extended -Optional

Media Weight

Media Thickness

Media Type

Media Color

Media Forms Part

## ENVELOPE FEEDER DEVICE TABLE

An inserter component may have multiple envelope feeder devices associated with it. For example, different mailpiece characteristics may require selective insertion into envelopes of an appropriate size. Each entry in the table is an envelope feeder device object definition with an ordered set of attributes. Only those attributes applicable for the insert feeder device need be specified.

### Envelope Feeder Device -Required

Index

Type

Feed Control Mode

Dim Unit

Media Dim Feed Dir

Media Dim XFeed Dir

Capacity Unit

Max Capacity

Current Level

Status

Media Name

Action On Misses

Action On Doubles

### Device Type Extended -Optional

Name

Vendor Name

Model

Version

Serial Number

Description

Security

### Input Media Extended -Optional

Media Weight

Media Thickness

Media Type

Media Color

## INSERTER OUTPUT DEVICE TABLE

An inserter component may have multiple output devices associated with it. For example, postal meters, conveyors, traying devices and bar code sprayers. Each entry in the table is an output device object definition with an ordered set of attributes. Only those attributes applicable for the output device need be specified.

## INSERTER OUTPUT DEVICE -REQUIRED

Index

Type

Capacity Unit

Max Capacity

Current Level

Status

### Device Type Extended -Optional

Name

Vendor Name

Model

Version

Serial Number

Description

Security

## INSERTER OUTPUT SUPPLY TABLE

Inserter Output devices can have one or more supplies associated with them. For example, ink for a bar code sprayer and postage credit for postage frankers. A device may also have more then one source for a given type of supply e.g., multiple supply sources of ink for tray separation markers. Each unique source of supply for a output device is an entry in the Inserter Output Supply Table. Each supply entry can have its own characteristics associated with it such as colorant and current supply level.

Index

Device Index

Colorant Index

Class (container or receptacle)

Type

Description

Supply Unit

Max Capacity

Current Level

## OUTPUT DEVICE COLORANT TABLE

Output devices can have one or more colorants associated with it. Each colorant supported by the device is an entry in this table.

Index

Device Index

Role (process or spot)

Value

Tonality

## ALERT TABLE

The alert table lists all the critical and non-critical alerts currently active in the Inserter. A critical alert is one that stops the inserter immediately such that it can not continue until the critical alert condition is eliminated. Non-critical alerts are those items that do not stop inserting but may do so at some future time. The table

contains information on severity, component, detailed location within the component, alert code and description for each critical alert that is currently active within the component.

Index

Severity Level

Training Level

Group

Group Index

Location

Alert Code

Description

Time Stamp

Delete Row

Query

### CURRENT JOB GROUP

This group identifies the current job running on the inserter. It contains information that is also found in the Insert Job Table. The presence of duplicate information can lead to inconsistencies. The systems integrator in charge of the LMO installation is responsible for determining whether inconsistencies can occur. If so, the systems integrator can make either this group (or the duplicated attributes in the Insert Job table) optional.

Product Name

Instance Qualifier

Insert Job Name

Insert Job Size Unit

Insert Job Size

## FINISHER CLASS OBJECTS -STANDALONE DEVICES

Finisher devices may be present in the print path of a printer or as document feeder devices in the production assembly line of an inserter or they may exist as standalone components in a system. In each case, their description is modelled using common groups defined by the DMTF Finisher Work Group.

The following common groups are used to model finishers that are defined as stanalone components. The attributes of thes groups are the same as those previously defined in the inserter class objects descriptions..

ComponentID Group

General Group

Finisher Device Table

Finisher Input Table

Finishing Features Table

Finisher Output Table

Finisher Supply Table

Finisher Colorant Table

## PROCESS CLASS OBJECTS

Value Add Programs in a large mailing or print-on-demand document finishing system can be modelled using the following common group definition. This group may be defined and instrumented in a standalone component, or it may be included as group in another MIF which has been defined to manage software units of work in a multiprocess workflow environment.

Process ID

Process Name

Process Version

Process Type

Process Input Parameter Set

Process Output Parameter Set

Process Start

Process End

Process Accounting Data

## RECONCILIATION AGENT CLASS OBJECTS

The standard group definitions defined in this section are used by instumented component agents provided by LMO System mangers to reconcile products and print jobs in operational LMO system environments. Once the instrumented reconciliation component is installed, standard DMI management interfaces are used to pass parameters to activate and control the reconcilation process.

### RECONCILIATION COMPONENTID GROUP

This group makes general information provided by the installer of this component available to management applications. It is a required group in all Management Information Format (MIF) component definitions.

Manufacturer

Product

Version

Serial Number

Installation Date

Verify

### RECONCILE PRODUCT GROUP

This group specifies attributes that apply to reconciling Products in an LMO system.

Product Name

Product Instance Qualifier

Operation

Current Status

Message

Delete Row

Query

## RECONCILE PRINT JOB GROUP

This group specifies attributes that apply to reconciling Print Jobs in an LMO system.

Product Name

Product Instance Qualifier

Print Job Name

Operation

Current Status

Message

Delete Row

Query

## RECONCILE JOB CONTROL GROUP

This group specifies job control attributes for instrumented reconciliation agents.

Index

Component Name

Group Name

Attribute List

Condition

Path Name

File Control

Time Out

Command

Delete Row

Query

## RECONCILE REPRINT CONTROL GROUP

This group specifies reprint control attributes for instrumented reconciliation agents.

Index

Component Name

Group Name

Attribute List

Condition

Path Name

File Control

Time Out

Command

Delete Row

Query

### RECONCILE PRODUCT CONTROL GROUP

This group specifies product control attributes for instrumented reconciliation agents.

Index

Component Name

Group Name

Attribute List

Condition

Path Name

File Control

Time Out

Command

Delete Row

Query

## LMO UTILITY CLASS OBJECTS

The standard group definitions defined in this section are used by instumented component agents provided by LMO System mangers to perform utility functions in operational LMO system environments. Once the instrumented component is installed, standard DMI management interfaces are used to pass parameters to activate and control the specified operations.

### LMO UTILITY COMPONENTID GROUP

This group makes general information provided by the installer of this component available to management applications. It is a required group in all Management Information Format (MIF) component definitions.

Manufacturer

Product

Version

Serial Number

Installation Date

Verify

### INPORT EXPORT GROUP

This group specifies attributes that apply to importing and exporting attribute values of a specified component in an LMO system..

Component Name

Group Name

Attribute List

Condition

Path Name

File Control

Current Status

Message

Operation

Delete Row

Query

## QUERY ROWS GROUP

This group specifies attributes to be queried in a specified component and group in an LMO system.

Component Name

Group Name

Attribute List

Condition

Query Rows

Delete Row

Query

## INDICATE CONTROL GROUP

This group specifies attributes for controlling on/off indications in specified components and groups in an LMO system.

Component Name

Group Name

Attribute Name

Operation

## WATCH CONTROL TABLE

This table specifies control attributes for monitoring specific attribute values for change in specified components and groups. Each row in the table defines the following control parameters.

Component Name

Group Name

Attribute Name

Key List

Current Value

Accumulate Value

Time Started

Time Stopped

Time Disabled

Duration Watching

Duration Disabled

Polling Interval

Current Status

Message

Operation

Delete Row

Query

## WATCH TABLE

This table specifies change information about attribute values in specified components and groups that are being watched. . Each row in the table defines the following information on changes.

Component Name

Group Name

Attribute Name

Key List

Value

Previous Value

Start Count

Accumulator

Instance Count

Average

Delete Row

Query

## HISTORY CONTROL TABLE

This table specifies history control for saving  attribute values in specified components and groups that are being watched. . Each row in the table defines the following control information.

History Name

Component Name

Group Name

Key List

Watch Attributes

Save Attributes

Polling Interval

File Name

Last Poll Time

Current Status

Message

Operation

Delete Row

Query

# OBJECT FORMAT

Objects are described in a simple format called the Management Information Format, or MIF. The MIF defines components, groups and their associated attributes. The MIF language syntax and the rules for encoding MIF files are described in The Desktop Management Interface Specification previously referenced in the DMI Overview.

Standard MIF files have been defined by the DMTF Finisher Work Group. for the following components:

- **LMO Product MIF**
- **Mail Job MIF**
- **Printer MIF with Inline Finishers**
- **Inserter MIF with Inline Finishers**
- **Finisher MIF Standalone**

The MIF syntax files for these standard components may be obtained from the DMTF ftp library on request

MIF objects for standard group and attributes are defined in this document using the following syntax notations:.

## OBJECT SYNTAX

| | |
|---|---|
| **OBJECT** | A simple descriptive name assigned to the object. This simple name is used to identify a group or attribute in a MIF. |
| **SYNTAX** | The data type of the MIF object. |
| **DEFINITION** | A description of the MIF object. |
| **ACCESS** | One of read-only, read-write, write-only, or not accessible. This is an attribute of the object to be assigned specifying access control over the object If the object is further used as a key for accessing entries in a table,that is also indicated here. |

## DATA TYPES

The data types used in this document conform to the data types allowed in the MIF. The types are:

| | |
|---|---|
| **integer** | A 32 bit signed integer. |
| **integer64** | A 64 bit signed integer. |

| | |
|---|---|
| **gauge** | A 32 bit unsigned integer. that may increase or decrease. When a guage reaches its maxiimum value (32,767), it continues to report the maximum value until the value decreases below the maximum. An automobile's speedometer is an example of a guage. |
| **counter** | A 32 bit unsigned integer that never decreases. A counter increases to the maximum possible value, then rolls over to zero. An automobile's odometer is an example of a counter. |
| **counter64** | A 64 bit unsigned integer that never decreases.. See counter. |
| **string(n) or**<br><br>**displaystring(n )** | A displayable string of *n* octets. |
| **octetstring** | A string of n octets, not necessarily displayable. |
| **date** | A 28-octet displayable string in the following format:<br><br>yyymmddHHMMss.uuuuuu+000.<br><br>The time fields are year/month/day/hours seconds and microseconds offset from UTC. |

For string types, the value *n* represents the maximum number of octets in the string. The actual number of octets may be shorter than this maximum value.

## GLOBAL ENUMERATIONS

The following enumerations are global to the LMO Product and LMO Mail Job Mifs.

```
start enum
        name = "Status"
//
        type = integer
        1 = "Other"
        2 = "Unknown"
  // Host set
       10 = "Created"
       11 = "Spooled"
       12 = "Held"
       13 = "Started"
       14 = "Normal End"
       15 = "Abnormal End"
  // Print set
```

```
    20 = "Printing"
    21 = "Printed"
    22 = "Reprinted"
// Insert set                                /
    30 = "Inserting"
    31 = "Inserted"
    32 = "Removed"
    33 = "Diverted"
    34 = "Mutilated"
    35 = "Missing"
    36 = "Recovered"
// Manual set
    50 = "Quality"
    51 = "Inventory"
    55 = "Mailing"
    56 = "Mailed"
// LMO System set
    60 = "Reconciling"
    61 = "Reconciled"
// Customer extensions
// 1xxx = "         "
// Vendor extensions
// 2xxx = "         "
// LMO System Manager extensions
// 3xxx = "         "
  end enum
```

# Product Class Objects

The standard groups defined in this section are used to model the attributes associated with multi-process work flow applications for large mailing operations and print-on-demand document finishing environments.

## PRODUCT COMPONENTID GROUP- MANDATORY

*Table 1 Product ComponentId Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| ComponentID | DisplayString(64) | This is a required group for all MIF components. It contains required attributes that must be specified for each component installed in the Management Information Format data base.<br><br>The ComponentID group is the first group in a DMTF component definition and must be assigned the Group Id value of 1.<br><br>The class value for this group is: DMTF\|ComponentId\|1.0. | read-only |

*Table 2 Product ComponentId Group Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Manufacturer | DisplayString(64) | The name of the organization that produces this component. | read-only |
| Product | DisplayString(64) | The name of this component or product. | read-only |
| Version | DisplayString(64) | The version string definition for this component. | read-only |
| Serial Number | DisplayString(64) | The serial number for this component. | read-only |
| Installation | Date | The time and date of the last install of this component on this system. | read-only |
| Verify | Enumeration | Verification level for this component. Asking for the verify attribute causes the component instrumentation code to perform checks to verify that the component is still in the system and working properly. It should return one of the following values:<br>(0) An error occurred; Check status code.<br>(1) Component does not exist.<br>(2) Verify not supported.<br>(4) Component exists, functionally untested.<br>(5) Component exists, functionally unknown.<br>(6) Component exists, functionally no good.<br>(7) Component exists, functionally good. | read-only |

## LMO PRODUCT TABLE

*Table 3 LMO Product Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Product Table | Sequence of Product Entries | A Large Mailroom Operation may have many application products active at the same time. The Product Table contains an entry for each product. Each entry is an ordered set of attributes describing the general characteristics of the product. The entries in this table are indexed by Product Name and Product Instance Qualifier. | read-write |

*Table 4 LMO Product Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Product Name | DisplayString(63) | The name of the product assigned by the application. This name is used by the Graphical User Interface programs and other management applications when referring products that are active in the system. For example, AAA may be the product name assigned to the application for mailing AAA statements. | read-write key |
| Product Instance Qualifier | DisplayString(63) | A text string qualifying the name of the product assigned by the application. The product instance qualifier name provides the ability to manage multiple instances of the same product at the same time. For example, the instance name qualifier for a product named AAA could be a string that specifies the quarter and cycle for this run of the product (AAA 4thQtr. 2ndCycle). | read-write key |
| Description | DisplayString(255) | A text string describing an application product that is active in the system. This object may be requested by various interfaces that list descriptions of components currently being managed by the Mailing Operations Manager. This field is also used by the LMO GUI to automate display of help information for the product. | read-write |
| Owner | DisplayString(255) | A text string identifying the owner of the product. This object may simply be the name of an individual or contain additional information such as address and telephone number determined to be of value in identifying ownership for the application product. | read-write |
| Date Created | Date | The date and time that this product entry was created and identified to the system. The time is described in a standard displayable format defined by the DMTF Date data type in terms of year/month/day/hours/minutes/seconds/and microseconds offset from UTC. | read-write |
| Input Origin | DisplayString(255) | The origin of the application data associated with this product. For example a mailing | read-write |

| | | | |
|---|---|---|---|
| | | product may be defined to process output from a specific branch office. The use of this object is product specific. | |
| Segment Size Unit | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Lines of Print<br>(4) Pages<br>(5) Mailpieces | The unit of measure used to estimate and report on the size of product segments. | read-write |
| Average Segment Size | Integer | The average size of a product segment in Segment Size Units. | read-write |
| Estimated Number of Segments | Integer | The estimated number of segments in this product. This attribute may be used to schedule work in an installation and to monitor percent completeness as individual product segments are reported complete. | read-write |
| Current Number of Segments | Integer | The current number of segments created for this product. This attribute may be used to monitor completeness of work. The ratio of this value to the estimated number of segments is an estimate of percent of product segmentation completed. The difference between the estimated number of segments and this value is the number of segments yet to be processed. | read-write |
| Final Number of Segments | Integer | The actual number of segments that were created for this product. This value may be used for accounting purposes and reconciliation of work associated with this product. | read-write |
| Accounting Data Format | Integer | An integer identifier specifying the format used for capturing and reporting accounting information for this product. Format identifiers are defined by users of the LMO system and may vary from product to product. Accounting data format specifications defined how to interpret the value specified in the Accounting Data. | read-write |
| Accounting Data | String(511) | Accounting Information for this product recorded in the format specified by the Accounting Data Format attribute. | read-write |
| Current Status | Status | The current status of this product entry | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | An attribute allowing instrumented management applications to delete this row in the table | read-write |
| Query | String(10) | Query information about keys for this product table entry | read-write |

## PRODUCT HISTORY TABLE

*Table 5 Product History Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Product History Table | Sequence of ProductHistory Entries | This table provides a status history of products as they proceed through the system | read-write |

*Table 6Product History Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Product Name | DisplayString(63) | The name of the product to which this history entry applies. | read-write key |
| Product Instance Qualifier | DisplayString(63) | The Instance Qualifier associated with the name of the product to which this history entry applies. The combination of Product Name and Product Instance Qualifier is required to insure unique key access to table entries when multiple instances of a product are active in the system at the same time. | read-write key |
| Time Stamp | Date | The date and time that this row was placed in the table. | read-write key |
| Current Status | Status | The status of this product entry. | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | An attribute allowing instrumented management applications to delete this row in the table | read-write |
| Query | String(10) | Query information aboyt keys for this table entry | read-write |

## PROCESS PATH TABLE

*Table 7 Process Path Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Process Path Table | Sequence of Process Path Entries | The sequence of steps that a product goes through to produce print data for mailing may be automated by some systems. The entries in this table are indexed by Product Name, Product Instance Qualifier and Process Name. Only those products that are to be managed in this fashion have any entries in this table. | read-write |

*Table 8 Process Path Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Product Name | DisplayString(63) | The name of the product to which this process step applies. | read-write key |
| Product Instance Qualifier | DisplayString(63) | The Instance Qualifier associated with the name of the product to which this process step applies. The combination of Product Name and Product Instance Qualifier is required to insure unique key access to table entries when multiple instances of a product are active in the system at the same time. | read-write key |
| Process Name | DisplayString(63) | The name of the process to be invoked for this instance of the product**.** | read-write key |
| Process Order | Integer | The process sequence number for a particular instance of the product. | read-write |
| Process Accounting Data | Octetstring | The accounting data returned on the completion of the process. The format and content of this data is process defined. | read-write |
| Current Status | Status | The status of this process entry. | read-write |

| | | | |
|---|---|---|---|
| Delete Row | Enumeration<br>0 = False<br>1 = True | An attribute allowing instrumented management applications to delete this row in the table | read-write |
| Query | String(10) | Query information aboyt keys for this table entry | read-write |

## PROCESS HISTORY TABLE

*Table 9 Process History Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Process History Table | Sequence of Process History Entries | A number of value add processing steps may be associated with a product. The history of status changes for each process step can be recorded in this table.. | read-write |

*Table 10 Process History Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Product Name | DisplayString(63) | The name of the product to which this process history entry applies. | read-write key |
| Product Instance Qualifier | DisplayString(63) | The Instance Qualifier associated with the name of the product to which this process step applies. The combination of Product Name and Product Instance Qualifier is required to insure unique key access to table entries when multiple instances of a product are active in the system at the same time. | read-write key |
| Process Name | DisplayString(63) | The name of the process to which this process history entry applies. | read-write key |
| Time Stamp | Date | The date and time that this row was placed in the table. | read-write key |
| Current Status | Status | The status of this process entry. | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | An attribute allowing instrumented management applications to delete this row in the table | read-write |
| Query | String(10) | Query information about keys for this table entry | read-write |

## PRINT HISTORY TABLE

*Table 11 Print History Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Print History Table | Sequence of Print History Entries | The print output from a product may be segmented into manageable units of work. Each unit of work is a unique print job. The status of print jobs can be recorded in this table | read-write |

*Table 12 Print History Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Product Name | DisplayString(63) | The name of the product to which this print history entry applies. | read-write key |
| Product Instance Qualifier | DisplayString(63) | The Instance Qualifier associated with the name of the product to which this print history applies.. | read-write key |
| Print Job Name | DisplayString(63) | The name of the print job to which this print history entry applies. | read-write key |
| Time Stamp | Date | The date and time that this row was placed in the table | read-write key |
| JES Assigned ID | DisplayString(8) | A text string specifying the ID assigned by the Job Entry System or Print Manager when print files are submitted for spooling. This attribute is used to correlate release/hold and cancel requests for mail job printing by the print job spooler. | read-write |
| Current Device | DisplayString(63) | The name of the device associated with this entry. | read-write |
| Current Status | Status | The status of this print job entry. | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | An attribute allowing instrumented management applications to delete this row in the table | read-write |
| Query | String(10) | Query information about keys for this table entry | read-write |

## INSERT HISTORY TABLE

*Table 13 Insert History Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Insert History Table | Sequence of Insert History Entries | The print output from one or more products may be input into an insert job. The manageable unit of work for inserters is the insert job. | read-write |

*Table 14 Insert History Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Product Name | DisplayString(63) | The name of the product to which this insert history entry applies. | read-write key |
| Product Instance Qualifier | DisplayString(63) | The Instance Qualifier associated with the name of the product to which this insert history entry applies. | read-write key |
| Insert Job Name | DisplayString(63) | The name of the insert job to which this insert history enty applies. | read-write |
| Time Stamp | Date | The date and time that this row was placed in the table | read-write key |
| Current Device | DisplayString(63) | The name of the device associated with this entry. | read-write |
| Current Status | Status | The status of this insert job entry. | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | An attribute allowing instrumented management applications to delete this row in the table | read-write |
| Query | String(10) | Query information about keys for this table entry | read-write |

# MAIL JOB CLASS OBJECTS

The standard group definitions defined in this section are used to model the attributes associated with managing the printing and inserting operations for large mailing and print-on-demand document finishing environments. Each of the manageable units of work for a mail job are defined in this section along with its associated control parameters, bill of materials and current status.

## MAIL JOB COMPONENTID GROUP- MANDATORY

*Table 15 Mail Job ComponentId Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| ComponentID | DisplayString(64) | This is a required group for all MIF components. It contains required attributes that must be specified for each component installed in the Management Information Format data base.<br><br>The ComponentID group is the first group in a DMTF component definition and must be assigned the Group Id value of 1.<br><br>The class value for this group is: DMTF\|ComponentId\|1.0. | read-only |

*Table 16 Mail Job ComponentId Group Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Manufacturer | DisplayString(64) | The name of the organization that produces this component. | read-only |
| Product | DisplayString(64) | The name of this component or product. | read-only |
| Version | DisplayString(64) | The version string definition for this component. | read-only |
| Serial Number | DisplayString(64) | The serial number for this component. | read-only |
| Installation | Date | The time and date of the last install of this component on this system. | read-only |
| Verify | Enumeration | Verification level for this component. Asking for the verify attribute causes the component instrumentation code to perform checks to verify that the component is still in the system and working properly. It should return one of the following values:<br>(0) An error occurred; Check status code.<br>(1) Component does not exist.<br>(2) Verify not supported.<br>(4) Component exists, functionally untested.<br>(5) Component exists, functionally unknown.<br>(6) Component exists, functionally no good.<br>(7) Component exists, functionally good. | read-only |

## PRINT JOB TABLE

*Table 17 Print Job Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Print Job Table | Sequence of Print Job Entries | Print output from products is segmented into manageable units of work called Print Jobs. The Print Job Table contains an entry for each print job describing its general characteristics, associated control objects, estimated and actual size, current state, and other pertinent management information. Product Name, Product Instance Qualifier, and Print Job Name are used as keys to entries in this table. | read-write |

*Table 18 Print Job Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Product Name | DisplayString(63) | The name of the product to which this print job entry applies. | read-write key |
| Product Instance Qualifier | DisplayString(63) | The Instance Qualifier for the name of the product associated with this print job entry. | read-write key |
| Print Job Name | DisplayString(63) | The name of the print job assigned by the application. This name is unique within the domain of print job names for a given product instance and may be used as a key to access a particular print job entry in the table. | read-write key |
| Print File Name | DisplayString(8) | The name of the file containing the printdata for this Print Job. This attribute is used by print, archive, and reprint service processes that operate on mailjob print data. | read-write key |
| Description | DisplayString(255) | A text string describing a print job that is active in the system. | read-write |
| Required Media Name | DisplayString(63) | The name of the required media or form to be used when printing this job. | read-write |
| Associated FormDef Name | DisplayString(8) | The name of the FormDef object used to control the printing process for the Print Job. This name is captured at print time and later used by reprint components to re-establish the print environment before reprinting mutilated statements. | read-write |
| Associated PageDef Name | DisplayString(8) | The name of the PageDef object used to format line data into pages for the Print Job. This name may be captured at print time and later used by reprint components to re-establish the print environment before reprinting mutilated statements. This attribute only applies to processing LINEMODE or MIXED LINEMODE print file formats. | read-write |
| Associated JobTicket Name | DisplayString(63) | The name of the Job Ticket used to control the inserting process for the Print Job. This name may also be used to re-establish the inserting environment before reprocessing mutilatedmailpieces. | read-write |
| Associated Document Index Name | DisplayString(8) | The name of the Document Index file to use to locate individual mailpiece pages within an archived print data set. This attribute is | read-write |

| | | | |
|---|---|---|---|
| | | accessed by re-print components to retrieve pages of individual statements for reprinting of mutilated mailpieces. | |
| Print File Format | DisplayString(63) | A text string specifying the document format of print data contained in the print file for this print job. ISO AFP values for this attribute are LINEDATA, MIXED, and MODCA. Other document-format strings defined by ISO/DPA Standard 10175 should be used when appropriate. | read-write |
| JES Assigned ID | DisplayString(8) | A text string specifying the ID assigned by the Job Entry System or Print Manager when print files are submitted for spooling. This attribute is used to correlate release/hold and cancel requests for mail job printing by the print job spooler. | read-write |
| Date Created | Date | The date and time that this print job was created and identified to the system. | read-write |
| Mail Job Size | Counter | An integer count of the number of mailpiece page groups contained within this print job. | read-write |
| Print Job Size Unit | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Lines of Print<br>(4) Pages<br>(5) Mailpieces | The unit of measure used to estimate and report on the size of this print job. | read-write |
| Estimated Print Job Size | Integer | The estimated size of this print job in Print Job Size Units. This attribute may be used by installations to schedule work for both printers and inserters of print data. | read-write |
| Actual Print Job Size | Integer | The actual size of the print output produced by this print job expressed in Print Job Size Units. | read-write. |
| Current Device | DisplayString(63) | The name of the current printer or inserter device that is processing the print output of this print job. | read-write |
| Current Status | Status | The current status of this print job entry. | read-write |
| First Mailpiece Identifier | DisplayString(63) | The name of the first mailpiece in this print job. | read-write |
| Last Mail piece Identifier | DisplayString(63) | The name of the last mailpiece in this print job. | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | An attribute allowing instrumented management applications to delete this row in the table | read-write |
| Query | String(10) | Query information about keys for this table entry | read-write |

## INSERT JOB TABLE

*Table 19 Insert Job Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Insert Job Table | Sequence of Insert Job Entries | The Insert Job table contains an entry for each active insert job in the LMO system. Insert jobs control finishing and inserting of physical materials into envelopes or containers for mailing and delivery. There | read-write |

| | | | |
|---|---|---|---|
| | | may be one or more insert job definitions associated with a given product. For example, machine setups, bill of materials and job ticket instructions can vary within a product for different postal classes of service or special account mailings e.g., platinum card holder billing. Each entry in the table is a unique insert job specification. Product Name, Product Instance Qualifier, and Insert Job Name are used as keys to entries in this table. | |

*Table 20 Insert Job Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Product Name | DisplayString(63) | The name of the product associated with this insert job. | read-write key |
| Product Instance Qualifier | DisplayString(63) | The Instance Qualifier associated with the name of the product to which this insert job applies. The combination of Product Name and Product Instance Qualifier is required to insure unique key access to table entries when multiple instances of a product are active in the system at the same time. | read-write key |
| Insert Job Name | DisplaysString(63) | The name assigned to this insert job. This name is unique within the domain of insert job names for a given product and is one of the keys used to access entries in the Insert Job Table. | read-write key |
| Description | DisplayString(255) | A free form text description of this insert job. This description may be presented to operators and other users of the system by the LMO Graphical User Interface and other management applications. | read-write |
| Associated Job Ticket Name | DisplayString(63) | The name of the Job Ticket used to control the inserting process for this insert job. | read-write |
| Date Created | Date | The date and time that this insert job was created and identified to the system. | read-write |
| Insert Job Size Unit | Enumeration:<br>(1) Other<br>(2) Unknown<br>(4) Pages<br>(5) Mailpieces | The unit of measure used to estimate and report on the size of this insert job. | read-write |
| Estimated Insert Job Size | Integer | The estimated size of this insert job in Insert Job Size Units. This attribute may be used by installations to schedule work for inserters. | read-write |
| Actual Insert Job Size | Integer | The actual size of the insert output produced by this insert job expressed in Insert Job Size Units. | read-write. |
| Current Device | DisplayString(63) | The name of the current inserter device that is processing the print output of this insert job. This value only has meaning when the current status of the Insert Job is Inserting. | read-write |
| Current Status | Status | The current status of this insert job, | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | An attribute allowing instrumented management applications to delete this row in the table | read-write |
| Query | String(10) | Query information about keys for this table | read-write |

| | |
|---|---|
| | entry |

## BILL OF MATERIALS TABLE

*Table 21 Bill of Materials Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Bill of Materials Table | Sequence of Bill of Material Entries | This table lists the bill of materials associated with insert jobs. Each entry in the table identifies a specific type of insert such as print output from a print job, call inserts setup by operators or named media stock supplied from inventory. Multiple bill of material entries of the same or different types may be associated with a given Insert Job. Bill of materials for multiple Insert jobs may be specified in the table. Product Name, Product Instance Qualifier, Insert Job Name and Material Name are used as keys to entries in this table | read-write |

*Table 22 Bill of Materials Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Product Name | DisplayString(63) | The name of the product associated with this bill of material entry.. | read-write key |
| Product Instance Qualifier | DisplayString(63) | The Instance Qualifier associated with the name of the product to which this bill of material entry applies. The combination of Product Name and Product Instance Qualifier is required to insure unique key access to table entries when multiple instances of a product are active in the system at the same time. | read-write key |
| Insert Job Name | DisplayString(63) | The name of the insert job with which this bill of material entity is associated. | read-write key |
| Material Name | DisplayString(63) | The name of the material for the type of material referenced by **Type**. For example, if the type is Print Job, then the value is a Print Job Name; if the type is a Call Insert, then the value is the Call Name referenced by the vendor defined Job Ticket to accommodate late bindings, and if the type is Named Media, then the value is the name of preprinted media that possibly may be maintained in inventory. | read-write key |
| Type | Enumeration:<br>(1) Other<br>(3) Print Job<br>(4) Call Insert<br>(5) Named Media<br>(6) Envelope | The type of material that is referenced by this bill of material entry. | read-write |
| Description | DisplayString(255) | A free form text description of this material insert in a form meaningful to inserter operators and management applications. | read-write |
| Number Required | Integer | The amount of this material required for this | read-write |

| | | | |
|---|---|---|---|
| | | insert job. | |
| Number Used | Integer | The amount of this material used for this insert job. | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | An attribute allowing instrumented management applications to delete this row in the table | read-write |
| Query | String(10) | Query information about keys for this table entry | read-write |

## MAILPIECE XREFERENCE TABLE

*Table 23 Mailpiece XReference Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Mailpiece XReference Table | Sequence of Mailpiece XReference Entries | This table provides a mapping of Mailpiece Identifiers to the Process Assigned Identifier, the Insert Job Name, and the Print Job Name for each mailpiece, to enable reprint and tracking. | read-write |

*Table 24 Mailpiece Xreference Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Product Name | DisplayString(63) | This name of the product associated with this mailpiece entry. | read-write key |
| Product Instance Qualifier | DisplayString(63) | The Instance Qualifier for the name of the product associated with this mailpiece entry. The combination of Product Name and Product Instance Qualifier is required to insure unique key access to table entries when multiple instances of a product are active in the system at the same time. | read-write key |
| MailPiece Identifier | DisplayString(63) | The name of the Mailpiece assigned by the application. For example an account number. | read-write key |
| Process Assigned Identifier | DisplayString(63) | A sequence number assigned to this mailpiece in the print job file as a result of performing a value add process; for example, presorting mailpiece data by zip code, walk sequence or traying sequence. | read-write |
| Print Job Name | DisplayString(63) | The name of the Print Job that contains the print data associated with this mailpiece entry. | read-write |
| Insert Job Name | DisplayString(63) | The name of the insert job that contains this mailpiece. | read-write |
| Mailpiece Size Unit | Enumeration<br>(1) Other<br>(2) Unknown<br>(3) Lines of Print<br>(4) Pages<br>(5) Ounces<br>(6) Tenths of Ounces<br>(7) Pounds<br>(8) Grams<br>(9)Kilograms | The unit of measure used to report on the size of a mailpiece | read-write |
| Mailpiece Size | Integer | The size of this mailpiece expressed in | read-write |

| | | | |
|---|---|---|---|
| | | Mailpiece Size Units | |
| Mailpiece Postage Unit | DisplayString(63) | The unit of measure used to report on the size of the mailpiece postage | read-write |
| Mailpiece Postage | DisplayString(63) | The amount of postage required to mail this mailpiece expressed in Mailpiece Postage Units | read-write |
| Current Status | Status | The current status of thismailpiece entry, | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | An attribute allowing instrumented management applications to delete this row in the table | read-write |
| Query | String(10) | Query information about keys for this table entry | read-write |

## MAILPIECE HISTORY TABLE

*Table Mailpiece History Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Mailpiece History Table | Sequence of Mailpiece History Entries | This table provides a history of mailpieces as they proceed through the system | read-write |

*Table 25 Mailpiece History Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Product Name | DisplayString(63) | The name of the product to which this mailpiece history entry applies. | read-write key |
| Product Instance Qualifier | DisplayString(63) | The Instance Qualifier associated with the name of the product to which this mailpiece history entry applies. | read-write key |
| Mailpiece Identifier | DisplayString(63) | The name of the mailpiece assigned by the application | read-write |
| Time Stamp | Date | The date and time that this row was placed in the table | read-write key |
| Current Status | Status | The status of this insert job entry. | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | An attribute allowing instrumented management applications to delete this row in the table | read-write |
| Query | String(10) | Query information about keys for this table entry | read-write |

# PRINTER CLASS OBJECTS

The standard group definitions in this section are used to model the attributes associated with printer devices and finisher products that are inline in the print path of the printer.

## PRINTER MODEL

Figure 5. Printer Model. The DMTF Printer Workgroup has developed the above model which is useful in understanding the various objects making up a printer component definition.

## PRINTER COMPONENTID GROUP- MANDATORY

*Table 26 Printer ComponentId Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| ComponentID | DisplayString(64) | This is a required group for all MIF components. It contains required attributes that must be specified for each component installed in the Management Information Format data base.<br><br>The ComponentID group is the first group in a DMTF component definition and must be assigned the Group Id value of 1.<br><br>The class value for this group is: DMTF\|ComponentId\|1.0. | read-only |

*Table 27 Printer ComponentId Group Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Manufacturer | DisplayString(64) | The name of the organization that produces this component. | read-only |
| Product | DisplayString(64) | The name of this component or product. | read-only |
| Version | DisplayString(64) | The version string definition for this component. | read-only |
| Serial Number | DisplayString(64) | The serial number for this component. | read-only |
| Installation | Date | The time and date of the last install of this component on this system. | read-only |
| Verify | Enumeration | Verification level for this component. Asking for the verify attribute causes the component instrumentation code to perform checks to verify that the component is still in the system and working properly. It should return one of the following values:<br>(0) An error occurred; Check status code.<br>(1) Component does not exist.<br>(2) Verify not supported.<br>(4) Component exists, functionally untested.<br>(5) Component exists, functionally unknown.<br>(6) Component exists, functionally no good.<br>(7) Component exists, functionally good. | read-only |

## PRINTER GENERAL GROUP -MANDATORY

*Table 28 Printer General Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| General | DisplayString(64) | This group specifies general attributes that apply to the overall printer and its states. | read-only |

*Table 29 Printer General Group Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Config Changes | Counter | A count of the configuration changes for this printer. This value is incremented each time a configuration change is made to this printer component. This value may be compared against a previous read value to determine if any configuration changes have been made to the printer.<br><br>Management applications may cache infrequently changed configuration information about sub-units on the printer. This object should be incremented whenever the agent wishes such applications to invalidate that cache and re-download all of this configuration information, thereby signally a change of printer configuration.<br><br>Configuration changes represent changes in capabilities. These changes represent modifications which may affect the set of jobs that can be printed by a system, increasing or decreasing the likelihood of successful printing. Examples might include the addition or removal of input devices, media paths and interpreters among other items.<br><br>For example, if an input tray that contained paper of different dimensions was added, this counter should be incremented. As an additional example, this counter would not be incremented when an input tray is removed or the level of input is changed. | read-only |
| Current Localization | Integer | This value is an index into the printer localization table specifying the current language, country and character set used in description string values for objects and attributes. | read-write |
| Reset | Enumeration<br>(3) Not Resetting<br>(4) Power Cycle Reset -cold start<br>(5)Reset to NVRAM - warm start<br>(6) Reset to Factory Defaults -resets NVRAM | Setting this value to Power Cycle Reset, Reset NVRAM or Reset to Factory Defaults will result in the resetting of the printer. Setting the value to Not Resetting will have no effect on the printer.. | read-write |
| Input Default Index | Integer | The value of the index corresponding to the | read-write |

| | | | |
|---|---|---|---|
| | | default input device in the **Input Table**. This value selects the default source of the input media. A value of 1 selects the first input in the Input Table. | |
| Output Default Index | Integer | The value of the index corresponding to the default output device in the **Output Table**. This value selects the default destination of the print media. A value of 1 selects the first device in the Output Table. | read-write |
| Marker Default Index | Integer | The value corresponding to the default marker device in the **Marker Table**. a value of 1 selects the first marker device in the Marker Table. | read-write |
| Media Path Default Index | Integer | The value of the index corresponding to the default media path in the **Media Path Table**. A value of 1 selects the first path in the Media Path Table. | read-write |
| Console Localization | Integer | An index into the **Localization Table** giving the language, country and character set to be used for the console. a value of 1 selects the first localization in the table. | read-write |
| Console Number of Display Lines | Integer | The number of lines on the printer's physical display .This value is zero if there are no lines on the physical display or if there is no physical display. | read-write |
| Console Number of Display Characters | Integer | The number of characters per line displayed on the physical display. | read-write |
| Console Disable | Ennumeration<br>(1) Other<br>(3) Enabled<br>(4) Disabled | A lock to provide one at a time access by co-operationg management applications to the operations of the operator console. | read-write |
| Current Operator | String(127) | The name of the operator or person responsible for the printer. This string may contain additional information such as phone number. | read-write |
| Service Person | String(127) | The name of the last person responsible for serviceing this printer. This string may contain additional information such as phone number. | read-write |
| Status | Enumeration | This value indicates the overall status of this printer component. The enumeration values have the following meanings::<br>(1) Other<br>(2) Unknown<br>(3) Available<br>(4) Unavailable Critical Alert (5) Available Non Critical Alert<br>(6) Unavailable Both Critical and Non Critical Alerts<br>(7) On Request<br>(8) Unavailable<br>(9) Busy Temporarily Unavailable | read-only |

## PRINTER LOCALIZATION TABLE

*Table 30 Printer Localization Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Localization Table | Sequence of Localization Entries | This table identifies the natural language, country and character sets supported by the printer | read-only |

*Table 31 Printer Localization Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by this printer to identify a specific entry in the localization table. Although these values may change due to a major reconfiguration of the device (e.g. the addition of new character sets to the printer), values are expected to remain stable across succesive printer power cycles | read-only |
| Language | String(2) | A two character language code defined by ISO639. | read-only |
| Country | String(2) | A two character country code defined by ISO3166. | read-only |
| Localization Character Set | CodedCharSet Enumeration | The coded character set used for this localization. Registered CodedCharSet values are defined by DMTF workgroups consistent with ISO standards. | read-only |

## PRINTER INPUT TABLE -MANDATORY

*Table 32 Printer Input Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Input Table | Sequence of Input Device Entries | List of input devices capable of providing printing media resources to this printer including information regarding their identity, possible and current configuration, and status. | read-only |

*Table 33 Printer Input Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by this printer to identify an associated input device. Although these values may change due to a major reconfiguration of the device (e.g., the addition of new input devices to the printer) values are expected to remain stable across successive printer powercycles. | read-only |
| Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Sheet Auto Removable Tray<br>(4) Sheet Feed Non Auto Removable Tray | This value indicates the type of technology (discriminated primarily by feeder mechanism type) employed by the input device. | read-only |

| | | | |
|---|---|---|---|
| | (5) Manual Feed<br>(6) Continuous Roll<br>(7) Continuous Fan Fold | | |
| Dim Unit | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Thousandths of Inches<br>(4) Micrometers | The unit of measure for specifying dimensional values for this input device. | read-only |
| Declared Media Dim Feed Dir | Integer<br>(-1) Other<br>(-2) Unknown | This object provides the value of the declared dimension in the feed direction of the media that is placed or will be placed in this input device. The feed direction is the direction in which the media is fed on this device. The dimension is measured in Dim Units. If this device can reliably sense this value, the value is sensed and is read-only access. Otherwise the value is read-write access and maybe written by management or control panel applications. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates *unknown*. | read-write |
| Declared Media Dim XFeed Dir | Integer (-1) Other<br>(-2) Unknown | This object provides the value of the declared dimension in the cross feed direction of the media that is placed or will be placed in this input device. The cross feed direction is ninety degrees relative to the feed direction on this device. The dimension is measured in **Dim Units**. If this device can reliably sense this value, the value is sensed and is read-only access. Otherwise the value is read-write access and maybe written by management or control panel applications. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |
| Chosen Media Dim Feed Dir | Integer<br>(-1) Other<br>(-2) Unknown | This object provides the value of the granted dimension in the feed direction of the media that is placed or will be placed in this input device. Feed dimension measurements are taken parallel to the feed direction of the device and measured in **Dim Units**. If the printer supports the declared dimension, the granted dimension is the same as the declared dimension. If not, the granted dimension is set to the closet dimension that the printer supports when the declared dimension is set. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates *unknown*. | read-only |
| Chosen Media Dim XFeed Dir | Integer<br>(-1) Other<br>(-2) Unknown | This object provides the value of the granted dimension in the cross feed direction of the media that is placed or will be placed in this input device. The cross feed direction is ninety degrees relative to the feed direction | read-write |

| | | | |
|---|---|---|---|
| | | on this device. The dimension is measured in **Dim Units**. If the printer supports the declared dimension, the granted dimension is the same as the declared dimension. If not, the granted dimension is set to the closet dimension that the printer supports when the declared dimension is set. The value (-1) means *other and* specifically indicates that the device places no restriction on this parameter. The value (-2) indicates *unknow*n. | |
| Capacity Units | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Ten Thousands of Inches<br>(4) Micrometers<br>(5) Sheets<br>(16) Feet<br>(17) Meters | The unit of measure for specifying capacity values for this device. | read-only |
| Max Capacity | Integer<br>(-1) Other<br>(-2) Unknown | The maximum capacity of the input source in input capacity units. There is no convention associated with the media itself so this value is claimed capacity. If the input device can reliably sense this value, the value is sensed by the printer and is read-only. Otherwise the value may be written by a management or control console application. The value (-1) means *other and* specifically indicates that the device places no restriction on this parameter. The value (-2) indicates *unknown*. | read-write |
| Current Level | Integer<br>(-1) Other<br>(-2) Unknown | The current capacity of the input source in input Capacity units. If the input device can reliably sense this value, the value is sensed by the printer and is read-only. Otherwise the value may be written by a management or control console application. The value (-1) means *other and* specifically indicates that the device places no restriction on this parameter. The value (-2) indicates *unknown*. | read-write |
| Status | Enumeration: | This value indicates the status of this input device. The values have the following meanings:<br>(1) Other<br>(2) Unknown<br>(3) Available<br>(4) Unavailable Critical Alert<br>(5) Available Non Critical Alert<br>(6) Unavailable Both Critical and Non Critical Alerts<br>(7) On Request<br>(8) Unavailable<br>(9) Busy Temporarily Unavailable | read-write |
| Media Name | String(63) | The name of the current media contained in this input device. Examples are ISO-a4 white and Hillary's Pink Presidential Letterhead. See ISO standard name | read-only |

| | |
|---|---|
| | references for recommended values. |

## PRINTER INPUT TYPE ATTRIBUTES EXTENDED -OPTIONAL

*Table 34 Printer Input Device Type Extended*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Input Type Extended | Augments Device Type: Sequence of Attributes | A list of information suitable for extended input device management. | read-only |
| Name | DisplayString(63) | The name assigned to this input device. | read-only |
| Vendor Name | DisplayString(63) | The name of the vendor of this device. | read-only |
| Model | DisplayString(63) | The model name of this input device. | read-only |
| Version | DisplayString(63) | The version string for this input device. | read-only |
| Serial Number | DisplayString(63) | The serial number assigned to this input device. | read-only |
| Description | DisplayString(63) | A free form text description of this input device. | read-only |
| Security | Enumeration:<br>(1) Other<br>(2) Reserved<br>(3) On<br>(4) Off<br>(5) Not Present | Indicates whether this device has some security associated with it, and if so, whether it is on or off. | read-write |

## PRINTER INPUT MEDIA ATTRIBUTES EXTENDED -OPTIONAL

*Table 35 Printer Input Media Extended*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Input Media Extended | Augments Media Type: Sequence of Attributes | A list of attributes used to specify media associated with an input device. | read-only |
| Medium Weight | Integer (-1) Other<br>(-2) Unknown | The weight of the media associated with this input device in grams per meter squared. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |
| Media Type | DisplayString(63) | The name of the type of medium associated with this input device. Standard values are standardized strings from ISO 10175 (DPA) and ISO10180 (SPDL) which are:<br>stationary<br>transparency<br>envelope<br>envelope-plain<br>envelope-window<br>continuos-long<br>continuos-short<br>tab-stock<br>labels<br>multi-layer<br>Implementors may add additional string values. The naming conventions in ISO standards are recommended to advoid potential name clashes. | read-write |

| | | | |
|---|---|---|---|
| Media Color | DisplayString(63) | The name of the color of the medium associated with this input device. standard values are standardized strings from ISO 10175 (DPA) and ISO10180 (SPDL) which are:<br>white,<br>pink,<br>yellow.<br>buff,<br>goldenrod,<br>blue green,<br>transparent.<br>Implementors may add additional string values. The naming conventions in ISO standards are recommended to avoid potential name clashes. | read-write |
| Medium Forms Parts | Integer<br>(-2) Unknown | The number of parts associated with this medium if it is a multi-part form. The value (-2) indicates *unknown*. | read-write |

## MARKER DEVICE TABLE

*Table 36 Marker Device Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Marker Table | Sequence of Marker Device Entries | A list of marker devices in this printer capable of rendering information onto print media including information about their attributes and usage. | read-only |

*Table 37 Marker Device Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by this printer to identify an associated marker device. Although this value may change due to a major reconfiguration of the printer (e.g., the addition of new marker devices to the printer), values are expected to remain stable over successive printer powercycles. | read-only |
| Mark Tech | Enumeration:<br>(1) Other<br>(2) Unknown<br>(n) registered values | The type of technology employer by this marker. Registered enumerations defined by the DMTF Printer work group are:<br>(1) Other<br>(2) Unknown<br>(3) Electric Photographic LED<br>(4) Electric Photographic Laser<br>(5) Electric Photographic Other<br>(6) Impact Moving Head Dot Matrix 9 Pin<br>(7) Impact Moving Head Dot Matrix 24 Pin<br>(8) Impact Moving Head Dot Matrix Other<br>(9) Impact Moving Head Fully Formed<br>(10) Impact Band<br>(11) Impact Other<br>(12) Ink Jet Aqueous<br>(13) Ink Jet Solid | read-only |

| | | | |
|---|---|---|---|
| | | (14) Ink Jet Other<br>(15) Pen<br>16) Thermal Transfer<br>(17) Thermal Sensitive<br>(18) Thermal Diffusion<br>(19) Thermal Other<br>(20) Electroerosion<br>(21) Electrostatic<br>(22) Photographic Microfiche<br>(23) Photographic Image Setter (24) Photographic Other<br>(25) Ion Deposition<br>(26) E Beam<br>(27) Typesetter | |
| Counter Unit | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Ten Thousandths Of Inches<br>(4) Micrometers<br>(5) Characters<br>(6) Lines<br>(7) Impressions<br>(8) Sheets<br>(9) Dot Row<br>(11) Hours<br>(16) Feet<br>(17) Meters | The unit that will be used by this printer when reporting countervalues for this marker device. The time units of measure are provided for a device like a strip recorder that does not or can not track the physical dimensions or the media or does not use character, lines or sheets. | read-only |
| PowerOnCount | Counter | This value is an accumulative count from the last power on of this marker device in **CounterUnits**. | read-only |
| Life Count | Counter | This value is an accumulative count over the life of this marker device in Counter Units. | read-only |
| Process Colorant | Counter<br>(-1) Other<br>(-2) Unknown | The number of process colorants supported by this marker device. A process color value of 1 implies monochrome. The value of this object and the value of the **spot colorant** object can not both be 0. The value (-1) means *other and* specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-only |
| Spot Colorant | Counter<br>(-1) Other<br>(-2) Unknown | This value is the number of spot colorants supported by this marker device. The value of this object and the value of the **processcolorant** can not both be 0. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates *unknown*. | read-only |
| Addressability Unit | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Ten Thousandths of Inches<br>(4) Micrometers | The unit of measure of distance. | read-only |
| Addressability Feed Dir | Integer<br>(-1) Other<br>(-2) Unknown | The number of addressable marking positions in the Feed direction in units of measure specified by **Addressability Unit**. The value | read-only |

| | | | |
|---|---|---|---|
| | | (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | |
| Addressability XFeed Dir | Integer<br>(-1) Other<br>(-2) Unknown | The number of addressable marking positions in the Cross Feed direction in units of measure specified by **Addressability Unit**. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates | read-only |
| North Margin | Integer<br>(-1) Other<br>(-2) Unknown | The margin, in units identified by **Addressability Unit**, from the leading edge of the medium as the medium flows through the marking engine with the side to be image facing the observer. The leading edge is the North edge and the other edges are defined by the normal compass layout of directions with the compass facing the observer. Printing within the area bounded by all four margins is guaranteed for all interpreters. The value (-2) means unknown. | read-only |
| South Margin | Integer<br>(-1) Other<br>(-2) Unknown | The margin from the South edge (see North Margin) of the medium in units identified by **Addressability Unit**. Printing within the area bounded by all four margins is guaranteed for all interpreters. The value (-2) means unknown. | read-only |
| West Margin | Integer<br>(-1) Other<br>(-2) Unknown | The margin from the West edge (see North Margin) of the medium in units identified by **Addressability Unit print**ing within the area bounded by all four margins is guaranteed for all interpreters. The value (-2) means unknown. | read-only |
| East Margin | Integer<br>(-1) Other<br>(-2) Unknown | The margin from the East edge (see North Margin) of the medium in units identified by **Addressability Unit** Printing within the area bounded by all four margins is guaranteed for all interpreters. The value (-2) means unknown. | read-only |
| Status | Enumeration: | This value indicates the overall status of this marker device. The values have the following meanings::<br>(1) Other<br>(2) Unknown<br>(3) Available<br>(4) Unavailable Critical Alert<br>(5) Available Non Critical Alert<br>(6) Unavailable Both Critical and Non Critical Alerts<br>(7) On Request<br>(8) Unavailable<br>(9) Busy Temporarily Unavailable | read-only |

## MARKER SUPPLY TABLE

*Table 38 Marker Supply Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Marker Supply Table | Sequence of Marker Supply Entries | A list of marker supplies used in this printer, their description, capacities and current level. | read-only |

*Table 39 Marker Supply Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by the printer to identify this marker supply container or receptacle. Although these values may change due to a major reconfiguration of the printer (e.g., the addition of new marker supplies to the printer), values are expected to remain stable across successive printer power cycles. | read-only |
| Marker Index | Integer | The value of the **Marker Index** corresponding to the marker device with which this marker supply entity is associated. | read-only |
| Colorant Index | Integer | The value of the Colorant Index corresponding to the colorant with which this marker supply entity is associated. The value shall be 0 if there is no **Marker Colorant Table**. | read-write |
| Class | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Supply that is consumed<br>(4) Receptacle that is filled | This value indicates whether this supply entity represents a supply container that is consumed or a receptacle that is filled. | read-write |
| Type | Enumeration:<br>(1) other<br>(2) unknown<br>(3) toner<br>(4) waste toner<br>(5) ink<br>(6) ink cartridge<br>(7) ink ribbon<br>(8) waste ink<br>(9) OPC (organic photo conductor) (10) developer<br>(11) fuser oil<br>(12) solid wax<br>(13) ribbon wax<br>(14) waste wax | This value identifies the supply type of this supply. | read-only |
| Description | String(255) | The description of this supply container or receptacle in the localization specified by **General Current Localization**. | read-only |
| Supply Unit | Enumeration:<br>(1) other<br>2) unknown<br>(3) ten thousandths of inches | The unit of measure of this marker supply container or receptacle. | read-only |

| | | | |
|---|---|---|---|
| | (4) micrometers<br>(12) thousandths of ounces<br>(13) tenths of grams<br>(14) hundreds of fluid ounces<br>(15) tenths of milliliters | | |
| Max Capacity | Integer<br>(-1) Other<br>(-2) Unknown | The maximum capacity of this supply container or receptacle is expressed in Supply Units. If this supply container/ receptacle can reliably sense this value, the value is sensed by the printer and is read-only; otherwise the value may be written by management or control panel application. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates *unknown*. | read-write |
| Current Level | Integer<br>(-1) Other<br>(-2) Unknown | The current level if this supply entity is a container; the remaining space if this supply entity is a receptacle. If this supply container/ receptacle can reliably sense this value, the value is sensed by the printer and is read-only; otherwise the value may be written by management or control panel application. The value<br>(-1) means *other and* specifically indicates that the device places no restriction on this parameter. The value (-2) indicates *unknown*. | read-write |

## MARKER COLORANT TABLE

*Table 40 Marker Colorant Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Marker Colorant Table | Sequence of Marker Colorant Entries | A list of all the marker colorants available on the printer. | read-only |

*Table 41 Marker Colorant Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by the printer to identify this marker colorant. Although these values may change due to a major reconfiguration of the printer (e.g., the addition of new colorants to the printer), values are expected to remain stable across successive printer power cycles. | read-only |
| Marker Index | Integer | The value of the **Marker Index** corresponding to the marker device with which this marker colorant entry is associated. | read-only |
| Role | Enumeration:<br>(1) Other<br>(2) Unknown | The role played by this colorant. | read-write |

| | | | |
|---|---|---|---|
| | (3) Process<br>(4) Spot | | |
| Value | String(63) | The name of the color of this colorant using standardized string names from ISO 10175 (DPA)and ISO 10180 (SPDL) which are:<br>other,<br>unknown,<br>white,<br>red,<br>green,<br>blue,<br>cyan,<br>magenta,<br>yellow and<br>black.<br>Implementors may add additional string values. The naming conventions in ISO standards are recommended in order to avoid potential name clashes. | read-write |
| Tonality | Integer | The distinct levels of tonality realizable by a marking device when using this colorant. This value does not include the number of levels of tonal difference that an interpreter can obtain by techniques such as half toning. This value must be at least 2. | read-write |

## PRINTER OUTPUT TABLE -MANDATORY

*Table 42 Output Device Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Output Table | Sequence of Ouput Device Entries | A list of output devices capable of accepting and delivering printed media from printing process associated with this printer including information regarding their identity, possible and current configuration, and status. | read-only |

*Table 43 Output Device Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by this printer to identify an associated output device. Although these values may change due to a major reconfiguration of the device (e.g., the addition of new output devices to the printer), values are expected to remain stable across successive printer powercycles. | read-only |
| Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Removable Bin<br>(4) Unremoveable Bin<br>(5) Continuous Roll Device<br>(6) Mailbox<br>(7) Continuous Fan | The type of technology employed by this output device. | read-only |

| | | | |
|---|---|---|---|
| | Fold | | |
| Capacity Unit | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Ten Thousandths of Inches<br>(4) Micrometers<br>(8) Sheets<br>(16) Feet<br>(17) Meters | The unit of measure for specifying capacity values for this device. | read-only |
| Max Capacity | Integer,<br>(-1) Other<br>(-2) Unknown | The maximum capacity of this device in Output Capacity Units. There is no convention associated with the media itself so this value is claimed capacity. If the device can reliably sense this value, the value is sensed by the printer and is read-only; otherwise the value may be written by management or control panel applications. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |
| Remaining Capacity | Integer | The remaining capacity of the possible capacity in **Output Capacity units**. If the device can reliably sense this value, the value is sensed by the printer and is read-only; otherwise the value may be written by management or control panel applications. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates: | c. read-write |
| Status | Enumeration | This value indicates the specific state of this output device:<br>(1) Other<br>(2) Unknown<br>(3) Available<br>(4) Unavailable Critical Alert<br>(5) Available Non Critical Alert<br>(6) Unavailable Both Critical and Non Critical Alerts<br>(7) On Request<br>(8) Unavailable<br>(9) Busy Temporarily Unavailable | read-only |

## PRINTER OUTPUT TYPE ATTRIBUTES EXTENDED -OPTIONAL

*Table 44 Printer Output Device Type Extended*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Output Device Type Extended | Augments Output Type: Sequence of Attributes | A list of information suitable for extended output device management. | read-only |
| Name | DisplayString(63) | The name assigned to this output device. | read-only |
| Vendor Name | DisplayString(63) | The name of the vendor of this device. | read-only |
| Model | DisplayString(63) | The model name of this output device. | read-only |

| | | | |
|---|---|---|---|
| Version | DisplayString(63) | The version string for this output device. | read-only |
| Serial Number | DisplayString(63) | The serial number assigned to this Output device. | read-only |
| Description | DisplayString(255) | A free form text description of this output device in the localization specified by **General Current Localization.** | read-only |
| Security | Enumeration:<br>(1) Other<br>(2)Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | Indicates that this device has some security associated with it. | read-only |

## PRINTER OUTPUT DIMENSION ATTRIBUTES EXTENDED -OPTIONAL

*Table 45 Output Device Dimensions Extended*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Output Dims Extended | Augments Output Dim: Sequence of Attributes | A list of information suitable for extended output device management. | read-only |
| Dim Unit | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Ten Thousandths of Inches<br>(4) Micrometers | The unit of measure for specifying dimensional values for this output device. | read-only |
| Max Dim Feed Dir | Integer<br>(-2) Unknown | The maximum dimensions supported by this output device for measurements taken parallel relative to the feed direction associated with that device. If this device can reliably sense this value, the value is sensed by the printer and is read-only; otherwise the value may be written by a management or control panel application. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |
| Max Dim XFeed Dir Direction | Integer<br>(-1) Other<br>(-2) Unknown | The maximum dimensions supported by this output device for measurements taken ninety degrees relative to the feed direction associated with that device in DIM Units. If this device can reliably sense this value, the value is sensed by the printer and is read-only; otherwise the value may be written by a management or control panel application. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |
| Min Dim Feed Dir | Integer<br>(-1) Other<br>(-2) Unknown | The minimum dimensions supported by this output device for measurements taken parallel relative to the feed direction associated with that device. If this device can reliably sense this value, the value is sensed by the printer and is read-only; otherwise the value may be written by a management or control panel application. The value (-1) means other and | read-write |

| | | | |
|---|---|---|---|
| | | specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | |
| Min Dim XFeed Dir Direction | Integer<br>(-1) Other<br>(-2) Unknown | | read-write |

## PRINTER OUTPUT FEATURE ATTRIBUTES EXTENDED -OPTIONAL

*Table 46 Printer Output Features*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Output Device Feature Extended | Augments Printer Output Device : Sequence of Attributes | The Output Features objects extend the Output Device Table with standardized and commonly useful, but possibly optional objects not required of a minimally conforming implementation. | read-only |
| Stacking Order | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) First to Last<br>4) Last to First | The current state of the stacking order for this output device. First To Last means that as pages are output, the front of the next page is placed against the back of the previous page. Last to first means as pages are output, the back of the next page is placed against the front of the | read-write |
| Page Delivery Orientation | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Face Up<br>(4) Face Down | The reading surface that will be up when pages are delivered to the output device. Values are Face Up and Face Down. | read-write |
| Bursting | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that the output unit supports bursting, and if so, whether the feature is enabled. Bursting is the process by which continuous media is separated into individual sheets typically by bursting along pre-formed perforations. | read-write |
| Decollating | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that the output device supports decollating, and if so, whether the feature is enabled. Decollating is the process by which the parts of a multi-part form are separated and sorted into separate stacks for each part. | read-write |
| Page Collated | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that the output device supports page collation, and if so, whether the feature is enabled. | read-write |
| Offset Stacking | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that the output device supports offset stacking, and if so, whether the feature is enabled. | read-write |

## FINISHER DEVICE TABLE -OPTIONAL

*Table 47 Finisher Device Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Finisher Device Table | Sequence of Finisher Device Entries | A list of finishing devices installed inline in the print path of the printer. This is a common group defined by the DMTF Finisher Work Group. See the Finisher Device Table description in the Inserter component for details of these entries. | read-only |

## PRINTER ALERT TABLE

*Table 48 Alert Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Alert Table | List of Active Printer Alerts | The alert table lists all the critical and non-critical alerts currently active in the printer. A critical alert is one that stops the printer from printing immediately and can not continue until the critical alert condition is eliminated. Non-critical alerts are those items that do not stop printing but may at some future time. The table contains information on the severity, component, detail location within the component, alert code and description of each critical alert that is currently active within the printer.<br><br>There is one printer alert table per printer component. The format of alert tables and the rules for their management are described in **Appendix A: Alerts**. | read-only |

# INSERTER CLASS OBJECTS

The standard group definitions defined in this section are used to model the manageable attributes of devices such as finsishers, insert feeders, envelope feeders, and various output devices when they are inline in the production assembly path of an inserter component.

## INSERTER MODEL

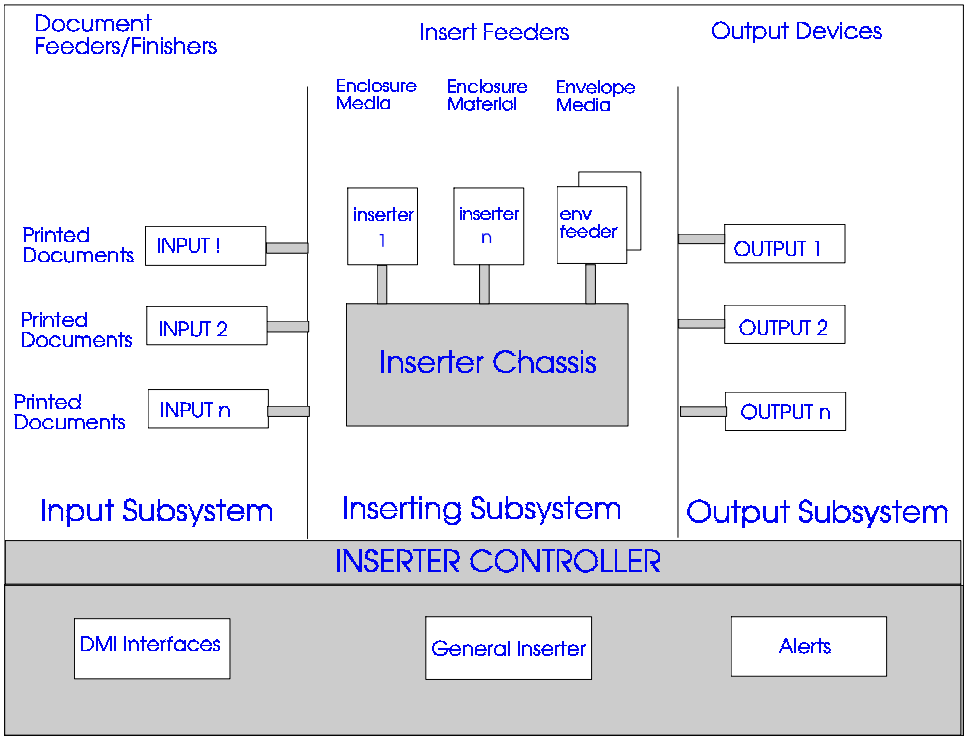



Figure 6. Inserter Model. The LMO standards group has developed the above model which is useful in understanding the classifications of the various objects making up an inserter component definition:

### Table 43 Inserter Model Functions

This table characterizes the types of functions performed by devices managed in each of the logical subsystem groupings of an inserter component.

*Table 49 Inserter Model Functions*

| Input Subsystem | Inserting Subsystem | Output Subsystem |
|---|---|---|
| Document Feeding | Material Merge | Envelope Sealing |
| Document Finishing | Collation | Postal Metering |
| Document Imprinting | Envelope Insertion | Bar Code Printing |
| Document Accumulation | | Traying / Stacking |
| Exception Diverts | | Exception Diverts |

## INSERTER COMPONENTID GROUP -MANDATORY

*Table 50 Inserter ComponentId Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| ComponentID | DisplayString(64) | This is a required group for all MIF components. It contains required attributes that must be specified for each component installed in the Management Information Format data base.<br><br>The ComponentID group is the first group in a DMTF component definition and must be assigned the Group Id value of 1.<br><br>The class value for this group is: DMTF\|ComponentId\|1.0. | read-only |

*Table 51 Inserter ComponentId Group Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Manufacturer | DisplayString(64) | The name of the organization that produces this component. | read-only |
| Product | DisplayString(64) | The name of this component or product. | read-only |
| Version | DisplayString(64) | The version string definition for this component. | read-only |
| Serial Number | DisplayString(64) | The serial number for this component. | read-only |
| Installation | Date | The time and date of the last install of this component on this system. | read-only |
| Verify | Enumeration | Verification level for this component. Asking for the verify attribute causes the component instrumentation code to perform checks to verify that the component is still in the system and working properly. It should return one of the following values:<br>(0) An error occurred; Check status code.<br>(1) Component does not exist.<br>(2) Verify not supported.<br>(4) Component exists, functionally untested.<br>(5) Component exists, functionally unknown.<br>(6) Component exists, functionally no good.<br>(7) Component exists, functionally good. | read-only |

## INSERTER GENERAL GROUP:

*Table 52 Inserter General Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| General | DisplayString(64) | This group specifies general attributes that apply to the overall inserter and its states. | read-only |

*Table 53 Inserter General Group Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Config Changes | Counter | A count of the configuration changes for this inserter. This value is incremented each time a configuration change is made to this inserter component. This value may be compared against a previous read value to determine if any configuration changes have been made to the inserter.<br><br>Management applications may cache infrequently changed configuration information about sub-units on the inserter. This object should be incremented whenever the agent wishes such applications to invalidate that cache and re-download all of this configuration information, thereby signally a change of inserter configuration.<br><br>Configuration changes represent changes in capabilities. These changes represent modifications which may affect the set of jobs that can be handled by a system, increasing or decreasing the likelihood of successful inserting. Examples might include the addition or removal of finisher devices, envelope feeders and output devices among other items.<br><br>For example, if an envelope feeder that contained stock of different dimensions was added, this counter should be incremented. As an additional example, this counter would not be incremented when the level of input is changed. | read-only |
| Current Localization | Integer | This value is an index into the inserter localization table specifying the current language, country and character set used in description string values for objects and attributes. | read-write |
| Status | Enumeration: | This value indicates the overall status of this inserter component. The values have the following meanings:<br>(1) Other<br>(2) Unknown<br>(3) Ready Available<br>(4) Critical<br>(5) NonCritical<br>(6) Both Critical and NonCritical<br>(7) On Request<br>(8) Unavailable<br>(9) Busy Temporarily Unavailable | read-only |

## INSERTER LOCALIZATION TABLE

*Table 54 Inserter Localization Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Localization Table | Sequence of Localization Entries | This table identifies the natural language, country and character sets supported by the inserter. | read-only |

*Table 55 Inserter Localization Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used to identify a specific entry in the localization table. Although these values may change (e.g. the addtion of new character sets to the inserter), values are expected to remain stable over successive power cycles. | read-only |
| Language | String(2) | A two character language code defined by ISO639. | read-only |
| Country | String(2) | A two character country code defined by ISO3166. | read-only |
| Localization Character Set | CodedCharSet | The coded character set used for this localization. Registered CodedCharSet values are defined by DMTF Work Groups consistent with ISO standards. | read-only |

## FINISHER DEVICE TABLE -MANDATORY

*Table 56 Finisher Device Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Finisher Device Table | Sequence of Finisher Device Entries | A list of document finisher devices that accept printed media resources and produce sets of pages for insertion into envelopes by this inserter including information regarding their identity, possible and current configuration, and status. | read-only |

*Table 57 Document Finisher Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by this inserter to identify an associated document finisher device. Although these values may change due to a major reconfiguration of the inserter (e.g., the addition of new devices to the inserter) values are expected to remain stable across successive powercycles. | read-only |
| Name | DisplayString(63) | The name assigned to this document finisher device | read-only |
| Vendor Name | DisplayString(63) | The name of the vendor of this device. | read-only |
| Model | DisplayString(63) | The model name of this device. | read-only |
| Serial Number | DisplayString(63) | The serial number assigned to this device. | read-only |
| Description | DisplayString(255) | A free form text description of this document finishing device. | read-only |
| Status | Enumeration: | This value indicates the overall status of the | read-write |

| | | | |
|---|---|---|---|
| | | finisher device. The enumeration values have the following meanings:<br>(1) Other<br>(2) Unknown<br>(3) Available<br>(4) Unavailable Critical Alert<br>(5) Available Non Critical Alert<br>(6) Unavailable Both Critical and Non Critical Alerts<br>(7) On Request<br>(8) Unavailable<br>(9) Busy Temporarily Unavailable | |

## FINISHER INPUT TABLE -MANDATORY

*Table 58 Finisher Input Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Input Table | Sequence of Input Entries | A list of input devices capable of providing printing media resources to this finisher including information regarding their identity, possible and current configuration, and status. | read-only |

*Table 59 Finisher Input Device Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | The index of this input for the finisher device to which this input device is associated. | read-write |
| Finisher Index | Integer | The index of the document finisher device to which this input device is associated. | read-write |
| Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Sheet Auto Removable Tray<br>(4) Sheet Feed Non Auto Removable Tray<br>(5) Manual Feed<br>(6) Continuous Roll<br>(7) Continuous Fan Fold<br>(8) Continuous Webb | This value indicates the type of technology (discriminated primarily by feeder mechanism type) employed by this finisher input device. | read-write |
| Feed Control Mode | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Feed Only<br>(4) Read and Feed | This attribute contains values used to control the feed mode for this finisher input media.<br>*Feed-only*- This value is used to select the normal feeding mode for the device. No additional information is needed to control the feeding process.<br>*Read-and-feed*- This value indicates that feeding is controlled by scanning information on the input media. When this mode is selected, the reader devices associated with this input entry define the criteria that must be met to trigger the feeder operation. | read-write |
| Dim Unit | Enumeration: | The unit of measure for specifying | read-write |

| | | | |
|---|---|---|---|
| | (1) Other<br>(2) Unknown<br>(3) Ten Thousandths of Inches<br>(4) Micrometers | dimensional values for this finisher input device. | |
| Media Dim Feed Dir | Integer<br>(-1) Other<br>(-2) Unknown | This object provides the value of the dimension in the feed direction of the media that is placed or will be placed in this input device. Feed dimension measurements are taken parallel to the feed direction of the device and measured in **Dim Units**. The value (-1) means *other* andspecifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |
| Media Dim XFeed Dir | Integer<br>(-1) Other<br>(-2) Unknown | This object provides the value of the dimension in the cross feed direction of the media that is placed or will be placed in this input device. The cross feed direction is ninety degrees relative to the feed direction on this device. The dimension is measured in Dim Units. The value (-1) means *other and* specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |
| Capacity Units | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Ten Thousands of Inches<br>(4) Micrometers<br>(5) Sheets<br>(16) Feet<br>(17) Meters | The unit of measure for specifying capacity values for this finisher input device. | read-write |
| Max Capacity | Integer<br>(-1) Other<br>(-2) Unknown | The maximum capacity of the input source in input **capacity units**. There is no convention associated with the media itself so this value is claimed capacity. If the input device can reliably sense this value, the value is sensed by the finisher and is read-only. Otherwise the value may be written by a management or control console application. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |
| Current Level | Integer<br>(-1) Other<br>(-2) Unknown | The current capacity of the input source in input Capacity units. If the input device can reliably sense this value, the value is sensed by the finisher and is read-only. Otherwise the value may be written by a management or control console application. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |
| Status | Enumeration: | This value indicates the specific status of this finisher input device. The values have the following meanings:: | read-write |

| | | | |
|---|---|---|---|
| | | (1) Other<br>(2) Unknown<br>(3) Available<br>(4) Unavailable Critical Alert<br>(5) Available Non Critical Alert<br>(6) Unavailable Both Critical and Non Critical Alerts<br>(7) On Request<br>(8) Unavailable<br>(9) Busy Temporarily Unavailable | |
| Media Name | String(63) | The name of the current media contained in this input device. Examples are Print Job Output A and Triple A Billing Statements. | read-write |

## FINISHER INPUT TYPE ATTRIBUTES EXTENDED -OPTIONAL

*Table 60 Finisher Input Device Type Extended*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Input Type Extended | Augments Device Type Sequence of Attributes | A list of information suitable for extended input device management. | read-only |
| Name | DisplayString(63) | The name assigned to this input device. | read-only |
| Vendor Name | DisplayString(63) | The name of the vendor of this device. | read-only |
| Model | DisplayString(63) | The model name of this input device. | read-only |
| Version | DisplayString(63) | The version string for this input device. | read-only |
| Serial Number | DisplayString(63) | The serial number assigned to this input device. | read-only |
| Description | DisplayString(255) | A free form text description of this input device. | read-only |
| Security | Enumeration:<br>(1) Other<br>(2) Reserved<br>(3) On<br>(4) Off<br>(5) Not Present | Indicates whether this device has some security associated with it, and. if so, whether it is on or off. | read-write |

## FINISHER INPUT MEDIA ATTRIBUTES EXTENDED -OPTIONAL

*Table 61 Finisher Input Media Extended*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Input Media Extended | Augments Media Type: Sequence of Attributes | A list of attributes used to specify media associated with an input device. | read-only |
| Media Weight | Integer<br>(-2) Unknown | The weight of the media associated with this input device in grams per meter squared. The value (-2) indicates unknown. This object can be used to calculate weight of an individual pages processed by the document finisher. This value, when multiplied by the number of pages in a finished set, can be used to determine the total weight of a set before it is inserted into the mailing envelope. | read-write |
| Media Thickness | Integer<br>(-2) Unknown | This object identifies the thickness of the input media processed by this document | read-write |

| | | | |
|---|---|---|---|
| | | finisher measured in micrometers. This value may be used by devices (or operators) to setup proper machine tolerances for the feeder operation. The value (-2) is used when the media thickness is unknown or not used in the setup for the finisher. | |
| Media Type | DisplayString(63) | The name of the type of medium associated with this input device. Standard values are standardized strings from ISO 10175 (DPA) and ISO10180 (SPDL) which are:<br>stationary<br>transparency<br>envelope<br>envelope-plain<br>envelope-window<br>continuos-long<br>continuos-short<br>tab-stock<br>labels<br>multi-layer<br>Additional strings with names that do not conflict with standard names may also be used. | read-write |
| Media Color | DisplayString(63) | The name of the color of the medium associated with this input device. Standard values are standardized strings from ISO 10175 (DPA) and ISO10180 (SPDL) which are:<br>white,<br>pink,<br>yellow,<br>buff,<br>goldenrod,<br>blue green,<br>transparent.<br>Additional strings that do not conflict with standard names may also be specified. | read-write |
| Medium Forms Parts | Integer<br>(-2) Unknown | The number of parts associated with this medium if it is a multi-part form. The value (-2) indicates unknown. | read-write |

## FINISHING FEATURES TABLE

*Table 62 Finishing Features Finishing Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Finishing Feature Table | Sequence of Finishing Features Entries | A list of finishing features associated with the processing of input media including information regarding functions supported, possible configuration options and status for each input source. | read-only |

*Table 63 Finishing Feature Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|

| | | | |
|---|---|---|---|
| Finisher Index | Integer | The index of the finisher device entry to which this set of finishing feature functions apply. | read-write |
| Input Source Index | Integer | The index of the finisher input entry to which this set of finishing feature functions apply. | read-write |
| Stitching | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that stitching is supported as a finishing feature option for this input source and whether the feature is enabled. | read-write |
| Stitching Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Staple<br>(4) Staple Top Left<br>(5) Staple Bottom Left<br>(6) Staple Top Right<br>(7) Staple Bottom Right<br>(8) Saddle Stitch<br>(9) Edge Stitch<br>10) Edge Stitch Right<br>(11) Edge Stitch Top<br>(12) Stitch | This object indicates the type of stitching that is presently enabled for page sets in the input media source associated with this entry. **Staple** indicates the staple type is the finisher defined stapling default; stitch indicates the stitch type is the finisher defined stitching default. | read-write |
| Max Pages Stitched | Integer | This object specifies the maximum number of pages that can be in a page set to be stitched by this finishing device. This value may be used for device setups and/or bypassing (diverting) stitching operations for sets that are too large to be handled. | read-write |
| Folding | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that folding is supported as a finishing feature option for this input source and whether the feature is enabled. | read-write |
| Folding Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Z Fold<br>(4) ½ Fold<br>(5) Letter Fold | This object indicates the type of folding that is presently enabled for page sets in the input media source associated with this entry. | read-write |
| Max Pages Folded | Integer | This object specifies the maximum number of pages that can be in a page set to be folded by this finishing device. This value may be used for device setups and/or bypassing (diverting) folding operations for sets that are too large to be handled. | read-write |
| Binding | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that binding is supported as a finishing feature option for this input source and whether the feature is enabled. | read-write |
| Binding Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Bind | This object indicates the type of binding that is presently enabled for page sets in the input media source associated with this entry. **Bind** indicates the bind type is the finisher defined | read-write |

| | | | |
|---|---|---|---|
| | (4) Tape<br>(5) Plastic<br>(6) Velo<br>(7) Perfect<br>(8) Spiral | binding default. | |
| Max Pages Bound | Integer | This object specifies the maximum number of pages that can be in a page set to be bound by this finishing device. This value may be used for device setups and/or bypassing (diverting) binding operations for sets that are too large to be handled. | read-write |
| Binding Color | DisplayString(63) | The name of the color of the binding associated with this finishing feature. Standard values are standardized strings from ISO 10175 (DPA) and ISO10180 (SPDL) which are:<br>other,<br>unknown,<br>white,<br>red,<br>green,<br>blue,<br>pink,<br>yellow,<br>black,<br>buff,<br>goldenrod.<br>Additional strings with names that do not conflict with standard names may also be used. The naming conventions in ISO Standards are recommended to avoid potential name clashes. | read-write |
| Trimming | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that trimming is supported as a finishing feature option for this input source and whether the feature is enabled. | read-write |
| Trimming Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(13) Trim | This object indicates the type of trimming that is presently enabled for page sets in the input media source associated with this entry. Trim indicates the trim type is the finisher defined trimming default. | read-write |
| Max Pages Trimmed | Integer | This object specifies the maximum number of pages that can be in a page set to be timed by this finishing device. This value may be used for device setups and/or bypassing (diverting) trimming operations for sets that are too large to be handled. | read-write |
| Die Cutting | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that die cutting is supported as a finishing feature option for this input source and whether the feature is enabled. | read-write |
| Die Cutting Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(14) Die Cut | This object indicates the type of die cutting that is presently enabled for page sets in the input media source associated with this entry. **Die Cut** indicates the die cut type is | read-write |

| | | | |
|---|---|---|---|
| | | the finisher defined die cut default. | |
| Max Pages Die Cut | Integer | This object specifies the maximum number of pages that can be in a page set to be die cut by this finishing device. This value may be used for device setups and/or bypassing (diverting) die cut operations for sets that are too large to be handled. | read-write |
| Punching | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that punching is supported as a finishing feature option for this input source and whether the feature is enabled. | read-write |
| Punch Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(10) Punch | This object indicates the type of punching that is presently enabled for page sets in the input media source associated with this entry. **Punch** indicates the punch type is the finisher defined punching default. | read-write |
| Max Pages Punched | Integer | This object specifies the maximum number of pages that can be in a page set to be punched by this finishing device. This value may be used for device setups and/or bypassing (diverting) die cut operations for sets that are too large to be handled. | read-write |
| Perforating | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that perforating is supported as a finishing feature option for this input source and whether the feature is enabled. | read-write |
| Perforation Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(10) Perforate | This object indicates the type of perforation that is presently enabled for pages in the input media source associated with this entry. **Perforate** indicates the perforation type is the finisher defined perforating default. | read-write |
| Slitting | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that slitting is supported as a finishing feature option for this input source and whether the feature is enabled. | read-write |
| Slitting Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Slit<br>(4) Slit and Separate<br>(5) Slit and Merge | This object indicates the type of slitting operation that is presently enabled for the input media associated with this entry. **Slit** indicates the slitting type is the finisher defined slitting default. | read-write |
| Separation Cut | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that separation cuts are supported as a finishing feature option for this input source and whether the feature is enabled. | read-write |
| Separation Cut Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(30) Separation Cut | This object indicates the type of separation cut that is presently enabled for the input media associated with this entry. **Separation Cut** indicates the separation cut type is the finisher defined separation cut default. | read-write |

| | | | |
|---|---|---|---|
| Imprinting | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that imprinting is supported as a finishing feature option for this input source and whether the feature is enabled. | read-write |
| Imprinting Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(30) Imprinting | This object indicates the type of imprinting cut that is presently enabled for the input media associated with this entry. Imprinting indicates the imprinting type is the finisher defined imprinting default. | read-write |
| Wrapping | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that wrapping is supported as a finishing feature option for this input source and whether the feature is enabled. | read-write |
| Wrapping Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Wrap<br>(4) Shrink Wrap<br>(5) Paper Wrap | This object indicates the type of wrapping that is presently enabled for page sets in the input media source associated with this entry. Wrap indicates the wrapping type is the finisher defined wrapping default. | read-write |
| Max Pages Wrapped | Integer | This object specifies the maximum number of pages that can be in a page set to be wrapped by this finishing device. This value may be used for device setups and/or bypassing (diverting) wrapping operations for sets that are too large to be handled. | read-write |
| Banding | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that banding is supported as a finishing feature option for this input source and whether the feature is enabled. | read-write |
| Banding Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Band | This object indicates the type of banding that is presently enabled for page sets in the input media source associated with this entry. **Band** indicates the banding type is the finisher defined banding default. | read-write |
| Max Pages Banded | Integer | This object specifies the maximum number of pages that can be in a page set to be banded by this finishing device. This value may be used for device setups and/or bypassing (diverting) wrapping operations for sets that are too large to be handled. | read-write |

## FINISHER OUTPUT DEVICE TABLE -MANDATORY

*Table 64 FinisherOutput Device Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Output Table | Sequence of Ouput Device Entries | A list of output devices capable of accepting and delivering documents from finishing processes associated with this finisher including information regarding their identity, | read-only |

| | | | |
|---|---|---|---|
| | | possible and current configuration, and status. | |

Table 65Finisher Output Device Entry

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by this finisherfinisher to identify an associated output device. Although these values may change due to a major reconfiguration of the device (e.g., the addition of new output devices to thefinisher), values are expected to remain stable across successive power cycles. | read-only |
| Finisher Index | Integer | The value of the **Finisher Index** corresponding to the Finisher device with which this output device is associated.. | read-only |
| Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Removable Bin<br>(4) Unremoveable Bin<br>(5) Continuous Roll Device<br>(6) Mailbox<br>(7) Continuous Fan Fold<br>(8) Conveyer<br>(9) Smart Cart | The type of technology employed by this output device. | read-only |
| Capacity Unit | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Ten Thousandths of Inches<br>(4) Micrometers<br>(8) Sheets<br>(16) Feet<br>(17) Meters | The unit of measure for specifying capacity values for this device. | read-only |
| Max Capacity | Integer,<br>(-1) Other<br>(-2) Unknown | The maximum capacity of this device in Output Capacity Units. There is no convention associated with the media itself so this value is claimed capacity. If the device can reliably sense this value, the value is sensed by the finisher and is read-only; otherwise the value may be written by management or control panel applications. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |
| Remaining Capacity | Integer | The remaining capacity of the possible capacity in **Output Capacity units**. If the device can reliably sense this value, the value is sensed by the finisher and is read-only; otherwise the value may be written by management or control panel applications. The value (-1) means other and specifically indicates that the device places no | read-write |

| | | | |
|---|---|---|---|
| | | restriction on this parameter. The value (-2) indicates: | |
| Status | Enumeration | This value indicates the specific state of this output device:<br>(1) Other<br>(2) Unknown<br>(3) Available<br>(4) Unavailable Critical Alert<br>(5) Available Non Critical Alert<br>(6) Unavailable Both Critical and Non Critical Alerts<br>(7) On Request<br>(8) Unavailable<br>(9) Busy Temporarily Unavailable | read-only |

## FINISHER OUTPUT TYPE ATTRIBUTE EXTENDED -OPTIONAL

*Table 66Finisher Output Device Type Extended*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Output Device Type Extended | Augments Output Type: Sequence of Attributes | A list of information suitable for extended output device management. | read-only |
| Name | DisplayString(63) | The name assigned to this output device. | read-only |
| Vendor Name | DisplayString(63) | The name of the vendor of this device. | read-only |
| Model | DisplayString(63) | The model name of this output device. | read-only |
| Version | DisplayString(63) | The version string for this output device. | read-only |
| Serial Number | DisplayString(63) | The serial number assigned to this Output device. | read-only |
| Description | DisplayString(255) | A free form text description of this output device in the localization specified by **General Current Localization.** | read-only |
| Security | Enumeration:<br>(1) Other<br>(2)Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | Indicates that this device has some security associated with it. | read-only |

## FINISHER OUTPUT DIMENSION ATTRIBUTES EXTENDED -OPTIONAL

*Table 67 Finisher Output Device Dimensions Extended*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Output Dims Extended | Augments Output Dim Sequence of Attributes | A list of information suitable for extended output device management. | read-only |
| Dim Unit | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Ten Thousandths of Inches<br>(4) Micrometers | The unit of measure for specifying dimensional values for this output device. | read-only |
| Max Dim Feed Dir | Integer<br>(-1) Other<br>(-2) Unknown | The maximum dimensions supported by this output device for measurements taken parallel relative to the feed direction associated with | read-write |

| | | | |
|---|---|---|---|
| | | that device. If this device can reliably sense this value, the value is sensed by the finisher and is read-only; otherwise the value may be written by a management or control panel application. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | |
| Max Dim XFeed Dir Direction | Integer<br>(-1) Other<br>(-2) Unknown | The maximum dimensions supported by this output device for measurements taken ninety degrees relative to the feed direction associated with that device in DIM Units. If this device can reliably sense this value, the value is sensed by the finisher and is read-only; otherwise the value may be written by a management or control panel application. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |
| Min Dim Feed Dir | Integer<br>(-1) Other<br>(-2) Unknown | The minimum dimensions supported by this output device for measurements taken parallel relative to the feed direction associated with that device. If this device can reliably sense this value, the value is sensed by the finisher and is read-only; otherwise the value may be written by a management or control panel application. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |
| Min Dim XFeed Dir Direction | Integer<br>(-1) Other<br>(-2) Unknown | The minimum dimensions supported by this output device for measurements taken ninety degrees relative to the feed direction associated with that device in **DIM Units**. If this device can reliably sense this value, the value is sensed by the finisher and is read-only; otherwise the value may be written by a management or control panel application. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |

## FINISHER OUTPUT FEATURES ATTRIBUTES EXTENDED -OPTIONAL

*Table 68 Finisher Output Features Extended*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Output Device Feature Extended | Augments Printer Output Device Table Sequence of Attributes | The Output Features objects extend the Output Device Table with standardized and commonly useful, but possibly optional objects not required of a minimally conforming implementation. | read-only |
| Stacking Order | Enumeration:<br>(1) Other | The current state of the stacking order for this output device. First To Last means that as | read-write |

| | | | |
|---|---|---|---|
| | (2) Unknown<br>(3) First to Last<br>4) Last to First | pages are output, the front of the next page is placed against the back of the previous page. Last to first means as pages are output, the back of the next page is placed against the front of the | |
| Page Delivery Orientation | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Face Up<br>(4) Face Down | The reading surface that will be up when pages are delivered to the output device. Values are Face Up and Face Down. | read-write |
| Bursting | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that the output unit supports bursting, and if so, whether the feature is enabled. Bursting is the process by which continuous media is separated into individual sheets typically by bursting along pre-formed perforations. | read-write |
| Decollating | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that the output device supports decollating, and if so, whether the feature is enabled. Decollating is the process by which the parts of a multi-part form are separated and sorted into separate stacks for each part. | read-write |
| Page Collated | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that the output device supports page collation, and if so, whether the feature is enabled. | read-write |
| Offset Stacking | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | This object indicates that the output device supports offset stacking, and if so, whether the feature is enabled. | read-write |

## FINISHER SUPPLY TABLE

*Table 69 Finisher Supply Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Finisher Supply Table | Sequence of Supply Entries | A list of supplies used, their description, capacities and current level. | read-only |

*Table 70 Finisher Supply Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by the finisher to identify this finishing supply container or receptacle. Although these values may change due to a major reconfiguration of the finisher (e.g., the addition of new finishing supplies to the finisher), values are expected to remain stable across successive power cycles. | read-write |
| Finisher Index | Integer | The value of the **Finisher Index** corresponding to the finishing device with | read-write |

| | | | |
|---|---|---|---|
| | | which this marker supply entity is associated. | |
| Colorant Index | Integer | The value of the Colorant Index corresponding to the colorant with which this finishing supply entity is associated. The value shall be 0 if there is no Finisher Colorant Table. | read-write |
| Class | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Supply that is consumed<br>(4) Receptacle that is filled | This value indicates whether this supply entity represents a supply container that is consumed or a receptacle that is filled. | read-write |
| Type | Enumeration:<br>(1) other<br>(2) unknown<br>(3) toner<br>(4) waste toner<br>(5) ink<br>(6) ink cartridge<br>(7) ink ribbon<br>(8) waste ink<br>(9) OPC (organic photo conductor)<br>(10) developer<br>(11) fuser oil<br>(12) solid wax<br>(13) ribbon wax<br>(14) waste wax<br>(15) water<br>(16) waste water<br>(17) glue water additive<br>(18) waste paper<br>(19) binding tape<br>(20) banding tape<br>(21) stitching wire<br>(22) shrink wrap<br>(23) paper wrap | This value identifies the supply type of this supply. | read-write |
| Description | String(255) | The description of this supply container or receptacle in text useful for operators and management applications. | read-write |
| Supply Unit | Enumeration:<br>(1) other<br>(2) unknown<br>(3) ten thousandths of inches<br>(4) micrometers<br>(12) thousandths of ounces<br>(13) tenths of grams<br>(14) hundreds of fluid ounces<br>(15) tenths of milliliters | The unit of measure of this finisher supply container or receptacle. | read-write |
| Max Capacity | Integer (-1) Other<br>(-2) Unknown | The maximum capacity of this supply container or receptacle is expressed in Supply Units. If this supply container/ receptacle can reliably sense this value, the value is sensed | read-write |

| | | | |
|---|---|---|---|
| | | by the finisher and is read-only; otherwise the value may be written by management or control panel application. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | |
| Current Level | Integer<br>(-1) Other<br>(-2) Unknown | The current level if this supply entity is a container; the remaining space if this supply entity is a receptacle. If this supply container/ receptacle can reliably sense this value, the value is sensed by the finisher and is read-only; otherwise the value may be written by management or control panel application. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |

## FINISHER COLORANT TABLE

*Table 71 Finisher Colorant Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Finisher Colorant Table | Sequence of Colorant Entries | A list of all the colorants available on the Finisher. | read-only |

*Table 72 Finisher Colorant Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by the finisher to identify this finisher colorant. Although these values may change due to a major reconfiguration of the finisher (e.g., the addition of new colorants to the finisher), values are expected to remain stable across successive finisher power cycles. | read-write |
| Finisher Index | Integer | The value of the **Finisher Index** corresponding to the finisher device with which this colorant entry is associated. | read-write |
| Role | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Process<br>(4) Spot | The role played by this colorant. | read-write |
| Value | String(63) | The name of the color of this colorant using standardized string names from ISO 10175 (DPA)and ISO 10180 (SPDL) which are:<br>other,<br>unknown,<br>white,<br>red,<br>green,<br>blue,<br>cyan, | read-write |

| | | | |
|---|---|---|---|
| | | magenta,<br>yellow and<br>black.<br>Implementors may add additional string values. The naming conventions in ISO 9070 are recommended in order to avoid potential name clashes. | |
| Tonality | Integer | The distinct levels of tonality realizable by a marking device when using this colorant. This value does not include the number of levels of tonal difference that an interpreter can obtain by techniques such as half toning. This value must be at least 2. | read-write |

## INSERT FEEDER TABLE -MANDATORY

*Table 73 Insert Feeder Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Insert Feeder Table | Sequence of Insert Feeder Devices | A list of feeder devices that select printed media from an input source for insertion along with other documents in the envelope for mailing. Attributes of inserter feeder devices include information regarding their identity, possible and current configuration, and status. | read-only |

*Table 74 Insert Feeder Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by this inserter to identify an associated insertfeeder device. Although these values may change due to a major reconfiguration of the inserter (e.g., the addition of new feeder devices to the inserter) values are expected to remain stable across successive powercycles. | read-write |
| Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Stream Feeder<br>(4) Gripper (5) Rotary | This value indicates the type of technology discriminated primarily by feeder mechanism type employed by the device. rotary | read-write |
| Feed Control Mode | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Feed Only<br>(4) Read and Feed | This attribute contains values used to control the feed mode for this insert feeder.<br>feed-only- This value is used to select the normal feeding mode for the device. No additional information is needed to control the feeding process.<br>read-and-feed- This value indicates that feeding is controlled by scanning information on the input media. When this mode is selected, the Reader devices associated with this entry define the criteria that must be met to trigger the feeder operation. | read-write |
| Dim Unit | Enumeration:<br>(1) Other | The unit of measure for specifying dimensional values for this insert feeder | read-write |

| | | | |
|---|---|---|---|
| | (2) Unknown<br>(3) Ten Thousandths of Inches<br>(4) Micrometers | device. | |
| Media Dim Feed Dir | Integer<br>(-1) Other<br>(-2) Unknown | This object provides the value of the dimension in the feed direction of the media that is placed or will be placed in this insert device. Feed dimension measurements are taken parallel to the feed direction of the device and measured in **Dim Units**. The value<br>(-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |
| Media Dim XFeed Dir | Integer<br>(-1) Other<br>(-2) Unknown | This object provides the value of the dimension in the cross feed direction of the media that is placed or will be placed in this insert device. The cross feed direction is ninety degrees relative to the feed direction on this device. The dimension is measured in Dim Units. The value (-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |
| Capacity Units | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Ten Thousands of Inches<br>(4) Micrometers<br>(5) Sheets | The unit of measure for specifying capacity values for this insert feeder device. | read-write |
| Max Capacity | Integer.<br>(-1) Other<br>(-2) Unknown | The maximum capacity of the input source in input **capacity units**. There is no convention associated with the media itself so this value is claimed capacity. If the input device can reliably sense this value, the value is sensed by the inserter and is read-only. Otherwise the value may be written by a management or control console application. The value<br>(-2) indicates unknown. | read-write |
| Current Level | Integer<br>(-1) Other<br>(-2) Unknown | The current capacity of the input source in input **capacity units**. If the input device can reliably sense this value, the value is sensed by the inserter and is read-only. Otherwise the value may be written by a management or control console application. The value (-2) indicates unknown. | read-write |
| Status | Enumeration | This value indicates the specific status of this input device. The values have the following meanings:<br>(1) Other<br>(2) Unknown<br>(3) Available<br>(4) Unavailable Critical Alert<br>(5) Available Non Critical Alert<br>(6) Unavailable Both Critical and Non Critical Alerts<br>(7) On Request | read-write |

| | | | |
|---|---|---|---|
| | | (8) Unavailable<br>(9) Busy Temporarily Unavailable | |
| Media Name | String(63) | The name of the insert media contained in this feeder device. Examples are Sales Discount Coupon and Proxy Voter Card. | read-write |
| Action On Misses | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Ignore on Miss<br>(4) Divert on Miss | This object specifies the action taken by the inserter when a miss is detected by the insert feeder. This value may be used to insure insertion of critical media for the mailing. | read-write |
| Action On Doubles | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Ignore on Doubles<br>(4) Divert on Doubles | This object specifies the action taken by the inserter when a feeding of multiple media sheets is detected by the insert feeder. This value may be used for quality control and to insure the total weight of the mailing piece does not exceed a postage rate threshold. | read-write |

## INSERT FEEDER TYPE ATTRIBUTE EXTENDED -OPTIONAL

*Table 75 Insert Feeder Type Extended*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Insert Feeder Type Extended | Augments Feeder Type Sequence of Attributes | A list of information suitable for extended input device management. | read-only |
| Name | DisplayString(63) | The name assigned to this input device. | read-only |
| Vendor Name | DisplayString(63) | The name of the vendor of this device. | read-only |
| Model | DisplayString(63) | The model name of this insert feeder device. | read-only |
| Version | DisplayString(63) | The version string for this insert feeder device. | read-only |
| Serial Number | DisplayString(63) | The serial number assigned to this insert feeder device. | read-only |
| Description | DisplayString(255) | A free form text description of this input feeder device. | read-only |
| Security | Enumeration:<br>(1) Other<br>(2) Reserved<br>(3) On<br>(4) Off<br>(5) Not Present | Indicates whether this device has some security associated with it, and. if so, whether it is on or off. | read-write |

## INSERT MEDIA ATTRIBUTE EXTENDED -OPTIONAL

*Table 76 Insert Media Extended*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Insert Media Extended | Augments Media Type Sequence of Attributes | A list of media attributes media associated with an insert feeder device. | read-only |
| Media Weight | Integer<br>(-2) Unknown | The weight of the media associated with this insert feeder in grams per meter squared. The value (-2) indicates unknown. This object can be used to calculate weight of individual | read-write |

| | | | |
|---|---|---|---|
| | | sheets processed by the insert feeder. | |
| Media Thickness | Integer<br>(-2) Unknown | This object identifies the thickness of the input media processed by this insert feeder measured in micrometers. This value may be used by devices (or operators) to setup proper machine tolerances for the feeder operation. The value (-2) is used when the media thickness is unknown or not used in the setup for the inserter. | read-write |
| Media Type | DisplayString(63) | The name of the type of medium associated with this input device. Recommended values are standardized strings from ISO 10175 (DPA) and ISO10180 (SPDL) where appropriate; otherwise descriptive strings may be specified by implementations. Examples are: stationary envelope flyer | read-write |
| Media Color | DisplayString(63) | The name of the color of the medium associated with this input device. Standard values are standardized strings from ISO 10175 (DPA) and ISO10180 (SPDL) which are:<br>white,<br>pink,<br>yellow,<br>buff,<br>goldenrod,<br>blue,<br>green.<br>Implementors may add additional string values. The naming conventions in ISO standards are recommended in order to advoid potential name clashes. | read-write |
| Medium Forms Parts | Integer<br>(-2) Unknown | The number of parts associated with this medium if it is a multi-part form. The value (-2) indicates unknown. | read-write |

## ENVELOPE FEEDER TABLE -MANDATORY

*Table 77 Envelope Feeder Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Envelope Feeder Table | Sequence of Envelope Feeder Devices | A list of feeder devices that select envelope media to be used for the insertion of mail piece bill of materials for mailing. Attributes of envelope feeders include information regarding their identity, possible and current configuration, and status. | read-only |

*Table 78 Envelope Feeder Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by this inserter to identify an associated envelope feeder device. Although these values may change due to a major reconfiguration of the inserter | read-write |

| | | | |
|---|---|---|---|
| | | (e.g., the addition of new feeder devices to the inserter) values are expected to remain stable across successive powercycles. | |
| Type | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Stream Feeder<br>(4) Gripper<br>(5) Rotary | This value indicates the type of technology discriminated primarily by feeder mechanism type employed by the device. rotary | read-write |
| Feed Control Mode | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Feed Only<br>(4) Read and Feed | This attribute contains values used to control the feed mode for this envelope feeder.<br>feed-only- This value is used to select the normal feeding mode for the device. No additional information is needed to control the feeding process.<br>read-and-feed- This value indicates that feeding is controlled by scanning information on the media to be inserted. When this mode is selected, the Reader devices associated with this entry define the criteria that must be met to trigger the feeder operation. | read-write |
| Dim Unit | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Ten Thousandths of Inches<br>(4) Micrometers | The unit of measure for specifying dimensional values for this envelope feeder device. | read-write |
| Media Dim Feed Dir | Integer<br>(-2) Unknown | This object provides the value of the dimension in the feed direction of the envelope that is placed or will be placed in this feeder device. Feed dimension measurements are taken parallel to the feed direction of the device and measured in **dim units**. The value (-2) indicates unknown. | read-write |
| Media Dim XFeed Dir | Integer<br>(-2) Unknown | This object provides the value of the dimension in the cross feed direction of the envelope that is placed or will be placed in this feeder device. The cross feed direction is ninety degrees relative to the feed direction on this device. The dimension is measured in **dim units**. The value (-2) indicates unknown. | read-write |
| Capacity Units | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Ten Thousands of Inches<br>(4) Micrometers<br>(15) Envelopes | The unit of measure for specifying capacity values for this insert feeder device. | read-write |
| Max Capacity | Integer .<br>(-1) Other<br>(-2) Unknown | The maximum capacity of the input source in input **capacity units**. There is no convention associated with the media itself so this value is claimed capacity. If the input device can reliably sense this value, the value is sensed by the inserter and is read-only. Otherwise the value may be written by a management or control console application. The value (-2) indicates unknown. | read-write |
| Current Level | Integer | The current capacity of the input source in | read-write |

| | | | |
|---|---|---|---|
| | (-1) Other<br>(-2) Unknown | input **capacity units**. If the input device can reliably sense this value, the value is sensed by the inserter and is read-only. Otherwise the value may be written by a management or control console application. The value (-2) indicates unknown. | |
| Status | Enumeration: | This value indicates the status of this envelope feeder device. The values have the following meanings:<br>(1) Other<br>(2) Unknown<br>(3) Available<br>(4) Unavailable Critical Alert<br>(5) Available Non Critical Alert<br>(6) Unavailable Both Critical and Non Critical Alerts<br>(7) On Request<br>(8) Unavailable<br>(9) Busy Temporarily Unavailable | read-write |
| Media Name | String(63) | The name of the insert media contained in this feeder device. Examples are XYZ Large and XYZ Letter. | read-write |
| Action On Misses | Enumeration:<br>(1) Other<br>(2) Unknown<br>(4) Divert on Miss | This object specifies the action taken by the inserter when a miss is detected by the envelope feeder. | read-write |
| Action On Doubles | Enumeration:<br>(1) Other<br>(2) Unknown<br>(4) Divert on Doubles | This object specifies the action taken by the inserter when a feeding of multiple envelopes is detected by the inserter. | read-write |

## ENVELOPE FEEDER TYPE ATTRIBUTE EXTENDED -OPTIONAL

*Table 79 Envelope Feeder Type Extended*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Insert Feeder Type Extended | Augments Envelope Feeder Type Sequence of Attributes | A list of information suitable for extended device management. | read-only |
| Name | DisplayString(63) | The name assigned to this envelope feeder device. | read-only |
| Vendor Name | DisplayString(63) | The name of the vendor of this envelope feeder device. | read-only |
| Model | DisplayString(63) | The model name of this envelope feeder device. | read-only |
| Version | DisplayString(63) | The version string for this envelope feeder device. | read-only |
| Serial Number | DisplayString(63) | The serial number assigned to this envelope feeder device. | read-only |
| Description | DisplayString(255) | A free form text description of this envelope feeder device. | read-only |
| Security | Enumeration:<br>(1) Other<br>(2) Reserved<br>(3) On | Indicates whether this device has some security associated with it, and. if so, whether it is on or off. | read-write |

| | | | |
|---|---|---|---|
| | (4) Off<br>(5) Not Present | | |

## ENVELOPE MEDIA ATTRIBUTE EXTENDED -OPTIONAL

*Table 80 Envelope Media Extended*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Envelope Media Extended | Augments Media Type Sequence of Attributes | A list of media attributes media associated with an envelope feeder device. | read-only |
| Media Weight | Integer<br>(-2) Unknown | The weight of the envelope associated with this insert feeder in grams per meter squared. This object can be used in the weight calculations for the mailpiece including its envelope. The value (-2) indicates unknown. | read-write |
| Media Thickness | Integer<br>(-2) Unknown | This object identifies the thickness of the envelope media processed by this envelope feeder measured in micrometers. This value may be used by devices (or operators) to setup proper machine tolerances for the feeder operation. The value (-2) is used when the media thickness is unknown or not used in the setup for the feeder. | read-write |
| Media Type | DisplayString(63) | The name of the type of medium associated with this input device. Recommended values are standardized strings from ISO 10175 (DPA) and ISO10180 (SPDL) where appropriate; otherwise descriptive strings may be specified by implementations. Examples are: envelope, envelope plain, envelope window, b5 envelope. | read-write |
| Media Color | DisplayString(63) | The name of the color of the envelope media associated with this feeder device. Valid values are standardized strings from ISO 10175 (DPA) and ISO10180 (SPDL) which are: white, pink, yellow, buff, goldenrod, blue, green. | read-write |

## INSERTER OUTPUT DEVICE TABLE -OPTIONAL

*Table 81 Inserter Output Device Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| OutputDevice Table | Sequence of Associated Output Device Entries | A list of output devices installed in the current mailpiece path of this inserter. | read-only |

*Table 82 Inserter Output DeviceEntry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by this inserter to identify an associated output device. | read-write |

| | | | |
|---|---|---|---|
| | | Although these values may change due to a major reconfiguration of the device (e.g., the addition of new devices to the inserter path) values are expected to remain stable across successive powercycles. | |
| Type | DisplayString(63) | A unique value used to identify the type of associated output device. Commonly used values will be registered in ISO standards. Proposed names include the following: postal-meter conveyer traying-device bar-code-sprayer | read-write |
| Capacity Unit | Enumeration: Registered Values | The units of measure for the capacity appropriate to this output device and its operation. For examples, currency units for postal-meters. | read-write |
| Max Capacity | Integer .<br>(-1) Other<br>(-2) Unknown) | The maximum capacity of this output device expressed in **capacity Units**. | read-write |
| Current Level | Integer<br>(-2) Unknown | The remaining capacity of the possible capacity expressed in capacity units. | read-write |
| Status | Enumeration: | This value indicates the specific status of the output device. The values have the following meanings:<br>(1) Other<br>(2) Unknown<br>(3) Available<br>(4) Unavailable Critical Alert<br>(5) Available Non Critical Alert<br>(6) Unavailable Both Critical and Non Critical Alerts<br>(7) On Request<br>(8) Unavailable<br>(9) Busy Temporarily Unavailable. This value indicates the status of the output device. | read-write |

## INSERTER OUTPUT TYPE ATTRIBUTE EXTENDED -OPTIONAL

*Table 83Inserter Output Device Type Extended*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Output Device Type Extended | Augments Output Type: Sequence of Attributes | A list of information suitable for extended output device management. | read-only |
| Name | DisplayString(63) | The name assigned to this output device. | read-only |
| Vendor Name | DisplayString(63) | The name of the vendor of this device. | read-only |
| Model | DisplayString(63) | The model name of this output device. | read-only |
| Version | DisplayString(63) | The version string for this output device. | read-only |
| Serial Number | DisplayString(63) | The serial number assigned to this Output device. | read-only |
| Description | DisplayString(255) | A free form text description of this output device in the localization specified by **General Current Localization.** | read-only |
| Security | Enumeration:<br>(1) Other | Indicates that this device has some security associated with it. | read-only |

| | | | |
|---|---|---|---|
| | (2)Unknown<br>(3) On<br>(4) Off<br>(5) Not Present | | |

## INSERTER OUTPUT DEVICE SUPPLY TABLE

*Table 84 Output Device Supply Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Output Device Supply Table | Sequence of Supply Objects | A list of supplies used, their description, capacities and current level. | read-only |

*Table 85 Output Device Supply Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by the inserter to identify this output supply container or receptacle. Although these values may change due to a major reconfiguration of the inserter (e.g., the addition of new supplies to the output device), values are expected to remain stable across successive power cycles. | read-write |
| Index | Integer | The value of the **Index** corresponding to the inserter output device with which this supply entity is associated. | read-write |
| Colorant Index | Integer | The value of the Colorant Index corresponding to the colorant with which this supply entity is associated. The value shall be 0 if there is no Output Colorant Table. | read-write |
| Class | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Supply that is consumed<br>(4) Receptacle that is filled | This value indicates whether this supply entity represents a supply container that is consumed or a receptacle that is filled. | read-write |
| Type | Enumeration:<br>(1) other<br>(2) unknown<br>(3) toner<br>(4) waste toner<br>(5) ink<br>(6) ink cartridge<br>(7) ink ribbon<br>(8) waste ink<br>(9) OPC (organic photo conductor) (10) developer<br>(11) fuser oil<br>(12) solid wax<br>(13) ribbon wax<br>(14) waste wax<br>(15) water<br>(16) waste water<br>(17) glue water | This value identifies the supply type of this supply. | read-write |

| | | | |
|---|---|---|---|
| | additive<br>(18) waste paper | | |
| Description | String(255) | The description of this supply container or receptacle in text useful for operators and management applications. | read-write |
| Supply Unit | Enumeration:<br>(1) other<br>(2) unknown<br>(3) thousandths of inches<br>(4) micrometers<br>(12) thousandths of ounces<br>(13) tenths of grams<br>(14) hundreds of fluid ounces<br>(15) tenths of milliliters | The unit of measure of this supply container or receptacle. | read-write |
| Max Capacity | Integer<br>(-2) Unknown | The maximum capacity of this supply container or receptacle is expressed in **supply units**. If this supply container/ receptacle can reliably sense this value, the value is sensed by the device and is read-only; otherwise the value may be written by management or control panel application. The value (-2) indicates unknown. | read-write |
| Current Level | Integer<br>(-1) Other<br>(-2) Unknown | The current level if this supply entity is a container; the remaining space if this supply entity is a receptacle. If this supply container/ receptacle can reliably sense this value, the value is sensed by the finisher and is read-only; otherwise the value may be written by management or control panel application. The value<br>(-1) means other and specifically indicates that the device places no restriction on this parameter. The value (-2) indicates unknown. | read-write |

## OUTPUT DEVICE COLORANT TABLE

*Table 86 Output Device Colorant Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Output Device Colorant Table | Sequence of Colorant Entries | A list of all the colorants available on Inserter output devices.. | read-only |

*Table 87 Output Device Colorant Entry*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | A unique value used by the inserter to identify this output device colorant. Although these values may change due to a major reconfiguration an output device (e.g., the addition of new colorants to the output device), values are expected to remain stable | read-write |

| | | | |
|---|---|---|---|
| | | across successive inserter power cycles. | |
| Index | Integer | The value of the **Index** corresponding to the output device with which this colorant entry is associated. | read-write |
| Role | Enumeration:<br>(1) Other<br>(2) Unknown<br>(3) Process<br>(4) Spot | The role played by this colorant. | read-write |
| Value | String(63) | The name of the color of this colorant using standardized string names from ISO 10175 (DPA)and ISO 10180 (SPDL) which are:<br>other,<br>unknown,<br>white,<br>red,<br>green,<br>blue,<br>cyan,<br>magenta,<br>yellow and<br>black.<br>Implementors may add additional string values. The naming conventions in ISO standards are recommended in order to avoid potential name clashes. | read-write |
| Tonality | Integer | The distinct levels of tonality realizable by a marking device when using this colorant. This value does not include the number of levels of tonal difference that an interpreter can obtain by techniques such as half toning. This value must be at least 2. | read-write |

## INSERTER ALERT TABLE

*Table 88 Alert Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Alert Table | List of Active Inserter Alerts | The alert table lists all the critical and non-critical alerts currently active in the inserter. A critical alert is one that stops the inserter from inserting immediately and can not continue until the critical alert condition is eliminated. Non-critical alerts are those items that do not stop inserting but may at some future time. The table contains information on the severity, component, detail location within the component, alert code and description of each critical alert that is currently active within the inserter.<br><br>There is one inserter alert table per inserter component. The format of alert tables and the rules for their management are described in **Appendix A: Alerts**. | read-only |

## CURRENT JOB GROUP

*Table Current Job Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Current Job Group | DisplayString(64) | This group identifies the current job running on the inserter | read-write |

*Table 89 Current Job Group Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Product Name | DisplayString(63) | The name of the product associated with this insert job | read-write key |
| Product Instance Qualifier | DisplayString(63) | The Instance Qualifier for the name of the productassociated with this insert job | read-write key |
| Insert Job Name | DisplayString(63) | The name assigned to this insert job. This name is unique within the domain of insert names for a given product. | read-write |
| Insert Job Size Unit | Enumeration<br>1 = Other<br>2 = Unknown<br>4 = Pages<br>5 = Mailpieces | The unit of measure used to estimate and report on the size of this insert job. | read-write |
| Insert Job Size | Integer | The actual size of the insert output produced by this insert job expressed in Insert Job Size Units | read-write |

# RECONCILIATION AGENT CLASS OBJECTS

The standard group definitions defined in this section are used by instrumented component agents provided by LMO System managers to reconcile products and print jobs in operational LMO system environments. Once the instrumented component is installed, standard DMI management interfaces are used to pass parameters to activate and control the reconciliation process.

## RECONCILIATION COMPONENTID GROUP- MANDATORY

*Table 90 Reconciliation ComponentId Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| ComponentID | DisplayString(64) | This is a required group for all MIF components. It contains required attributes that must be specified for each component installed in the Management Information Format data base.<br><br>The ComponentID group is the first group in a DMTF component definition and must be assigned the Group Id value of 1.<br><br>The class value for this group is: DMTF\|ComponentId\|1.0. | read-only |

*Table 91 Reconciliation ComponentId Group Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Manufacturer | DisplayString(64) | The name of the organization that produces this component. | read-only |
| Product | DisplayString(64) | The name of this component or product. | read-only |
| Version | DisplayString(64) | The version string definition for this component. | read-only |
| Serial Number | DisplayString(64) | The serial number for this component. | read-only |
| Installation | Date | The time and date of the last install of this component on this system. | read-only |
| Verify | Enumeration | Verification level for this component. Asking for the verify attribute causes the component instrumentation code to perform checks to verify that the component is still in the system and working properly. It should return one of the following values:<br>(0) An error occurred; Check status code.<br>(1) Component does not exist.<br>(2) Verify not supported.<br>(4) Component exists, functionally untested.<br>(5) Component exists, functionally unknown.<br>(6) Component exists, functionally no good.<br>(7) Component exists, functionally good. | read-only |

## RECONCILE PRODUCT GROUP

*Table 92 Reconcile Product Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Reconcile Product | Sequence of Attributes | This group specifies atributes for reconciling Products in an LMO system environment. | read-write |

*Table 93 Reconcile Product Group Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Product Name | DisplayString(63) | The name of the product to which this entry applies. | read-write key |
| Product Instance Qualifier | DisplayString(63) | The Instance Qualifier for the name of the product associated with this entry. | read-write key |
| Operation | Enumeration<br>0= Do Nothing<br>1= Reconcile & Delete<br>2 = Reconcile<br>3 = Delete | The reconcile operation to perform. | read-write |
| Current Status | Enumeration<br>0 = No Status<br>1 = Busy<br>2 = Done<br>3 = Error | The current status of the operation. | read-write |
| Message | DisplayString(200) | An error or warning message from the operation | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | Delete this row of the table | read-write |
| Query | String(10) | Query information about keys for this table entry | read-write |

## RECONCILE PRINT JOB GROUP

*Table 94 Reconcile Print Job Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Reconcile Print Job | Sequence of Attributes | This group specifies attributes for reconciling print jobs in an LMO system environment. | read-write |

*Table 95 Reconcile Print Job Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Product Name | DisplayString(63) | The name of the product associated with this entry. | read-write key |
| Product Instance Qualifier | DisplayString(63) | The Instance Qualifier associated with the name of the product to which this insert job applies. The combination of Product Name and Product Instance Qualifier is required to insure unique key access to table entries when multiple instances of a product are active in the system at the same time. | read-write key |
| Print Job Name | DisplaysString(63) | The name of the print job to be reconciled. | read-write key |

| | | | |
|---|---|---|---|
| Operation | Enumeration<br>0 = Do Nothing<br>1 = Reconcile & Delete<br>2 = Reconcile<br>3 = Delete<br>4 = Reprint | The reconcile operation to perform | read-write |
| Current Status | Enumeration<br>0 = No Status<br>1 = Busy<br>2 = Done<br>3 = Error | The current status of the operation | read-write |
| Message | DisplayString(200) | An error or warning message from the operation | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | Delete this row of the table | read-write |
| Query | String(10) | Query information about keys for this table entry | read-write |

## RECONCILE JOB CONTROL GROUP

*Table 96 Reconcile Job Control Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Reconcile Job Control | Sequence of Attributes | This group specifies control attributes for reconciling jobs in an LMO system environment. | read-write |

*Table 97 Reconcile Job Control Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | The Index of this entry | read-write key |
| Component Name | DisplayString(50) | The component name for data to be exported | read-write |
| Group Name | DisplayString(50) | The group name for the data to be exported. | read-write |
| Attribute List | DisplayString(256) | A list of attributes to be exported. | read-write |
| Condition | DisplayString(512) | The contion for rows to be exported. The initial condition is product name and instance qualifier. Any value in this attribute is added to that base. | read-write |
| Path Name | DisplayString(64) | The name of the file to export into. | read-write |
| File Control | Enumeration<br>0 = Overwrite<br>1 = Append | The method for writing data to the export file. | read-write |
| Time Out | Integer | Length in seconds to wait for export to complete | read-write |
| Command | DisplayString(256) | The command to execute after export. | read-write |
| Delete Row | Enumeration<br>0 =False<br>1 = True | Delete this row of the table | read-write |
| Query | String(10) | Query information about keys for this table entry | read-write |

## RECONCILE REPRINT CONTROL GROUP

*Table 98 Rreconcile Reprint Control Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Reconcile Reprint Control | Sequence of Attributes | This group specifies control attributes for reconciling reprints in an LMO system environment. | read-write |

*Table 99 Reconcile Job Control Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | The Index of this entry | read-write key |
| Component Name | DisplayString(50) | The component name for data to be exported | read-write |
| Group Name | DisplayString(50) | The group name for the data to be exported. | read-write |
| Attribute List | DisplayString(256) | A list of attributes to be exported. | read-write |
| Condition | DisplayString(512) | The contion for rows to be exported. The initial condition is product name and instance qualifier. Any value in this attribute is added to that base. | read-write |
| Path Name | DisplayString(4) | The name of the file to export into. | read-write |
| File Control | Enumeration<br>0 = Overwrite<br>1 =Append | The method for writing data to the export file. | read-write |
| Time Out | Integer | Length in seconds to wait for export to complete | read-write |
| Command | DisplayString(256) | The command to execute after export. | read-write |
| Delete Row | Enumeration<br>0 =False<br>1 = True | Delete this row of the table | read-write |
| Query | String(10) | Query information about keys for this table entry | read-write |

## RECONCILE PRODUCT CONTROL GROUP

*Table 100 Rreconcile Product Control.Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Reconcile Print Job | Sequence of Attributes | This group specifies control attributes for reconciling products in an LMO system environment. | read-write |

*Table 101 Reconcile Product Control Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | The Index of this entry | read-write key |
| Component Name | DisplayString(50) | The component name for data to be exported | read-write |
| Group Name | DisplayString(50) | The group name for the data to be exported. | read-write |
| Attribute List | DisplayString(256) | A list of attributes to be exported. | read-write |
| Condition | DisplayString(512) | The contion for rows to be exported. The | read-write |

| | | | |
|---|---|---|---|
| | | initial condition is product name and instance qualifier. Any value in this attribute is added to that base. | |
| Path Name | DisplayString(4) | The name of the file to export into. | read-write |
| File Control | Enumeration<br>0 = Overwrite<br>1 =Append | The method for writing data to the export file. | read-write |
| Time Out | Integer | Length in seconds to wait for export to complete | read-write |
| Command | DisplayString(256) | The command to execute after export. | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | Delete this row of the table | read-write |
| Query | String(10) | Query information about keys for this table entry | read-write |

# UTILITY AGENT CLASS OBJECTS

The standard group definitions defined in this section are used by instrumented component agents provided by LMO System managers to monitor, control, import, export and query attributes for specified components active in operational LMO system environments. Once the instrumented component is installed, standard DMI management interfaces are used to pass parameters to activate and control the utility functions.

## UTILITY COMPONENTID GROUP- MANDATORY

*Table 102 Reconciliation ComponentId Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| ComponentID | DisplayString(64) | This is a required group for all MIF components. It contains required attributes that must be specified for each component installed in the Management Information Format data base.<br><br>The ComponentID group is the first group in a DMTF component definition and must be assigned the Group Id value of 1.<br><br>The class value for this group is: DMTF\|ComponentId\|1.0. | read-only |

*Table 103 Utility ComponentId Group Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Manufacturer | DisplayString(64) | The name of the organization that produces this component. | read-only |
| Product | DisplayString(64) | The name of this component or product. | read-only |
| Version | DisplayString(64) | The version string definition for this component. | read-only |
| Serial Number | DisplayString(64) | The serial number for this component. | read-only |
| Installation | Date | The time and date of the last install of this component on this system. | read-only |
| Verify | Enumeration | Verification level for this component. Asking for the verify attribute causes the component instrumentation code to perform checks to verify that the component is still in the system and working properly. It should return one of the following values:<br>(0) An error occurred; Check status code.<br>(1) Component does not exist.<br>(2) Verify not supported.<br>(4) Component exists, functionally untested.<br>(5) Component exists, functionally unknown.<br>(6) Component exists, functionally no good.<br>(7) Component exists, functionally good. | read-only |

## IMPORT EXPORT GROUP

*Table 104 Import Export Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Import Export | Sequence of Attributes | This group specifies atributes for importing and exporting attribute data in an LMO system environment. | read-write |

*Table 105 Import Export Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Component Name | DisplayString(50) | The component name for data to be imported or exported | read-write |
| Group Name | DisplayString(50) | The group name for the data to be imported or exported. | read-write |
| Attribute List | DisplayString(256) | A list of attributes to be imported or exported. If this value is blank or a star(*), then all attributes are imported/exported. Otherwise, this is a comma separated list of attributes to import/export from the indicated component and group. | read-write |
| Condition | DisplayString(512) | The contion for rows to be exported. If this value is blank or a star(*), then all rows from the group are exported. Otherwise this condition is used to identify rows to export whose general form is <attr name> =<value> **and** <attr name> = <value>. Parenthesis are allowed, conditionals allowed are "=,<>,<,>,<=,>= and logical operations allowed are AND,OR, and NOT. | read-write |
| Path Name | DisplayString(4) | The path to the file to read/write the import/export data.. | read-write |
| File Control | Enumeration<br>0 = Overwrite<br>1 =Append | The method for writing data to the export file. | read-write |
| Current Status | Enumeration<br>0 = No Status<br>1 = Busy<br>2 = Done<br>3 = Error | Current status of the operation | read-write |
| Message | DisplayString(4) | An error or warning message from the operation | read-write |
| Operation | Enumeration<br>0 = Do Nothing<br>2 = Export Data<br>3 = Import Data<br>4 = Export & Delete Data | The operation to perform | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | Delete this row of the table | read-write |
| Query | String(10) | Query information about keys for this table entry. | read-write |

## QUERY ROWS GROUP

*Table 106 Query Rows Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Query Rows | Sequence of Attributes | This group specifies atributes for quering data attributes in specified components and groups.. | read-write |

*Table 107 Query Rows Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Component Name | DisplayString(50) | The component name for the information to be queried. | read-write |
| Group Name | DisplayString(50) | The group name for the information to be queried. | read-write |
| Attribute List | DisplayString(256) | A list of attributes to be queried. If this value is blank or a star(*), then all attributes arequeried. Otherwise, this is a comma separated list of attributes to be queried from the indicated component and group. | read-write |
| Condition | DisplayString(512) | The contion for rows to be queried. If this value is blank or a star(*), then all rows from the group arequeried.. Otherwise this condition is used to identify rows to query whose general form is <attr name> =<value> **and** <attr name> = <value>. Parenthesis are allowed, conditionals allowed are "=,<>,<,>,<=,>= and logical operations allowed are AND,OR, and NOT. | read-write |
| Query Rows | DisplayString(10) | Query the defined data. When a get is done on this attribute, it returns a string that contains a comma-delimited list of the defined attribute values. with line feeds between rows. | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | Delete this row of the table | read-write |
| Query | String(10) | Query information about keys for this table entry. | read-write |

## INDICATE CONTROL GROUP

*Table 108 Indicate Control Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Indicate Control | Sequence of Attributes | This group specifies atributes for controlling indications is specified components and groups | read-write |

*Table 109 Indicate Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Component Name | DisplayString(50) | The name of the component which contains the group and attribute for which we want to turn on/off indications | read-write |

| | | | |
|---|---|---|---|
| Group Name | DisplayString(50) | The name of the group for which we want to tur on/off indications. | read-write |
| Attribute Name | DisplayString(50) | The name of the attribute for which we want to turn on/off indications. This field is not used if the operation is Indicate on Add, Indicate on Delete, No Indicate on Add, No Indicate on Delete. | read-write |
| Operation | Enumeration<br>0 = Do Nothing<br>1 = Indicate on Add<br>2 = Indicate on Delete<br>3 = Indicate on Set<br>4 = No Indicate on Add<br>5 = No Indicate on Delete<br>6 = No indicate on Set | The type of indication to perform | read-write |

## WATCH CONTROL GROUP

*Table 110 Watch Control Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Watch Control Table | Sequence of Attributes | This group specifies control atributes used by insrtumented component agents for watching specific attributes in specific components and groups for changes in state. | read-write |

*Table 111 Watch Control Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Component Name | DisplayString(50) | The name of the component which contains the attribute you want to watch | read-write |
| Group Name | DisplayString(50) | The name of the group which contains the attribute you want to watch | read-write |
| Attribute Name | DisplayString(50) | The name of the attribute to watch. Whenever the attribute changes state, an entry is added/updated in the Watch Table | read-write |
| Key List | DisplayString(150) | The keys to use to uniquely identify the one row within the group that we want to watch. This is a comma separated list of <value> clauses. There should be no spaces around any commas. | read-write |
| Accumulate Value | DisplayString(50) | The value to be added to the Accumulator attribute in the Watch Table for this attribute each time it changes state. If the value of this attribute is ACCUMULATE TIME, then we add the amount of time the attribute was in this state to the Accumulator. Otherwise, this is the name of an attribute within this group that is read at the beginning of a state and the end of a state, and the difference is added to the Accumulator. The type of the indicated attribute must be integer. | Read-write |
| Time Started | Date | The Date/Time we started watching this attribute. | read-write |
| Time Stopped | Date | The Date/Time we stopped watching this | read-write |

| | | | |
|---|---|---|---|
| | | attribute | |
| Time Disabled | Date | The Date/Time we stopped watching this attribute due to a disable operation | read-write |
| Duration Watching | Counter | The amount of time we were enabled while watching this attribute. | read-write |
| Polling Interval | Counter | The polling interval in seconds for this attribute. If this value is zero, then the attribute must be instumented and we will use asynchronous indicates to identify state changes for the attribute. | read-write |
| Current Status | Enumeration<br>0 = No Status<br>1 = Started<br>2 = Stopped<br>3 = Disabled<br>4 = Error | The current status of this watch | read-write |
| Message | DisplayString(200) | This field will be filled with any messages during operation of the watch processing. Usually this will be an error message that is set at the same time the status goes to Error. But, it may also be an informative message during normal operation. | read-write |
| Operation | Enumeration<br>0 = Do Nothing<br>1 = Start Watching<br>2 = Stop Watching<br>3 = Restart Watch<br>4 = Enable Watching<br>5 = Disable Watching<br>6 = Delete Watch<br>7 = Display Watch | The operation to perform for this watch. | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | Delete this row of the table | read-write |
| Query | String(10) | Query information about keys for this table entry. | read-write |

## WATCH TABLE

*Table 112 Watch Control Group*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Watch Table | Sequence of Attributes | This group contains the results from watching specific attributes in a component/group.. | read-write |

*Table 113Watch Control Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Component Name | DisplayString(50) | The name of the component which contains the attribute being watched. | read-write |
| Group Name | DisplayString(50) | The name of the group which contains the attribute being watched | read-write |
| Attribute Name | DisplayString(50) | The name of the attribute being watched.. | read-write |
| Key List | DisplayString(150) | The keys to use to uniquely identify the one row within the group that we want to watch. This is a comma separated list of <value> clauses. There should be no spaces around | read-write |

| | | | |
|---|---|---|---|
| | | any commas. | |
| Value | DisplayString(50) | The state of the attribute for this entry in the table. For each state that is detected for this attribute, an entry is added to this table. | Read-write |
| Previous Value | DisplayString(50) | The state that this attribute was in prior to reaching its current state (if known). | read-write |
| Start Count | Counter | The starting count for this instance of this attribute. If the Accumulate Value in the control entry when watching was ACCUMULATE_TIME, this is the time when the attribute entered this state. Otherwise, this is the value read from the indicated attibute at the time the attribute entered this state. | read-write |
| Accumulator | Counter | The accumulator for this attribute. Each time the attribute changes state, we read the current Accumulate Value, subtract the Start Count, and add the difference to this attribute. So , if ACCUMLATE_TIME was used, this indicates the total time spent in this state. Otherwise, it is the total from the indicated attribute. | read-write |
| Instance Count | Counter | This is the number of times the attribute was in this state. | read-write |
| Average | DisplayString(30) | This is the average computed by the Accumulator divided by the Instance Count. | read-write |
| Current Status | Enumeration<br>0 = No Status<br>1 = Started<br>2 = Stopped<br>3 = Disabled<br>4 = Error | The current status of this watch | read-write |
| Message | DisplayString(200) | This field will be filled with any messages during operation of the watch processing. Usually this will be an error message that is set at the same time the status goes to Error. But ,it may also be an informative message during normal operation. | read-write |
| Operation | Enumeration<br>0 = Do Nothing<br>1 = Start Watching<br>2 = Stop Watching<br>3 = Restart Watch<br>4 = Enable Watching<br>5 = Disable Watching<br>6 = Delete Watch<br>7 = Display Watch | The operation to perform for this watch. | read-write |
| Delete Row | Enumeration<br>0 = False<br>1 = True | Delete this row of the table | read-write |
| Query | String(10) | Query information about keys for this table entry. | read-write |

## HISTORY CONTROL TABLE

*Table 114 History Control Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|

| | | | |
|---|---|---|---|
| History Control Table | Sequence of Attributes | This group contains attributes fo controlling processing of history data for watched attributes in specific components and groups... | read-write |

*Table 115 History Control Attributes*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| History Name | DisplayString(50) | The name of this history. | read-write |
| Component Name | DisplayString(50) | The name of the component which contains the attributeyou want to watch. | read-write |
| Group Name | DisplayString(50) | The name of the group which contains the attributeyou want to watch. | read-write |
| Key List | DisplayString(150) | The keys to use to uniquely identify the one row within the group that we want to watch. This is a comma separated list of <value> clauses. There should be no spaces around any commas. | read-write |
| Watch Attributes | DisplayString(500) | The name of the attributes within the group to watch. This is a comma separated list of attribute names.. | read-write |
| Save Attributes | DisplayString(500) | The names of the attributes within the group to save whenever any of the watched attributes changes | Read-write |
| Polling Interval | Counter | The polling interval in seconds for this watch. | read-write |
| File Name | DisplayString(200) | The name of the file reconciled history is exported into or the name of the MIF file to create. | read-write |
| Last Poll Time | Date | The last time this history polled. | read-write |
| Current Status | Enumeration<br>0 = No Status<br>1 = Started<br>2 = Stopped<br>3 = Error<br>4 = Purged<br>5 = Reconciled<br>6 = MIF Created | The current status of this history. | read-write |
| Operation | Enumeration<br>0 = Do Nothing<br>1 = Start<br>2 = Stop<br>3 = Purge<br>4 = Reconcile<br>5 = Reconcile & Delete<br>6 = Create MIF<br>7 = Reset<br>8 = Display | The operation to perform for this history. START starts watching the desired attributes. STOP stops the watching. PURGE deletes all data from the history table and the active history table for this history (it must first be stopped). RECONCILE exports all non-active rows from the history table to the indicated file. RECONCILE &DELETE exports and then deletes all non-active rows from the history table to the indicated file. The RECONCILE operations may be performed while the history is started. CREATE MIF creates a MIF file that contains the definitions for the history table and active history table. This file is then installed with the DMI Service Layer and the LMOINSTR.CFG file is modified to instrument the active history table. After creation and instalation of the MIF file, both the lmomontr and the lmoinstr code must be | read-write |

| | | | |
|---|---|---|---|
| | | restarted so that the new MIF will be utilized. RESET deletes all data from the history table and the active history table, stops any polling that might be in process, and sets the status to STOPPED. DISPLAY prints the rows in the history table and the active history table to the window in whic h lmomontr was started. The typical sequence of operations is: CREATE MIF, restart lmomontr and lmoinstr, START, RECONCILE, STOP, RESET, ETC. | |
| Delete Row | Enumeration<br>0 = False<br>1 = True | Delete this row of the table | read-write |
| Query | String(10) | Query information about keys for this table entry. | read-write |

# APPENDIX A. ALERTS

## OVERVIEW

An Alert is the representation of a reportable event. An event is a change in the state of a component. Some of those state changes are of interest to management applications and are therefore reportable. Typically, these are the events that affect the component's ability to function. Alerts usually occur synchronously to the operation of the LMO server.

Alerts can be classified into two basic categories, critical and NonCritical. A critical alert is one that is triggered by entry into a state in which processing is stopped and can not continue until the condition that caused critical alert is eliminated. "Out of paper", "toner empty" and "output bin" full are examples of critical alerts that apply to a printer. Non-critical alerts are triggered by those events that enter a state in which processing is not stopped. Such a non-critical state may, at some future time, lead to a state in which processing may be stopped. Examples of this kind of non-critical alerts are "input media low", "toner low" and "output bin nearly full". A non-critical alert may also simply provide information, such as signaling a configuration change in the component.

## INSTRUMENTATION

Component instrumentation is responsible for detecting reportable events, making an entry in an ALERT TABLE and, if and only if the event is a critical event, invoking a DMIINDICATE to cause an asynchronous event to be reported. The Alert Table contains information on the severity, location, alert code, and description of each critical alert that is currently active within the printer. Each reportable event, both critical and non-critical, causes an entry to be made in the Alert Table.

Summary information about the state of the overall component and individual devices within the component is also be provided in status states that are instrumented. For components that maintain status states the following information is reported:

1 other

2 unknown

3 Ready and available

4 In Critical state

5 In Non-critical, warning state

6 State is a Combination of Critical and Warning

7 On request (intervention required)

8 Unavailable

9 Busy/Temporarily Unavailable

For example, an input device that is jammed on a printer could show a status of 6 -both a critical state, jammed, and a non critical state, low paper.

## ALERT TABLES

The Alert Table is a single table in which all active alerts are represented. This section provides an overview of the table and description of how it is managed. The basic content of the alert table is the severity (critical or non-critical) of the alert, the Group and entry where a state change caused the alert, additional information about the alert (a more detailed location, an alert code, and a description), and an indication of the level of training needed to service the alert.

The Alert Table contains some information that is redundant, for example that an event has occurred, and some information that is only represented in the Alert Table, for example the additional information. A single table is

used because a single operational event could cause more than one alert, for example paper jams in more than one place in a media path. Associating the additional information information with alerts makes multiple reports possible.

Every time an alert occurs in a component, the component makes one or more entries into its Alert Table. The component determines if an event is to be classified as critical or non-critical. If the severity of the Alert is "critical", the component sends DmiIndicate event notification to the server indicating that the table has changed. Whether or not a an asynchronous event is sent, the management application can poll the component on a regular basis and read and parse the table to determine what conditions have changed, in order to obtain information to on both critical and non-critical events.

## ALERT TABLE MANAGEMENT

The alert tables are sparsely populated tables. This means the tables will only contain entries of the alerts that are currently active and the number of rows, or entries in the table will be dynamic. More than one event can be added or removed from the event tables at a time depending on the implementation of the component.

There are basically two kinds of events that produce alerts: binary change events and simple change events. Binary change events come in pairs: the leading edge event and the trailing edge event. The leading edge event enters a state from which there is only one exit; for example, going from running to stopped with a feeder jam. The only exit from this state is fixing the feeder jam and it is clear when that is accomplished. The trailing edge event is the event which exits the state that was entered by the leading edge event; in the example above fixing the feeder jam is the trailing edge event.

It is relatively straightforward to manage binary change events in the Alert Table. Only the leading edge event makes an entry in the alert table. This entry persists in the Alert Table until the trailing edge event occurs at which point this event is signal by the removal of the leading edge event entry in the Alert Table. That is, a trailing edge event does not create an entry; it removes the corresponding leading edge event. With binary events it is possible to compute the maximum number that can occur at the same time and construct an Alert Table that would hold that many events. There would **be no** possibility of table overflow and no information about outstanding events would be lost.

Unfortunately, there are some events that are not binary changes. This other category of event, the simple change event, is illustrated by the configuration change event. With this kind of event the state of the machine has changed, but to a state which is (often) just as valid as the state that was left and from which no return is necessary. For example, an operator may change the paper that is in the primary input source from letter to legal. At some time in the future the paper may be changed back to letter, but it might be changed to executive instead. This is where the problem occurs. It is not obvious how long to keep simple change event entries in the Alert Table. It they were never removed, the Alert Table would continue to grow indefinitely.

There are several approved solutions to the problem of removing simple change events. One approach is to not put simple change events into the Alert Table. Then, as observed above, the table can be designed to never overflow. This would imply, however, that simple events like supply adds would have to be reported synchronously in order to make the event information available to management applications Another approved approach is to use a policy for removing alert entries when the table is about to overflow. This policy says that entries for Critical Alerts will not be removed as long as there is an entry for a Non-critical Alert that can be removed. After that, it is up to the vendor to decide which Non-critical Alert entry to remove. If there are no Non-Critical Alert entries to remove, then an entry for a Critical Alert can be removed, again vendor's choice.

Note that because the Alert Index is a counter there will be gaps in the values in the table which can be detected by the management application. This may indicate to the management application that it may want to re-acquire the current component states and check for state changes it did not observe in polling the Alert Table.

## ALERT TABLE FORMAT

The format of the Component Alert Table is shown below.

## COMPONENT ALERT TABLE

*Table 116 Alert Table*

| Group Name | Syntax | Definition | Access |
|---|---|---|---|
| Alert Table | List of Active Component Alerts | The alert table lists all the critical and non-critical alerts currently active in the component. A critical alert is one that stops the component from processing immediately and can not continue until the critical alert condition is eliminated. Non-critical alerts are those items that do not stop processing but may at some future time. The table contains information on the severity, component, detail location within the component, alert code and description of each critical alert that is currently active within the component. | read-only |

*Table 117 Alert Template*

| Attribute Name | Syntax | Definition | Access |
|---|---|---|---|
| Index | Integer | The index value used to determine which alerts have been added or removed from the alert table. This is an incrementing integer starting from zero every time the component is installed. When the component adds an alert to its table, that alert is assigned the next higher integer value from the last item entered into the table. If the index value reaches its maximum value, the next item entered will cause the index value to roll over and start at zero again. The first event placed in the alert table after installation of the component shall have an index value of 1.<br><br>NOTE: The management application will read the alert table when an asynchronous event notification occurs or at a periodic rate and then parse the table to determine if any new entries were added by comparing the last known index value with the current highest index value. The management application will then update its copy of the alert table.<br><br>When the component discovers that an alert is no longer active, the component shall remove the row for that alert from the table and shall reduce the number of rows in the table. The component may add or delete any number of rows from the table at any time. | read-only |
| Severity Level | Enumeration:<br>(1) Other<br>(3) Critical<br>(4) Warning | The level of severity of this alert table entry. The component determines the severity level assigned to each entry in the table if not explicitly specified for the current LMO installation. | read-only |
| Training Level | Enumeration: | The level of training required to handle this | read-only |

| | | | |
|---|---|---|---|
| | (1) Other<br>(2) Unknown<br>(3) Untrained<br>(4) Trained<br>(5) Field Service<br>(6) Management | alert. The training level is an enumeration that is determined and assigned by the component manufacturer based on the information or training required to handle this alert. The component will break alerts into these different training levels. It is the responsibility of the management application in the system to determine how a particular alert is handled and how and to whom that alert is routed. The following are the four training levels of alerts:<br><br>(1) FIELD SERVICE -alerts that typically require training and technical knowledge of the printer and its devices. An example of a technical person would be a manufacturer's Field Service representative, or other person formally trained by the manufacturer or similar representative.<br><br>(2) TRAINED -alerts that require an intermediate or moderate level of knowledge of the component and its devices. A typical example of an alert that a trained operator can handle is replacing toner cartridges.<br><br>(3) UNTRAINED -alerts that can be fixed without prior knowledge of the component and its devices either because the action to correct the alert is obvious or the component can help the untrained person to fix the problem. Typical examples of such alerts is loading paper and emptying output bins on a low end printer.<br><br>(4) MANAGEMENT -Alerts that have to do with the overall operation and configuration of the component. Examples of management events are configuration changes of devices. | |
| Group | Enumeration: Component Model Specific: e. g<br>(1) Other<br>(3) Input<br>(4) Output<br>(5) Finisher<br>(6) Media Path<br>(7) General Printer<br>(8) Marker | The type of device within the component that this alert is related. This is an integer enumeration equal to the ID assigned to the Group or Table within the component definition. Input, Output, and Markers are examples of groups defined in the Printer component model. | read-only |
| Group Index | Integer | An index of the row within the principal table in the group identified by Alert Group that represents the device within the component that caused this alert. The combination of the Group and GroupIndex defines exactly which component device caused the alert; for example Input #1, Output #2, and marker # 1. | read-only |
| Location | Integer | The device location that is defined by the vendor manufacturer to further refine the | read-only |

| | | | |
|---|---|---|---|
| | | location of this alert within the designated device. The location is used in conjunction with the Group and Group Index values: for example, there is an alert at Input #2 at location number 7. Marker # 1. | |
| Alert Code | Enumeration: registered values | The code that describes the type of alert for this entry. There are different codes for each devicetype; for example, Media Supply Low and MediaSupply Empty are alert codes for input devices. | read-only |
| Description | DisplayString(255) | A description of this alert entry in the current localization. The description is provided by component further elaborate on the enumerated alert or provide information in the case where the code is classified other or unknown. The component is required to return a description string but the string may be a null string. | read-only |
| Delete Row | Enumeration<br>0 = False<br>1 = True | An attribute allowing instrumented management applications to delete this row in the table | read-write |
| Query | String(10) | Query information about keys for this product table entry | read-write |

# APPENDIX B. DMI CLIENT NETWORK INTERFACE

In a Large Mailroom system environment there are many devices and processes to be managed and many management applications requiring access and control over manageable data. Most, if not all, of these are remote from the system containing the LMO Server. To accommodate this requirement the LMO System provides a client DMI Network interface to the functions provided by the DMI Service Layer that runs in the server.

The DMI client interface is accomplished by porting DMI function calls to each of the required client platforms and supporting all of the functions of the MI and CI through a transparent error free connection. The function calls are implemented on a Remote Procedure Call (RPC) base using TCP-IP as the transport carrier. The underlying RPC support is handled by the DMI Client code and is transparent to the DMI programmer.

Like the DMI itself, the client interface must be capable of operating in a consistent manner across a variety of operating systems, hardware platforms and different architectures. In addition, it must be capable of allowing a client implemented on one architecture to interoperate with a server implemented on another. For example, DOS and UNIX Clients must both be able to interoperate with an OS/2 server, and a given client should be able to interoperate with multiple servers.

The DMI Client Interface meets these requirements by standardizing DMI interface extensions which preserve the semantics and syntax of the MI and CI while enabling specific servers to be selected for communication.

The DMI Network interface is available in each of the following client systems in the first release of the LMO System:

- MVS
- DOS
- Windows
- Unix

A brief description of the DMI Network Interface extensions is given in the sections below. A more complete description of the DMI Client Network Interface and its associated software is found in the documentation provided with the LMO Systems Manager.

## DMI NETWORK INTERFACE EXTENSIONS

### DMI Client API Definitions

```
#if defined(__OS2__) || defined(OS2)
#  define NET_FAR
#  define NET_FUNC_ENTRY _System
#endif
#if defined(__WINDOWS__) || defined(_WINDOWS) || defined(MSDOS)
#  define NET_FAR _FAR
#  define NET_FUNC_ENTRY _export
#endif
#if defined(__AIX__) || defined(AIX)
#  define NET_FAR
#  define NET_FUNC_ENTRY
#endif
```

```
#ifdef __cplusplus
extern "C" {
#endif
```

**ROUTINE: ConnectToServer()**

**PURPOSE:**
the user's API entry point to connect to the desired server

**INPUT:**
hostName
- a string naming the server to connect to

bStartCallBackServer
- if non-zero, then the code will start callback service for this server
- callback service must be started for this client to be a CI interface to the DMI
- callback service must be started for this client to be a MA interface that can receive indicates from the DMI
- DOS and WINDOWS can not start call back service in same task. See EnterNetDispatch foralternative methods of starting call back service.

timeOut
- timeout value in seconds for all calls to this server

**OUTPUT:**
0 - success in connecting to the server
non-zero - one of the NETSL_xxx codes listed below

```
DMI_UNSIGNED NET_FUNC_ENTRY
ConnectToServer(
   const char NET_FAR * hostName,
   int bStartCallBackServer,
   unsigned long timeOut);
```

**ROUTINE: DisconnectFromServer()**

**PURPOSE:**
the user's API entry point to disconnect from the desired server

**INPUT:**
hostName
- a string naming the server to disconnect from

**OUTPUT:**
0 - success in disconnecting the server
non-zero - one of the NETSL_xxx codes listed below

DMI_UNSIGNED NET_FUNC_ENTRY
DisconnectFromServer(const char NET_FAR *hostName);

**ROUTINE: NetSyncInvoke()**

**PURPOSE:**
the user's API entry point to perform a synchronous DmiInvoke to the server which was last connected

**INPUT:**
dmiCommand - pointer to the DMI command block

**OUTPUT:**
0 - success in invoking to the server
non-zero - one of the NETSL_xxx codes listed below or one of the error codes returned from DmiInvoke()

DMI_UNSIGNED NET_FUNC_ENTRY
NetSyncInvoke( void NET_FAR * dmiCommand);

**ROUTINE: NetSyncInvokeServer()**

**PURPOSE:**
the user's API entry point to perform a synchronous DmiInvoke to the server named in the argument

**INPUT:**
serverName - name of the server to send the command to
the server must have been previously connected
dmiCommand - pointer to the DMI command block

**OUTPUT:**
0 - success in invoking the server
non-zero - one of the NETSL_xxx codes listed below or one of the error codes returned from DmiInvoke()

DMI_UNSIGNED NET_FUNC_ENTRY

```
NetSyncInvokeServer(
   const char NET_FAR *serverName,
   void NET_FAR * dmiCommand);
```

**ROUTINE: NetIndicate()**

**PURPOSE:**
the user's API entry point to send an indicate to the server
which was last connected

**INPUT:**
dmiIndicate - pointer to the DMI indicate block

**OUTPUT:**
0 - success in sending the indicate to the server
non-zero - one of the NETSL_xxx codes listed below or one of
the error codes returned from DmiIndicate()

```
DMI_UNSIGNED NET_FUNC_ENTRY
NetIndicate(void NET_FAR *dmiIndicate);
```

**ROUTINE: NetIndicateServer()**

**PURPOSE:**
the user's API entry point to send an indicate to the server
named in the argument

**INPUT:**
serverName - name of the server to send the indicate to
the server must have been previously connected
dmiIndicate - pointer to the DMI indicate block

**OUTPUT:**
0 - success in sending the indicate to the server
non-zero - one of the NETSL_xxx codes listed below or one of
the error codes returned from DmiIndicate()

```
DMI_UNSIGNED NET_FUNC_ENTRY
NetIndicateServer(
   const char NET_FAR *serverName,
   void NET_FAR *dmiIndicate);
```

**ROUTINE: GetNetErrorMsg()**

**PURPOSE:**
the user's API entry point to retrieve an error message
the message retrieved is copied into the user's buffer and
is the last error found by the network client code.
this message is only meaningful if one of the other APIs
returned one of the NETSL_xxx codes below

**INPUT:**
buf - pointer to where the message should be copied to
bufLen - length of the above space

```
void NET_FUNC_ENTRY
GetNetErrorMsg(char NET_FAR *buf, int bufLen);

#if defined(__WINDOWS__) || defined(_WINDOWS)
```

**ROUTINE: StartWsa()**

**PURPOSE:**
for windows, this starts the WSA server
the ConnectToServer() routine automatically starts WSA if necessary
so the user does not need to use this API if calling ConnectToServer

**INPUT**
:

```
int NET_FUNC_ENTRY
StartWsa(void);
```

**ROUTINE: StopWsa()**

**PURPOSE:**
for windows, this stops the WSA server if it was running and
this is the last user of it
the DisconnectFromServer() routine automatically stops WSA if necessary
so the user does not need to use this API if calling
DisconnectFromServer

**INPUT:**

```
void NET_FUNC_ENTRY
```

```
StopWsa(void);
#endif

#if defined(__WINDOWS__) || defined(_WINDOWS) || defined(MSDOS)
```

**ROUTINE: EnterNetDispatch()**

**PURPOSE:**
enters the dispatch loop (by calling svc_run())
this routine should never return
it's use is only for WINDOWS, where we cannot start a new thread,
instead, the user will connect with callback, do anything else
he wants to do, and then call this routine to infinitely process
callback functions

```
void NET_FUNC_ENTRY
EnterNetDispatch(void);
#endif

#ifdef __cplusplus
}
#endif

#define NETERR       0x00020000
#define NETSL_ALLOC_ERROR              (NETERR + 0)
#define NETSL_NOT_CONNECTED_TO_SERVER       (NETERR + 1)
#define NETSL_RPC_INVOKE_ERROR             (NETERR + 2)
#define NETSL_ALREADY_CONNECTED                  (NETERR + 3)
#define NETSL_RPC_INDICATE_ERROR (NETERR + 4)
#define NETSL_SERVER_NOT_CONNECTED    (NETERR + 5)
#define NETSL_WSASTARTUP_ERROR             (NETERR + 6)
#define NETSL_CALLBACK_NOT_ALLOWED   (NETERR + 7)
#define NETSL_CALLBACK_ERROR               (NETERR + 8)
#define NETSL_CONNECT_ERROR                (NETERR + 9)
#define NETSL_DISCONNECT_ERROR             (NETERR + 10)
#define NETSL_CLIENT_NOT_CONNECTED    (NETERR + 11)
#define NETSL_CLIENT_INVOKE_ERROR       (NETERR + 12)
#define NETSL_SWAP_ERROR              (NETERR + 13)
#define NETSL_SOCKINIT_ERROR               (NETERR + 14)
```

**DMI CLIENT-SERVER COMMUNICATION SEQUENCES**

The sequence of communications between the client and server: is summarized below.

- server creates a TCP transport and registers it's program/version

- server then calls svc_run() to infinitely process client commands

**CLIENT CONNECT TO SERVER:**

- if client wants to create for callback:
  - finds an available TCP port number
  - starts a callback thread that calls svc_run to process commands from the server
- client creates a TCP connection to the server
- client sends a CLIENT_CONNECT command to the server which contains:
  - callback program number (if any)
  - callback version number (if any)
  - callback port number (if any)
  - byte order of the client machine
- server responds with a client ID for the connection, this ID is used by all subsequent communication from the client to the server

**CLIENT DISCONNECT FROM SERVER:**

- client sends a CLIENT_DISCONNECT command to the server which contains:
  - client ID of the client (which was returned from the CLIENT_CONNECT command
- server responds with NULL and frees any resources used by the client
- client then destroys the TCP connection to the server

**CLIENT DMI INVOKE OF SERVER:**

- client sends a SYNC_INVOKE command to server which contains:
  - client ID of the client
  - length of DMI command block
  - data of the DMI command block (length bytes)
- server retrieves the client ID, length and command data
- server finds info about the client and if necessary byte-swaps the command block into server byte order
- server allocates a confirm buffer
- if the -f option is given when starting the NETSL code, the the command block is checked to see if we can bypass the DMISL. If we can bypass, we invoke LMODB2 directly to perform the operation. If we cannot bypass, the server calls sDmiInvoke() to process the command
- server checks result, and if success:
  - if the command was a MA register, it adds a MA entry for the desired client
    - if the client is registering to receive indicates (i.e. has supplied an indicate function pointer), the server "allocates" an event function (it only has so many to allocate) and supplies this function pointer to the DMISL in the register

command. Then, when the DMISL calls that function, an INDICATE command is sent to the associated client (i.e. callback to the client).
- if the command was a MA unregister, it removes the appropriate MA entry of the client
- if the command was a CI register, it adds a CI entry for the desired client
  - the CiInvoke function pointer (from the client) is replaced by the server's CiInvoke routine, thus, when the DMISL calls the server's CiInvoke routine to perform a command, the server will find the appropriate CI entry based on the component ID in the CI command, and will the route the command on to the associated client (i.e. callback to the client)
- if the command was a CI unregister, it removes the appropriate CI entry of the client

- server then optionally packs the confirm buffer
- server byte-swaps the command and confirm into the client byte order if necessary
- server sends a response to the client with:
  - status code from the DmiInvoke
  - length of the DMI command block
  - data of the DMI command block (above length bytes)
  - length of the DMI confirm block
  - data of the DMI confirm block (above length bytes)
- client awakes out of the clnt_call(SYNC_INVOKE) and places the result command and confirm buffers into the appropriate memory
- client checks result, and if success:
  - if the command was a MA register, it adds a MA entry for the desired server
  - if the command was a MA unregister, it removes the appropriate MA entry of the server
  - if the command was a CI register, it adds a CI entry for the desired server
  - if the command was a CI unregister, it removes the appropriate CI entry of the server

NOTE:
- the server does not respond to the clnt_call until the DMI responds with the result of the command. If the command takes a long time to process, other requests are postponed until the one in progress completes. To do otherwise would require callback to the client and asynchronous invokes of the DMI.
- as a consequence of this waiting for processing to complete, the client may time out in it's connection with the server if a command takes a long time to process. (the timeout is set via the ConnectToServer call)

**SERVER DMI CI INVOKE OF CLIENT:**

- server sends a CI_INVOKE command to client which contains:
  - CI handle of the component (returned by the DmiCiRegister)
  - length of DMI command block
  - data of the DMI command block (length bytes)
  - length of DMI confirm block
  - data of the DMI confirm block (length bytes)
- client retrieves the CI handle and command length, then allocates memory and reads the command into the memory. It the gets the confirm length, allocates memory for it and reads the confirm data into the buffer.
- client finds info about the CiInvoke routine to call
- client calls desired routine with the command and confirm
- client sends a response to the server with:
  - status code from the DmiCiInvoke
  - length of the DMI command block
  - data of the DMI command block (above length bytes)
  - length of the DMI confirm block
  - data of the DMI confirm block (above length bytes)
- server awakes out of the clnt_call(CI_INVOKE) and places the result command and confirm buffers into the appropriate memory
- server returns to the DMISL

NOTE:

- the client does not respond to the clnt_call until the CiInvoke processing finishes. If the command takes a long time to process, other requests are postponed until the one in progress completes. This could be corrected by having the client respond immediately that it received the request and then the server would return to the DMISL. When the client completed the command it would send a command to the server indicating such, the server would then pass that info on to the DMISL.
- as a consequence of this waiting for processing to complete, the server may time out in it's connection with the client if a command takes a long time to process. (the timeout is set to 30 sec.)

**SERVER INDICATE OF CLIENT:**

- DMISL calls the "allocated" indicate function in the server code
- server byte-swaps the indicate buffer if necessary for the client
- server sends a INDICATE command to the client which contains:
  - MA handle of the component (returned by the DmiMgmtRegister)
  - length of DMI indicate block
  - data of the DMI indicate block (length bytes)
- client retrieves the MA handle and indicate length, then

allocates memory and reads the indicate into the memory.
- client finds info about the indicate routine to call
- client calls desired routine with the indicate buffer
- client sends a response to the server with:
  - NULL
- server awakes out of the clnt_call(INDICATE) and returns to the DMISL

NOTE:
- the client does not respond to the clnt_call until the indicate routine finishes processing. If processing takes a long time, other requests are postponed until the one in progress completes. This could be corrected by having the client respond immediately that it received the request and then the server would return to the DMISL. When the client completed the command it would send a command to the server indicating such, the server would then pass that info on to the DMISL.

**CLIENT INDICATE OF SERVER:**

- client sends an INDICATE command to the server which contains:
  - ID of the client making the call
  - length of DMI indicate block
  - data of the DMI indicate block (length bytes)
- server retrieves the clint ID and indicate length, then allocates memory and reads the indicate into the memory.
- server byte-swaps the indicate buffer if necessary for the client
- server calls the DMISL with the indicate
- server sends a response to the client with:
  - NULL
- client awakes out of the clnt_call(INDICATE) and returns

NOTE:
- the server does not respond to the clnt_call until the DMI indicate routine finishes processing. If processing takes a long time, other requests are postponed until the one in progress completes. This could be corrected by having the server respond immediately that it received the request and then the client would return to the DMISL. When the server completed the command it would send a command to the client indicating such. This would require callback on the client.

**CLIENT INSTALL MIF TO SERVER:**

- client sends an INSTALL_MIF command to the server which contains:
  - ID of the client making the call
  - length of MIF file data
  - data of the MIF file
  - length of message buffer

- server retrieves the client ID and MIF data length, then
  allocates memory and reads the MIF data into the memory.
  It then retrieves the length of the message buffer and allocates
  memory for the message buffer.
- server calls the DMISL API DmiInstall to install the MIF.
  Any messages from the DmiInstall callback routine are placed
  into the message buffer.
- server sends a response to the client with:
  - component ID of installed MIF
  - DmiLibStatus returned from the DmiInstall
  - SLStatus returned from the DmiInstall
  - length of any messages placed in message buffer
  - message buffer data
- client awakes out of the clnt_call(INSTALL_MIF) and returns

**CLIENT UNINSTALL COMPONENT FROM SERVER:**

- client sends an UNINSTALL_MIF command to the server which contains:
  - ID of the client making the call
  - ID of component to uninstall
- server retrieves the client ID and component ID
- server calls the DMISL to uninstall the desired component
- server sends a response to the client with:
  - result code of the uninstall
  - ID of the component uninstalled
- client awakes out of the clnt_call(UNINSTALL_MIF) and returns

# APPENDIX C. INSTRUMENTED COMPONENT AGENTS

Object-oriented modeling of the Large Mailing environment enables large volumes of attribute data associated with each step in the mailing process to be captured and maintained in a data base for access by management applications. For example, attributes about products, mail jobs, insert jobs, printers, inserters, supplies, machine setups and finishing instructions may all be found in the object model of various components of an LMO system

Since a typical printing and mailing process for a large production application has many processing steps and generates millions of envelopes for mailing, the tasks of locating particular units of work, monitoring status and importing\exporting data can be quite complex requiring accessing multiple components and attribute groups. This problem is made even more difficult by the need of management applications to have a detailed understanding of how the overall system model is constructed in order to find the answer to common questions such as where is Mailpiece XYZ, what are the supply levels of my devices, and what is the process history for a product.

A novel technique has been defined for LMO System Managers of instrumented components and groups such that they may be used for passing parameters and returning values for complex queries. These instrumented groups serve as Agent programs to other Management applications and are invoked simply by issuing **Set** and **Get** commands using the standard DMI interfaces.

The following instrumented agents have been defined for general use in LMO systems. A description of each agent and their use is provided in the sections that follow:

Reconcile Agent Component
History Agent
Watch Agent
Import\Export Agent
Query\Delete Groups Agent

## RECONCILE AGENT COMPONENT

There are three types of reconcile:

**Reconcile Product**:
- used to reconcile an entire product
- started by setting "LMO Reconcile|Reconcile Product|Operation" to "Reconcile" or "Reconcile & Delete"

**Reprint**:
- used to reprint mutilated mailpieces for a print job
- started by setting "LMO Reconcile|Reconcile Print Job|Operation" to "Reprint"

**Reconcile Print Job**:
- used to reconcile a print job
- started by setting "LMO Reconcile|Reconcile Print Job|Operation" to "Reconcile" or "Reconcile & Delete"

Each type of reconcile (Reprint job, Reconcile print job, Reconcile product) uses a "Control" table to define what data to export and any system command to be executed. Each row of the control tables defines an export/system command pair that is executed during reconciliation. The three control tables are:

Reconcile Reprint Control
Reconcile Job Control
Reconcile Product Control

### RECONCILE AGENT CONTROL TABLES

The attributes of the control tables are:

**Index** A unique key for the row

**Component Name** The name of the component containing the data to be exported.
If the name is empty, nothing is exported

**Group Name** The name of the table to export

**Attribute List** A list of the names of the attributes to export (* for all).

**Condition** Additional conditions on the export. The default conditions are:

For Product:
(Product Name = <prod name> AND
Product Instance Qualifer = <prod IQ>)

For Job and Reprint:
(Product Name = <prod name> AND
Product Instance Qualifer = <prod IQ> AND
Print Job Name = <job name>)

If you supply something in the condition attr, it is concatenated on the end of the above condition. For example to export mutilated mailpieces: condition = "AND Current Status = Mutilated"

**Path Name** The full name of file to export into.

**File Control** A value controlling how the data is written to file.

0 - overwrite the file

1 - append to the file

**Time Out** The number of seconds to wait for the export to finish..

**Command** A system command to be executed after the export; if empty, no command is issued. This can be a program or script command.<prod name>, <inst qual> (and <pjob name>) are passed as the first 3 arguments to the command; replaceable parameters are available (see below). Note: the code waits for the command to finish. If you want to spawn off a process and not wait, the command must start another process in background. The program should should return zero on sucess, any other value sets the "ERROR" status of the reconcile.

**Delete Row** Deletes this row from the table

The rows are currently exported in "random" order which is actually a first-in/first exported order. They are NOT executed in the order of the indices since the DMI cannot order rows. If the export or command fails for a row, the agent goes to the next row after setting the Status to Error.

## RECONCILE AGENT COMMUNICATION SEQUENCES

The sequence of communications for reconcile functions are summarized below.

REPRINT JOB:

- the user sets the <Product Name>,<Instance Qualifier>,<Print Job Name> in the <Reconcile Print Job> group
- the user then sets the <Operation> attribute to "Reprint"

The data to be exported (and optionally deleted) is defined by the rows in the "Reconcile Reprint Control" table of the reconcile MIF as described above.

RECONCILE PRINT JOB:

- the user sets the <Product Name>,<Instance Qualifier>,<Print Job Name> in the <Reconcile Print Job> group
- the user then sets the <Operation> attribute to one of the following values (when a "set" to this attribute is performed, a notify is issued from LMODB2 that is intercepted by LMORECON which performs the reconcile tasks):

"Reconcile & Delete"

- this exports and then deletes data

"Reconcile"

- this just exports data

The data to be exported (and optionally deleted) is defined by the rows in the "Reconcile Job Control" table of the reconcile MIF as described above. In both instances, the code sets the <Current Status> of the indicated <Print Job> to "Reconciled"

RECONCILE PRODUCT:

- the user sets the <Product Name>,<Instance Qualifier> in the <Reconcile Product> group
- the user then sets the <Operation> attribute to one of the following values (when a "set" to this attribute is performed, a notify is issued from LMODB2 that is intercepted by LMORECON which performs the reconcile tasks):

"Reconcile & Delete"

- this exports and then deletes data associated with the product

"Reconcile"

- this just exports the data associated with the product

The data to be exported (and optionally deleted) is defined by the rows in the "Reconcile Product Control" table of the reconcile MIF as described above.

#### SYSTEM COMMANDS

The system command,part of the control entry),operates as such:

**AIX**:

- only executable files are accessible (i.e. we do an exec), thus REXX scripts are not available (though you could probably set up an intermediary shell script to call the desired REXX script).
- the arguments are set up into the appropriate argv entries

**OS/2:**

- either executable programs or REXX scripts are allowed. The code checks the command name (the first token of the command string) for ".cmd" in the name - if it is there, it assumes it is a REXX script. Otherwise, it calls spawn() to execute the command. If the spawn fails, it will try it as REXX.
- for REXX, it operates the same as if you typed the command at the prompt - i.e. the arguments all come into REXX as one string (i.e. one argument).
- for executable programs, the arguments are passes individually in the argv array

The command operates pretty much exactly the same as if you typed the command at the operating system prompt.

There are three (two for reconcile product) default parameters that are passed to the command before the parameters that are given on the command line:

Product Name
Product Instance Qualifier
Print Job Name (not passed in for reconcile product)

There are replaceable parameters available:

@FILE - file name from the control entry
@PROD - product name from reconcile entry
@PIQ - product instance qualifier from reconcile entry
@PJOB - print job name from reconcile entry (not available for reconcile product)
@<comp>|<group>|<attr>(<keys>)
- gets the indicated value and uses it as the value for the argument
- <keys> can contain any of the above replaceable params

**RECONCILE EXAMPLES**

Given below are are some sample reconciliation specifications.

| Product Reconciliation Example 1 |
|---|

```
Reconcile Product:
   Product Name                  = TTQS001
   Product Instance Qualifier    = STRESS02
Reconcile Product Control:
  Component Name                 = LMO Product
  Group Name              = Product Table
  Attribute                      = *
  Condition               =
  Path Name               = demo/prod.exp
  File Control            = Overwrite
  Time Out                = 300
  Command                 =
```

This will set up an export with the above values, except that the condition will actually be: (Product Name = TTQS001 AND Product Instance Qualifier = STRESS02). It will wait up to 300 secs for the export to finish. The system command is empty, so nothing is done.

| Product Reconciliation Example 2 |
|---|

```
Reconcile Product:
  Product Name            = TTQS001
  Product Instance Qualifier     = STRESS02
```

Reconcile Product Control:

| | |
|---|---|
| Component Name | = LMO Product |
| Group Name | = Product Table |
| Attribute List | = * |
| Condition | = OR Product Name = TempProd |
| Path Name | = demo/prod.exp |
| File Control | = Overwrite |
| Time Out | = 300 |
| Command | = prodrecon -c |

This will set up an export with the above values, except that the condition will actually be: (Product Name = TTQS001 AND Product Instance Qualifier = STRESS02) OR Product Name = TempProd. After the export finishes (assuming no time out), the system command will be executed. The effective command will be: prodrecon TTQS001 STRESS02 -c.

### Reconcile Job Example

Reconcile Job:

| | |
|---|---|
| Product Name | = TTQS001 |
| Product Instance Qualifier | = STRESS02 |
| Print Job Name | = TTQSPJ01 |

Reconcile Job Control:

| | |
|---|---|
| Component Name | = LMO Mail Job |
| Group Name | = Mailpiece XReference Table |
| Attribute List | = * |
| Condition | = AND Current Status = Mutilated |
| Path Name | = demo/mute.exp |
| File Control | = Overwrite |
| Time Out | = 300 |
| Command | = pjobrecon -f @FILE |

This will set up an export with the above values, except that the condition will actually be: (Product Name = TTQS001 AND Product Instance Qualifier = STRESS02 AND Print Job Name = TTQSPJ01) AND Current Status = Mutilated. After the export finishes (assuming no time out), the system command will be executed. The effective command will be: pjobrecon TTQS001 STRESS02 TTQSPJ01 -f demo/mute.exp

### Reconcile Reprint Job Example

Reprint Job:

| | |
|---|---|
| Product Name | = TTQS001 |
| Product Instance Qualifier | = STRESS02 |
| Print Job Name | = TTQSPJ01 |

Reconcile Job Control:

| | |
|---|---|
| Component Name | = LMO Mail Job |
| Group Name | = Mailpiece XReference Table |

```
Attribute List              = *
Condition                   = AND (Current Status = Mutilated OR
                              Current Status = Diverted)
Path Name                   = demo/mute.exp
File Control                = Overwrite
Time Out                    = 300
Command                     = reprint @FILE "@LMO Product|
                                Print Job Table|Print File Name
                                (@PROD,@PIQ,@PJOB)"
```

This will set up an export with the above values, except that the condition will actually be:

```
(Product Name = TTQS001 AND Product Instance Qualifier = STRESS02
 AND Print Job Name = TTQSPJ01) AND
 (Current Status = Mutilated OR Current Status = Diverted)
```

After the export finishes (assuming no time out), the system command will be executed. The effective command will be: reprint TTQS001 STRESS02 TTQSPJ01 demo/mute.exp JOBNAME. This assumes that the value the "Print File Name" in the Print Job Table for the given print job is "JOBNAME"

## WATCH AGENT COMPONENT

The LMO Watch Agent watches and keep statistics on certain attributes. The agent provides a Watch Control Table to define what data is to be watched and a Watch Table to maintain the resulting statistics gatered from the watch.

### WATCH CONTROL TABLE

The attributes of the control table are:

**Component Name** the name of the component which contains the attribute we want to watch

**Group Name** the name of the group within that component which contains the attribute we want to watch

**Attribute Name** the name of the attribute within that group that we want to watch

**Key List** a list of key values that uniquely identify the one row of the group that we want to watch. The format is: <value>[,<value>] No blanks should surround the commas since, during indicates, we must build a list to match this one, and the building routine does not add blanks anywhere. String values can be double quoted or not (they only need to be quoted if they contain commas). The Key List must contain values for all keys of the desired watch attribute and they must be in the same order as denoted by the service layer.

**Current Value** the current value of the attribute being watched. If we are polling the attribute, then this is the value last polled, otherwise, it is the value last indicated by an event from the instrumentation.

**Accumulate Value** this indicates what value should be accumulated while the attribute is in one state. If this is "ACCUMULATE_TIME", then the number of seconds spent in each state is accumulated. Otherwise, this is the name of another attribute with the "Group Name" group that is read and used as the value accumulated. The attribute named must be either INTEGER, COUNTER, or GAUGE.

**Time Started** the date/time we started watching this attribute.

**Time Stopped** the date/time we stopped watching this attribute.

**Time Disabled** the date/time we disabled watching of this attribute.

**Duration Watching** the amount of time in seconds that we were actually watching the attribute. This is the

Time Stopped - Time Started - Duration Disabled.

**Duration Disabled** the total amount of time in seconds that we were disabled while watching this attribute.

**Polling Interval** if this attribute should be polled rather than using indicates from instrumentation, then this gives the polling interval in seconds. If this is zero, then we set up for indications.

**Current Status** the current status of this watch:

0 - No Status - no known status for the watch
1 - Started - we are started and watching
2 - Stopped - we are done watching
3 - Disabled - we are started and not watching
4 - Error - an error was detected

**Operation** the operation to perform on this control entry:

0 - Do Nothing
1 - Start Watching - start watching the defined attribute
2 - Stop Watching - stop watching the attribute
3 - Restart Watch - re-start the watch (same as Start Watching)
4 - Enable Watching - restarts watching after being disabled
5 - Disable Watching - disable watching
6 - Delete Watch - deletes all entries in the Watch Table associated with this watch.
7 - Display Watch - displays the Watch Table for this watch (this is only for debugging).

**Delete Row** deletes this row from the Watch Control Table

**Note:** Since this is a normal MIF, all attributes can be read or written, however, some attributes are only used internally by the monitor code. Unpredictable results will occur if the user sets attributes used internally by the monitor code or if the user sets control attributes while the watch is "active" (i.e. not stopped).

### WATCH CONTROL TABLE USAGE

The following attributes should be set before starting a watch and should not be changed while the watch is in progress:

Component Name
Group Name
Attribute Name
Key List
Accumulate Value
Polling Interval
Operation

The following attributes are used internally by the watch code:

Current Value
Current Status
Delete Row
Time Disabled

The following attributes can be read after stopping:

Time Started
Time Stopped
Duration Watching
Duration Disabled

**WATCH TABLE**

| | |
|---|---|
| **Component Name** | the name of the component which contains the attribute this entry refers to |
| **Group Name** | the name of the group within that component which contains the attribute this entry refers to |
| **Attribute Name** | the name of the attribute within that group that this entry refers to |
| **Key List** | same as the Key List entry in the Watch Control table. |
| **Value** | the value (state) of the attribute for this entry |
| **Previous Value** | the prior value the attribute had before entering this state. |
| **Start Count** | the value the Accumulate Value had when this attribute entered this state. For ACCUMULATE_TIME, this is the time when the attribute entered this state. Otherwise, it is the value read from the indicated Accumulate attribute when this attribute entered this state. |
| **Accumulator** | the total "deltas" of the accumulate values while this attribute was in this state. For ACCUMULATE_TIME, this is the total number of seconds spent in the state. Otherwise, when the attribute changes out of this state, the indicated accumulate attribute is read and the Start Count is subtracted from it, and the result is added to this Accumulator. |
| **Instance Count** | the number of times the attribute was in this state. |
| **Average** | a string showing Accumulator / Instance Count that is set whenever the others are |

**Delete Row** deletes this row from the table. This attribute is used by the Alert code when the "Delete Watch" Operation is performed on the Control Table.

The attributes of the Watch Table containing statistics and status of the data being watched are:

**WATCH TABLE USAGE**

All attributes are maintained internally by the watch code, user programs should only read the rows in this table, it should never set any attributes.

The attributes in the watch table are always valid once the watch has started, however, the "active" row will not contain the accumulated value for the current instance of that attribute.

The following attributes can be read after stopping:

Component Name
Group Name
Attribute Name
Key List
Value
Previous Value
Accumulator
Instance Count
Average

**WATCH EXAMPLE**

To watch the amount of time <Current Status> of the print job identified by "product 1","instance 1","print job 1" stays in each state:

Watch Control Table:

<Component Name> <== LMO Mail Job
<Group Name> <== Print Job Table
<Attribute Name> <== Current Status
<Key List> <== "product 1","instance 1","print job 1"
<Accumulate Value> <== ACCUMULATE_TIME
<Polling Interval> <== 0
<Operation> <== 1 (Start Watching)

This turns on indicates for the attribute (because no polling interval). When the <Current Status> attribute changes state for this print job, an indicate is issued and caught by the monitor code. It then "closes" the current entry in the <Watch Table> and "opens" a new entry in the <Watch Table>.

When you want to stop watching the attribute set the operation to:

<Operation> <== 2 (Stop Watching)

This will close all entries in the <Watch Table> for this watch. You can then read all of the entries in the <Watch Table> with the same <Component Name>,<Group Name>,<Attribute Name> and <Key List> to see the total time, number of times, and average time spent in each state that <Current Status> was in while the monitor was active.

The percentage of total time spent in one state can be calculated from <Watch Table>|<Accumulator> divided by <Watch Control Table>|<Duration Watching>.

## HISTORY AGENT COMPONENT

The LMO History Agent provides a way to keep an "audit history" of when attributes change
state.

Essentially, this provides a means of performing "keep track of whenever attribute X changes state".
For instance, if we want to keep track of all the times any print job status changes, this agent will keep a history of the time the status entered and exited a state. You could then look in the history table and be able to see that the status for print job 1 entered the created state at time A, left that state and entered the printing state at time B, went to printed at time C, went to the printing state (again) at time D, went to printed (again) at time E, etc.

The values of all keys from the target MIF, the values of the watched attributes and the values of the save attributes are all saved in the history table whenever any one of the watch attrs changes. The values saved are read when the row is "opened".Only non-keys can be watched (since keys never change state).
The list of keys, list of watch attrs, and list of save attrs must be mutually exclusive.

The frequency of polling the attributes being watched is defined for each separate history.This feature is designed to be used to watch attributes that do not change state quickly, (i.e. it is probably not appropriate for attributes that change state many times a second or minute). Nor is it designed to watch tables which can have lots of rows. The polling of the attributes to see if they have changed state is relatively expensive, and the more rows that it must check, the more costly it will be.

The attribute(s) to watch must be in the same group.The attribute(s) to save can be in any group.

### HISTORY CONTROL TABLE

The attributes of the History Control Table are:

| | |
|---|---|
| **Name** | Name of this history entry |
| **Component** | Component to watch |
| **Group** | Group within that component |
| **Key List** | Key values for that group |
| **Watch Attrs** | Attr names to watch for changes |
| **Save Attrs** | Attr names to save when change |
| **Poll Interval** | Polling interval in secs. |
| **File Name** | File name for export and create |
| **Last Poll Time** | Last time polled |
| **Status** | Status of history |

started
stopped
purged
exported
MIF created
error

| | |
|---|---|
| **Message** | Error message |
| **Operation** | Operation to perform |
| | start |
| | stop |
| | purge |
| | export |
| | create MIF |
| | reset |
| **Delete Row** | Delete this row |

Syntax:

<watch attrs> ::= <attr>[,<attr>]
<attr> ::= attribute name within the component/group defined
<save attrs> ::= <save attr>[,<save attr>]
<save attr> ::= <attr>| <non-local attr>
vertical bars below necessary
<non-local attr> ::= <comp>|<group>|<attr name>[<key values>]
<comp> ::= <static comp> | <indirect comp>
<indirect comp> ::= @<attr>[<example comp>] ;; brackets necessary
<key values> ::= (<key value>[,<key value>]) ;; parentheses necessary
<key value> ::= <static value>| <indirect value>
<indirect value> ::= @<attr>

**HISTORY AGENT USAGE**

To use the History Agent function, the user sets:

| | |
|---|---|
| Name | the name of this history watch |
| Component | name of component to keep history on |
| Group | name of the table withing that component being watched |
| Key List | list of key values for the table<br>If list is empty, watch all rows in the table<br>otherwise, the list is used as an initial subset<br>of the keys for the group (i.e. you do not need<br>to give all key values, but if any are given, they<br>must be given in order) |
| Watch Attrs | list of attribute names within group to watch |
| Save Attrs | other attribute names within group or in other comp/group<br>that are read and saved whenever any of the Watch<br>Attrs change. |

Poll Interval polling interval in seconds
File Name name of the file to be created by the export operation, or
the name of the MIF file to create
Operation
start - starts keeping history as defined
stop - stops keeping history
purge - empties the history tables, does not delete the control entry
export - exports the data in the history table to a file
create MIF - creates a MIF file to correspond to this entry
reset - resets this history - status goes to STOPPED, purges all
data and makes sure polling is stopped
Delete Row deletes this row in the history control table

Other attrs:
Last Poll Time used by the history agent to keep track of when it last polled
the table values
Status indicates current status of this history entry
Started the history is currently running
Stopped the history is not running
Purged data has been purged from the history table
Exported data has been exported to a file
MIF created the MIF has been successfully created for this history
Error an error has occurred during processing
Message if status is set to error, this gives a message
about the error (note that some errors create multiple
messages, this will be filled with the last message).
It may also be filled with a useful warning message.

**HISTORY AGENT EXAMPLE**

To keep history on the status changes of print jobs for a specific product the following attributes may be set:

History Control Table:

Name <== "Print Job History"
Component <== "LMO Mail Job"
Group <== "Print Job Table"
Key List <== "product 1"
Watch Attrs <== "Current Status"
Save Attrs <== "Print File Name,Actual Print Job Size,
Current Device"
Poll Interval <== 30

**COMMUNICATION SEQUENCE**

First set Operation to "Create MIF". This would create a MIF:

Component "Print Job History"
Group "History Table" (instrumented)
keys 1,2,3,4
1 - "Product Name"
2 - "Product Instance Qualifier"
3 - "Print Job Name"
4 - "Start Time"
5 - "Stop Time"
6 - "Current Status"
7 - "Print File Name"
8 - "Actual Print Job Size"
9 - "Current Device"
Group "Active History Table" (instrumented)
keys 1,2,3,4
1 - "Product Name"
2 - "Product Instance Qualifier"
3 - "Print Job Name"
4 - "Start Time"
5 - "Last Poll Time"
6 - "Current Status"
7 - "Delete Row" (must be SPECIAL_DELETE in lmoinstr.cfg)

- the system would then install this MIF with the SL

Then set Operation to "Start". This will start the watching of the defined attributes.

## UTILITY AGENT GROUPS

The utility MIF contains instrumented groups that are used by many of the agent programs previously described and by any other DMI Management application that desires to use them. These additional instrumented groups are describe below.

### IMPORT EXPORT GROUP

The Import/Export group has the following attributes:

**Component Name**
the component name of the information to import/export
**Group Name**
the group name within the component of the information to import/export
**Attribute List** (not used by delete)
- the list of attributes to export or import
- the format of this string is:
"" - export/import all columns
* - export/import all columns
<MIF attr name> [,<MIF attr name>] - export/import the listed columns
- if the value is "SQL_STATEMENT", then the "condition" defines a SQL statement
that is executed directly, no parsing takes place.

**Condition** (not used by import)
- the search condition for export or delete.
- if the first 8 characters of the condition are DISTINCT, then the SQL query is generated such that result table contains no duplicate rows.
- this string is pretty close to SQL syntax except that MIF attribute names are used, if strings are quoted, double quotes should be used, and enums are expressed by name rather than value
- the format is:
"" - we want all rows in the table
"*" - we want all rows in the table
or, when we want specific rows:
<where_clause> ::= <expr> |
[ ( ] <expr> <bool_oper> <expr> [ ) ]

<expr> ::= [ ( ] <MIF attr name> <cond_oper> <value> [ ) ] |
NOT <expr>
<cond_oper> ::= "=" | ">" | "<" | "<>" | "<=" | ">=" | "LIKE"
<bool_oper> ::= "AND" | "OR"
<value> ::= MY_COMPONENT_NAME |
MY_COMPONENT_ID |
MY_GROUP_NAME |

MY_GROUP_ID |
MY_ATTRIBUTE_NAME |
MY_ATTRIBUTE_ID |
VALUE_OF <MIF attr name> |
<user-supplied value>
<user-supplied value> ::= <decimal number> |
<string> |
"<string>"

-Note that strings can be quoted or not. The string only needs to be quoted when it contains any of the pre-defined delimiters or keywords (e.g. =,<,AND).
-very little checking is done for valid format, basically it just changes <MIF attr name> into the appropriate DB2 column name and the <value> into a quoted string if the type of the column requires it. Thus, it does not check for such things as matched parentheses or that <user-supplied value> for an integer is actually a string of digits after converting names and values, it just passes the resulting string on to DB2 which may reject it for various reasons

**Path Name** (not used by delete)
- the file name to use for importing/exporting

**File Control** (not used by import, delete)
- controls the way the file is opened:
  0 - Overwrite - the file is truncated and opened
  1 - Append - the file is appended to

**Current Status**
- indicates the status of the current operation:
  0 - No Status
  1 - Busy
  2 - Done
  3 - Error

**Message**
- filled with a message when Current Status is set to "Error"

**Operation**
- used to actually start the operation
- an enum that should be set after the other fields of this group have been set up. The setting actually starts the operation:
  1 - "Export Data"
  2 - "Import Data"
  3 - "Export and Delete Data"
  4 - "Delete Data"

**Delete Row**
- when set to 1, deletes this row from the table

The data exported to the file is formatted as follows:

- the file is ASCII comma-delimited
- each line represents one row of data that matches the Condition above.
- within each line, the values of the attributes defined by the Attribute List above are separated by commas
- if the attribute value is a string, it is surrounded by double quotes.
- if the attribute is numeric, it is not quoted and is printed in decimal.
- each row ends with a new-line (\n)

The data imported must be in the same format as described above:

- The number and type of each value in a row of the file should match the respective column name in the attribute list. They do not need to be in MIF order unless the attribute list is *, in which case they do need to be in MIF order. Otherwise, the order must match the order given in the attribute list.
- The column list must include at least all keys for the group (either implicitly with a star, or explicitly). And the keys must define a unique row for the table (either to be updated or to be added).
- There can be more values in each row of the input file than listed in the attribute list, the extra values are ignored.
- The import code first tries to update a row whose key values match the values read from the file for the associated key column. The other non-key columns from the column list are updated if the row exists.
- If the row does not exist, then a new row is created in the table. If the values for the new row are not complete (i.e. undefined columns) then default values are placed in those columns.

**IMPORT EXPORT EXAMPLES**

- to perform an export of all "Mutilated" mailpieces for a given product:

```
Component Name   <== "LMO Mail Job"
Group Name       <== "MailPiece Xreference Table"
Path Name        <== "mute.lst"
Attribute List   <== "Mailpiece Identifier, Process Assigned Identifier,
                     Product Name, Product Instance Qualifier,
                     Print Job Name"
Condition        <== "Product Name = product 1 AND Status = Mutilated"
File Control     <== "Overwrite"
Operation        <== "Export Data"
```

- to perform an export of all MUTED mailpieces for two products:

```
Condition        <== "((Product Name = product 1) OR
                     (Product Name = product 2)) AND Status = MUTED"
```

Example export conditions:

- NOT (<name> = <value>)

- <name> <> <value>
- (<name> < "<value>") AND (<name> >= "<value>")
- <name> LIKE "<value>%"

Import Example:

- to perform an import (or update if any rows already exist based on the keys for the group) of all columns into the product table:
  Component Name <== "LMO Product"
  Group Id <== "Product Table"
  Path Name <== "product.lst"
  Attribute List <== "*"
  Condition <== N/A
  Operation <== "Import Data"

**QUERY ROWS GROUP**

The Query Rows instrumented group queries the contents of specific attributes for specified rows in a table.. The Query Rows group has the following attributes.

**Component Name**
the component name of the information to be queried

**Group Name**
the group name within the component of the information to be queried

**Attribute List**
- the list of attributes to return in the query response
- the format is same as described above for import/export group
- if the value is "SQL_STATEMENT", then the "condition" defines a SQL statement that is executed directly, no parsing takes place.

**Condition**
- the search condition that will define the rows to be queried
- this string has the same format and meaning as described above for the import/export group

**Query Rows**
- used to actually get the query data
- the user should "GET" this attribute and what will be returned is the same information as though this was an EXPORT, but the information will be returned as a string where each column value is separated by a comma, string values are double quoted, and each row is separated by a line-feed character.
- the string returned will probably be larger than the size of this attribute (i.e. the size of the Query Rows attribute is 10, but the returned string can be much longer). The code returns as much queried data as will fit in the DMI confirm buffer.
- the user should not "GET" this attribute until the other attributes have been set up.

# APPENDIX D. DMISET TAG LOGICAL ELEMENTS

## OVERVIEW

Tags can be placed in AFP data streams by applications or value add programs to associate attributes such as account numbers and mailpiece characteristics to specific pages or groups of pages in the print file.

Tags are defined by Tag Logical Elements (TLEs) structured fields. Each TLE structured field specifies an attribute name, attribute value pair.

TLEs can be created in three ways. The first method is to use the AFP Conversion and Index Facility (ACIF) to automatically create tags and index objects for pages based on scanning the print data and looking for matches defined by the ACIF user to conditionally generate the tags. The second method is to use the AFP Work Bench API to insert TLEs in print data generated by an application. The third method is to provide value add software capable of building TLE structured field records in the AFP format and inserting them in appropriate locations in the print stream.

In the first release, LMO host support will make use of TLE structured fields to set attributes in components being managed through the Desktop Management Interface DMI. DMISET attribute tags may be inserted in the print data using any of the three methods previously described.

TLEs containing DMISET attributes will be interpreted in a PSF (Print Service Facility) exit program. Based on the value field of the TLE, the exit program, in turn, will generate the corresponding DMI calls to set attributes in the Management Information Format (MIF) data base for LMO components. The DMISET values are generic to the MIF structure and can therefore be used to set any attribute, in any component where the attribute is defined to have read/write access in the system.

## RULES FOR USING DMISET TLES

The following rules apply to the definition of DMISET TLEs in print data.

1. LMO will make use of the existing TLE structure, as used by ACIF. Specifically, there is ONE attribute/value pair per TLE.
2. The TLE Attribute field will contain the characters ‘DMISET’, in either ISO 8859-1 (single-byte) or Unicode (double-byte) format. (See below)
3. All the values required to perform a DMI SetAttribute will be contained in the single TLE Value field.
4. The first ‘token’ (see below) of the TLE Value field will be the delimiter to be used for parsing the remainder of the TLE Value field.
5. The entire collection of fields (listed below) cannot exceed 250 bytes (255 - 2_byte_length - 1_byte_type - 2 reserved bytes).
6. Names, NOT IDs, are specified in TLE Value fields. For example, ComponentName rather than ComponentID is specified to reference a specific component. Integers are expressed as display strings, not binary data, in TLE Value fields.
7. The sequence of subfields in the TLE Value field will be as follows:

    . delimiter

    . ComponentName

    . delimiter

    . GroupName

    . delimiter

. Number of Keys

. delimiter

. Key1_value

. delimiter

. Key2_value

. delimiter

. Keyn_value

. delimiter

. Attribute1_Name

. delimiter

. Value1

. delimiter

. Attribute2_Name

. delimiter

. Value2

. delimiter

. Attribute_Name

. delimiter

## IMPLEMENTATION NOTES

PSF Exit processing will determine the format (encoding) of the data based on the value of the TLE Attribute, as follows:

- If the Attribute is NOT equal to 'DMISET' in either single or double byte encoding, the record is not processed (by LMO).
- If the Attribute is equal to the single-byte form of 'DMISET', the first BYTE of the TLE Value field is considered to be the delimiter, and the data is scanned one byte at a time.
- If the Attribute is equal to the double-byte form of 'DMISET', the first TWO bytes of the TLE Value field are considered to be the delimiter, and the data is scanned two bytes at a time.
- Once the parsing format is determined, the input parameters are parsed, MIF meta-file definitions are used to convert values to their required formats, and a DMI Invoke is executed to set the named DMI attribute to the specified value.

# GLOSSARY

**attribute.** A characteristic of an object.

**capacity units.** The units in which the capacity of a device is measured or reported. For example the capacity units for a roll feed input device may be 100s of feet, while the capacity units for an input tray may be reported in terms of pages or inches of pages.

**consumable resource units.** The units in which the consumable resources used by a device is measured or reported. Examples of consumable resource types include toner, ink, and stitching wire.

**document.** When not otherwise qualified, a document is assumed to be the electronic form of a document, in a form acceptable to print servers, viewers, archivers and other document processing components to which the document is (to be) sent.

**document-format.** The generic name for any data syntax in which a final form electronic document may be encoded. For example, in AFP architecture the document format for final form electronic document interchange is MODCA, Mixed Object Document Content Architecture.

**event.** An occurrence of significance to a process or device. Device error, state change, supply add and process complete are examples of events

**finisher device.** A finisher is a device that performs operations on a medium other than marking. Some examples of finishing processes performed by a finisher device are stapling, punching, trimming, binding, and banding. A given finisher device may support more than one finishing process.

**Inserter.** A collection of devices which can accumulate related papers, fold them, add selective inserts, stuff the collected papers in an envelope, seal and meter them, and stack the finished mailpiece.

**Insert Job.** A mailjob that is ready for the inserter.

**Job Ticket.** A set of specific finishing instructions for processing a particular mail job. a job ticket may contain setup instructions for the inserter, operating instructions for the inserter, and structure information for the inserter to correlate the paper structure to the instructions. An *architected* job ticket is one that conforms to a consortium standard. a *non - architected* is one that conforms to the standard of a particular vendor.

**LMO.** Large Mailing Operation.

**Large Mailing Operation.** A collection of hardware and software components designed to work together to automate output processes associated with generation, printing, mailing and tracking large volumes of mailpieces destined for delivery to individual recipients, often via postal services.

**Mailpiece.** The envelope for mailing and its entire contents.

**Mail Job.** A collection of mail pieces that represent a unit of work in an LMO system. A Mailjob results from segmenting output from product applications into manageable pieces for printing, inserting, postal processing or re-print of MailPiece data.

**Mail Job Table.** A table of MailJob associated with a product.

**Marker Device.** A mechanism that produces marks on the medium.

**Mutes.** Mutes or mutilated documents are mailpieces which have been damaged or fail to meet quality criteria such that they are not suitable for mailing in their present condition.

**Process Path.** An ordered sequence of processes to be performed on a MailJob.

**Product.** Unless qualified otherwise, the term Product refers to a customer application producing print output to be processed by the LMO system. Attributes of a product, such as product name, date of output creation, owner, mail job segmentation and current processing state are maintained as work flows through the LMO system.

**Reconciliation.** Verification that all mailpieces in a job have been trayed for mailing.

**Traying.** Placing completed mailpieces into containers provided by the U.S. Postal Service for post office processing.

**Value Add (VA).** A software process whereby the print dataset is enhanced. Examples are correcting invalid addresses, adding postal bar codes, and sorting mailpiece statements by postal zip codes.