OfficeVision/VM Migration Tool
Tips and Techniques

Michael Hilborn
Willi Monschau

IBM Lotus Integration Center
ilicna@us.ibm.com

IBM Lotus Integration Center EMEA
ilicemea

Contact points:

IBM Lotus Integration Center North- and Latin America

Dallas, Texas
ilicnaus.ibm.com
phone: (817)962-3100

IBM Lotus Integration Center EMEA

Boeblingen, Germany
ilicemea
phone: (49)7031-16-6052

First Edition, December 1997.

The changes to the document will be described here. .

First edition	December 1997
Notelog Performance Support	January 1998
Migrating VM Databases	February 1998
Migrating OV/VM Conference Rooms	March 1998
Migrating OV/VM Bulletin Boards	April 1998
Migrating 'To Do' Files	April 1998
Maintaining ‘Read/Unread’ Status	April 1998
Document Migration Method 3	July 1998
Migrating AWAY Entries	September 1998

Introduction

Today's OV/VM Migration Tool allows the customer to migrate a broad range of OV/VM and VM/ESA data. The migration of some specific data has been deliberately exempted because, for the time being, there was not a general solution that would satisfy all or at least a majority of customers.

Things that are not yet included are:

Document migration except those contained in the in-basket
User specific migration configuration
Migrate Document Meetings
Migration of CMS notebooks as notelogs
Migration of reminders
Support for holiday file
Time Zone Support
Migration of addenda dictionaries

This document and the accompanying procedures intend to provide information and samples for parties interested in extending the functionality of the OV/VM Migration Tool.

Information in here is provided on an as-is basis. Procedures are samples and are not maintained in any way. They are only provided as a starting point for your own development. These procedure usually
only show simple solutions, are not thoroughly tested and do not include all the required error handling and backout processing.

Whoever will use these samples needs to have good skills in coding in REXX, managing Ids and disks on a VM system, and managing and using OV/VM.

Document Migration

Current Situation

In the first release of theOfficeVision/VM Migration Tool, documents are only handled if they are referenced in the in-basket.

This is usually not sufficient for customers who really use the OV/VM document store.

The usage of the OV/VM document store varies between different customer situations. What we see is:

1. It is just used as a central place to store documents that are usually not shared between users.
2. Documents stored are usually shared by sending them as references to other users.
3. Documents are stored by central functions and are heavily shared by sending to large distribution lists. Normally used as bulletin board function. Sometimes accessed through special interfaces where these documents are referenced so that they are not required to be sent to the whole user community.
4. Sub-database functionality is used. Users only access their own sub-database or their department's sub-database. Documents are sometimes restricted with access to only a limited number of users.
5. More....

All these different implementations of the document database result in different requirements for migration. There is often a requirement for Notes side application development to be able to provide similar functionality with limited chance for a migration tool to be included.

In some cases though, the Office Vision/VM Migration Tool can be used to at least do the transport of data into a Notes database that can then be accessed directly by the users or that can be available for document extraction and importation into a special application database.

For documents that are only stored by single users or documents which are not shared often - cases 1 and 2 - a migration could be done as part of the normal migration process. Here, either all documents of a user or documents that a user has marked for processing could be migrated for the user and stored into a database that will end up at the user's workstation. If such documents are assumed to be public then they could also end up in a central database on a server.

Documents that are heavily shared between users and, especially for a bulletin board type of usage, should be migrated from a central document server userid that would migrate all documents in the database or in a sub-database to a central Notes database.

The same is true if documents are stored in departmental sub-databases. They should probably be migrated from a central document server id and end up in server based databases which are only accessible by defined groups of users.

Limitations

Although we think that in the following chapters we do provided methods that can help with migrating documents to notes databases, this does not mean that we consider them to be the complete solutions.

The thing to keep in mind here is that the number of documents and the amount of data associated with them can easily reach a level that is too much for a single Notes database.

So, a controlled process needs to be put in place to make sure that whatever is migrated from an OV/VM database is usable as a Notes database at the end.

This could mean that the target would not be just one database but multiple databases. This could be done manually by providing different names for target databases while migration goes on.

It also could mean that special databases need to be developed for a certain customer situation or that, at the end of the migration, the documents would need to be moved into other databases, maybe even into Domino.doc.

Document Migration - Method 1

Problem Description

In the method described here, we wish to migrate documents which the user can access through a document log. The standard OfficeVision/VM Migration Tool only permits the migration of documents which are mailed to your in-basket.

As users are migrated, OFSINDEX OFSDATA or perhaps another document index file is referenced, documents are extracted from the OV/VM document database and are placed in the MDF for the user. The MDF indicates that the documents are destined for a server-based database and the ACL is set to permit only each user to access their own documents.

Extension Description

As each user is migrated, their documents are migrated into a server-based database called DOCS.NSF. In order for the user to access their migrated documents, they would add an icon representing this database to their Notes workspace. Choosing File->Database->Open, then
select the server containing the database would show the list of databases on that server. They would then choose the database entitled ‘OV/VM Documents’.

Documents are extracted from OV/VM and then formatted. They are **not** attachments to a memo. They will be placed into one of these categories:

Softcopy - DCF or RFT final or draft documents
Hardcopy - Hardcopy (paper) documents
File - Files stored in the database (these will be attachments)

Additionally, their status will be set to one of:

Ready for Review - if a draft document
Review Complete - if a final document
Other - for files and hardcopy

These OV/VM API's are used:

EPUHIDOC - to extract the document from the database
EPRHFORM - to format a draft script document
XFORM - to format an RFT document

NLS constants and default settings are contained at the start of LNMOVDOC EXEC.

The code also implements these new options in LNMMIGR CONFIG:

MIGRATE_OVVMDOCS
Set MIGRATE_OVVMDOC to YES if documents in an index file are to be migrated.
MIGRATE_DOCNAME
Set MIGRATE_DOCNAME to PC filename of server based document NSF.
MIGRATE_DOCTITLE
Set MIGRATE_DOCTITLE to database title as end user sees it.

MIGRATE_DOCFILE
Set MIGRATE_DOCFILE to filename and type of user's document index file.
MIGRATE_DOCNTF
Set MIGRATE_DOCNTF to document database template name.
MIGRATE_DOCCATS
Set MIGRATE_DOCCATS to YES if documents should be categorized. Categories are DCF, RFT and File.
MIGRATE_DOCSTATUS
Set MIGRATE_DOCSTATUS to YES if document status should be set. Status can be Ready for Review, Review Complete and Other.
MIGRATE_DOCDATES
Set MIGRATE_DOCDATES to a date in yyddd format. When a document is about to be migrated, the date in the document cron is compared with MIGRATE_DOCDATES. Only documents with cron dates the same as this date or after will be migrated.

Exits Supplied

Exit LNMEX03 is called to process a document extraction which failed. This is documented in the manual. The following exits are additional to LNMOVDOC:

LNMEX20 Return one or more document index entries

If this exec doesn't exist, the document index file listed in option MIGRATE_DOCFILE is read. Otherwise, you must return data in the CMS stack in the same format as is contained in OFSINDEX OFSDATA.

Input Filename and type from MIGRATE_DOCFILE.

Output Document index records must be in the CMS stack. The following data from the index records are used:

1. cron - columns 1 to 12
2. sender userid - columns 33 to 40
3. sender nodeid - columns 41 to 48
4. subject - columns 66 to 110

Return Codes <0 means the migration of this user will be terminated. Otherwise, processing continues.

LNMEX21 Return a category for a document

If this exec doesn't exist, the default category for the document will be set. Otherwise, you must return data in the CMS stack.

Input Filename, type and mode of the document on disk, cron, filename and type of the document in the database and the default category.

Output New category must be in the CMS stack.

Return Codes <0 means the migration of this user will be terminated. Otherwise, processing continues.

LNMEX22 Return a status for a document

If this exec doesn't exist, the default status for the document will be set. Otherwise, you must return data in the CMS stack.

Input Filename, type and mode of the document on disk, cron, filename and type of the document in the database and the default status.

Output New status must be in the CMS stack.

Return Codes <0 means the migration of this user will be terminated. Otherwise, processing continues.

LNMEX23 Return one or more DocumentReaders.

If this exec doesn't exist, the document reader for the document will be set to the migrating user's userid. Otherwise, you must return data in the CMS stack.

Input Filename, type and mode of the document on disk, cron, filename and type of the document in the database and the default DocumentReader.

Output One or more DocumentReaders must be in the CMS stack. These must be in ‘useridnodeid’ format, one per stacked line.

Return Codes <0 means the migration of this user will be terminated. Otherwise, processing continues.

LNMEX24 Return one or more DocumentAuthors.

If this exec doesn't exist, the document author for the document will be set to the migrating user's userid. Otherwise, you must return data in the CMS stack.

Input Filename, type and mode of the document on disk, cron, filename and type of the document in the database and the default DocumentAuthor.

Output One or more DocumentAuthors must be in the CMS stack. These must be in ‘useridnodeid’ format, one per stacked line.

Return Codes <0 means the migration of this user will be terminated. Otherwise, processing continues.

LNMEX25 Return the name of the sender of the document.

If this exec doesn't exist, the name of the sender of the document will be extracted from the document. This will be from line 2 or, if the document is a meeting notice, line 5. Otherwise, you must return data in the CMS stack. In general, the name of the sender imbedded in the document comes from the author profile of the sender. If, when you migrate documents, the sender field in Notes is set to :q.?:eq. it means that the default processing couldn't find a sender name - you will have to use this exit to determine it yourself.

Input Filename, type and mode of the document on disk, cron, filename and type of the document in the database.

Output Name of the sender must be in the CMS stack.

Return Codes <0 means the migration of this user will be terminated. >0 means you wish for LNMOVDOC to extract it itself (may or may not be successful). Otherwise, the sender on the stack is used.

LNMEX26 Review and/or reformat document.

If this exec doesn't exist, the document will be migrated in the format that LNMOVDOC produces. Otherwise, you can review or re-format the document as you wish.

Input Filename, type and mode of the document on disk, cron, filename and type of the document in the database.

Output If the document is reformatted, it must have the same filename, type and mode as the input file.

Return Codes <0 means the migration of this user will be terminated. >0 means you wish the document **not** to be migrated. Otherwise, LNMOVDOC will migrate the document.

Implementation

This extension consists of a new add-on exec (LNMOVDOC), new options for LNMMIGR CONFIG and a change to the master controlling exec (LNMMUSER).

1. The file LNMOVDOC EXEC must be placed on the 200 code disk.
2. The new options must be added to LNMMIGR CONFIG and modified accordingly. A sample of these options is contained in a file LNMMIGR UPDATE1.
3. The following lines of code (found in file LNMMUSER DOCUPDAT) must be added to LNMMUSER EXEC before the statement
   'EXEC LNMLOG 900' routine fm "Trailer" (on or about line 166):

```
/* Start of document database support   */

   if translate(lnmcfg("MIGRATE_OVVMDOCS")) = "YES"
     then do
       'EXEC LNMLOG 900' routine fm "OV/VM Document"       'EXEC LNMOVDOC' fm
       If rc <> 0
         Then return -1
     end

/* End   of document database support   */
```

Advantages

1. Simplicity over other methods
2. Migrates only documents in the document index file

   Extraneous documents are not migrated

Disadvantages

1. Migrates only documents in the document index file
   Other documents which the user could access are not migrated

2. Duplicate copies of documents **will** be migrated
   There is no attempt to not migrate a document which has already been migrated.

Document Migration - Method 2

Problem Description

In the method described here, we wish to migrate documents which the user can access through a document log. We also do not wish to migrate a document more than once as well as exploit Notes' access control lists. The standard OfficeVision/VM Migration Tool only permits the migration of documents which are mailed to your in-basket.

As users are migrated, OFSINDEX OFSDATA or perhaps another document index file is referenced, documents are extracted from the OV/VM document database and a check is made to see if the document has already been migrated by another user. If it has, the document is not placed in the MDF. Otherwise, it is and the document is then migrated. The MDF indicates that the documents are destined for a server-based database and the ACL is set appropriately.

Extension Description

This method is essentially the same as method 1 except that an attempt is made to not migrate documents twice.

A ‘migrated document’ table is maintained by a service machine. If a document hasn't been migrated, then it will be migrated and with the appropriate ACL. DocumentAuthors and DocumentReaders will be set to the users who can access the OV/VM document (output from OFFICE CRON). However, if the document has been finalized, the DocumentAuthors list is empty and the document can not be edited on Notes.

If the document has been migrated, then the date of the document to be migrated is compared with the date of the document which was migrated. This will handle the case of a draft document which has been prior migrated and since been revised. If the dates are different then it will be migrated. If the dates are the same, then a check is made to make sure that the migrating user's userid is in the access list for the document. If it isn't, then the document will be migrated and it will be a duplicate. Otherwise, the document is skipped.

The code also implements an option in addition to those described for method 1:

MIGRATE_DOCSERVER
Set MIGRATE_DOCSERVER to the userid of the document server machine

Note: Do not have this option in LNMMIGR CONFIG if you don't want to use this document migration method.

Exits Supplied

Method 2 contains the same exit points as Method 1.

Implementation

This extension consists of a new add-on exec (LNMOVDOC), new options for LNMMIGR CONFIG, a change to the master controlling exec (LNMMUSER) and a document server userid.

1. The file LNMOVDOC EXEC must be placed on the 200 code disk.
2. The new options as described for method 1 must be added to LNMMIGR CONFIG and modified accordingly. The option MIGRATE_DOCSERVER must also be added. Asample of these options is contained in a file LNMMIGR UPDATE2.
3. The following lines of code (found in file LNMMUSER DOCUPDAT) must be added to LNMMUSER EXEC before the statement
   'EXEC LNMLOG 900' routine fm "Trailer" (on or about line 166):

```
/* Start of document database support   */

   if translate(lnmcfg("MIGRATE_OVVMDOCS")) = "YES"
     then do
       'EXEC LNMLOG 900' routine fm "OV/VM Document"
       'EXEC LNMOVDOC' fm
       If rc <> 0
         Then return -1
     end

/* End   of document database support   */
```

4. A document server userid must be created:

   1. Its A-disk should be 5 cylinders or large enough to contain the file CRON DATES (see below).
   2. Program CRONVM EXEC should be on the A-disk.
   3. The PROFILE EXEC must access the 200 code disk.
   4. Execute CRONVM and then type #CP DISC

5. The supplied exit LNMEX99 must be used to send the list of migrated documents to the document server userid. This should be placed on the 200 code disk.

Note: CRON DATES contains the document date by cron for each migrated document. It is updated at the end of each user migration. Its size in bytes will be 26 times the number of unique crons. CRON DATES is managed by the document server userid and it is created as user migrations occur. You do not create this file.

Advantages

1. Documents are migrated only once (in general)
2. Migrates only documents in the document index file
   Extraneous documents are not migrated
3. Finalized documents can not be changed in Notes
   (Exit LNMEX24 can override this)

Disadvantages

1. Complex implementation with another service machine and possibility of errors.
2. Migrates only documents in the document index file
   Other documents which the user could access are not migrated

3. Duplicate copies of documents may be migrated:
   If two users are migrated simultaneously with access to the same document, there will likely be duplicate documents migrated.
4. If a draft document is updated, different versions will be in Notes.
5. If a document is mailed to a user but the in-basket is never opened, the document will not be accessible to that user.

Note: The OfficeVision/VM Migration Tool may open the mail though, if in-basket mail is requested to be migrated. In this case, the document index log *will* contain the document cron and the document will be migrated if it hasn't already been migrated.

Document Migration - Method 3

Problem Description

There are different possible reasons that might require a migration to be performed from a central Id.
Some of which are:

A large number of documents is to be migrated. If this is done during the normal migration of the users it could result in very large migration data files and also could slow down the user migration.
The OV/VM Document Database is used as a bulletin board where documents are not sent to the users but are made available through an interface that only provides the references (cron) for access. In this case no information is found in the user's index files.

Extension Description

This sample is assumed to be used to migrate all documents in one go, from a central Id. It does not take care of modifications and new documents as they would appear in a coexistence environment. If it is required to use this solution in a coexistence environment the code will need to be extended to keep track of the document/user migration status and thus be able to handle a live document database. The thing to still keep in mind is that this solution only can provide a migration that feeds documents into Notes databases. Nothing is available yet to get changes or additions back to OV/VM.

This solution uses a VM Id to collect the information to be processed from the OV/VM database and the user's index files. If all documents are to be migrated, data from the user's index files need not be accessed. Because the user's index file, OFSINDEX OFSDATA is used to verify a user's access to a document, only those documents are migrated that the user is aware of. If a document was sent to a user but the user never opened the mail, the information is not stored in the user's index file and therefore the user can not be authorized for the document on Notes.

During the actual extract of the documents, a migration data file (MDF) will be created and transferred to a Notes server that runs a Domino Migration Engine. The MDF file will indicate that the documents are to be stored in a server based database. Each user will be added to the access control list (ACL) of the target database to permit general access it. The access rights will be refined for each document as all authorized users will be added to the DocumentReader filed. Given this, only users that had referenced a document on OV/VM will see it on Notes.

The handling of the documents in this solution will be similar to what is described for Method 1 and Method 2. Basically RFT and DCF documents will be formatted after extract and be provided as text documents in the Notes database. Files/PC-Files in the OV/VM database will be stored as attachments to documents.

The exits as they are supplied with the previous methods are not implemented in this solution. Although adding them to LNMDOCSV would not be too difficult as it has the same structure as LNMOVDOC that is used in the previous methods.

Migrating documents can result in moving large amounts of data. To keep the MDF files manageable, a routine has been included that checks the size of the file and interrupts the extract of documents when a limit that is set in LNMMIGR CONFIG is reached. The MDF file will then be completed with a trailer record and transferred. Restart information is saved to disk and the administrator will be notified. Once the MDF file is accepted by the Domino Migration Engine the process can be restarted. It will pick up where it was interrupted and create the next MDF file.

LNMMIGR CONFIG also contains options to select documents based on a start and end day range. The date in the document's cron is used for comparison purposes.

Prerequisites:

The VM Id will require read access to all the disks of the OV/VM Database machine.

The VM Id will also require access to all user 191 disks if the entire OV/VM document database won't be migrated.

It is recommended that the OV/VM Database be reorganized before document migration is started. During the migration, the xxxDBM should be shut down to prevent from having data and disks updated while they are accessed by extract or migration.The VM Id is defined as a user in the migration administration database with a specific Source Location as shown in the following sample:.

Source Location		
Userid:	DOCADMIN	Batch ID:
Source Location:	DOCUMENT	Migration ID: DOCADMIN@DOCUMENT
Migration Status:	Migration request sent	Source ID:
Notes		Mail
First name:		Domain: SDFPSL01
Middle initial:		Mail server: SDFMIG2/SDFPSL01
Last name: DOCADMIN		Mail file: Mail\DOCADMIN.NSF
User name: DOCADMIN/SDFPSL01		
		Forwarding address: DOCADMINDOCUMENT @ GATEWAY2
Short name: docadmin		Registration Profile: BASIC

The Source Location requires the bypass standard communications option. See the next figure for a sample of the source location document. This means there are no key files generated. After registration, a user from this source location will be automatically put into a status of Migration Request Sent.

Basics	
--	
Source Location	**DOCUMENT**
Description	Special Location for Documents
Processing Options	
--	
Incoming Work Directory Specify value	c:\migra\incoming\document
Outgoing Work Directory Specify value	c:\migra\outgoing\document
Server for server-based databases	Local
Character Set	IBMCP850
Max Migration Errors Use default	10
Name Resolution	Enabled
Directory Migration Pre-processing	Enabled
Name Collision Match	on User name
Name Collision Options - for Registered Notes Users	Use existing Person document
- for other Person documents	Update if duplicate
Post-Registration Communication Option	**Bypass communications**

The VM Id will perform the following tasks:

Extract Document information from the OV/VM Document Database
Extract User/OFSINDEX information from the user’s disks if the entire document database is not to be migrated
Compile a list of documents to be migrated with general document data, the physical location of the document and a list of authorized users.
Extract the documents that have been identified during the previous processes and store the information in a Migration Data File.

The extract of user information needs a list of users to work on if all documents won't be migrated. A file is required that contains UserId and A-Disk address, usually 191. This list could be extracted from the OFSUAD, the OV/VM User Registration file or from any other source. The sample also could be changed to use OFSUAD directly. The problem with this approach could be that it also would include resources and other ids that do not have their own disks.

During the migration, the list of documents will be processed and the documents are extracted directly from the database storage.

Implementation

VM Rexx Procedures provided for the preparation of a list documents to be migrated:

LNMDBEXT - Extract document information from the OV/VM Database
LNMUSEXT - Extract document information from a user's index file
LNMUSCMP - Compress document information to one record per document
LNMDOLST - Create a document/location/users list for the migration

VM Rexx Procedures provided for the migration of the documents in a predefined list:

LNMDRUN - Procedure to kick off the migration
LNMSTDOC - Set up the environment, call migration, transfer MDF file
LNMMDOC - Create MDF file, header, trailer and calls LNMDOCSV
LNMDOCSV - Performs the document extract and adds to the MDF file
LNMLOG - Modification for LNMLOG to make it easier for the admin
LNMHEADR - Modification for LNMHEADR to set special Migration Id

LNMMIGR UPDATE3 - Additions to the configuration file LNMMIGR CONFIG as shown in the following sample.

```
#---------------------------------------------------------------------
# The following configuration options can be set to control the migration
# of user's documents contained in the OV/VM document database and
# referenced by a document index file on the user's disk.

# Set MIGRATE_OVVMDOC to YES if documents in an index file are to be migrated
MIGRATE_OVVMDOCS = YES

# Set MIGRATE_DOCNAME to PC filename of server based document NSF.
MIGRATE_DOCNAME  = DOCS

# Set MIGRATE_DOCTITLE to database title as end user sees it.
MIGRATE_DOCTITLE = OV/VM Documents

# Set MIGRATE_DOCFILE to filename and type of user's document index file.
MIGRATE_DOCFILE  = DOCFILE  FILE

# Set MIGRATE_DOCNTF to document database template name.
MIGRATE_DOCNTF   = DOCLIB4.NTF

# Set MIGRATE_DOCCATS to YES if documents should be categorized.
# Categories are DCF, RFT and File.
MIGRATE_DOCCATS  = YES

# Set MIGRATE_DOCSTATUS to YES if document status should be set.  Status can be
# Ready for Review, Review Complete and Other. #
MIGRATE_DOCSTATUS= YES

# Set DOCUMENT_CENTRAL to YES if this configuration file is used
# by a document migration machine.
DOCUMENT_CENTRAL = YES

# Set DOCUMENT_USER to the userid that is set up in the Domino Migration
# Engine for central document migration
DOCUMENT_USER = DOCADMIN

# Set DOCUMENT_NODE to the source location that is set up in the Domino
# Migration Engine for central document migration
DOCUMENT_NODE = DOCUMENT

# Set DOCUMENT_MAX_MB to the maximum number of MBs to be contained in
# a single MDF file during central document migration
DOCUMENT_MAX_MB = 1

# Set DOCUMENT_STARTDAY to the first day that is accepted for a
# document to be migrated. The format is YYDDD and is compared
# against the cron.
DOCUMENT_STARTDAY = 88001

# Set DOCUMENT_ENDDAY to the last day that is accepted for a document
# to be migrated. The format is YYDDD and is compared against the cron.
DOCUMENT_ENDDAY = 99366


# Set  Set DOCUMENT_SUBDBS to one or more subdatabase numbers whose documents you
want to migrate.  If not defined, or the value is ALL or *, all documents will be selected
DOCUMENT_SUBDBS = ALL
```

Configuring the Right Environment

Before you execute any of the VM REXX procedures, you need to have the right environment established:

1. Your VM Id which will perform the document migrations must be set up:
   a. To have read access to all user's 191 disks. (Not necessary if you are migrating all documents from the database.)
   b. To have read access to all of the disks owned by the OV/VM database manager (191, 161, 5FF, 5FE, 5xx, etc)
   c. To have read access to the 200 code disk
   d. To have access to the OV/VM 399 disk
   e. To have write access to a VM Migration Server work disk (for example, the 201 disk)
   f. To be able to run a file transfer
      Loader tables may have to be set (SET LDRTBLS 5)
      Access to any disks containing file transfer software (just like the VM Migration Server)
   g. To be defined as a user in the migration administration database. The userid (in the database) should be DOCADMIN. (The VM Id may be called anything.) The source location should be DOCUMENT.
   h. It should have at least 32M defined as memory.
   i. It should have a work disk (may be the A-disk) defined to be quite large. The migration data file for all documents will be created on this disk.
   j. The list of users whose documents are to be migrated should be created and be accessible. (Not necessary if you are migrating all documents from the database.) This will be file LISTA USERS with the format:
      Userid columns 1-8
      Address columns 10-12 (blank implies 191)
      Password columns 14-21 (blank implies RACF in use)
2. The OV/VM database should be re-organized. The database manager should be shut down and logged off.
3. A source location document called DOCUMENT in the migration database should be created. It must have :q.Bypass Communications:eq. as an option.
4. The following document migration VM Rexx procedures must be on the 200 code disk:
   LNMDBEXT
   LNMUSEXT
   LNMUSCMP
   LNMDOLST
   LNMDRUN
   LNMSTDOC
   LNMMDOC
   LNMDOCSV
5. The PUFF MODULE supplied must also be on the 200 code disk. Instructions are included in file PUFF INSTR in case the supplied one doesn’t work on your system and you need to generate one based on your current system.
6. Modification to LNMLOG EXEC on the 200 code disk must be done. File LNMLOG UPDATE contains code which must be added to the file LNMLOG EXEC. It should go in close to the end of the file, right before the line that contains 'CP LOGOFF'.

7. Modification to LNMHEADR EXEC on the 200 code disk must be done. File LNMHEADR UPDATE contains code which must be added to the file LNMHEADR EXEC. It needs to go in after line 29. This is the line containing "ovnode = strip(ovnode)".

8. The configuration file (LNMMIGR CONFIG) needs to be updated to include added options.

   MIGRATE_OVVMDOCS must be set to YES.

   MIGRATE_DOCFILE must be set to DOCFILE FILE. (Document migration methods 1 and 2 have this set to OFSINDEX OFSDATA.)

   DOCUMENT_CENTRAL must be set to YES for Method 3.

   DOCUMENT_USER must be set to DOCADMIN. This is the userid established for document migration.

   DOCUMENT_NODE must be set to DOCUMENT. This is the node referenced in the migration administration database

   Review the other new options and make changes as appropriate.

Preparing for Document Migration

Document Migration is a five step process. The first four are in preparation for moving the documents to the Domino Migration Engine.

If you are migrating the entire document database and don't have access to all user's 191 disks, you will only perform steps 1, 4 and 5.

1. **LNMDBEXT EXEC**

   Extracts document information from the DBMs index file and checks if documents are available on the DBMs disks. Two settings in LNMMIGR CONFIG define start and end dates for the selection. The setting DOCUMENT_SUBDBS can also be used to select only documents from one or more sub databases. Note that the sub database number is stored in columns 259-261 into DOC FILE. This is read by routine LNMDOCSV but is currently not used. You could tailor LNMDOCSV to use this in some manner if you wanted.

   DOC IND is created on the work disk. This file contains the information extracted from the DBMs index file. This includes file name and location of the document on the DBMs data disk and the list of users that the document had been mailed to.

   As the information might be extracted from multiple index files, it needs to be sorted by cron for future processing. **This is not yet automated.**

   Prerequisites

   Read access to all OV/VM DBM disks, including 191, 161 and the different document storage disks is required.

   Input

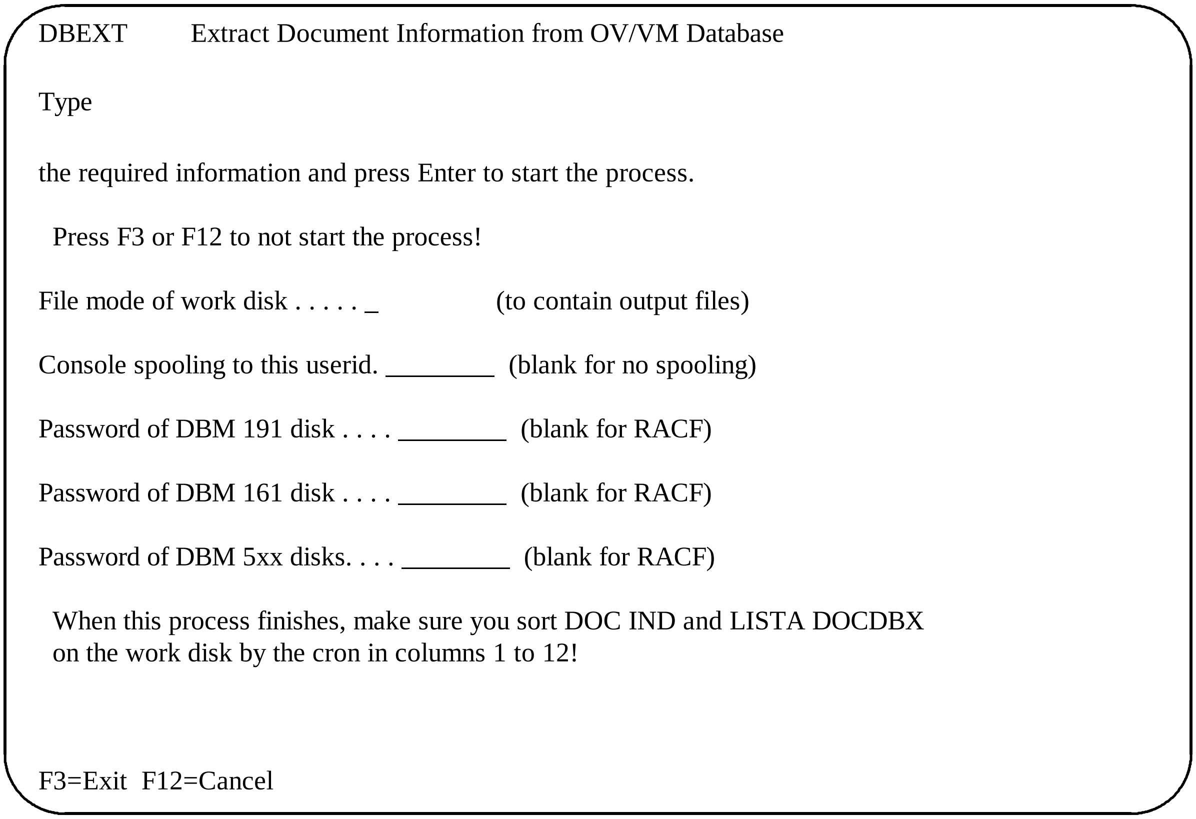
   OFSINSTL FILE - found on OV/VM Admin 399. Provides the userid of the OV/VM Database Machine and the local nodeid.
   OFSLPASS FILE - on xxxDBM's 191 disk. Contains the list of all the DBM's database disks. Usually 5FF, 5FE, 5FD, and so on. This information is required to access the disks and check their disk labels so that they can be used later on to extract the documents.
   OFSUAD FILE - the OV/VM user registration file, located on DBM's 191 disk. Besides other information, this file contains userids and the Document Access Numbers (DAN) which are stored with the documents to identify the users.
   INDEX DATAxxx and UPDATES DATAxxx - Database index files located on DBMs 161 disk.

   Output

   DOC ERROR - contains documents that were marked in error.
   LISTA DOCDBX - there will be one record per document, containing cron and the physical location of the actual version and any previous versions of the document.

DOC IND - this is the file that contains all the information about all the valid documents in the database, based on start and end dates as they were provided to the procedure.

Panel Input



```
DBEXT        Extract Document Information from OV/VM Database

Type

the required information and press Enter to start the process.

  Press F3 or F12 to not start the process!

File mode of work disk . . . . . _               (to contain output files)

Console spooling to this userid. ________  (blank for no spooling)

Password of DBM 191 disk . . . . ________  (blank for RACF)

Password of DBM 161 disk . . . . ________  (blank for RACF)

Password of DBM 5xx disks. . . . ________  (blank for RACF)

  When this process finishes, make sure you sort DOC IND and LISTA DOCDBX
  on the work disk by the cron in columns 1 to 12!



F3=Exit  F12=Cancel
```

2. **LNMUSEXT EXEC**

This procedure works on a liest of users that is stored in a file called LISTA USERS.

The 191 disks of the users are linked and accessed and the information from OFSINDEX OFSDATA is extracted.

The resulting file - DOCLOG DOC - contains one record per document per user. If a document is referenced by multiple users there will be multiple lines for the document.

The information in this file will be in the order of userids and needs to be sorted by cron.
This is not yet automated.

Prerequisites

Read access to all the user’s 191 disks is required.

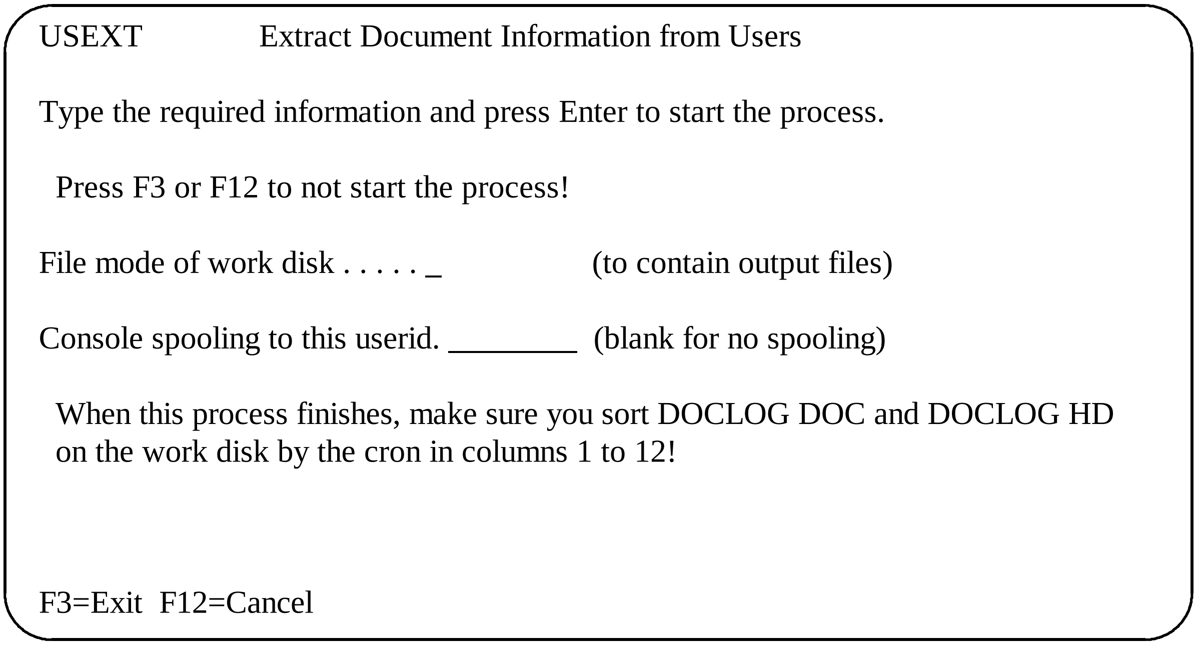
Input

LISTA DOCDBX - there will be one record per document, containing cron and the physical location of the actual version and any previous versions of the document.
LISTA USERS - this file needs to be provided by the administrator. It contains userid in 1 to 8 and A-disk address, usually 191 in columns 10 to 12 for all users to be handled. If a read password is required for linking it may be supplied in columns 14 to 21. Leave blank if RACF is in use.
OFSINDEX OFSDATA - a users document index.
* OFS$* - additional index files

Output

DOCLOG ERROR - errors encountered during the process will be logged in here.
DOCLOG HDC - information about hardcopy/paper documents are stored in this file. These documents are identified by a 3-charId of 'HDC'.
DOCLOG NO - list documents that were in a user's index file but were not found in LISTA DOCDBX, the list of documents found in the database.
DOCLOG EMPTY - lists the users with no documents.
DOCLOG DOC - this is the main output file. It contains one record per user/document. So can contain the same document multiple times for different users. Must be sorted by cron.

Panel Input



```
USEXT          Extract Document Information from Users

Type the required information and press Enter to start the process.

  Press F3 or F12 to not start the process!

File mode of work disk . . . . . _              (to contain output files)

Console spooling to this userid. ________  (blank for no spooling)

  When this process finishes, make sure you sort DOCLOG DOC and DOCLOG HD
  on the work disk by the cron in columns 1 to 12!



F3=Exit  F12=Cancel
```

3. **LNMUSCMP EXEC**

LNMUSCMP reads the file - DOCLOG DOC - containing multiple lines per document and creates the file - DOCLIST FILE - hich will contain single lines per document with a liest of users at the end of each line.

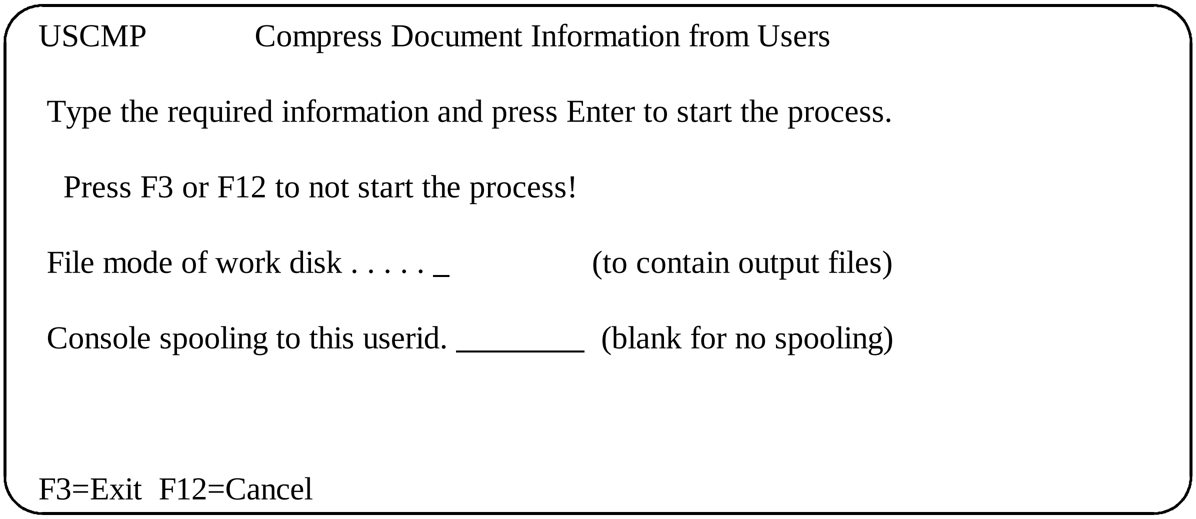
Input

DOCLOG DOC - contains one record per user/document. So can contain the same document multiple times for different users. The file was sorted by document cron after LNMUSEXT was run.

Output

DOCLIST FILE - this file contains one record per document with a list connected that contains all users who reference the document.

Panel Input



```
USCMP            Compress Document Information from Users

 Type the required information and press Enter to start the process.

   Press F3 or F12 to not start the process!

 File mode of work disk . . . . . _               (to contain output files)

 Console spooling to this userid. ________  (blank for no spooling)



F3=Exit  F12=Cancel
```

LNMDOLST EXEC

This will combine the information retrieved from the OV/VM Database with the data coming from the different users.

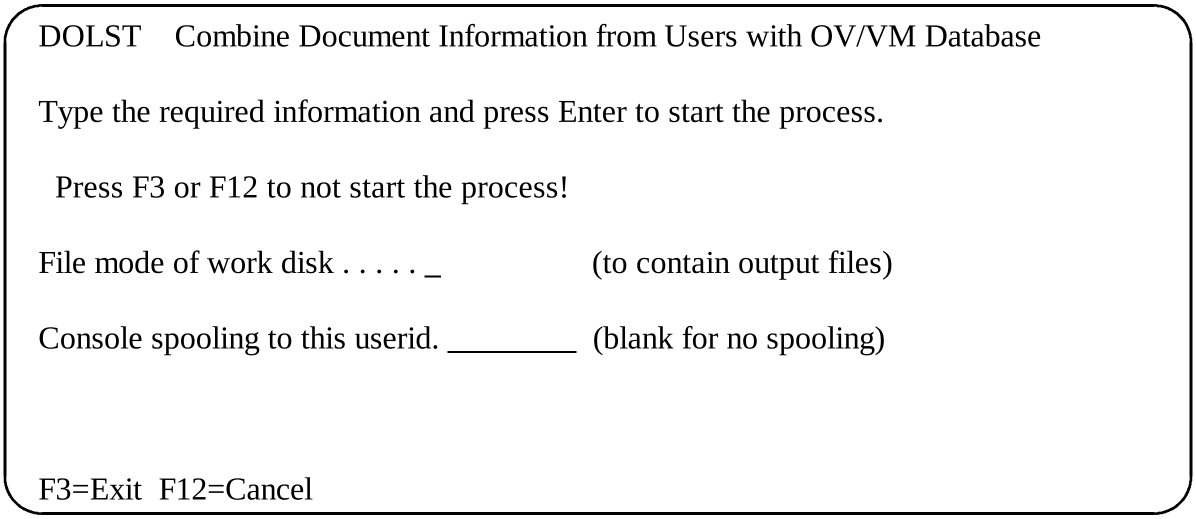
Input

DOC IND - contains all the information about a document as it was retrieved from the OV/VM database, including the data required to access the document directly from the database. The file has been sorted by document cron in an intermediate step.
DOCLIST FILE - this file will has one record per document with a list of all users who reference the document.

Output

DOCFILE FILE - combined file with all the document information and a list of users who reference the document.

Panel Input



```
DOLST    Combine Document Information from Users with OV/VM Database

Type the required information and press Enter to start the process.

  Press F3 or F12 to not start the process!

File mode of work disk . . . . . _                (to contain output files)

Console spooling to this userid. ________  (blank for no spooling)



F3=Exit  F12=Cancel
```

Performing the Document Migration

The fifth step is to perform the migration by invoking LNMDRUN:

LNMDRUN EXEC

This procedure will read DOCFILE FILE (a document index file), extract the documents from the document database, assign ACLs based on the document distribution list, append the document to a migration data file and finally transfer the MDF to Notes.

Make sure you verify that KEYPATH and MIGRPATH are set correctly in LNMMIGR CONFIGSV so that file transfer will send the MDF to the directories specified in the source location document for DOCUMENT.

You will also have to make an appropriate adjustment to the MAXDATA size for MDFs in the Migration Default document. This normally controls the size of user's mail files but you will need to really increase this to take into account the size of the MDF containing your documents.

LNMDOCSV is called by LNMDRUN and it works on the list of documents to be migrated, reads them from the OV/VM Document Database and stores them into a Migration Data File.

Based on the DOCUMENT_MAX_MB setting the procedure will stop the migrations process when the size of the MDFile exceeds the given size. Restart information is saved to disk and the administration will receive a message explaining the situation. When migration is restarted the restart information is read from disk and the procedure will continue to move document information into a new MDFile.

Prerequisites

Access required to all DBM disks.

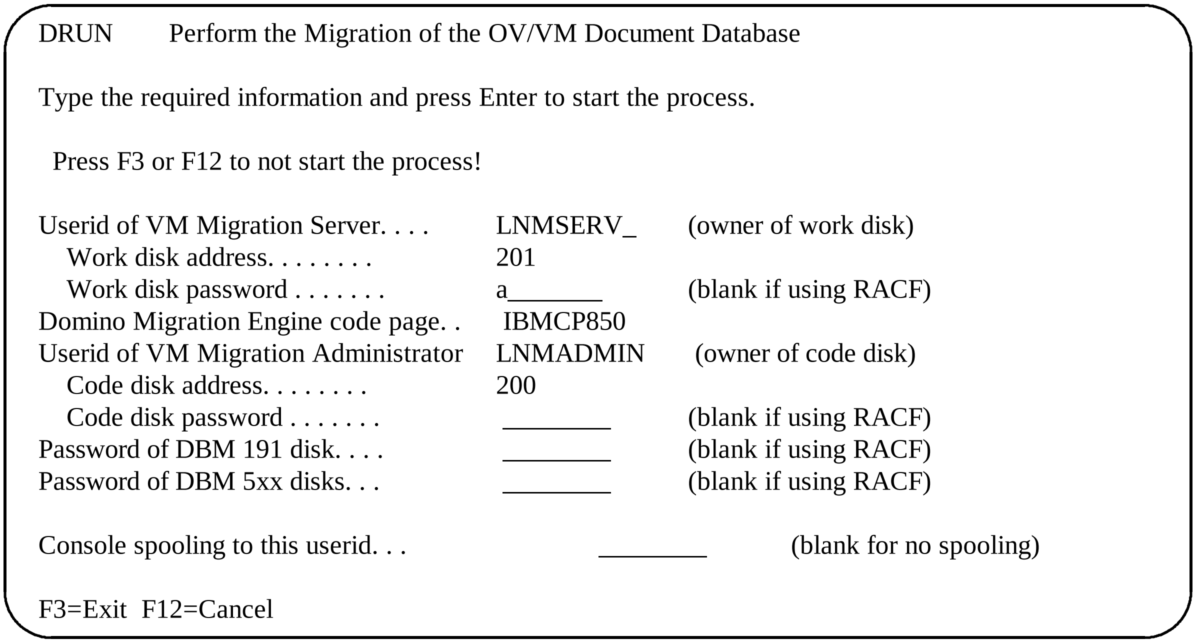
Input

DOCFILE FILE - combined file with all the document information and a list of users who reference the document. The list of users who can access the document starts in column 259 of each record.

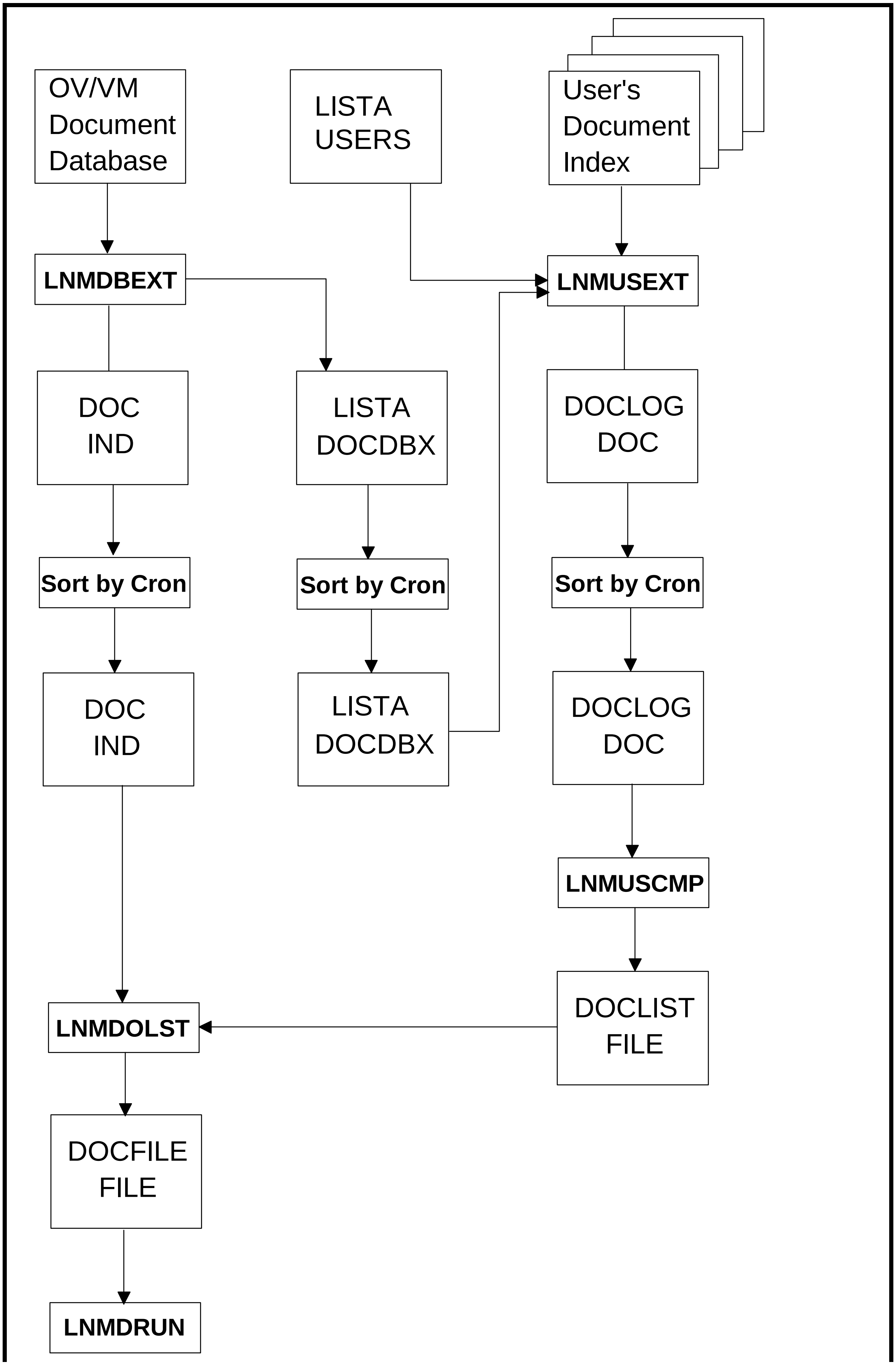
Output

DOCFILE ERRORLOG -Summary of errors detected during document processing
LNMMIGR RESTART - Restart file contains last cron processed when the MDF file size exceeded the option DOCUMENT_MAX_MB in LNMMIGR CONFIG.

Panel Input



```
DRUN      Perform the Migration of the OV/VM Document Database

Type the required information and press Enter to start the process.

  Press F3 or F12 to not start the process!

Userid of VM Migration Server. . . .      LNMSERV_      (owner of work disk)
    Work disk address. . . . . . . .      201
    Work disk password . . . . . . .      a_______      (blank if using RACF)
Domino Migration Engine code page. .      IBMCP850
Userid of VM Migration Administrator      LNMADMIN       (owner of code disk)
    Code disk address. . . . . . . .      200
    Code disk password . . . . . . .      ________      (blank if using RACF)
Password of DBM 191 disk. . . .           ________      (blank if using RACF)
Password of DBM 5xx disks. . .            ________      (blank if using RACF)

Console spooling to this userid. . .              ________        (blank for no spooling)

F3=Exit  F12=Cancel
```


OV/VM Document Database
LISTA USERS
User's Document Index
LNMDBEXT
LNMUSEXT
DOC IND
LISTA DOCDBX
DOCLOG DOC
Sort by Cron
Sort by Cron
Sort by Cron
DOC IND
LISTA DOCDBX
DOCLOG DOC
LNMUSCMP
LNMDOLST
DOCLIST FILE
DOCFILE FILE
LNMDRUN

Advantages

1. Documents are only migrated once, no duplicates.
2. Document migration does not influence user migration.
3. Can be modified to migrate centrally managed documents.

Disadvantages

1. Best to be used for migrating all documents at the end of a user migration process. Not really for ongoing migration in a coexistence situation.
2. If documents are mailed to a user but this user's in-basket got never opened, the user will not have access to the migrated document. True only if user's 191 disk needs to be accessed. No problem if entire document database is being migrated.

Customization of the Migration for each User

Current Situation

Customization of the migration is controlled by the file LNMMIGR CONFIG. This file is updated by the VM migration administrator and its parameters govern all users targeted for the current migration. For different migrations the administrator can update the file. However, there is no built in method to customize on a user by user basis.

Discussion

The migration programs read the file LNMMIGR CONFIG from the first accessed disk. Normally, this is the code disk belonging to the VM migration administrator. If you knew what migration parameters you wanted to change based on the userid, you could use exit LNMEX00 to create a custom LNMMIGR CONFIG on the user's A-disk. Exit LNMEX99 can be used to erase it at the end of the migration.

Possible Solution 1

The VM administrator could manage a file containing separate parameters for each user. You could envision one record per user. Each record would contain values for the parameters of LNMMIGR CONFIG. Exit LNMEX00 could read this file and locate the correct record and then create a local copy of LNMMIGR CONFIG on the user's A-disk. A variation which would reduce the number of records is to have a common record for groups of users, say, based on their userid. Maybe your userids are assigned based on departments and you can use some key in the userid to access a record in your "parameter" file.

Possible Solution 2

Sample program LNMUOPTS EXEC has been written to display options which the user can change. Tell them to execute this program before you migrate them.

LNMUOPTS will read LNMMIGR CONFIG. An options table at the end of the exec shows which options are to be displayed. Displayed options which have a NO value set by the administrator in LNMMIGR CONFIG will not be changeable by the user. Only LNMMIGR CONFIG options with YES or NO values are supported by LNMUOPTS.

Note the option table near the end of the exec. Each table entry consists of 3 values:

1. Y or N - Y means option is displayed
2. option name - must be the same as in LNMMIGR CONFIG
3. option description - this is displayed on the panel

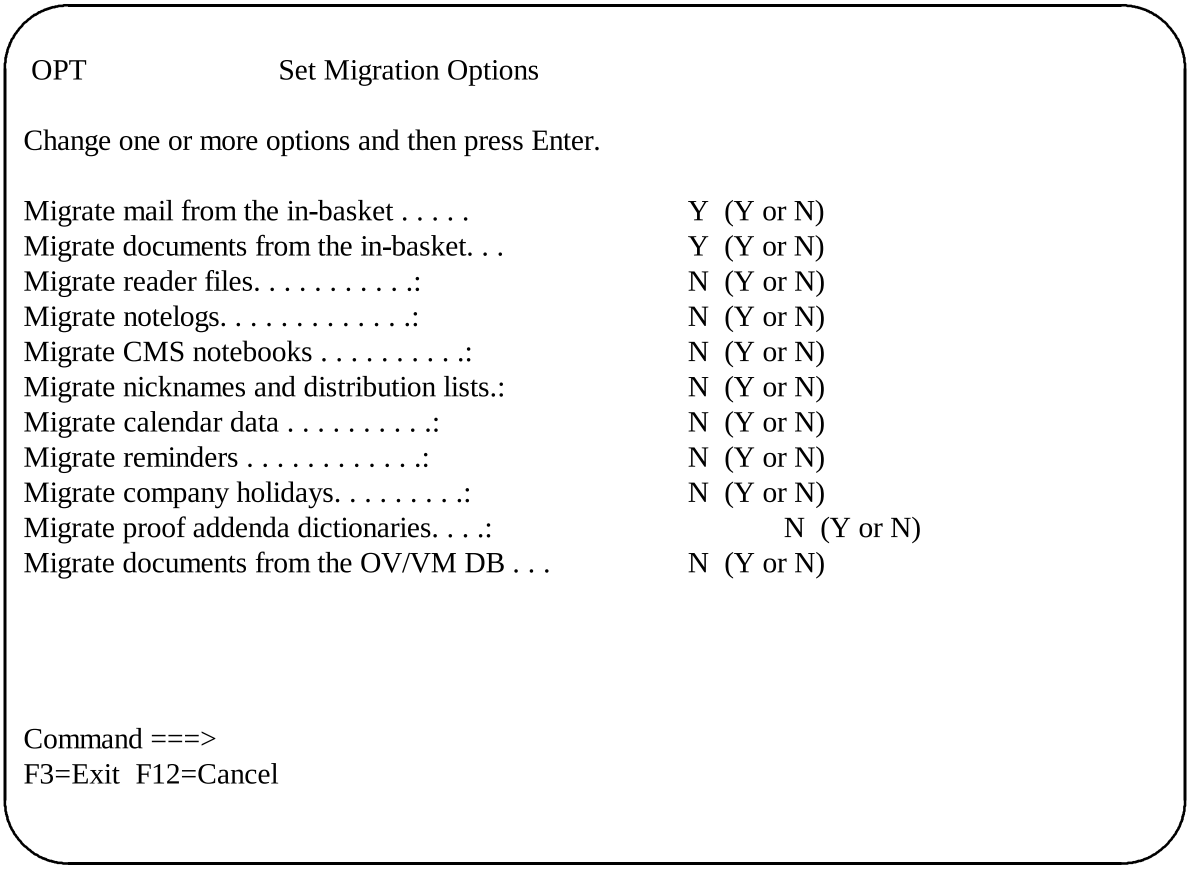
As the exec is written, options which have a NO value in LNMMIGR CONFIG can not be changed to YES. You can change this just by deleting lines 35 to 37:

```
If sysopts.optname='NO' Then
  fschar=':{'
Else
```

Note: The output of LNMUOPTS EXEC is an updated copy of LNMMIGR CONFIG on the user's A-disk. This file could be changed manually by the user!

The exec needs to be customized with the userid of the owner of the 200 code disk. Update line 3 accordingly.

Sample panel:



```
 OPT                    Set Migration Options

Change one or more options and then press Enter.

Migrate mail from the in-basket . . . . .          Y  (Y or N)
Migrate documents from the in-basket. . .          Y  (Y or N)
Migrate reader files. . . . . . . . . . .:         N  (Y or N)
Migrate notelogs. . . . . . . . . . . . .:         N  (Y or N)
Migrate CMS notebooks . . . . . . . . . .:         N  (Y or N)
Migrate nicknames and distribution lists.:         N  (Y or N)
Migrate calendar data . . . . . . . . . .:         N  (Y or N)
Migrate reminders . . . . . . . . . . . .:         N  (Y or N)
Migrate company holidays. . . . . . . . .:         N  (Y or N)
Migrate proof addenda dictionaries. . . .:                N  (Y or N)
Migrate documents from the OV/VM DB . . .          N  (Y or N)




Command ===>
F3=Exit  F12=Cancel
```

Hardcopy Document Support

Problem Description

The OfficeVision/VM Migration Tool will not migrate hardcopy documents since the data is not in the document database. Hardcopy data is kept in the document index log.

Extension Description

Code is available to migrate hardcopy document information as an attachment to a memo. In other words, they are treated and migrated the same as other documents.

The attachment will have a PC filename of the format Dydddnnnn just like other migrated documents. The PC filetype will be HDC.

Implementation

1. The file LNMHDC EXEC must be placed on the 200 code disk.
2. The following lines of code (found in file LNMPDOC UPDATE) must be added to LNMPDOC EXEC after the „end” statement (on or about line 46) for the first „when” clause which checks for „code”:

```
/* Hardcopy support start */
  When code = 67 & Substr(cron,6,3)='HDC'
    Then Do
      'EXEC LNMHDC' cron '$$LNM$$ OFSDATA A' configfm
      If rc<0 Then
        Exit -1
      docname='D'Substr(cron,2,4)||Substr(cron,9,4)||Substr(cron,6,3)
    End
/* Hardcopy support end   */
```

3. The file LNMMIGR FT2EXT must be updated to have a table entry for files with a filetype of HDC:

   ```
   HDC HDC T
   ```

   Note: A file transfer mode of T must be specified.

Migrating Document Meeting Notices

Problem Description

Meeting notices in the in-basket can be created as notes or documents. The calendar interface is used to generate meeting notices as notes. The document processing interface will create a meeting notice and mail the document to the in-basket. The OfficeVision/VM Migration Tool just views this as a standard document and attaches it to a memo destined to the Notes inbox. You can't add it to the calendar using the "Add to Calendar" button.

Extension Description

There is an exit called LNMEX02 which gets control when a document in the in-basket is processed. Sample code has been created which performs correct processing of the meeting notice. It relies on the fact that the installation has not changed the meeting notice form (file MEETING @DOCFMAT). It currently does not handle recurring meeting notices.

Implementation

The sample LNMEX02 must be placed on the 200 code disk.

Migrating CMS Notebooks as Notelogs

Problem Description

The base code will migrate CMS notebooks as attachments to a memo. The attachment is just an unstructured file. It would not be possible to reply to an individual note within the attachment.

Extension Description

This problem could be overcome if the notebooks were converted to an OV/VM notelog and then have the migration tool migrate them into a folder. Each individual note could then
be addressed and replied to. Exit LNMEX01 is provided to extend notelog processing. The exit gets control after all OV/VM notelogs are processed. A sample LNMEX01 has been coded to look for CMS notebooks and it converts them to OV/VM notelogs. The exit then just passes back the filenames of the newly created notelogs to the migration tool.

Implementation

The sample LNMEX01 must be placed on the 200 code disk.

Note: Since CMS notebooks will be migrated as notelogs by this exit, make sure that you have **MIGRATE_NOTELOGS** set to **YES** in LNMMIGR CONFIG!

REMINDER File Support

Problem Description

The file OFS $remind contains reminders that OV/VM users have created. The OfficeVision/VM Migration Tool does not read this file and therefore they are not placed on the Notes calendar. Note that OV/VM does not place reminders on the calendar system.

Extension Description

There is no exit available to process the file OFS $remind. However, changes are available to the base code which process this file.

The file OFS $remind is not migrated to Notes. The method behind the code is simply to add the reminder information to the migration data file and let it be imported into the Notes calendar. The calendar entry type will be set to "reminder" as you would expect.

Not all of the reminders in OFS $remind will be added. The only ones accepted are those after the current migrate date and time and before the "calendar_end" option in LNMMIGR CONFIG.

The code also implements a new option called MIGRATE_REMINDERS.

While the reminder file supports multiple reminders, each separated by a time interval, the processing code ignores these fields. This is because Notes places reminders in the calendar database and multiple reminders would show up as multiple entries during the day.

One further note needs to be made concerning reminders on the Notes calendar. In order for a reminder's alarm notification to be displayed, Notes requires some setup or action to be performed first. Failure to do so will mean that reminder alarm notifications **won't** be shown. One of two things must be performed to get the alarm notifications:

1. Before Notes is started, the NOTES.INI file must contain $EnableAlarms=1.
2. When Notes is started, open up the mail file, select Actions, Calendar Tools, then Calendar Profile. In the Scheduling Options section, select the "Enable alarm notifications" checkbox. Then select to have reminders alarmed. Click on the OK action box. Exit Notes and restart it. This last step **must** be done.

Implementation

1. The files LNMRMNDR EXEC and LNMRMNDR TABLE must be placed on the 200 code disk.
2. The new option MIGRATE_REMINDERS must be added to LNMMIGR CONFIG:

```
# The following option determines whether OFS $remind will
# be migrated.  If YES is specified, OFS $remind will be
# migrated.

MIGRATE_REMINDERS = YES
```

3. The following line of code must be added to LNMMUSER EXEC after the „if”statement on or about line 93:

```
   translate(lnmcfg("MIGRATE_REMINDERS")) = "YES" | , /* reminders */
```

4. The following lines of code (found in file LNMMUSER REMINDER) must be added to LNMMUSER EXEC before the statement

   „record = "..DE"” (on or about line 160)::

```
/* start of reminder support */
    if translate(lnmcfg("MIGRATE_REMINDERS")) = "YES"
      then do
        'EXEC LNMLOG 900' routine fm "Reminder"
        'EXEC LNMRMNDR' userid fm
        If rc <> 0
          Then return -1
      end
/* end   of reminder support */
```

HOLIDAY EPSFILE Support

Problem Description

The HOLIDAY EPSFILE lists the company holidays which users can add to their own personal calendars. The OfficeVision/VM Migration Tool does not read this file and, if these holidays are not already on your calendar, they will not be posted to your Notes calendar.

Extension Description

There is no exit available to process the HOLIDAY EPSFILE. However, changes are available to the base code which process this file.

The HOLIDAY EPSFILE itself is not migrated to Notes since there is no object or structure to accept it. The method behind the code is simply to add the holiday information to the migration data file and let it be imported into the Notes calendar, already posted. It will be posted as an "all-day" event. If the holiday is already on the OV/VM calendar, then the holiday information is ignored. So, in essence, the code adds the company holidays to the calendar.

Not all of the holidays in the HOLIDAY EPSFILE will be added. The only ones accepted are those within the "calendar_start" and "calendar_end" options in LNMMIGR CONFIG.

The code also implements a new option called MIGRATE_HOLIDAYS.

While the HOLIDAY EPSFILE supports a myriad of date formats, the code will only understand dates in US, European or International formats.

Exit Supplied

LNMEX19 Return one or more holiday file entries

If this exec doesn't exist, HOLIDAY EPSFILE is read. Otherwise, you must return data in the CMS stack in the same format as is contained in HOLIDAY EPSFILE.

Input Calendar start and end dates in the format yyyy/mm/dd. LNMHOLID will not process holidays outside of this date range.

Output Holiday dates must be in the CMS stack. Columns 1-8 must contain the date in the system date format. Column 20 starts the description of the holiday.

Return Codes <0 means the migration of this user will be terminated. Otherwise, processing continues.

Implementation

1. The file LNMHOLID EXEC must be placed on the 200 code disk.
2. The new option MIGRATE_HOLIDAYS must be added to LNMMIGR CONFIG:

```
# The following option determines whether HOLIDAY EPSFILE will
# be migrated.  If YES is specified, the holidays will be added
# into the Notes calendar.

MIGRATE_HOLIDAYS = YES
```

3. The following line of code must be added to LNMMUSER EXEC after the „if” statement on or about line 93:

```
translate(lnmcfg("MIGRATE_HOLIDAYS")) = "YES" | , /* holiday */
```

4. The following lines of code (found in file LNMMUSER HOLIDAY) must be added to LNMMUSER EXEC before the statement

   „record = "..DE"” (on or about line 160):

```
/* start of holiday support */
    if translate(lnmcfg("MIGRATE_HOLIDAYS")) = "YES"
      then do
        'EXEC LNMLOG 900' routine fm "Holiday"
        'EXEC LNMHOLID' userid fm
        If rc <> 0
          Then return -1
      end
/* end   of holiday  support */
```

Time Zone Support

Problem Description

Release 1 of the OfficeVision/VM Migration Tool has no support for time zones. This becomes important for meeting notices in the in-basket and notelogs as well as calendar entries. If the mail server or client are not in the same time zone as the VM host, the times are incorrect.

Extension Description

A time zone offset is applied to calendar entries as well as start and end times in meeting notices. These meeting notices may be in the in-basket or in notelogs. If support for reminders or document meeting notices is included, time zone offsets will also be applied.

A new option called TIME_ZONE_OFFSET is added to file LNMMIGR CONFIG which specifies the time zone offset between the Notes server running the DME and the Notes client.

Exit Supplied

LNMEX10 Return time zone offset in seconds

If this exec doesn't exist, the time zone offset is taken from LNMMIGR CONFIG. Otherwise, you must return the time zone offset in seconds as the return code.

Input None

Output None

Return Codes The return code is the offset in seconds. A negative return code indicates that the server is west of the client.

Implementation

1. The file LNMTMZN EXEC must be placed on the 200 code disk.

2. The file RXCDAYS MODULE must be placed on the 200 code disk.

3. The following lines of code (found in file LNMCLNDR UPDATE) must be added to LNMCLNDR EXEC after the „End” statement (on line 181)

```
/* start of time zone support            */
    If time_zone_offset<>0 Then
      Do
        tzdata=LNMTMZN(start_date)
        If tzdata=-1 Then
          Exit -1
        Parse Var tzdata time_zone_offset start_date
        tzdata=LNMTMZN(end_date)
        If tzdata=-1 Then
          Exit -1
        Parse Var tzdata . end_date
      End
/* end   of time zone support            */
```

4. The new option TIME_ZONE_OFFSET must be added to LNMMIGR CONFIG: This option and the comments are found in LNMMIGR TIMEZONE.

```
# The following option specifies the time zone offset between the
# migrating Notes client and the Notes server running the Domino
# Migration Engine.  The time zone offset can be specified in hours,
# minutes and seconds with an indicator for east or west that the Notes
# server is with respect to the Notes client.

# Example: For a server in EST and client in GMT (east by 5 hours):
#        TIME_ZONE_OFFSET = -05.00.00

# Example: For a server in GMT and client in CST (west by 6 hours):
#        TIME_ZONE_OFFSET = +06.00.00

# Example: For a server in CST and client in CST:
#        TIME_ZONE_OFFSET = +00.00.00

TIME_ZONE_OFFSET = +0.00.00
```

5. The same lines of code (found in file LNMCLNDR UPDATE) must be added to LNMPNLOG EXEC after the :q.|| end_ampm:eq. statement (on line 286)

Discussion

Since there is only one setting for TIME_ZONE_OFFSET which is global to the migration process, you should only migrate groups of users who are all in the same time zone. This could be overcome by customizing the migration for each user. But you would need to find out which time zone all of your users were on.

No consideration in the code is taken for OV/VM systems running Release 2 of the ESA Calendar Feature. This release supports time zones.

Here is an example of migrating an appointment without time zone support:

1. Assume the client is in GMT.
2. Assume the Domino Migration Engine (the first server we will be using) is in CST.
3. The time zone of the VM system is irrelevant.
4. The client's appointment is at 8AM.
5. The OfficeVision/VM Migration Tool on VM extracts the 8AM time and inputs 8AM to the DME.
6. The DME stores this appointment in GMT which would be 2PM. (8AM CST=2PM GMT)
7. The client on Notes (also on GMT) looks at the appointment and it indicates 2PM. Incorrect.

With time zone support we have:

1. Assume the client is in GMT.
2. Assume the Domino Migration Engine (the first server we will be using) is in CST.
3. The time zone of the VM system is irrelevant.
4. The client's appointment is at 8AM.
5. The time zone offset is set to -06:00:00.
6. The OfficeVision/VM Migration Tool on VM extracts the 8AM time, adds -6 to the time and inputs 2AM to the DME.
7. The DME stores this appointment in GMT which would be 8AM. (2AM CST=8AM GMT)
8. The client on Notes (also on GMT) looks at the appointment and it indicates 8AM. Correct.

PROOF ADDENDA Dictionary Support

Problem Description

The OfficeVision/VM Migration Tool will not migrate OV/VM's proofreading addenda dictionaries. These files are created during spell-checking. The main file is called PROOF ADDENDA but others may be created.

Extension Description

Code is available to migrate these files. Since the target environment only accepts one addenda dictionary (USER.DIC), all of the OV/VM addenda dictionaries are combined.
The resulting PC file then becomes an attachment to a Notes memo.

A user exit is available to aid in the selection of the addenda dictionary files. By default,
only addenda files on the A-disk are selected.

A new option called MIGRATE_ADDENDAS is implemented in LNMMIGR CONFIG to control the migration of these files.

Implementation

1. The file LNMADDEN EXEC must be placed on the 200 code disk.
2. The new option MIGRATE_ADDENDAS must be added to LNMMIGR CONFIG:

```
# The following option determines whether OV/VM addenda dictionaries
# will be migrated.  If YES is specified, files with a filetype of
# ADDENDA will be migrated.

MIGRATE_ADDENDAS = NO
```

1. The following line of code must be added to LNMMUSER EXEC after the „if” statement on or about line 93:

```
translate(lnmcfg("MIGRATE_ADDENDAS")) = "YES" | , /* addendas */
```

2. The following lines of code (found in file LNMMUSER ADDEN) must be added to LNMMUSER EXEC before the statement

   „record = "..DE"” (on or about line 160):

```
/* start of addenda support */
  if translate(lnmcfg("MIGRATE_ADDENDAS")) = "YES"
   then do
    'EXEC LNMLOG 900' routine fm "Addenda"
    'EXEC LNMADDEN' formname userid fm ovnode
    If rc <> 0
     Then return -1
   end
/* end   of addenda  support */
```

User Selection of Notelogs and Notebooks

Current Situation

If the Migration Administrator indicates that notelogs and notebooks are to be migrated, then they will :hp2.all:ehp2. by migrated. Only if each end user creates a file called $$LNM$$ NOTELOGS or $$LNM$$ CMSBOOKS (containing notelog or notebook names) will the migration be limited.

This requires xedit knowledge and, for users not trained to edit files, complex instructions.

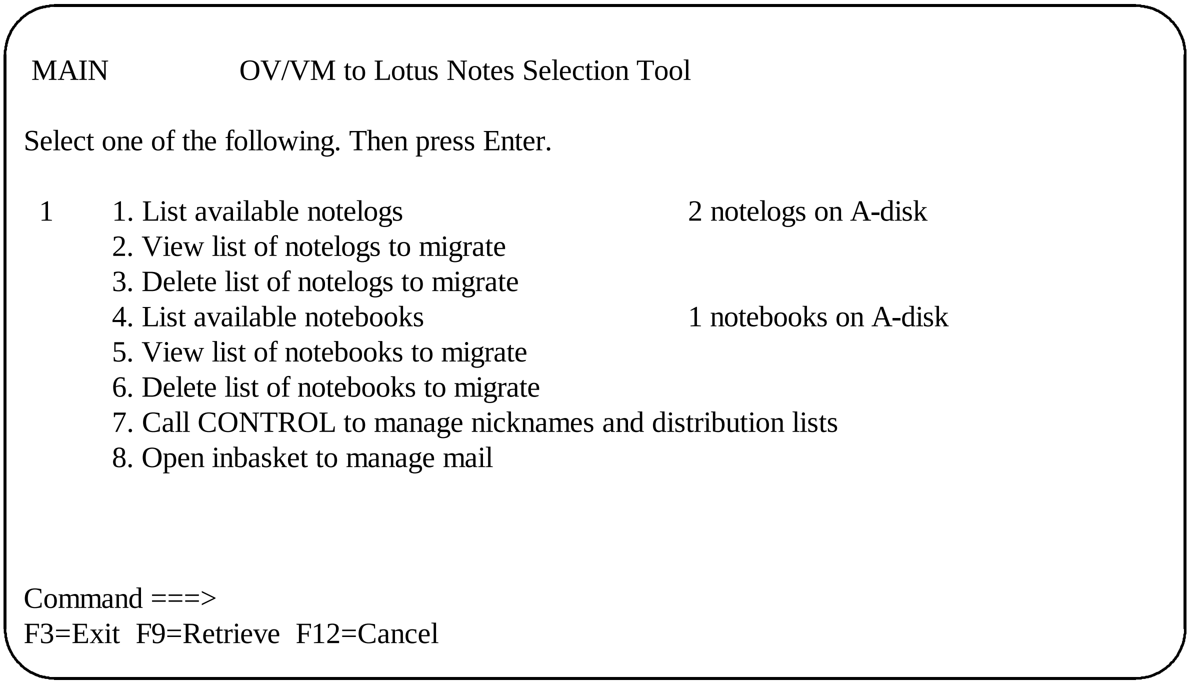
Possible Solution

A sample program called LNMSELCT EXEC has been written to aid in the selection of notelogs and notebooks. It will also allow for maintenance of these files so that old notes may be deleted. Access to the in-basket as well as nickname files is provided.

A variable on line 2 of the exec can be changed to flag notelogs and notebooks older than a specified number of days.

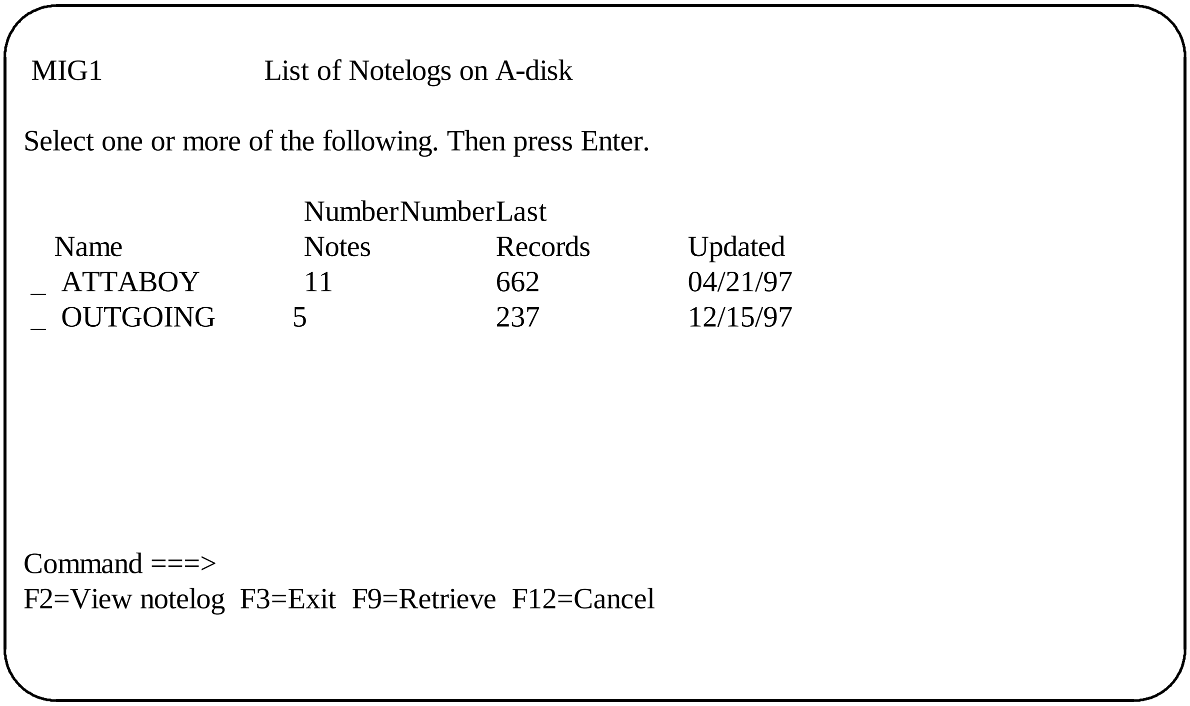
The file LNMSELCT PANLFILE contains some panel definitions. Line 2 gives the name of the program called when a notebook is to be changed.

Sample panel:



```
 MAIN            OV/VM to Lotus Notes Selection Tool

Select one of the following. Then press Enter.

 1    1. List available notelogs                  2 notelogs on A-disk
      2. View list of notelogs to migrate
      3. Delete list of notelogs to migrate
      4. List available notebooks                 1 notebooks on A-disk
      5. View list of notebooks to migrate
      6. Delete list of notebooks to migrate
      7. Call CONTROL to manage nicknames and distribution lists
      8. Open inbasket to manage mail



Command ===>
F3=Exit  F9=Retrieve  F12=Cancel
```

When a notelog is selected, the following panel is shown:



```
 MIG1                List of Notelogs on A-disk

Select one or more of the following. Then press Enter.

                         NumberNumberLast
  Name                   Notes          Records        Updated
 _  ATTABOY              11             662            04/21/97
 _  OUTGOING            5               237            12/15/97






Command ===>
F2=View notelog  F3=Exit  F9=Retrieve  F12=Cancel
```

Notelog Process Performance Support

Problem Description

Some customers have noticed performance problems while processing very large notelog files. A quick scan of the code indicates that the notelog is read multiple times and many temporary files are written during the entire process.

Extension Description

A replacement LNMPNLOG EXEC has been coded. The new routine reads the notelog once to locate the note header records and once to read all of the notelog lines into memory. All processing takes place in memory. It is recommended that LNMVM EXEC be modified so that the storage option is added to the XAUTOLOG commands and that the storage chosen be as larg
as possible. 32M would be recommended.

If a notelog is too large to fit into memory, the code will attempt to read half the file into memory. This process repeats until a successful read occurs. Then, in-memory processing will proceed but not at an optimum speed.

Sample runs indicate that elapsed time and I/O are reduced by half an both virtual and total time are reduced by more than half. The MDF is almost half the size since trailing blanks in the notes are removed. This will reduce file transfer time.

Implementation

Place the provided LNMPNLOG EXEC on the 200 code disk.
.LNMVM EXEC should be changed to add the storage option to the XAUTOLOG commands. These commands are on or about lines 672 and 679:

```
'CP XAUTOLOG ' user iplcmd ' # LNMSTART ' servid ...
```

Change this to:

```
'CP XAUTOLOG ' user iplcmd 'STORAGE 32M # LNMSTART ' servid ...
```

Migrating VM Databases

PROBLEM DESCRIPTION

There are multiple steps involved in migrating an application from a VM environment to a Notes environment. One of those steps may involve the migration of existing data in VM to Notes databases. The VM data may exist in CMS files or in complex databases.

EXTENSION DESCRIPTION

This extension is based on the methodology behind Document Migration Method 3. This methodology encompasses these elements:

- An administrator userid on VM is selected to perform the migration
- This userid will extract all data from VM
- It is responsible for creating a valid MDF
- It also uses file transfer to transfer the data to the Domino Migration Engine (DME)
- The DME has defined to it the administrator userid in a "Migration Request Sent" status along with a special Source Location document
- The special Source Location document has "Bypass communications" set for "Post-Registration Communication Option."

LNMIGRAT EXEC

Without going into prerequisites or details, the LNMIGRAT EXEC is used to perform the VM database migration. When invoked, it prompts you for details regarding the migration environment and a control file called an LNMIGRAT table which describes the VM database and the Notes database. The LNMIGRAT table can be created manually or the MIGRCMS EXEC can be used as a full-screen interface.

MIGRCMS EXEC

This exec was created to help manage one or more LNMIGRAT tables. These tables have a filetype of LNMIGRAT.

MIGRCMS will let you

- create tables
- update tables
- delete tables
- list existing tables

Each table describes 3 aspects:

1. Notes database elements
2. VM database file name and type
3. Information about fields in the VM Database and the Notes document

LNMIGRAT Table Format

MIGRCMS can be used to create this table or it can be created manually. Each record is in PIPE's varload format which looks like:

/varable-name/value-of-variable

Here are the records which MIGRCMS creates:

o /CMSFILE/fn ft

fn ft are the filename and type of the database to be migrated

o /LINESOFTAB.0/n

n is a count of the number of cross-reference records

o /LINESOFTAB.m/vmvarname notesvarname nameres

This is a cross-reference record. "m" will range from 1 to n as defined in /LINESOFTAB.0. vmvarname is the name to be assigned to a field in the VM database. notesvarname is the name of the corresponding field in the Notes document. namesres is Y if name resolution at the DME should be used. This will take a VM field in the format "userid@nodeid" and resolve it to a name.

o /DBNAME/notes-database-name

notes-database-names is the PC filename of the database. It has a file extension of NSF.

o /DBTITLE/database-title

database-title is the title of the database as seen on the desktop

o /DBNTF/database-template-name

This is the database-template-name

o /DBFORM/database-form-name

database-form-name is a form name you have defined. Case is important as well as exact spacing.

o /CMSINFO.0/n

n is a count of the number of VM field definition records. When n is zero, it means that you will use the PARSELINE exit routine in LNMIGRAT to parse out the records and assign values to REXX variables. These REXX variable names must be used also in LINESOFTAB.n records.

o /CMSINFO.m/vmvarname col-start length

This is a VM field definition record. "m" will range from 1 to n as defined in /CMSINFO.0. vmvarname is the name to be assigned to a field in the VM database. col-start is the starting column within the record and length is the field length. The length can be an asterisk in which case data to the end of the record will be assigned to the field. If your VM database contains columnar data, use these records. If not, set CMSINFO.0 to zero and use the PARSELINE exit routine in LNMIGRAT.

READFILE Exit Routine

LNMIGRAT will read your VM database using the READFILE routine. By default this routine doesn't do anything and returns a line count of zero. When the line count is zero, the database is just read into a stem variable called "LINE.".

If you don't want the VM database read this way (maybe you need to read multiple files or be selective in reading certain records), add your own code to READFILE. Just make sure the final records go into stem LINE., the linecount variable is set and LINE.0 is set to the number of records you want processed.

PARSELINE Exit Routine

The PARSELINE exit routine is called when CMSINFO.0 equals zero. This is set in the LNMIGRAT table. When you use MIGRCMS and you are defining all of the fields, CMSINFO.0 will be zero when you don't specify start columns and lengths for all fields.

Code PARSELINE when your VM database doesn't contain columnar data or you want to dynamically create field names instead of having them predefined in CMSINFO.n records.

PARSELINE must create REXX variable names the same as indicated in the LINESOFTAB.n records. These REXX names correspond to vmvarname in the record.

Setting ACL for Documents

LNMIGRAT has the capability to set an ACL on a document. In Notes, a user accessing the newly created database will only see records that they own. This corresponds to a VM database containing records for many userids but which disallows one userid from seeing or updating another user's records.

The SETACL exit routine must be coded to define who can get access to the database. Everyone you specify using SETACL gets Author access to the *database* (we haven't discussed document access yet). SETACL is called after every record has been processed. The variable "acltag" must be set in the format userid@nodeid. Maybe this data is in REXX variables defined for the record. Great, go ahead and use those names when you define acltag.

Once your users have been defined to get access to the database, you probably will want to permit the users only to access their own records. Well, you will have defined two fields in the Notes form to have type Readers and type Authors. These need to be loaded up with their Notes fullname. You need to define a VM database field to contain userid@nodeid and turn on the "name res" flag. Here is an example:

Suppose your VM database contains:

USERID	**ADDRESS**	**CITY**	**STATE**	**COUNTRY**
ALLENB	3390 Disk Drive	Poughkeepsie	NY	USA
SMITHJ	5 Maple Ave	Jonestown	NY	USA

You also know your source location node is VMNODE3 and have added in these lines to the procedure in LNMIGRAT called PROCESS_RECORD:

```
auth=userid'@VMNODE3'
rdr =userid'@VMNODE3'
```

Auth and rdr are our definitions for the document author and reader.

Your VM database field definitions look like:

VM Field	**Start**	**Field**	**Name**	**Notes Field**
Name	**Column**	**Length**	**Res?**	**Name**
userid__	___1	___8	N	Userid________________________
address_	__11	__16	N	Address_______________________
city____	__31	__12	N	City__________________________
state___	__44	___5	N	State_________________________
country_	__50	__10	N	Country_______________________
auth____	___1	___8	Y	Authors_______________________
rdr_____	___1	___8	Y	Readers_______________________

Note that auth and rdr are defined in our table and it looks like they get their values from columns 1-8 in the VM database. They do, but our two lines in PROCESS_RECORDS override this definition. See that the Names Resolution field is set to Y? The userid/nodeid combination is enough to allow a lookup in the Migration Name and Address book. The Notes fullname will then be loaded into the Notes fields Authors and Readers.

Prerequisites

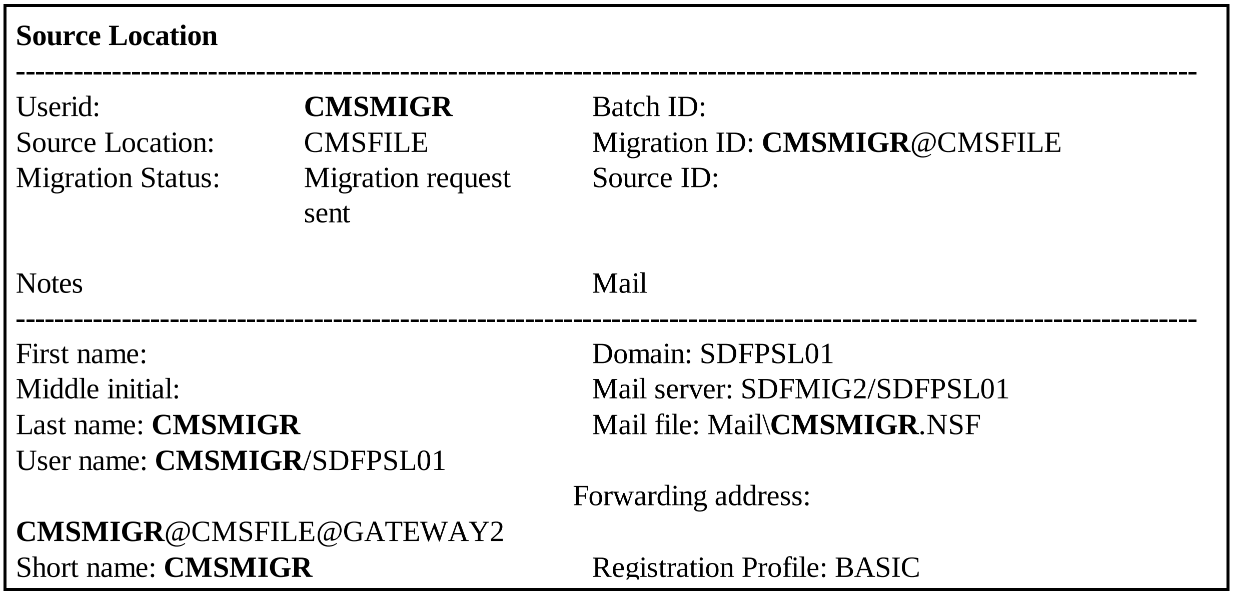
Before you migrate a VM database, there are three basic prerequisites:

1. Notes Database Design

You must already have designed your documents within your Notes database. The field names will be used in the LNMIGRAT table. Make sure the form name created is spelled out exactly in the DBFORM record.

2. Defining a 'user' for the VM database migration

The user performing the VM database migration (must not be the VM Migration Server) must have a person record defined within the DME Migration Name and Address Book. In the following figure, note where the userid (**CMSMIGR**) is used:



Source Location

Userid:	**CMSMIGR**	Batch ID:
Source Location:	CMSFILE	Migration ID: **CMSMIGR**@CMSFILE
Migration Status:	Migration request sent	Source ID:

Notes	Mail
First name:	Domain: SDFPSL01
Middle initial:	Mail server: SDFMIG2/SDFPSL01
Last name: **CMSMIGR**	Mail file: Mail\**CMSMIGR**.NSF
User name: **CMSMIGR**/SDFPSL01	
	Forwarding address:
CMSMIGR@CMSFILE@GATEWAY2	
Short name: **CMSMIGR**	Registration Profile: BASIC

Figure 8. VM Database Migration Admin Id

3. Source Location Document

The Source Location requires the bypass standard communications option. See the following figure for a sample of the source location document. This means there are no key files generated. After registration, a user from this source location will be automatically put into a status of Migration Request Sent.

```
Basics
---------------------------------------------------------------------------------------------------------------------
Source Location                 CMSFILE
Description                     Special Location for Documents

Processing Options
---------------------------------------------------------------------------------------------------------------------
Incoming Work Directory
Specify value                   c:\migra\incoming\document

Outgoing Work Directory
Specify value                   c:\migra\outgoing\document

Server for server-based
databases                       Local

Character Set                   IBMCP850

Max Migration Errors
Use default                     10

Name Resolution                 Enabled

Directory Migration
Pre-processing                  Enabled

Name Collision Match            on User name

Name Collision Options
  - for Registered
    Notes Users                 Use existing Person document

  - for other
    Person documents            Update if duplicate

Post-Registration
Communication Option            Bypass communications
```

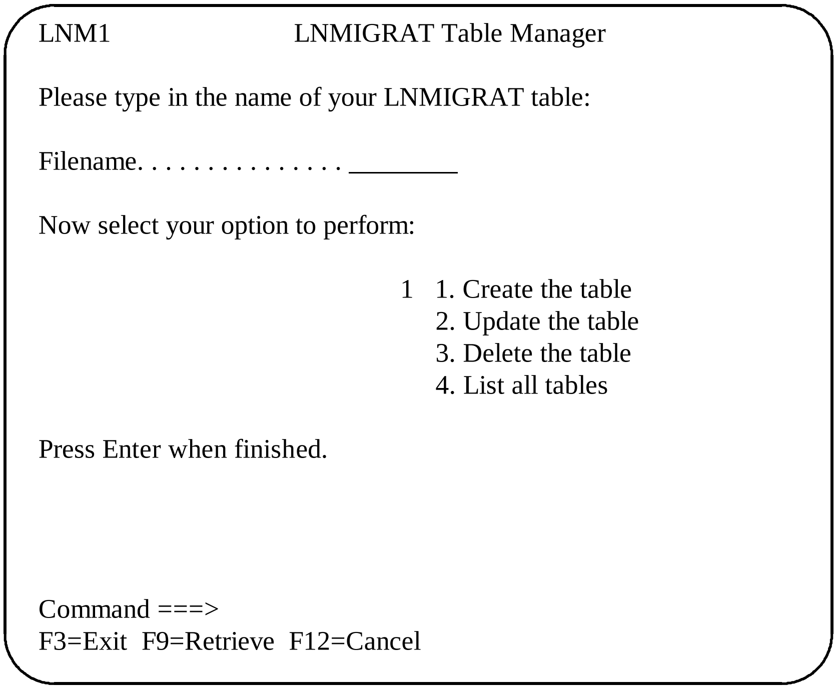
IMPLEMENTATION

1. The files LNMIGRAT EXEC and MIGRCMS EXEC must be placed on the 200 code disk.
2. Modification to LNMHEADR EXEC on the 200 code disk must be done. File LNMHEADR UPDATE2 contains code which must be added to the file LNMHEADR EXEC. It needs to go in before line 30. This is the line containing "'PIPE CMS ID var result'".
3. The configuration file (LNMMIGR CONFIG) needs to be updated to include the option MIGRATE_CMSFILE = YES.
4. Userid performing migration:
   a. SET LDRTBLS 5 must be executed so that file transfer works
   b. Access to the file transfer software must be established
   c. The userid must be able to successfully perform a file transfer
   d. Must have access to the VM database to migrate
   e. Must have write access to a VM Migration Server work disk

5. All CUA2001 package files (start with RXC)

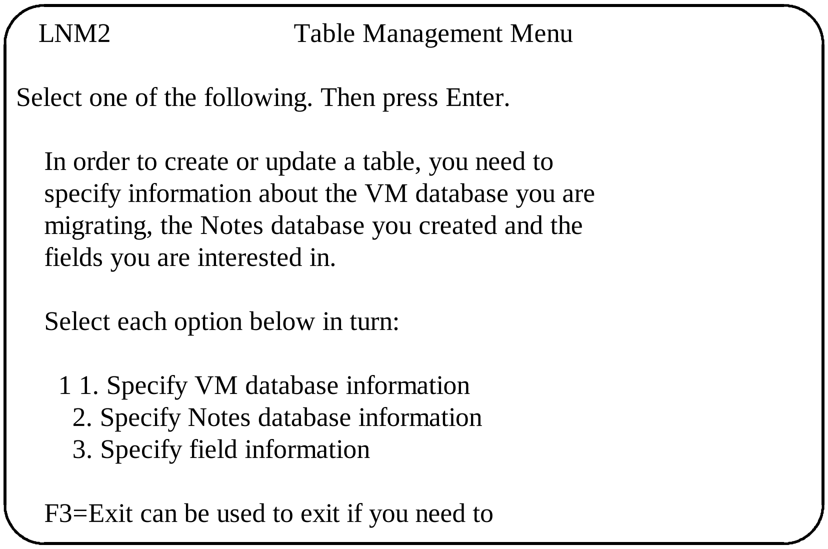
PERFORMING THE MIGRATION

Before the migration of the VM database is initiated, an LNMIGRAT table must be created. You can do this manually or invoke MIGRCMS. The initial panel look like:



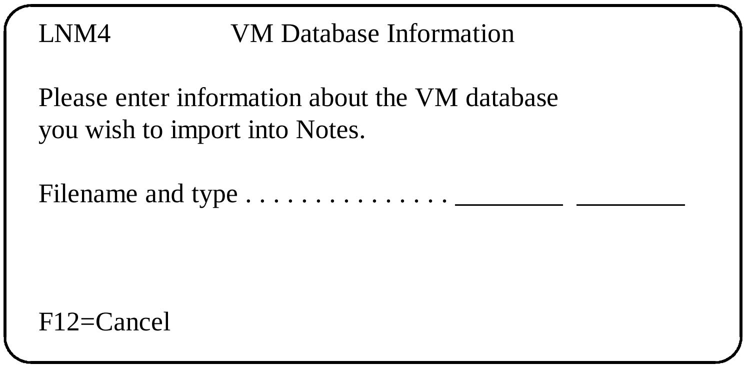
```
LNM1                    LNMIGRAT Table Manager

Please type in the name of your LNMIGRAT table:

Filename. . . . . . . . . . . . . . . ________

Now select your option to perform:

                                1   1. Create the table
                                    2. Update the table
                                    3. Delete the table
                                    4. List all tables

Press Enter when finished.



Command ===>
F3=Exit  F9=Retrieve  F12=Cancel
```

Just fill in the name of your table, select option 1 and press Enter. The next panel lets you select three options for defining the table contents. You must select each option in turn.



```
  LNM2                    Table Management Menu

Select one of the following. Then press Enter.

   In order to create or update a table, you need to
   specify information about the VM database you are
   migrating, the Notes database you created and the
   fields you are interested in.

   Select each option below in turn:

     1 1. Specify VM database information
       2. Specify Notes database information
       3. Specify field information

   F3=Exit can be used to exit if you need to
```

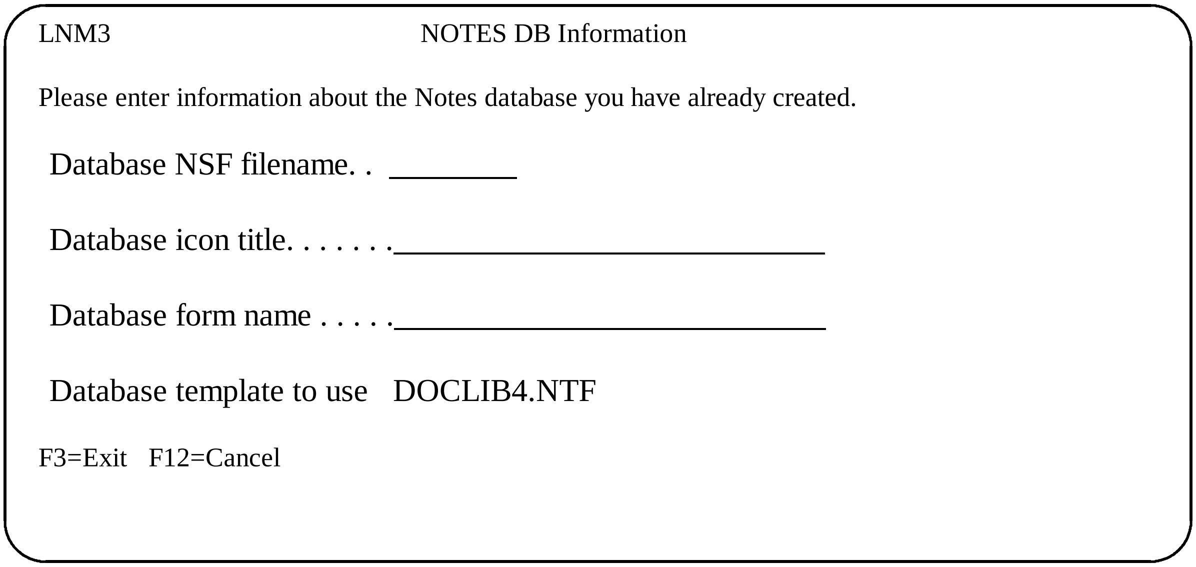
Option 1 shows you:



```
LNM4              VM Database Information

Please enter information about the VM database
you wish to import into Notes.

Filename and type . . . . . . . . . . . . . . . ________  ________



F12=Cancel
```

Just enter the VM database name and type.

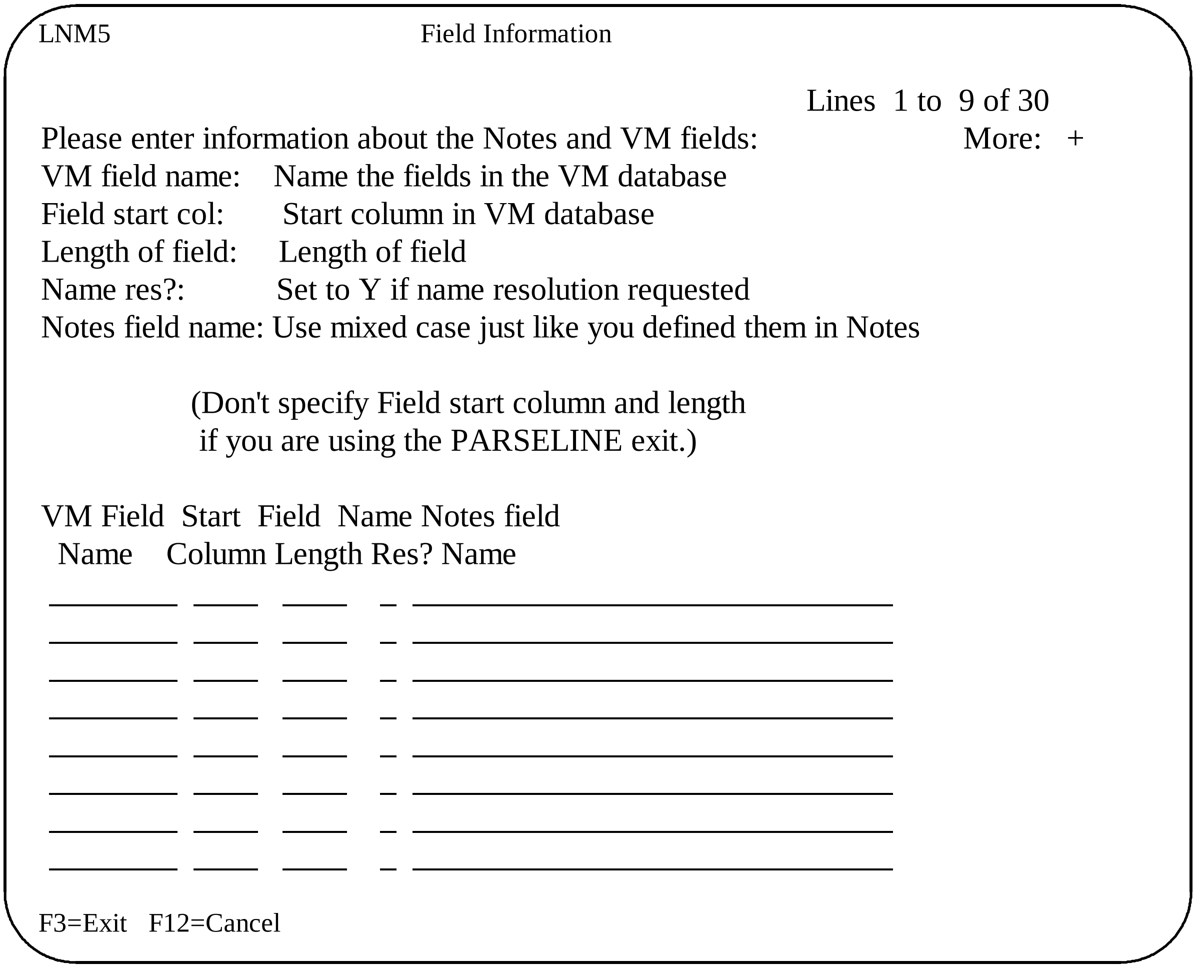
Option 2 shows you:



```
LNM3                              NOTES DB Information

Please enter information about the Notes database you have already created.

 Database NSF filename. .  ________

 Database icon title. . . . . . .____________________________

 Database form name . . . . .____________________________

 Database template to use   DOCLIB4.NTF

F3=Exit   F12=Cancel
```

Type in the requested information.

Option 3 shows you:

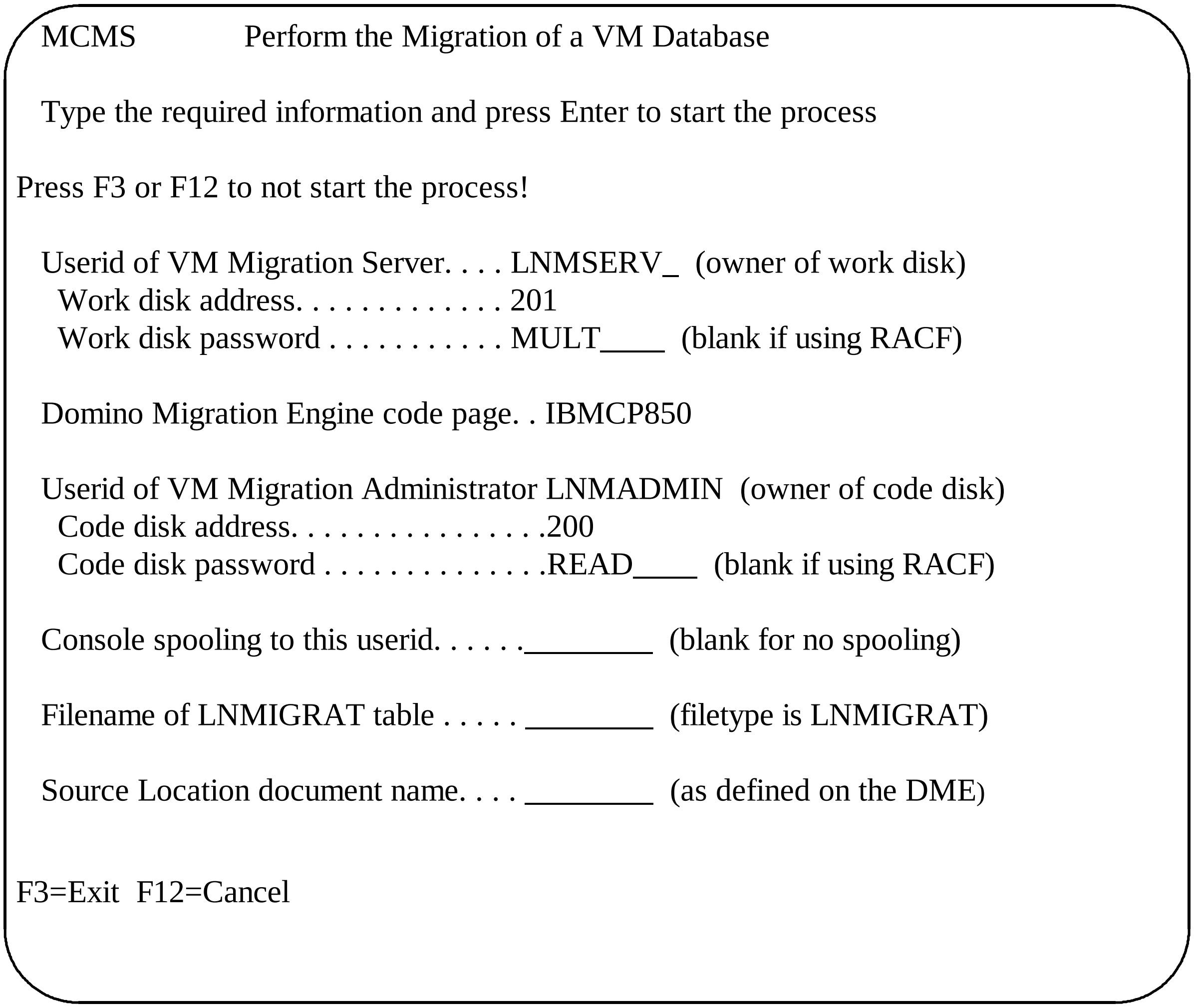


```
LNM5                         Field Information

                                                         Lines  1 to  9 of 30
Please enter information about the Notes and VM fields:              More:   +
VM field name:    Name the fields in the VM database
Field start col:   Start column in VM database
Length of field:   Length of field
Name res?:         Set to Y if name resolution requested
Notes field name: Use mixed case just like you defined them in Notes

              (Don't specify Field start column and length
               if you are using the PARSELINE exit.)

VM Field  Start  Field  Name Notes field
  Name    Column Length Res? Name
 ________ _____  _____   _  ________________________________
 ________ _____  _____   _  ________________________________
 ________ _____  _____   _  ________________________________
 ________ _____  _____   _  ________________________________
 ________ _____  _____   _  ________________________________
 ________ _____  _____   _  ________________________________
 ________ _____  _____   _  ________________________________
 ________ _____  _____   _  ________________________________

F3=Exit   F12=Cancel
```

For each field, type in the requested information. Don't specify start column and field length for any fields if you are using the PARSELINE exit. Name Res should be Y if you want the field resolved to a name from the Migration Name and Address book. You'll be expected to have the field loaded with data in the format "userid@nodeid."

LNMIGRAT EXEC

Start the migration by executing LNMIGRAT. The one panel included looks like:



```
 MCMS           Perform the Migration of a VM Database

 Type the required information and press Enter to start the process

Press F3 or F12 to not start the process!

 Userid of VM Migration Server. . . . LNMSERV_  (owner of work disk)
   Work disk address. . . . . . . . . . . . . 201
   Work disk password . . . . . . . . . . . MULT____  (blank if using RACF)

 Domino Migration Engine code page. . IBMCP850

 Userid of VM Migration Administrator LNMADMIN  (owner of code disk)
   Code disk address. . . . . . . . . . . . . . . .200
   Code disk password . . . . . . . . . . . . . .READ____  (blank if using RACF)

 Console spooling to this userid. . . . . .________  (blank for no spooling)

 Filename of LNMIGRAT table . . . . . ________  (filetype is LNMIGRAT)

 Source Location document name. . . . ________  (as defined on the DME)


F3=Exit  F12=Cancel
```

Just fill out the panel as indicated and press Enter.

Migrating OV/VM Conference Rooms

PROBLEM DESCRIPTION

Conference room calendar data as well as the list of available conference rooms are not migrated to the Notes resource database.

Background

In order to migrate conference room data from OV/VM to Notes, you must be familiar with this data both on OV/VM as well as on Notes.

In OV/VM, there is a file called CONFEREN EPSFILE which contains a list of resources. OV/VM makes no distinction between rooms and resources. Most customers just list conference rooms. This list contains the name of the resource and a description. Any additional information about the resource could be contained in the free-format description field. But OV/VM won't really know about it. There is also no hierarchical structure to these resources.

Notes, on the other hand, starts by letting the administrator define a set of sites. Each site can contain rooms or resources. Resources are further categorized. So, you could have a site called HQRTRS with rooms RM100, RM101, RM102 and resources Jet1, Limo1, Limo 2 and Van 1 under the category Transportation along with resources Overhead 1, Overhead 2 and Video 1 under the category Equipment.

Rooms also have a seating capacity. Both rooms and resources have availability days and times to limit when they can be reserved.

The OV/VM calendar data for resources and rooms has its counterpart in Notes. The resource database contains all of this data plus the site and resource definitions discussed prior. There can even be more than one resource database. In addition, the site and resource are also listed in the Public Name and Address book.

Finally, as reservations are made in the resource database, a busytime database is also kept up to date to aid in

free-time searches for times or resources.

As you surmise, one couldn't just migrate OV/VM conference rooms to Notes without providing more detailed information so that the rooms could properly match to those in Notes.

Extension Description

The extension code has the ability to allow you in OV/VM to define the structure as required by Notes. Further, it allows for site definitions, resource definitions and Public Name and Address book definitions to be migrated. Only once these definitions are migrated can the actual room reservations (calendar data) be migrated. This can all be done at once if required.

One important word of caution though. The extension code has the capability to migrate these definitions to Notes more than once. You will then get multiple definitions.

Migrating the conference room data follows the methodology of migrating VM databases. See page 50 for a short description.

MIGRCONF EXEC

This exec has been created to aid in the definition of sites and resources. It's very similar in operation to MIGRCMS. One or more conference room table definitions can be managed. These tables have a filetype of CONFROOM.

Each conference room table defines 4 aspects:

1. General database information
2. Resource database information
3. Site definitions
4. Resource definitions

General Database Information

Three items need to be specified:

1. Database NSF filename. This is the PC filename without the extension of NSF.
2. Database icon title. If the resource database already exists, this will be ignored. Otherwise, specify the title as you wish it to be displayed on the workspace icon.
3. Database template to use. This should be RESRC45.NTF. If you have another one, specify it here.

Resource Database Information

Three items need to be specified:

1. Resource domain. E.g., ILIC8
2. Resource server. This is the server name followed by the domain name. E.g., ILICLAB8/ILIC8.
3. Resource filename. This is the same as the database NSF filename except you need to specify the file extension. E.g., RESOURCE.NSF.

Site Definitions

MIGRCONF lets you list all of the sites in your company. Just type in each site name as required. Blanks are permitted and you may specify upper and lower case. However, HQRTRS and Hqrtrs are considered duplicate definitions which is an error.

Resource Definitions

Resources also include room definitions. Resources are defined within a site. MIGRCONF first presents a list of sites and, after you select one, you'll see a list of resources for that site. If none are defined, the list is empty and you can press a PF key to add in a room or resource. When you add in a resource you will be prompted for the OV/VM resource name as well as the corresponding Notes resource name. These don't have to be the same. One PF key is of note since it lets you bulk load resource data from a file. Typically this would be CONFEREN EPSFILE.

Bulk Loading Resource Definitions

If all of your resource definitions are in a file like CONFEREN EPSFILE, they can be loaded automatically. After you specify the input filename and type, you need to tell MIGRCONF how to process the file for the five key important pieces of information:

1. Whether the resource is a room or not. You can elect to have all items in the file as rooms, all as resources or let an exit (LNMEX30) decide.
2. Where the resource name is located in the file. You can specify columns or let the exit decide.
3. Where the resource description is located in the file. You can specify columns or let the exit decide.
4. What the resource availability times are. You can have a default set defined or let the exit decide.
5. What other comments are for the resource. You can have a default defined or let the exit decide.

For rooms, you will be prompted for information about the room capacity. You have four choices:

1. You can set a default value for all rooms.
2. You can specify columns in the file.
3. You can specify keywords in the file. For example: 20 chairs, seats 10 or capacity 15.
4. Let the exit decide.

For resources, you will be prompted for the category. You have two choices:

1. You can set a default for all resources.
2. Let the exit decide.

Exit LNMEX30

This exit is called by MIGRCONF to process a record during bulk load processing of CONFEREN EPSFILE or another file containing resource information. As each record is processed, LNMEX30 is called.

Parameters 1, 2 and 3 will be the filename, type and mode of the input file. Parameter 4 will be the record number being processed. Remaining parameters will be the input line.

All output must be in global variable group MIGRATE. Data returned must be in these variable names:

RESNAME - name of the resource

RESDESC - description of the resource

TIMES.n - availability times of the resource. n ranges from 1 to 7. TIMES.1 is for Sunday, TIMES.2 is for Monday, etc. If the resource is not available for a particular day, the value should be null. Otherwise, the value should be the start time followed by the end time. Times are in the format hh:mmAM or hh:mmPM. Example: Available Monday from 9AM to 5PM, TIMES.2 contains 09:00AM 05:00PM.

COMMENT - other comment about the resource

RESTYPE - 1 if a room. 2 if a resource.

CAPACITY - room capacity

CATEGORY - resource category

Any return code from the exit is ignored.

Prerequisites

1. Follow the prerequisites for migrating VM databases on page 54 where it details defining a 'user' for the migration. For the userid, specify the userid where you are running the migration from. For the source location, you may use RESOURCE or anything else you desire. Just make sure it's the same as the name in the source location document.
2. Make up a source location document as described on page 54 but use a source location name like RESOURCE. To have the resource database on a specific server other than the one running the DME, set "Server for server-based databases" to the full canonical name of the server.
3. Invoke MIGRCONF to define your sites and resources.

Implementation

1. These files need to go on the 200 code disk:

```
LNMRSRCE EXEC
LNMRSRC0 TABLE
LNMRSRC1 TABLE
LNMRSRC2 TABLE
LNMRSRC3 TABLE
MIGRCONF EXEC
All CUA2001 package files (start with RXC)
```

2. Modification to LNMHEADR EXEC on the 200 code disk must be done. File LNMHEADR UPDATE2 contains code which must be added to the file LNMHEADR EXEC. It needs to go in before line 30. This is the line containing "'PIPE CMS ID var result'".
3. The configuration file (LNMMIGR CONFIG) needs to be reviewed for the options CALENDAR_START and CALENDAR_END. Conference room data only between these dates are migrated. Also, data on the calendar which is not entirely in the availability range defined will not be migrated.
4. The configuration file (LNMMIGR CONFIG) needs to have a new option called MIGRATE_CMSFILES added and set to YES. Add in the line:

```
MIGRATE_CMSFILES = YES
```

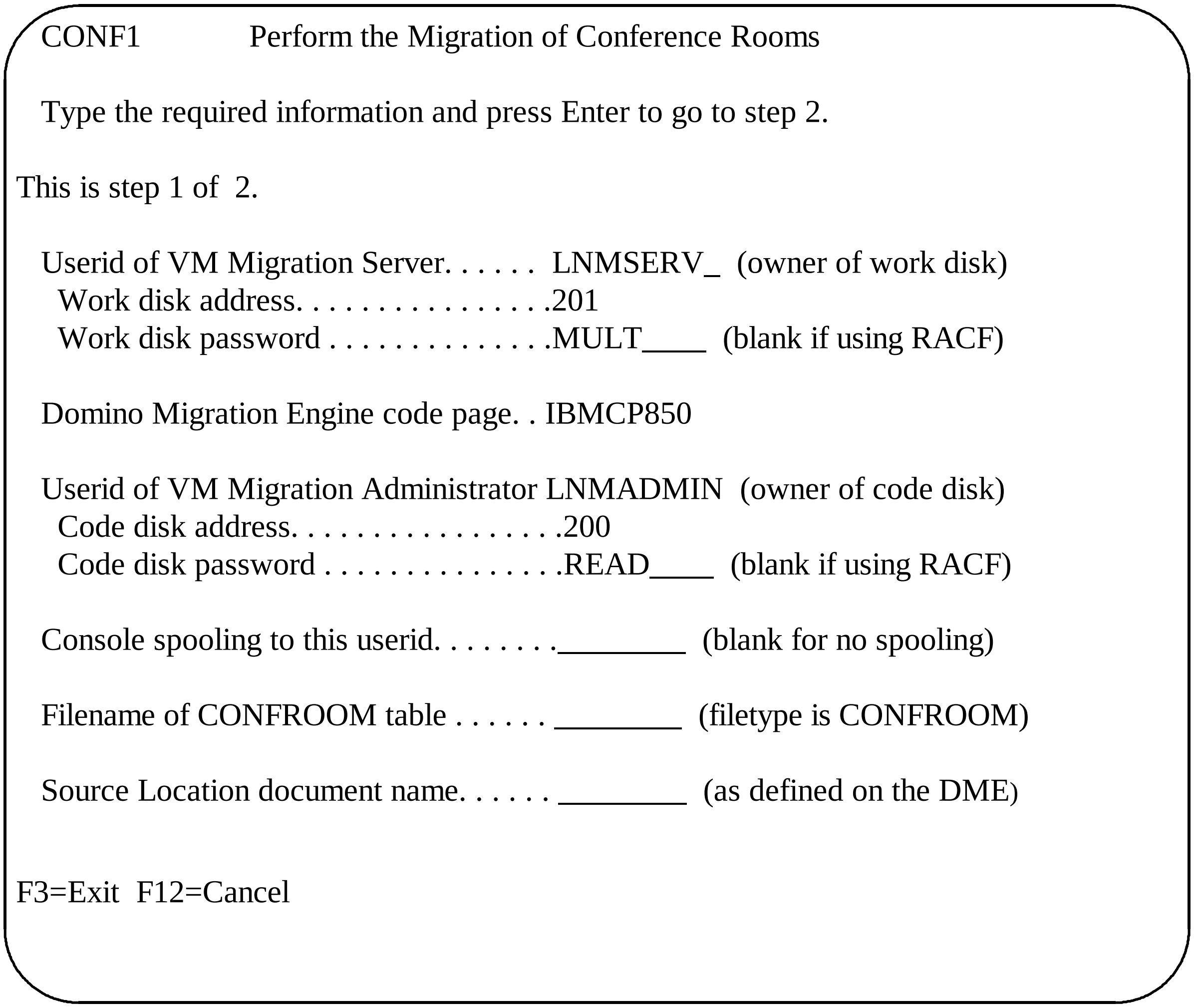
5. Userid performing migration:
   a. SET LDRTBLS 5 must be executed so that file transfer works
   b. Access to the file transfer software must be established
   c. The userid must be able to successfully perform a file transfer
   d. Must have access to the conference room's calendars.
   e. Must have write access to a VM Migration Server work disk
   f. Must have access to the 200 code disk.

Performing the Migration

Start the process by defining your resource database, sites and resources using the MIGRCONF tool. The VM Migration Server is not involved in any way when you migrate conference rooms. Log on to VM using a userid like SYSADMIN and invoke MIGRCONF. Make sure you have followed the prerequisites and the notes under the implementation section discussed prior.

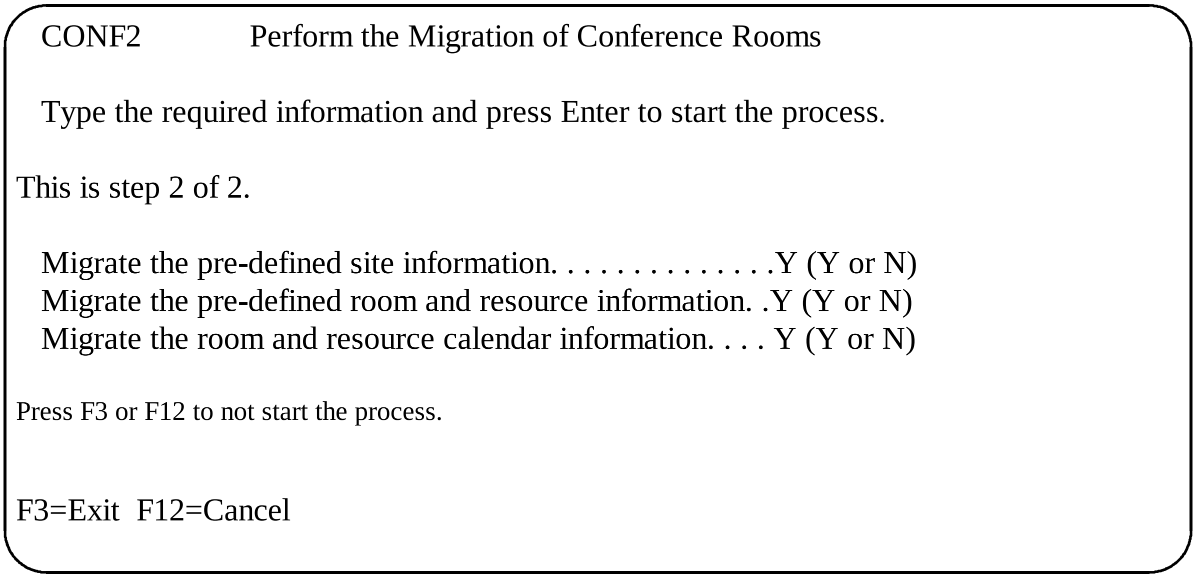
If you already have sites and resources defined in Notes, make sure you repeat these definitions in VM but **don't** actually migrate these site and resource definitions under step 2 of LNMRSRCE. If you do, you'll get duplicates. If they are not defined in Notes, then do make the definitions in VM and do migrate them. Remember that whatever you define in the resource section will determine which resource data gets migrated. For example, if you only define 5 conference rooms under one site, then you will just migrate calendar data for those 5 conference rooms.

Start the migration process by executing LNMRSRCE. Again, do this from SYSADMIN or another userid which meets the prerequisites. The first panel looks like:



```
  CONF1           Perform the Migration of Conference Rooms

  Type the required information and press Enter to go to step 2.

This is step 1 of  2.

  Userid of VM Migration Server. . . . . .  LNMSERV_  (owner of work disk)
    Work disk address. . . . . . . . . . . . . . . .201
    Work disk password . . . . . . . . . . . . . .MULT____  (blank if using RACF)

  Domino Migration Engine code page. . IBMCP850

  Userid of VM Migration Administrator LNMADMIN  (owner of code disk)
    Code disk address. . . . . . . . . . . . . . . . .200
    Code disk password . . . . . . . . . . . . . . .READ____  (blank if using RACF)

  Console spooling to this userid. . . . . . . .________  (blank for no spooling)

  Filename of CONFROOM table . . . . . . ________  (filetype is CONFROOM)

  Source Location document name. . . . . . ________  (as defined on the DME)


F3=Exit  F12=Cancel
```

The second panel looks like:



```
  CONF2          Perform the Migration of Conference Rooms

  Type the required information and press Enter to start the process.

This is step 2 of 2.

  Migrate the pre-defined site information. . . . . . . . . . . . . .Y (Y or N)
  Migrate the pre-defined room and resource information. .Y (Y or N)
  Migrate the room and resource calendar information. . . . Y (Y or N)

Press F3 or F12 to not start the process.


F3=Exit  F12=Cancel
```

Here you have a chance to decide what to migrate. You should migrate your site and resource information only once. Of course, if you already have them defined in Notes, you won't migrate them. If you try to do it twice, LNMRSRCE will detect this and ask you if you really want to do it. No check is made to see if the actual calendar information has already been migrated. In all cases, migrating more than once will result in duplicate information in the Notes resource database.

When the process finishes, an audit trail of the results is displayed. This is stored in the file MIGRATE AUDIT. The Migration Data File containing any definitions plus calendar data will be sent to the incoming directory on the Domino Migration Engine by SYSADMIN or whatever userid is executing LNMRSRCE. The resulting resource database will end up on the server where the Domino Migration Engine resides.

Exit LNMEX31

During processing of all events on the calendar, LNMRSRCE will call exit LNMEX31 if it exists. Its purpose is to return a phone number to place in the Notes reservation. Input parameter 1 is the resource name and the remaining parameters are the purpose lines in the reservation. If a phone number is returned, it should be in global variable group MIGRATE in the variable PHONENUMBER.

Post Migration Activities

Once you have migrated resource definitions and the events to the Notes calendar, there are three steps to complete the process:

1. The resource definitions migrated contain some data which is only partially processed by the Domino Migration Engine. You will have to create and run two agents on the resources you have migrated.

Open up the resource database and click on the create pulldown and select agents. Use a title of FixAvail, agent should run from menu and act on selected documents. Click on formula and enter a formula like:

```
FIELD SundayAvailable:=SundayAvailable;
@If(SundayAvailable=” ”;@SetField(“SundayAvailable”;””);””);
```

NOTE: The first set of double quotes contains a space. The last two have no spaces.

Repeat this for all days of the week so that your formula contains 14 statements. You may put the 7 FIELD statements at the beginning and the 7 @If statements at the end.

Make up a second agent and call it Fix$times which should run from the menu and act on selected documents. Its formula looks like:

```
@If(@Contains(@Text($Times1);”-”);@Return(“”);””);
FIELD $Times1 := @TextToTime(@Text(@Subset($Times1;1);”S1”) + “-” +
@Text(@Subset($Times1;-1);”S1”));
```

Repeat the last FIELD statement 6 more times using $Times2, $Times3, etc.

Now display all of the resources in your database and select the ones you just migrated. Run these two agents. Click on actions pulldown and then click on FixAvail. Re-select the same resources and run agent Fix$times.

2. The resource events (reservations) migrated contain some data which is also only partially processed by the Domino Migration Engine. You will have to create and run an agent on the reservations you have migrated.

   Open up the resource database and click on the create pulldown and select agents. Use a title of Fixtrange, agent should run from menu and act on selected documents. Click on formula and enter a formula like:

   ```
   @If(@Contains(@Text(TimeRange);”-”);@Return(“”);””);
   FIELD TimeRange := @TextToTime(@Text(@Subset(TimeRange;1);”S1”) + “-” +
   @Text(@Subset(TimeRange;-1);”S1”));
   FIELD OptimalStartTime := @TextToTime(@Text(@Subset(OptimalStartTime;1)) + “-” +
   @Text(@Subset(OptimalStartTime;-1)));
   FIELD TRTEST := @TextToTime(@Text(@Subset(TRTEST;1)) + “-” +
   @Text(@Subset(TRTEST;-1)));
   ```

   Now display all of the reservations in your database and select them all. (Edit pulldown and then Select All.) Run agent Fixtrange. Click on actions pulldown and then click on Fixtrange.

3. Finally, on the Notes server containing the resource database, run the command

   ```
   TELL SCHED VALIDATE
   ```

Notes

1. **Time Zones -** at the moment time zone support is not yet included.

2. **REXX Compiler -** don't compile LNMRSRC or MIGRCONF execs. If this needs to be done, contact me.

Migrating the OV/VM Bulletin Board

PROBLEM DESCRIPTION

The bulletin board system data within OV/VM are not migrated to an appropriate Notes database.

Background

OV/VM bulletin board data is accessed by the users with the BBOARD command. This command reads the file BBOARD EPQCTL which contains the list of bulletin board categories. The categories are displayed to the user. When a category is selected, a disk is accessed to read the items in that category. The items for that category are listed in a file with a filetype of EPQLIST. This list is then presented to the user. When an item is selected, it is read from the disk and displayed to the user. The bulletin board system keeps track of whether you have read an item or not by recording this information on the user's A-disk.

Extension Description

The extension code exploits the general nature of the extension code for migrating VM databases. This is described on page 50. You must read and understand this extension code before you migrate bulletin board data.

Migrating bulletin board entries includes:

migrating the categories
migrating the item titles for display in a Notes view and
migrating the items themselves

into a Notes document database. The default document template of DOCLIB4.NTF is used. You may have your own customized document database so you should be aware of exactly what data gets migrated. These Notes fields are migrated:

Categories	- category of item
Subject	- item title
Date	- start date of item
From	- userid/nodeid doing migration
Body	- item text

Once the bulletin board items are migrated you will notice that all items are in an unread status. There is no code in the extension to handle this feature. All Notes users will be able to access all items since it's a shared server-based database.

Prerequisites

You must meet the prerequisites for migrating VM databases. As well, when you run the migration, you need access to the main bulletin board file, BBOARD EPQCTL and have the ability to access the disks indicated in this file.

Implementation

Follow the implementation guidelines for migrating VM databases. Additionally:

1. Place the file BULLETIN LNMIGRAT on your A-disk. Invoke MIGRCMS and review the 3 optional parts contained in this table. Review and carefully change what you need in the Notes database section. Review the field information section. You may need to add to this section but if you make any updates or deletions you will need to make coding changes in LNMIGRAT EXEC to accommodate them.
2. Update the LNMIGRAT EXEC procedure called READFILE. Include the code contained in the file BULLETIN READFILE.

Performing the Migration

Start the migration by executing LNMIGRAT.

Migrating 'To Do' Files

Problem Description

Many installations have implemented a 'to do' type facility. This enables their users to track items to do or which are due to them by other people. No such data is migrated by the OV/VM Migration Tool since it isn't really defined within the OV/VM environment.

In most cases, a 'to do' entry contains these types of items:

A subject line
A due date
A start date
A completion date if the to do is completed
An assigned priority
A more descriptive body portion which goes into details about the to do
A userid to whom the 'to do' is assigned

This extension will map these items to the 'to do' facility within the Notes calendar.

Extension Description

There is no exit available to process 'to dos'. However, changes are available to the base code which process this file. The changes implement a new option called MIGRATE_TODOS.

When MIGRATE_TODOS is set to YES in LNMMIGR CONFIG, the 'to do' process will read the 'to do' file and load the data into the 'Tasks' form of the Notes inbox.

However, since the layout of a 'to do' file is not defined well enough, you will have to finish coding the READFILE and PARSELINE subroutines in LNMTODO EXEC. These subroutines are documented in the exec and have some sample code imbedded.

The Notes 'to do' facility really only has the 'to do' items listed above so if you have additional items in your 'to do' file, you may not be able to map them to the Notes version.

Implementation

1. The file LNMTODO EXEC must be placed on the 200 code disk.
2. The new option MIGRATE_TODOS must be added to LNMMIGR CONFIG:

```
# The following option determines whether ‘to do’s will
# be migrated.  If YES is specified, ‘to do’ entries will be
# migrated.

MIGRATE_TODOS = YES
```

3. The following line of code must be added to LNMMUSER EXEC after the ‘if”statement on or about line 93:

```
translate(lnmcfg("MIGRATE_TODOS")) = "YES" | , /* todos */
```

4. The following lines of code (found in file LNMMUSER TODO) must be added to LNMMUSER EXEC before the statement

   “record = "..DE"” (on or about line 160)::

```
/* start of todo support */
    if translate(lnmcfg("MIGRATE_TODOS")) = "YES"
      then do
        'EXEC LNMLOG 900' routine fm "Todo"
        'EXEC LNMTODO' userid fm ovnode
        If rc <> 0
          Then return -1
      end
/* end   of todo support */
```

Maintaining 'Read/Unread' Status

Problem Description

The current design of the migration tool doesn't keep the read/unread status of OV/VM inbasket items.

Extension Description

This extension is a set of modifications which need to be made to the inbasket processing routines to extract, process and save the status of a mail item.

The modifications are to three routines:

1. LNMPBSKT EXEC
2. LNMPNLOG EXEC
3. LNMPDOC EXEC

These modifications examine byte 192 in the records of the OFSMAIL PCDATA file. If this byte is hex code 00 or 41, then a UM tag is inserted into the MDF for the mail item. This signifies an unread mark.

Just as the inbasket contents are saved in the file $$LNM$$ OFSLOGfl in case the user is migrated again, the read/unread marks are saved in the file $$LNM$$ PCMARKS.

Implementation

The modifications and the instructions for making them are contained in the file URMARK CODE.

AWAY Support

Problem Description

Customers who are licenced for the PRPQ called AWAY Facility/VM (5699-FLP) likely want to migrate the existing AWAY entries on VM to the Notes “Out of the Office” facility. The OV/VM Migration Toolkit is not aware of this facility so code needs to be added to handle this situation.

Extension Description

There is no exit available to extract AWAY entries so, changes are available to the base code.

The code provided extracts all of the AWAY entries and determines the current one in effect. Other AWAY entries will be ignored. This is because, while the AWAY facility on VM lets the end user define multiple AWAY entries for future time periods, Notes only lets you define one.

The code also implements a new option called MIGRATE_AWAYS.

One further note needs to be made and that is with regard to enabling the “Out of the Office” agent. It is a requirement that each end user open their mail database and go to the All Documents view. There is a special item from “System” with a subject of “Please do not delete - Out of Office Profile”. You must double click on this item and then click on the action button which says “Enable out of Office Agent”.

Implementation

1. The files LNMAWAY EXEC and LNMAWAY TABLE must be placed on the 200 code disk.
2. The new option MIGRATE_AWAYS must be added to LNMMIGR CONFIG:

```
# The following option determines whether AWAY entries will
# be migrated.  If YES is specified, they will be migrated.
MIGRATE_REMINDERS = YES
```

3. The following line of code must be added to LNMMUSER EXEC after the „if”statement on or about line 93:

```
translate(lnmcfg("MIGRATE_AWAYS)) = "YES" | , /* AWAY */
```

4. The following lines of code (found in file LNMMUSER AWAY) must be added to LNMMUSER EXEC before the statement

   „record = "..DE"” (on or about line 160)::

```
/* start of AWAY support */
    if translate(lnmcfg("MIGRATE_AWAYS")) = "YES"
      then do
        'EXEC LNMLOG 900' routine fm "Away Entries"
        'EXEC LNMAWAY' userid fm
        If rc <> 0
          Then return -1
      end
/* end   of AWAY support */
```

Tips

Limiting the amount of data migrated

Question

Is there a way to limit the amount of data migrated or ways to control what gets migrated?

Answer

Yes. For one thing, on the Domino Migration Engine you can specify the maximum amount to migrate by size (bytes). You can always change the VM migration options so that you just don't migrate certain kinds of data (notes, documents, calendars, etc). The end user can also specify notelogs or notebooks to migrate by placing their filenames in a file on their A-disk.

Some installations provide for a chargeback scheme. Computing services are charged back to the end user. Consider charging a fixed amount for performing a migration. Some users may not want to have their data migrated if there is a cost associated with it. You could also charge per megabyte of data migrated. The amount of data migrated is not readily available to you but with modifications to the base code you can get to it. (The exec LNMVM contains a procedure "size_ok" which computes the amount of megabytes in the variable "ctmegs". Just record this amount in the log file or wherever you want. Hint: use LNMLOG right before the "Return 1" statement.)

How to size the work disks

Question

What size should the work disks be?

Answer

The VM Migration Server requires one or more work disks to be defined and the problem is "What size should the work disks be?"

We won't discuss how many work disks should be assigned. This is another matter.

Background

Work disks are owned by the VM Migration Server and are assigned to each user when a migration request is processed for that user. This work disk is used to hold all of the OV/VM data to be migrated to Lotus Notes.

Solution

First of all, all of the work disks need to be the same size. This is because the assignment of work disks to users is not controlled by the administrator. They are assigned on a first-come first-served basis.

The work disks need to be at least as large as the size of the largest migration data file being transferred to the Domino Migration Engine. You can take one of three approaches:

1. If you don't have any worries about disk space on your system, just assign 50 or 100 cylinders to your work disks.
2. Suppose most of your users only have 5 cylinder A-disks. This is common. Then size the work disks at 10 cylinders to cover the 5 they've got plus some calendar, document and reader file data. This may cover 95% of your users. For the remaining 5%, run their migration later using larger work disks (but maybe fewer of them).
3. If you are not comfortable with approaches 1 or 2, then you'll need to find out the size of the largest migration data file.

So, how big is that?

A migration data file consists of all of the data you are going to migrate plus some control information in the file. The control information will be such a small fraction when we are talking about large migration data files that we can ignore it.

So we need to know how much data we are going to migrate. I'd start with having the VM system programmer write a small exec to scan the VM directory and find out the sizes of all of the OV/VM user's A-disks. Except for documents in the inbasket, calendar data, reader files and inbaskets which are delegated, you now have an upper size limit on the amount of data to be migrated.

With proper link authority the VM system programmer could find out how much data on the A-disk was going to be migrated. I'd count up the total number of bytes in:

Notelogs and inbaskets (filetype=OFSLOGfl)
Nickname files (filetype=OFSMCNTL)
Distribution lists (filetype=OFSMLIST)
CMS notebooks and the CMS names file

With proper authority to look at the list of spool files, the VM system programmer could code up an exec to look at each OV/VM's reader queue and figure out the number of bytes of data based on the records in each file. Complex files might need to be transferred for closer examination or the spool file blocks might need to be interrogated to get the number of spool blocks.

For calendar data, the VM system programmer would link the calendar manager's disks to find out how much calendar data has been stored by the user. Each file on the disks is a user's calendar data for a month. For the ESA Calendar Feature, the files are on an SFS disk and are of a different format and organization.

If the user's A-disk doesn't contain an inbasket file (OFSMAIL OFSLOGfl), it could be on the mailbox manager's 5FF disk. You'll have to link to this disk and figure out the size of the inbasket file. I'd assume it was twice the size of the encrypted version.

Documents in the inbasket are a rather tough nut to crack. If you don't use documents, don't worry about them. Otherwise, you may have to read the inbasket and get the cron number and see if you can retrieve the document and get ist size. This is a lot of work and will take a lot of processing time.

Of course, in adding up the sizes of all of the OV/VM data you find you'll want to skip over the data you don't plan to migrate. This you'll find listed in the LNMMIGR CONFIG file. Additionally, this file lists the parameters which define HOW MUCH data to collect. For instance, the file will tell the migration tool not to migrate notes from the inbasket which are older than 6 months.

Recommendation

Take approach 2 above in which you survey the VM directory to find out the A-disk size of each OV/VM user. Build a distribution chart by lumping together sizes in 5 cylinder ranges. Determine the A-disk size which represents say, 80% of all users. Double this and make this to be the size for your work disks. This will take care of probably 90-95% of
the people.

How many work disks

Question

How many work disks should be assigned to the VM Migration Server?

Answer

The VM Migration Server requires one or more work disks to be defined and the problem is "How many work disks should there be?"

Background

Work disks are owned by the VM Migration Server and are assigned to each user when a migration request is processed for that user. This work disk is used to hold all of the OV/VM data to be migrated to Lotus Notes.

Discussion

The number of work disks directly determines the number of users who are autologged by the VM Migration Server at one time. If you want more autologged at the same time, add more work disks.

As each user is autologged, they run an OV/VM data collection program called LNMSTART EXEC. This exec will open the mail, access documents, collect calendar data and process notelogs. You would expect then, a performance hit to the OV/VM service machines involved: mail box manager, database manager and calendar manager. Each user being migrated will take substantial CPU and I/O resources.

As each user's migration process completes, the migration data file (MDF) is transferred to the Domino Migration Engine via file transfer. Note that with only one VM Migration Server you can transfer only one file to one Domino Migration Engine. This single-threaded processing obviously limits throughput.

Recommendation

Having too many work disks will impact your system performance. This will also cause OV/VM's service machines to be impacted to the extent of providing poorer service to other
users. Increasing the number of work disks does not necessarily speed up the number of users migrated since file transfer is single-threaded.

Only you know these factors:

The system load at the time migrations are performed
Speed of file transfers
Speed of processing migration data files

Your goal in determining the number of work disks is to keep the Domino Migration Engine busy processing migration data files. Start with a number such as five and determine via observation if the Domino Migration Engine is idle or not.

Increasing the rate of user migrations

Question
How do you increase the rate of user migrations?

Answer
The rate of user migrations is highly dependent on many factors.

Discussion
If you have decided to increase the rate of user migrations then you probably already have these in place:

1. A robust Lotus Notes infrastructure which can easily handle the increased load
2. A user training plan which might address education on Lotus Notes in a "just-in-time" manner
3. A support structure able to handle system outages, server failures and mail monitoring
4. A help desk adequately staffed with Notes and OV/VM knowledgeable people
5. The OV/VM Migration Tool in place and operational with a fair number of test runs, at least for timing purposes.

You are probably aware that migrating more users on the VM system will impact its performance and have taken steps to monitor this resource.

The rate of user migrations depends on:

The speed of the VM processor being used
The load on the VM processor (which might change over a 24 hour period)
The amount of OV/VM data each user has to migrate
The number of VM Migration Servers in use
The number of work disks assigned to each VM Migration Server
The speed of file transfer to the Domino Migration Engine
The number of Domino Migration Engines in use
The speed, size and capacity of the PC running the Domino Migration Engine
The number of migration tasks started on the Domino Migration Engine
Whether the Domino Migration Engine is running on a dedicated PC or not
The network speed which will gate the transfer of MAIL.NSF files from the Domino Migration Engine to your mail servers.

Some items in the preceding list you can't change or will find it difficult to change.

Recommendations

1. Perform the migrations when it is least busy on your VM system
2. Recommend to your users that they delete unwanted mail from their in-basket and notelogs
3. Carefully review the migration configuration parameters so that you migrate only the data you wish to migrate
4. Allocate a proper number of work disks to the VM Migration Server. More work disks mean more user migrations can proceed at once.
5. Establish as many Domino Migration Engines as possible
6. Establish as many VM Migration Servers as there are Domino Migration Engines.
7. Run the Domino Migration Engine on a dedicated Domino server

8. Insure that file transfer is as fast as possible
9. Configure each Domino Server with the fastest processor available and with as much memory as possible
10. Start each Domino Migration Engine with 3 migration tasks (NMSMIG)

How to set the SHORTNAME field to a value

Question

The shortname field is not being downloaded from my phone directory to the Notes NAB. How do I get it set?

Description

In many cases, a lookup in the Public NAB could be done on the SHORTNAME field which could contain the userid from VM.
This is useful for installations where the userid is easy to remember. If the userid is, say, a serial number and hard to remember, addressing on VM is very often done using a nickname. Knowing this, many installations will register users to Notes and specify this nickname in the SHORTNAME field.

Software available starting with the September 1997 refresh of the Domino Migration Engine allows for the Migration NAB matching algorithms to match on SHORTNAME instead of the FULLNAME field. If the host directory does not contain a SHORTNAME, Notes will generate one based on the first and last names.

Recommendations

In order to pass down the SHORTNAME to Notes, you will have to generate the shortname field in the host directory. Each record must contain this field of course, but from then on the directory migration program can handle it. Since there are different directory types supported, there are slightly different approaches:

You have a CALLUP/VM or PHONE (OV/VM) directory:

If shortname is a field already in the directory, then

1. set COPYOPT to YES in LNMMIGR CONFIG
2. set opt1 = ShortName xxxxxx (where xxxxxx is the name of the field in CallUp/VM or PHONE)
3. set TOTALOPT = 1 in LNMMIGR DEFAULTS
4. run LNMMDIR

If shortname is NOT already a field in the CallUp/VM or PHONE (OV/VM) directory, then

1. extract all of your data from CallUp/VM or PHONE and write it to a flat file
2. in the flat file for each record, include the shortname field
3. follow the directions for an installation defined flat file below

You have an installation defined flat file as a directory:

You are assumed to have a shortname field in this directory. If it's not there, write a program to read the flat file, add in the shortname field and then write a NEW flat file.

1. create file LNMMFLAT TABLE to describe the format of the flat file. Include the field definition of the shortname field
2. run LNMMDIR

How to migrate selected A-disk files

Question

There are some particular A-disk files I wish to migrate. How would I do this?

Answer

You have a situation where you need to migrate certain PC based files. They reside on the user's A-disk. An example would be WK3 files from Lotus 1-2-3 Mainframe. You wish to migrate only these reader files and no others.

Recommendations

The technique is to use exits LNMEX00 and LNMEX99. LNMEX00 is invoked at the start of the migration. Code it in REXX to do the following:

1. Change all existing reader files to a HOLD status (CP CHANGE READER ALL HOLD) If you wish to migrate all reader files, skip this step.
2. Use PIPE or LISTFILE to list all files with a filetype of WK3. Use the SENDFILE command to send the files to the reader.

LNMEX99 is invoked at the end of the migration. Code it in REXX to do the following:

1. Purge all of the WK3 files. Use PIPE or EXECIO to list all spool files and selectively delete all WK3 files.
2. Change all existing reader files to a NOHOLD status (CP CHANGE READER ALL NOHOLD)

With these exits in place the migration tool will migrate the WK3 files assuming you have

MIGRATE_READER = YES in LNMMIGR CONFIG.

You could also arrange to have your end users list all files they want migrated in a separate file. This is the technique used to migrate only a subset of notelogs or notebooks. Have them place the filenames and filetypes in a file say, $$LNM$$ PCFILES. Your exit LNMEX00 would then just read this file and send the listed files to the reader.

Are there some sample MDF and NSF sizes?

Question

Do you have any information on the relative sizes between migration data files and Notes NSF files?

Answer

The MDF size is directly related to to the amount of OV/VM data migrated. The resulting mail file is much larger because of Notes overhead.

Case Studies

Case 1:

Calendar data:	30 days worth (4 entries per day)
Reader files:	4
Documents:	4
Notes:	40

The MDF was .2M and the NSF was 3.01M

Case 2:

Calendar data:	180 days worth (4 entries per day)
Reader files:	86
Documents:	80
Notes:	88

The MDF was 1.81M and the NSF was 4.78M

Case 3:

Calendar data:	0
Reader files:	0
Documents:	0
Notes:	400 (totalling 9200 records)

The MDF was .8M and the NSF was 2.97M.

Case 4:

Calendar data:	180 days worth (4 entries per day)
Reader files:	0
Documents:	0
Notes:	400 (totalling 9200 records)

The MDF was 1.3M and the NSF was 3.56M.

Appendix A. Files containing the samples

Document Migration Method 1

LNMOVDOC EXEC
LNMMUSER DOCUPDAT
LNMMIGR UPDATE1

Document Migration Method 2

LNMOVDOC EXEC
LNMMUSER DOCUPDAT
LNMMIGR UPDATE2
LNMEX99 EXEC
CRONVM   EXEC

Document Migration Method 3

LNMDBEXT EXEC
LNMUSEXT EXEC
LNMUSCMP EXEC
LNMDOLST EXEC
LNMDRUN EXEC
LNMSTDOC EXEC
LNMMDOC EXEC
LNMDOCSV EXEC
LNMLOG UPDATE
LNMHEADR UPDATE
LNMMIGR UPDATE3
PUFF MODULE
CUA2001

Customization of the Migration for each user

LNMUOPTS EXEC
LNMSELCT PANLFILE
LNMSELCT EXEC
CUA2001

Hardcopy Document Support

LNMHDC EXEC
LNMPDOC UPDATE

Migrating Document Meeting Notices

LNMEX02 EXEC

Migrating CMS Notebooks as Notelogs

LNMEX01 EXEC

Reminder File Support

LNMRMNDR EXEC
LNMRMNDR TABLE
LNMMUSER REMINDER

HOLIDAY EPSFILE Support

LNMHOLID EXEC
LNMMUSER HOLIDAY

Time Zone Support

LNMTMZN EXEC
RXCDAYS MODULE
LNMCLNDR UPDATE
LNMMIGR TIMEZONE

PROOF ADDENDA Dictionary Support

LNMADDEN EXEC
LNMMUSER ADDEN

Notelog Process Performance Support

LNMPNLOG EXEC

Migrating VM Databases Support

LNMIGRAT EXEC
MIGRCMS EXEC
LNMHEADR UPDATE2
CUA2001

Migrating OV/VM Conference Rooms Support

LNMRSRCE EXEC
LNMRSRC0 TABLE
LNMRSRC1 TABLE
LNMRSRC2 TABLE
LNMRSRC3 TABLE
MIGRCONF EXEC
LNMHEADR UPDATE2
CUA2001

Migrating the OV/VM Bulletin Board Support

LNMIGRAT EXEC

BULLETIN LNMIGRAT
BULLETIN READFILE
MIGRCMS EXEC
LNMHEADR UPDATE2
CUA2001

Todo File Support

LNMTODO EXEC
LNMTODO TABLE
LNMMUSER TODO

Maintaining ‘Read/Unread’ Status

URMARK CODE

CUA 2001 Package

RXCTCHAR DEFSFILE
RXCWHELP DEFSFILE
RXCWIND DEFSFILE
RXCWINDE DEFSFILE
RXCWINDU DEFSFILE
RXCWL   DEFSFILE
RXCWUSER DEFSFILE
RXCLABEL EXEC
RXCSETUP EXEC
RXCTCHAR EXEC
RXCW   EXEC
RXCWDEF EXEC
RXCWG   EXEC
RXCWHELP EXEC
RXCWIND EXEC
RXCWINDH EXEC
RXCWL   EXEC
RXCWREG EXEC
RXCWTSET EXEC
RXCWUSER EXEC
RXCWUSER HELPFILE
RXCDAYS MODULE
RXCSCAN MODULE
RXCWIND PROGFILE
RXCWIND RXCTOPTS
RXCWIND SYSDEF
RXCUME   TXTAMENG
RXCDCCTL XEDIT
RXCLABEL XEDIT
RXCPROF XEDIT
RXCWCCTL XEDIT
RXCWFMT XEDIT

RXCWIND XEDIT
RXCWINDA XEDIT
RXCWINDC XEDIT
RXCWINDE XEDIT
RXCWINDM XEDIT
RXCWINDP XEDIT
RXCWINDR XEDIT
RXCWINDT XEDIT
RXCWINDU XEDIT
RXCWREF XEDIT