

Migration Specialties International, Inc.

217 West 2nd Street, Florence, CO 81226-1403

719-371-1711, Fax: 888-854-3417

E-mail: Info@MigrationSpecialties.com

www.MigrationSpecialties.com



Bruce Claremont, December 2005

ODS-2/ISO-9660 Dual Format CD Creation

Alpha OpenVMS V8.2

I needed to generate a Migration RPG distribution kit and decided it was time to figure out how to create a dual format distribution CD. Such a CD, formatted as ODS-2 and ISO-9660, would allow the disc to be read by OpenVMS systems and other, less worthy platforms.

Perusal of the Internet led to enough information to get the task done. I've compiled a list of useful Internet links at the end of this document. A special note of thanks goes to Brian Schenkenberger and the folks at TECSys Development, Inc. for providing a couple of web pages that served as my primary resources to accomplish this task. I also appreciate Verne Britton's efforts in proofing this article and ensuring its accuracy.

What follows is a detailed description of how I created the dual format Migration RPG distribution CD.

PROCESS SUMMARY

A dual format ODS-2/ISO-9660 CD can be generated using four basic steps:

1. Create a 650MB container file on an OpenVMS system.
2. Map an ISO-9660 file system to the container file.
3. Add the ODS-2 files that are to appear on the CD to the container file.
4. Burn an image of the container file to a CD.

TOOLS USED

- Alpha 1000 System
- OpenVMS V8.2
- LD Driver kit: The Logical Disk Driver has been integrated into OpenVMS since version 7.3-1, but is not enabled by default (guess how I know this). Be sure to place the following call in your SYSTARTUP_VMS.COM procedure:

```
$ @SYS$STARTUP:LD$STARTUP.COM
```

If you want to run the Logical Disk Driver under OpenVMS versions prior to 7.3-1, acquire and install the appropriate kit from an OpenVMS Freeware distribution.

- FILE Utility: An RMS file attributes management tool developed by Joe Meadows. (<http://vms.process.com/scripts/fileserv/fileserv.com?FILE>)

- MKISOFS: Pre-mastering program used to generate an ISO 9960 file system developed by Eric Youngdale.
(<http://www.tmesis.com/CDrom/MKISOFS.ZIP>)
- ODSISO: ODS-2/ISO-9660 co-map tool developed by TECSys Development, Inc.
(<http://support.tditx.com/~odsiso/download.html>)

CREATING A DUAL FORMAT CONTAINER FILE

1. I acquired and installed the tools listed in the Tools Used section. In my case, I installed the FILE, MKISOFS, and ODSISO kits under SYS\$SYSDEVICE and set up the following commands to reference them:

```
$ SET COMMAND SYS$SYSDEVICE: [FILE] FILE.CLD
$ MKISOFS := $SYS$SYSDEVICE: [MKISOFS] MKISOFS.EXE
$ ODSISO := @SYS$SYSDEVICE: [ODSISO] ODSISO_FIX1.COM
```

In configuring the ODSISO_FIX1.COM procedure, these lines:

```
$ odstools = f$elem(0, "]", f$environ("procedure") - "]" ["] + "]"
$ isotools = odstools
$ fileclld = "site$specific:file.cld"
```

were changed to this:

```
$ odstools = "SYS$SYSDEVICE: [ODSISO] "
$ isotools = "SYS$SYSDEVICE: [MKISOFS] "
$ fileclld = "SYS$SYSDEVICE: [FILE] FILE.CLD"
```

2. The ODSISO utility requires ISOROOT and SCRATCH directories. I created a root directory on a disk with lots of free space, then created these directories beneath it.

```
$ CREATE /DIR DKA100: [MRPG_CD]
$ CREATE /DIR DKA100: [MRPG_CD.ISOROOT]
$ CREATE /DIR DKA100: [MRPG_CD.SCRATCH]
```

3. All of the files that are to appear under the ISO file system need to be copied to the ISOROOT directory. This includes files that are to be shared between the ISO and ODS-2 file systems. In my case, I planned to share the PDF documentation files between the two file systems.

Make sure everything that is to appear under the ISO format is placed in the ISOROOT directory. Once the contents of ISOROOT have been mapped to the ODS-2 container file, they cannot be modified. If you need to make a change to the ISO contents, you'll need to rebuild the container file.

I copied the following files to the ISOROOT directory:

```
MIGRATION_RPG_LRM.PDF          MIGRATION_RPG_LRM.PS
MIGRATION_RPG_SFG.PDF          MIGRATION_RPG_SFG.PS
MIGRATION_RPG_USER_GUIDE.PDF   MIGRATION_RPG_USER_GUIDE.PS
MIGRPG082.ZIP                  README.TXT
```

File Restrictions

The following restrictions apply to file names and file formats when creating an ODS/ISO CD.

- The total length of file names cannot exceed 32 characters. This includes both the name and the extension. The VMS version number does not count towards this limit.
- Per ISO requirements, all files in the ISOROOT directory must be formatted as fixed-length, 512-byte records or Stream-LF. I handled this requirement as follows:
 - The zip archive, created with INFO-ZIP, was an ISO compliant fixed-length, 512-byte format by default. The archive contained a copy of the Migration RPG installation kit in OpenVMS BACKUP save set format (MIGRPG082.A).
 - The postscript (.PS) files were created under OpenVMS using VAX Document and were originally variable-length sequential files. A binary FTP transfer was used to place them on a Windows XP PC for conversion to PDF files. On the PC they were verified to work with the *GSview* postscript viewer, then binary transferred back to the ISOROOT directory. This process converted them to an ISO compliant fixed-length, 512-byte format.
 - The PDF files were generated from the VAX Document postscript files that had been transferred to the PC. They were converted from postscript to PDF format using *GSview*. The PDF files were binary transferred to the OpenVMS system for inclusion in the ISOROOT directory. This process produced ISO compliant fixed-length, 512-byte format PDF files.
 - The README.TXT file was created using the TPU editor on an OpenVMS system. The ODSISO kit includes a handy CVT_TO_STMLF.COM procedure, which was used to convert this file to Stream-LF format. Carriage return/line feed (CRLF; Hex 0D 0A) characters were hard coded at the end of each line in the text file before it was converted. This ensured that after it had been converted to Stream-LF format, it would display properly when TYPE'd or viewed with a Windows utility like Notepad.
 - The .GUIDE and .RELEASE_NOTES files were variable-length, sequential text files generated by RUNOFF. They were converted to Stream-LF format using the ODSISO CVT_TO_STMLF.COM procedure. RUNOFF inserts the embedded carriage return/line feed's necessary to ensure the files display properly after being converted.

Note: All transfers between the OpenVMS system and the PC were done using CKERMIT or FTP in binary mode.

4. Next, I created a text file containing publishing information that's incorporated into the ISO file system image. This information is used to populate the .mkisofsrc file used by the MKISOFS utility. I used TPU to create a text file in the DKA100:[MRPG_CD] root directory. The file was labeled ISO_RC.TXT and populated with the following lines:

```
APPI=mkisofs v1.05. Alpha OpenVMS 8.2
PUBL=Migration Specialties International, Inc., www.MigrationSpecialties.com
PREP=Migration Specialties International, Inc., www.MigrationSpecialties.com
COPY=Migration Specialties International, Inc., www.MigrationSpecialties.com
VOLI=MIGRPG082
```

5. Next up was creation of a file that would serve as a disk container. I used SYSGEN to create the file. 1,331,976 blocks equates to a 74 minute, 650MB CD. The .ISO file extension was chosen for compatibility with *Roxio* CD burning software.

```
$ MCR SYSGEN CREATE DKA100:[MRPG_CD]MRPG_CD.ISO /SIZE=1331976 /CONTIG
```

Note: Do not connect the container file at this time; the ODSISO utility will do this for you.

6. Now I was ready to map an ISO file system onto my container file. The ODSISO utility takes care of this:

```
$ ODSISO -
LDA21 - !P1: LD device
DKA100:[MRPG_CD]MRPG_CD.ISO - !P2: Container file
MIGRPG082 - !P3: VMS volume label
DKA100:[MRPG_CD.ISOROOT] - !P4: ISOROOT directory
DKA100:[MRPG_CD.SCRATCH] - !P5: Scratch directory
DKA100:[MRPG_CD]ISO_RC.TXT !P6: .mkisosrc file
```

- P1** LD device to which the container file is mapped.
- P2** Container file that will be used to create the CD image file.
- P3** VMS volume label for container file.
- P4** ISOROOT directory required by the MKISOFS Utility.
(*Note: The ISOROOT parameter must not contain any logical names!*)
- P5** The scratch directory used to hold the ISO image and temporary files
- P6** ISO publishing information file used as the .mkisofsrc file by the MKISOFS utility when generating the ISO image.

The ODSISO utility generates a container file initialized as an ODS-2 volume with an ISO file system image co-mapped onto it. Once ODSISO completes, the ISO files were visible under ODS-2 in the directory LDA21:[ISOROOT]. No further changes are allowed to the ISO files.

7. The next step was to add the files that were to appear under the ODS-2 file system. I started by mounting the disk container file:

```
$ MOUNT LDA21: MIGRPG082
```

8. I created an ODS-2 directory in the container file and populated it with files that will be visible only to OpenVMS:

```
$ CREATE /DIR LDA21:[MIGRPG082]
$ COPY MRPG:MIGRPG082.A LDA21:[MIGRPG082]
$ COPY MRPGDOC$:* .PS LDA21:[MIGRPG082]
```

9. This step and the next one were interesting. I used RENAME to "move" the PDF files I wanted to share between the ISO and ODS-2 formats from the ISOROOT directory on the container file to the ODS-2 directory. RENAME doesn't really move the files; it simply changes the ODS-2 file pointers. Files configured in this manner will be common between

the ODS-2 and ISO distribution, meaning they will be visible regardless of how the CD is mounted.

```
$ RENAME LDA21:[ISOROOT]*.PDF LDA21:[MIGRPG082]
```

10. I then used DELETE to "remove" the remaining files from the ISOROOT directory and the ISOROOT directory itself. This sounds counter-intuitive, but it works. Deleting the files removes their entries from the ODS-2 file index, but does not physically remove the files. The deleted ISO files will no longer be visible when the CD is mounted as an ODS-2 volume, but will be visible when it is mounted as an ISO volume.

```
$ DELETE LDA21:[ISOROOT]*.*;*
$ DELETE LDA21:[000000]ISOROOT.DIR;*
```

11. At this point, all modifications to the container file are finished and I dismounted the container file.

```
$ DISMOUNT LDA21:
```

12. I mounted the container file as a write protected ODS-2 volume and checked its contents:

```
$ MOUNT /NOWRITE LDA21: MIGRPG082
$ DIR LDA21:[*...]

Directory LDA21:[MIGRPG082]

MIGRATION_RPG_LRM.PDF;1          MIGRATION_RPG_LRM.PS;22
MIGRATION_RPG_SFG.PDF;1          MIGRATION_RPG_SFG.PS;25
MIGRATION_RPG_USER_GUIDE.PDF;1    MIGRATION_RPG_USER_GUIDE.PS;23
MIGRPG$082.RELEASE_NOTES;13      MIGRPG082.A;13
MIGRPG_INSTALLATION.GUIDE;107

Total of 9 files.
$ DISMOUNT LDA21:
```

13. I mounted the container file as an ISO volume and checked its contents:

```
$ MOUNT /NOWRITE /MEDIA_FORMAT=CDROM LDA21: MIGRPG082
$ DIR LDA21:[000000]

Directory LDA21:[000000]

MIGRATION_RPG_LRM.PDF;1          MIGRATION_RPG_LRM.PS;1
MIGRATION_RPG_SFG.PDF;1          MIGRATION_RPG_SFG.PS;1
MIGRATION_RPG_USER_GUIDE.PDF;1    MIGRATION_RPG_USER_GUIDE.PS;1
MIGRPG$082.RELEASE_NOTES;1        MIGRPG082.ZIP;1
MIGRPG_INSTALLATION.GUIDE;1        README.TXT;1

Total of 10 files.

$ DISMOUNT LDA21:
```

Note: The ISO volume label is case sensitive. If lowercase letters are used in the volume label and you are running on an ODS-2 disk, you will need to enclose the volume label in double quotes to have it processed correctly. I.E:

```
$ MOUNT /NOWRITE /MEDIA_FORMAT=CDROM LDA21: "MigRPG082"
```

Having verified the contents of my dual format container file, I disconnected the LD device.

```
$ LD DISCONNECT LDA21:
```

CREATING THE CD

I transferred the container file to a PC using binary mode FTP. Once on the PC, The container file serves as a CD image file, which can be burned to a CD using a raw or image format. In my case, the following steps work for version 6 of *Roxio Creator Classic*:

1. Launch *Creator Classic*.
2. Select the *File* pull-down and choose *Record Disc from Image...*
3. Select the image file to burn. In my case, this was *MRPG_CD.ISO*.
4. Burn the CD.

Tip: Under this version of *Roxio*, double clicking the *MRPG_CD.ISO* container file from within Windows Explorer launches directly into the CD burning screen.

CONCLUSION

I've used this process to successfully create several dual format CD's. Seeing how I would never remember all of this on my own, I did two additional things:

1. I documented the process in detail (you are reading the results).
2. I wrote a command procedure to further automate most of the process. The procedure is unique to my requirements, but can be used as an example and template. It is attached at the end of this document.

ABOUT THE AUTHOR

Bruce Claremont has a degree in Computer Science and is a certifiable VMS bigot, having worked with OpenVMS since 1983. In addition to OpenVMS skills, Bruce understands legacy programming languages like RPG and COBOL and knows how to code in Macro-32. He also knows a thing or two about software migration. He has worked all sides of the fence, as a customer, software engineer, system manager, support specialist, and project manager. He founded Migration Specialties International, Inc. in 1992 and, when not fantasizing about selling the business for millions and retiring to a life of leisure, continues to deliver OpenVMS and software migration related services. More information about Mr. Claremont and Migration Specialties can be found at www.MigrationSpecialties.com.

RESOURCE LIST

<http://support.tditx.com/~odsiso/index.html>

TECSys Development, Inc. support web page. Source of ODSISO and MKISOFS tools and instructions on ODSISO usage.

<http://www.tmesis.com/CDrom/>

Brian Schenkenberger's instructions and explanation of dual format CD creation.

http://h71000.www7.hp.com/openvms/journal/v6/disk_partitioning_with_lddriver.pdf

Excellent article on the LDdriver by Jur van der Burg.

<http://vms.process.com/fileserv-software.html>

Hunter Goatley's VMS Freeware File Server site, where I acquired a copy of Joe Meadows FILE Utility.

<http://www.stanq.com/vms-cd.html>

Stanley Quayle's page on writing OpenVMS compatible CD's

<http://www.djesys.com/vms/cdrom.html#dualfmt>

DJE Systems OpenVMS CD-ROM page.

http://h71000.www7.hp.com/faq/vmsfaq_007.html

Information on mounting an ISO-9660 CD on OpenVMS.

<http://h71000.www7.hp.com/openvms/freeware/index.html>

HP's OpenVMS Freeware page.

<http://www.linuxdevcenter.com/linux/cmd/m/mkisofs.html>

<http://cdrecord.berlios.de/old/private/man/mkisofs-1.12.html>

Sites containing MKISOFS Utility documentation.

MIGRATION RPG DUAL FORMAT CD BUILD PROCEDURE

```

$ IF F$MODE() .EQS. "BATCH" THEN SET VERIFY
$!-----
$! BLD_CD.COM
$!-----
$!   Build container file for dual format Migration RPG CD.
$!
$!   Created:   MSI, DEC-2005
$!
$   @SYSSUTILITIES:SET_ATTRIBUTES.COM "' 'F$ENVIRONMENT("PROCEDURE")' '
$   STATUS = 1
$   ON ERROR THEN GOTO ERROR_TRAP           !Standard abort trapping.
$   ON CONTROL_Y THEN GOTO ERROR_TRAP
$   IF _BATCH THEN SAY = "!"
$!
$   SAY ""
$   SAY "Migration RPG ODS-2/ISO-9660 Dual Format CD Build Procedure"
$   SAY ""
$   SAY "DKA100:[MRPG_CD.ISOROOT] Contents:"

```

```

$ DIR DKA100:[MRPG_CD.ISOROOT]
$ SAY ""
$ SAY "Have the appropriate files been placed in [.ISOROOT] and has"
$ SAY "their format been set to Stream-LF or Fixed-length, 512-byte?"
$ ASK "Y/N: " QST
$ IF .NOT. QST THEN GOTO END_PROCEDURE
$ PURGE DKA100:[MRPG_CD.ISOROOT]
$!
$ SAY ""
$ ASK "Kit Version (ex: 082): " VER
$ VER = F$EDIT(VER, "COLLAPSE, UPCASE")
$ IF F$SEARCH("DKA100:[MRPG_CD.ISOROOT]MIGRPG'VE'.ZIP") .EQS. ""
$ THEN
$     SAY _BELL, _BELL
$     SAY "ERROR: Bad version number or [.ISOROOT] not properly populated."
$     SAY "MIGRPG'VE'.ZIP not found."
$     GOTO ERROR_TRAP
$ ENDIF
$!
$! Create publishing information for ISO image.
$!
$ OPEN /WRITE ISO DKA100:[MRPG_CD]ISO_RC.TXT
$ WRITE ISO -
"APPI=mkisofs 'F$GETSYI("ARCH_NAME")' OpenVMS 'F$GETSYI("VERSION")' 'F$TIME()"
$ WRITE ISO -
"PUBL=Migration Specialties International, Inc., www.MigrationSpecialties.com
$ WRITE ISO -
"PREP=Migration Specialties International, Inc., www.MigrationSpecialties.com
$ WRITE ISO -
"COPY=Migration Specialties International, Inc., www.MigrationSpecialties.com
$ WRITE ISO "VOLI=MIGRPG'VE'"
$ CLOSE ISO
$ PURGE DKA100:[MRPG_CD]ISO_RC.TXT
$!
$ IF F$SEARCH("DKA100:[MRPG_CD]MIGRPG*.ISO") .NES. "" THEN -
DELETE DKA100:[MRPG_CD]MIGRPG*.ISO;*
$!
$ SAY ""
$ SAY "Creating container file: ", F$TIME()
$ MCR SYSGEN CREATE DKA100:[MRPG_CD]MIGRPG'VE'.ISO /SIZE=1331976 /CONTIG
$!
$ SAY ""
$ SAY "Creating ISO Overlay: ", F$TIME()
$ ODSISO -
LDA21 - !P1: LD Device
DKA100:[MRPG_CD]MIGRPG'VE'.ISO - !P2: LD container file
MIGRPG'VE' - !P3: VMS volume label
DKA100:[MRPG_CD.ISOROOT] - !P4: ISOROOT directory (NO LOGICALS ALLOWED)
DKA100:[MRPG_CD.SCRATCH] - !P5: Scratch directory
DKA100:[MRPG_CD]ISO_RC.TXT !P6: .mkisofsrc for ISO image
$!
$ SAY ""
$ SAY "Moving kit files to ODS-2 area: ", F$TIME()
$ MOUNT LDA21: MIGRPG'VE'
$ CREATE /DIR LDA21:[MIGRPG'VE']
$!
$ COPY BRUCE$:MIGRPG'VE'.A LDA21:[MIGRPG'VE']
$ COPY MRPG:MIGRPG$'VE'.RELEASE_NOTES LDA21:[MIGRPG'VE']
$ COPY MRPG:MIGRPG_INSTALLATION.GUIDE LDA21:[MIGRPG'VE']
$ COPY MRPGDOC$LRM:MIGRATION_RPG_LRM.PS LDA21:[MIGRPG'VE']
$ COPY MRPGDOC$SFG:MIGRATION_RPG_SFG.PS LDA21:[MIGRPG'VE']
$ COPY MRPGDOC$USER:MIGRATION_RPG_USER_GUIDE.PS LDA21:[MIGRPG'VE']
$!

```

```

$      SAY ""
$      SAY "Configuring ISO and ODS-2 files: ", F$TIME()
$!
$!      Set pointers for PDF files from ISO to ODS file system.
$!
$      RENAME LDA21:[ISOROOT]*.PDF LDA21:[MIGRPG'VER']
$!
$!      Remove ISO files from ODS-2 view.
$!
$      DELETE LDA21:[ISOROOT]*.*;*
$      DELETE LDA21:[000000]ISOROOT.DIR;*
$!
$!      Set ODS-2 file ownership and protections.
$!
$      SET FILE /OWNER=SYSTEM /PROT=(S:RE, O:RE, G:RE, W:RE) -
          LDA21:[000000]MIGRPG'VER'.DIR, -
          LDA21:[MIGRPG'VER']*.*
$!
$      DISMOUNT LDA21:
$!
$      SAY ""
$      SAY "Confirming ODS-2 volume contents: ", F$TIME()
$      MOUNT /NOWRITE LDA21: MIGRPG'VER'
$      DIR LDA21:[*...]
$      DISMOUNT LDA21:
$!
$      SAY ""
$      SAY "Confirming ISO volume contents: ", F$TIME()
$      MOUNT /NOWRITE /MEDIA=CDROM LDA21: MIGRPG'VER'
$      DIR LDA21:[000000]
$      DISMOUNT LDA21:
$      LD DISCONNECT LDA21:
$!
$      SAY _BELL
$      SAY "All done; MIGRPG'VER'.ISO container file is ready for transfer and burn."
$!
$ END_PROCEDURE:
$     IF F$TRNLNM("ISO") .NES. "" THEN CLOSE ISO
$     IF _BATCH THEN @SYS$UTILITIES:CLEAR_LDA_EXT.COM
$     EXIT 'STATUS'
$!
$ ERROR_TRAP:
$     IF .NOT. _BATCH
$     THEN
$         SAY BELL, "Procedure: ", F$ENVIRONMENT("PROCEDURE")
$         ASK "Procedure aborted by a <Ctrl^Y> or error. Enter 0 to exit: " QST
$         IF QST .NES. "0" THEN GOTO ERROR_TRAP
$     ENDIF
$!
$ CANCEL_PROCEDURE:
$     IF _BATCH .AND. F$ENVIRONMENT("DEPTH") .LE. 0 !procedure (depth=0) and
$     THEN !it is a batch procedure,
$         STATUS = 2 !have it exit with an
$     ELSE !error status (2) if it
$         STATUS = 3 !has terminated
$     ENDIF !abnormally.
$     GOTO END_PROCEDURE

```