Going Tapeless

Paul Kimpel

2020 UNITE Conference

Session 4025

Thursday, 19 November, 2:00 p.m. GMT-5

Going Tapeless

2020 UNITE Virtual Conference

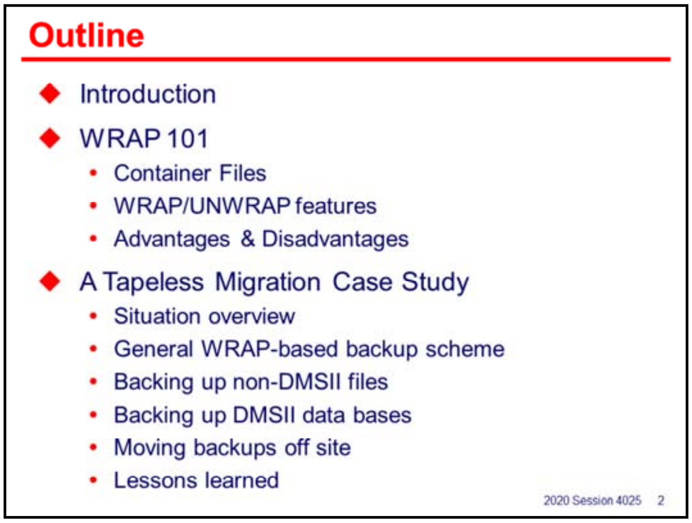
Session MCP 4025

Thursday, 19 November 2020, 2:00 p.m. GMT-5

Paul Kimpel
San Diego, California

http://www.digm.com

e-mail: paul.kimpel@digm.com



Outline

- Introduction
- WRAP 101
  - Container Files
  - WRAP/UNWRAP features
  - Advantages & Disadvantages
- A Tapeless Migration Case Study
  - Situation overview
  - General WRAP-based backup scheme
  - Backing up non-DMSII files
  - Backing up DMSII data bases
  - Moving backups off site
  - Lessons learned

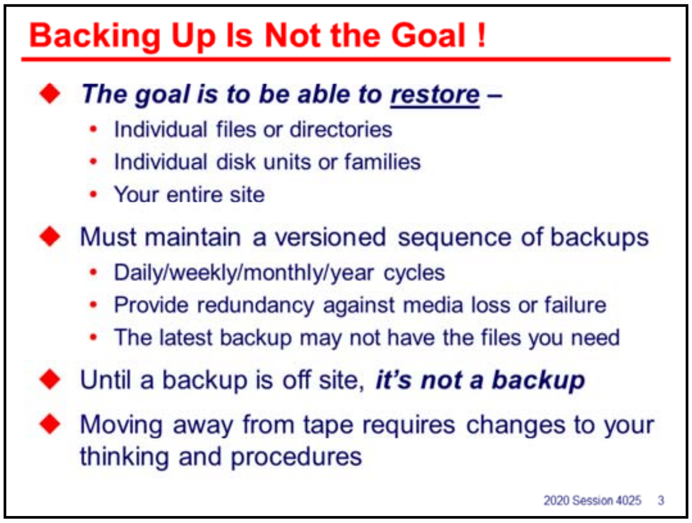
2020 Session 4025 2

Magnetic tape has been the primary backup medium used with computer systems for the past 50 years, and was the primary mass storage medium before that. Tape is still popular, but the decreasing cost of disk storage, increasing speed of wide-area networks, and the rise of Cloud storage as a backup medium are intruding on tape's turf. We are also starting to see challenges in the tape supply chain, with declining industry interest in manufacturing drives and media, as well as in new product development.

In this presentation, I will talk about some alternatives to magnetic tape for backing up ClearPath MCP systems, focusing on a feature of the MCP known as WRAP.

In the first part of the presentation, I will give an overview of the WRAP mechanism, discuss how it relates to backup with tape media, discuss "container files," describe the features of WRAP and its companion UNWRAP, and cover the advantages and disadvantages of WRAP over tape-based backup.

In the second part of the presentation, I will discuss a case study involving one of my customers who recently underwent a migration from LTO tape backup to WRAP. I will give an overview of their general situation and environment, discuss the general WRAP-based backup scheme we developed, and go into some detail on the issues of backing up DMSII and non-DMSII files, as well as moving backups off site. I will finish by covering some of the lessons we learned during this effort.



Backing Up Is Not the Goal !

- ***The goal is to be able to restore –***
  - Individual files or directories
  - Individual disk units or families
  - Your entire site
- Must maintain a versioned sequence of backups
  - Daily/weekly/monthly/year cycles
  - Provide redundancy against media loss or failure
  - The latest backup may not have the files you need
- Until a backup is off site, ***it's not a backup***
- Moving away from tape requires changes to your thinking and procedures

2020 Session 4025 3

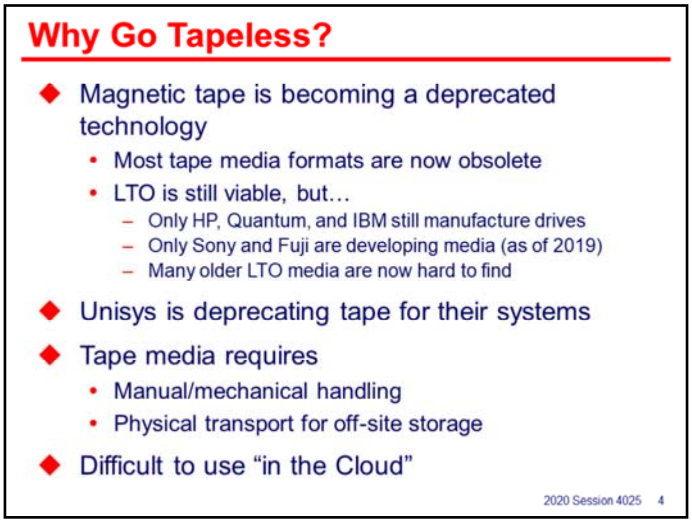
Let’s get something straight from the beginning – ***backing up your system is not the goal!*** In fact, backup, by itself, is totally pointless. The goal is to be able to repair or restore your system, applications, and data in the event something bad happens to it. That something could be as simple as restoring a file or directory that was removed accidentally. It could be repair of files after a software malfunction. You may have suffered physical damage to a disk unit, or perhaps to entire disk families. In the worst case, you could suffer a facility failure that forces you to move your processing to a different site. Thus, backing up files only serves a purpose when the backup can be used to repair or restore your data, application software, and system configuration assets.

Another point about backup is that it’s not sufficient just to make *a* copy of your files and stash them somewhere. You need to maintain a versioned sequence of backups. There are two reasons for this. The first is that backups are just like any other data asset – they can get lost, corrupted, or destroyed. The second is that sometimes a problem does not manifest itself until some time later. Your latest backup may have a perfect copy of corrupted data.

Thus, you need backups to your backups. One useful scheme is to maintain multiple cycles of backups. Make daily backups and keep them for some period of time. Then keep one daily backup per week for a longer period of time, one weekly backup per month for a still longer period of time, and perhaps a yearly backup for some number of years. The longer the time period you need to go back in order to recover your data, the more difficult the recovery will be, but at least you will have something to start with.

A third point is that, until a backup has left the building, it’s not a backup. Facility destruction does occur – fire, flood, etc. – and if your backups are destroyed along with your facility, you may as well not have bothered to make them to begin with.

Backup to magnetic tape has a long history and a rich repository of practices and techniques. If you move away from tape, some of your past practices may still be applicable, but many will not. You are going to need to change both your thinking and your procedures for backup to accommodate whatever new method you choose to use. The important thing to guide you in making that change is that *backup* is not the goal – it’s the ability to be able to *restore*.



Why Go Tapeless?

- Magnetic tape is becoming a deprecated technology
  - Most tape media formats are now obsolete
  - LTO is still viable, but…
    - Only HP, Quantum, and IBM still manufacture drives
    - Only Sony and Fuji are developing media (as of 2019)
    - Many older LTO media are now hard to find
- Unisys is deprecating tape for their systems
- Tape media requires
  - Manual/mechanical handling
  - Physical transport for off-site storage
- Difficult to use “in the Cloud”

2020 Session 4025 4

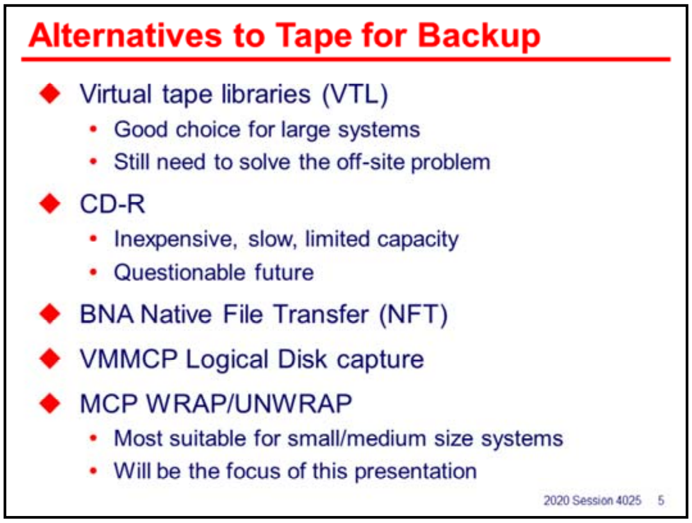
Why should you consider going away from magnetic tape?

The short answer is that before long you may have to. Magnetic tape is becoming a deprecated technology.

- Most tape media formats are now obsolete, and drives for them are either not available or increasingly difficult to maintain. “Round tape” has been obsolete for most of the past 20 years. 3480-class “square tape” is all but gone. 8mm tape is nowhere to be seen, and DAT/DDS has become almost as rare.
- LTO is still viable, but many of the older LTO media formats are becoming difficult to find. Only HP, Quantum, and IBM still manufacture LTO drives, and as of 2019, only Sony and Fuji were still developing media.
- Unisys announced last year that it is deprecating tape for their systems and will stop offering it within the next couple of years.

Another issue with tape media is that it requires either manual or mechanical handling. Drives must be loaded and unloaded, and the media must be physically transported off site for safe storage.

With the rising interest in cloud-based computing, using tape for backup raises serious issues. If your server is “in the Cloud,” it’s going to be really hard to attach a tape drive to it, let alone figure out how to handle the media for it. Cloud-based computing may be the real nail in the coffin for magnetic tape.



Alternatives to Tape for Backup

- Virtual tape libraries (VTL)
  - Good choice for large systems
  - Still need to solve the off-site problem
- CD-R
  - Inexpensive, slow, limited capacity
  - Questionable future
- BNA Native File Transfer (NFT)
- VMMCP Logical Disk capture
- MCP WRAP/UNWRAP
  - Most suitable for small/medium size systems
  - Will be the focus of this presentation

2020 Session 4025 5

So, if magnetic tape may be on its way out, what are the alternatives?

One popular choice, especially for large systems, is a Virtual Tape Library, or VTL. This is essentially a specialized file server that looks to the host like a standard tape drive. Instead of data being written to physical tape volumes, it is stored as files on the VTL, and the VTL software manages the data, allowing you to recall virtual volumes as necessary. Some VTLs offer de-duplication features, which detect when the same data is being backed up again, and maintain only one physical copy of it.

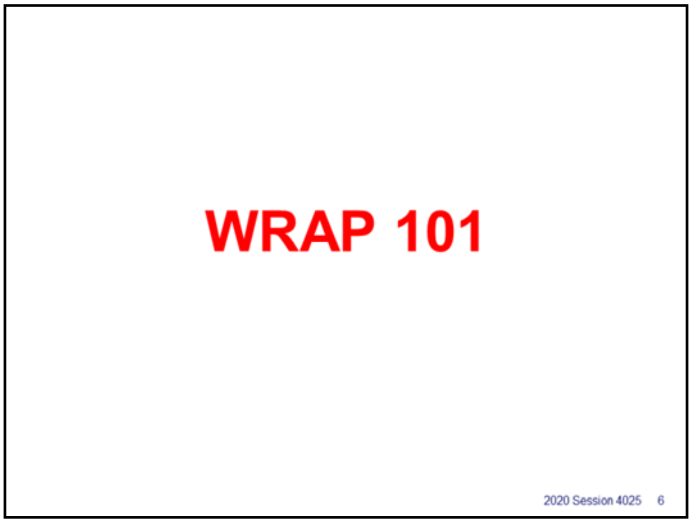
A VTL in a data center still has the problem of getting the backup off site. Some VTLs support physical tape drives for making off-site copies of the data, although this practice is likely to diminish over time, and doesn't work in a Cloud scenario. Many VTLs can also be networked, so that multiple units in geographically separate locations can back each other up. Extend that idea a little and you get to Cloud-based backup.

Another option, especially for small systems, is Compact Disk – CD-R or DVD-R. This is an inexpensive backup solution, but it's slow, and has a limited capacity. CD technology is also becoming deprecated, and is probably even less viable over the next few years than magnetic tape is.

For MCP systems, BNA Native File Transfer can be used much the same way that WFL COPY to tape can, but this is really a file copying mechanism, not a backup mechanism. It's hard to envision how you would do versioned backups using NFT without a lot of administrative overhead. Besides, a separate MCP system makes for a pretty expensive VTL.

For smaller MCP systems that use Logical Disks, which are just large files stored in a Windows file system, it's relatively easy to capture the entire logical disk image and store it somewhere else. Logical Disk images are also fairly compressible. The problem with this idea is that you need to completely shut down the MCP in order to capture these disk images safely, a restriction that simply won't work for most MCP shops.

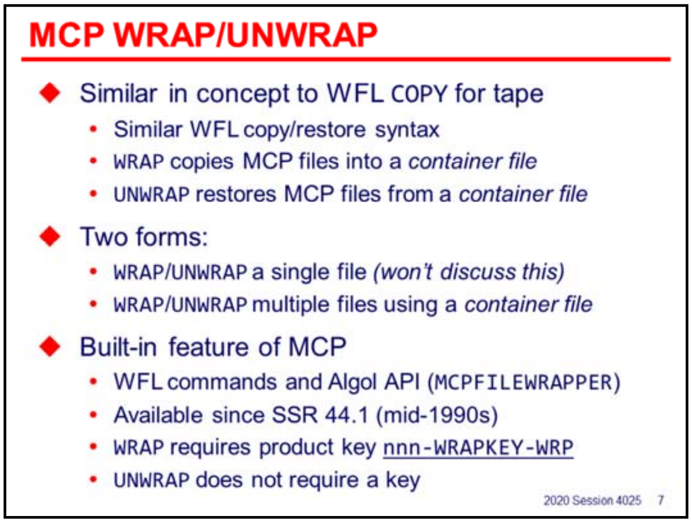
Finally, there is the MCP WRAP/UNWRAP mechanism. It is similar in concept to WFL COPY to tape, except that it copies MCP files to a byte stream instead of a tape volume. It is probably the most suitable mechanism for small-to-medium size systems. This is the approach we will focus on in this presentation.



WRAP 101

2020 Session 4025 6

With that introduction, let us dive into WRAP and see how it works.



MCP WRAP/UNWRAP

- Similar in concept to WFL COPY for tape
  - Similar WFL copy/restore syntax
  - WRAP copies MCP files into a *container file*
  - UNWRAP restores MCP files from a *container file*
- Two forms:
  - WRAP/UNWRAP a single file *(won't discuss this)*
  - WRAP/UNWRAP multiple files using a *container file*
- Built-in feature of MCP
  - WFL commands and Algol API (MCPFILEWRAPPER)
  - Available since SSR 44.1 (mid-1990s)
  - WRAP requires product key nnn-WRAPKEY-WRP
  - UNWRAP does not require a key

2020 Session 4025 7

WRAP and UNWRAP are similar in concept to WFL COPY commands for magnetic tape. Their WFL syntax is also similar to that for WFL COPY commands.

- WRAP commands copy MCP files into a *container file*, which we will discuss next.
- UNWRAP commands extract files from a container and restore them to the MCP file system.

WRAP and UNWRAP have two forms:

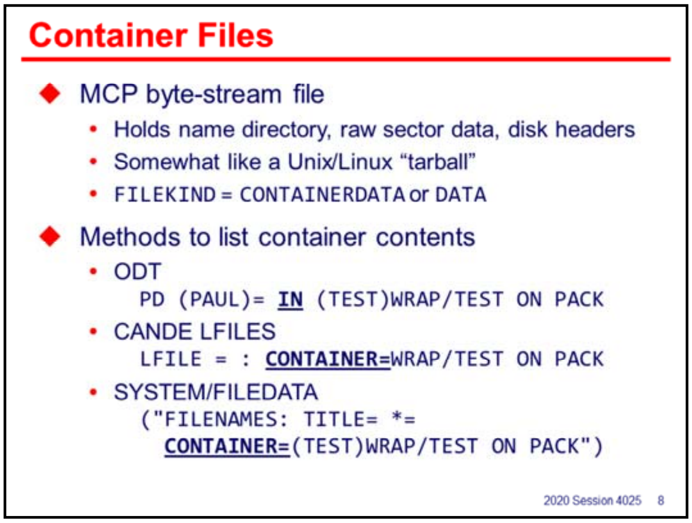
- WRAP a single MCP file into a single *wrapped file*. This is useful for converting a single file into a byte stream that can be transported across a network and later restored as an MCP file with all of its original attributes. I will not be discussing this form, as it is generally not suitable for backup.
- Wrap one or more MCP files into a *wrapped container file*. The container file is also a byte stream, and is somewhat analogous to a tape volume. This is the form I'll be concentrating on for the rest of the presentation.

WRAP and UNWRAP are built-in features of the MCP and are implemented along with WFL COPY as components of the Library/Maintenance facility. There is also an Algol API, `MCPFILEWRAPPER`, which is available in the `MCPSUPPORT` library.

WRAP has been around for quite a while. It was released as part of SSR 44.1 in the mid-1990s, and was originally designed as a means to deliver software over TCP/IP networks and the then-new Internet.

While WRAP is built into the standard MCP, you need a run-time key, "`nnn-WRAPKEY-WRP`," in order to use it for your own purposes. Without that key, WRAP is designed to be used only for submitting files to Unisys support. It will create a container file, but the file can be unwrapped only by Unisys.

UNWRAP, on the other hand, does not require a key, and can be used on any system.



Container Files

- MCP byte-stream file
  - Holds name directory, raw sector data, disk headers
  - Somewhat like a Unix/Linux “tarball”
  - FILEKIND = CONTAINERDATA or DATA
- Methods to list container contents
  - ODT

    ```
    PD (PAUL)= IN (TEST)WRAP/TEST ON PACK
    ```
  - CANDE LFILES

    ```
    LFILE = : CONTAINER=WRAP/TEST ON PACK
    ```
  - SYSTEM/FILEDATA

    ```
    ("FILENAMES: TITLE= *=
      CONTAINER=(TEST)WRAP/TEST ON PACK")
    ```

2020 Session 4025 8

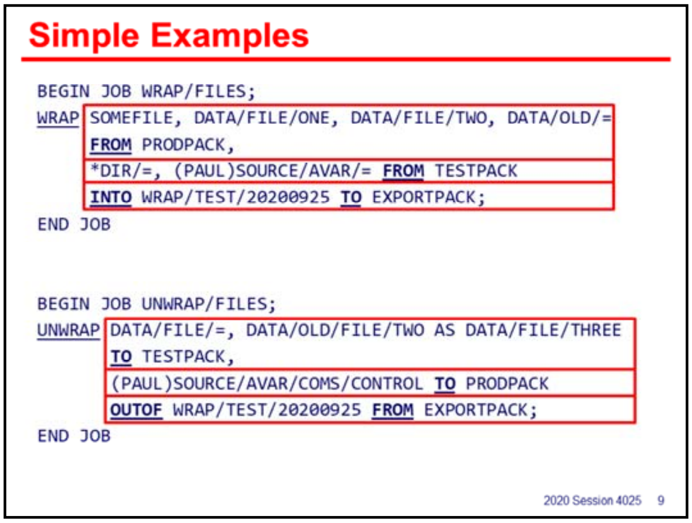
WRAP writes the files it is copying to an MCP “byte stream” file known as a *wrapped container file*. This is a file with records that are one 8-bit byte in length and no block structure. The data is written as a continuous stream of bytes. Logical I/Os can result in multiple physical I/Os and can cross area/row boundaries. The structure of this file is consistent with that of most other operating systems, including Windows, Unix, and Linux.

Internally, the structure of a wrapped container is somewhat like that of a Library/Maintenance tape. There is a directory containing the names of the copied files, the raw sector data for each file, and the disk header for each file, which holds the file’s metadata – its file attributes and disk allocation data. You can think of a container file as the MCP equivalent of a Unix/Linux “tarball” file.

Container files created by the MCP have a `FILEKIND` attribute value of `CONTAINERDATA`, but this is not a requirement for the file to be recognized as a container file. The file can be copied to another system or device and brought back, say by FTP or SMB file transfer, with the default `FILEKIND` of `DATA`, and the MCP will still be able to UNWRAP it.

There are multiple ways in which you can list the names of the files inside a container file:

- From the ODT, you can enter a command of the form `PD` <name> `IN` <container title>, where <name> is a file name or a directory/= name. The file names are listed on the ODT in the same manner as for other PD commands.
- The CANDE `LFILES` command (which runs `SYSTEM/FILEDATA` with an `ATTRIBUTES` request) can list the contents of a container file if you append the `CONTAINER`=<container title> modifier to the command.
- Most other task requests for `SYSTEM/FILEDATA` will accept a `CONTAINER` modifier as well.



Simple Examples

```
BEGIN JOB WRAP/FILES;
WRAP SOMEFILE, DATA/FILE/ONE, DATA/FILE/TWO, DATA/OLD/=
     FROM PRODPACK,
     *DIR/=, (PAUL)SOURCE/AVAR/= FROM TESTPACK
     INTO WRAP/TEST/20200925 TO EXPORTPACK;
END JOB


BEGIN JOB UNWRAP/FILES;
UNWRAP DATA/FILE/=, DATA/OLD/FILE/TWO AS DATA/FILE/THREE
       TO TESTPACK,
       (PAUL)SOURCE/AVAR/COMS/CONTROL TO PRODPACK
       OUTOF WRAP/TEST/20200925 FROM EXPORTPACK;
END JOB
```

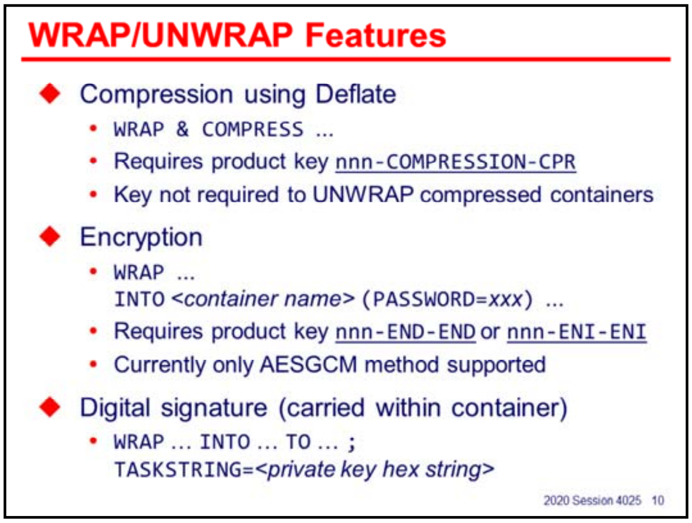
2020 Session 4025 9

The slide shows basic examples of WFL WRAP and UNWRAP commands. The syntax is similar to that of a WFL COPY statement:

- Following the `WRAP` keyword is a list of file and directory/= names and source family names. Since WRAP can copy files only from disk families and store its container file on a disk family, a `KIND` attribute in parentheses after the family name is not required. If `KIND` is specified, only the `DISK` or `PACK` mnemonics are accepted.
- Files may be renamed during the copy with an `AS` clause in the same manner as for COPY statements.
- After the list of file names and source families, a single `INTO` clause specifies the file name of the container file.
- After the `INTO` clause, there may optionally be a `TO` clause to specify the name of the disk family where the container file is to be written. If no `TO` clause is present, the destination family defaults to `DISK` with any family substitution applied.

The UNWRAP command essentially reverses the process:

- It takes a list of file and directory/= names specifying the files to be extracted from a container file and restored to the MCP file system.
- Each list of names may be followed by a `TO` clause to name the destination disk family. Since disk is the only possible destination, no `KIND` attribute in parentheses need be specified. If no destination family is specified, it defaults to `DISK` with family substitution applied.
- Files may be renamed with an `AS` clause when restored to the MCP file system as with COPY commands.
- After the list of file and destination family names, a single `OUTOF` clause specifies the file name of the container from which the files will be extracted.
- Following the `OUTOF` clause, there may optionally be a `FROM` clause to specify the name of the disk family where the container file is located. If no `FROM` clause is present, the source family defaults to `DISK` with any family substitution applied.



WRAP/UNWRAP Features

- Compression using Deflate
  - WRAP & COMPRESS ...
  - Requires product key nnn-COMPRESSION-CPR
  - Key not required to UNWRAP compressed containers
- Encryption
  - WRAP ...
    INTO *<container name>* (PASSWORD=xxx) ...
  - Requires product key nnn-END-END or nnn-ENI-ENI
  - Currently only AESGCM method supported
- Digital signature (carried within container)
  - WRAP ... INTO ... TO ... ;
    TASKSTRING=*<private key hex string>*

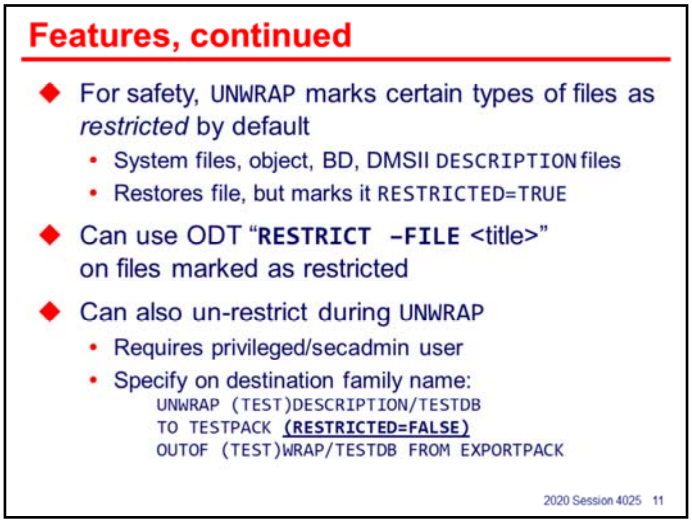
2020 Session 4025 10

WRAP has three optional features that modify the way a container file is created.

Container files may be compressed as they are written using the Deflate algorithm. You specify this by including "`& COMPRESS`" after the WRAP command. In order to create compressed containers, your system must have the run-time key "`nnn-COMPRESSION-CPR`" installed. If this key is not present, any request for compression is ignored. This key is not required to UNWRAP compressed containers, however.

Container files may also be encrypted as they are created. This is specified by including a `PASSWORD` attribute after the container file name of the `INTO` clause. In order to create encrypted containers, your system must have the run-time key "`nnn-END-END`" (for U.S. domestic customers) or "`nnn-ENI-ENI`" (for International customers) installed. If this key is not installed, any request for encryption is ignored with a warning message. Currently the only encryption method supported by WRAP is `AESGCM`.

Container files may be protected with a digital signature that is carried within the file itself. You specify this by appending a `TASKSTRING` attribute to the WRAP command. The value of that attribute is a hexadecimal string for the signature's private key. These keys are sensitive to each release level. By default the current release level is used. To WRAP or UNWRAP a container for a different release level, specify the level (e.g., 591) in the `TARGET` task attribute for the command.



Features, continued

- For safety, UNWRAP marks certain types of files as *restricted* by default
  - System files, object, BD, DMSII DESCRIPTION files
  - Restores file, but marks it RESTRICTED=TRUE
- Can use ODT "**RESTRICT -FILE** <title>" on files marked as restricted
- Can also un-restrict during UNWRAP
  - Requires privileged/secadmin user
  - Specify on destination family name:

```
UNWRAP (TEST)DESCRIPTION/TESTDB
TO TESTPACK (RESTRICTED=FALSE)
OUTOF (TEST)WRAP/TESTDB FROM EXPORTPACK
```

2020 Session 4025 11

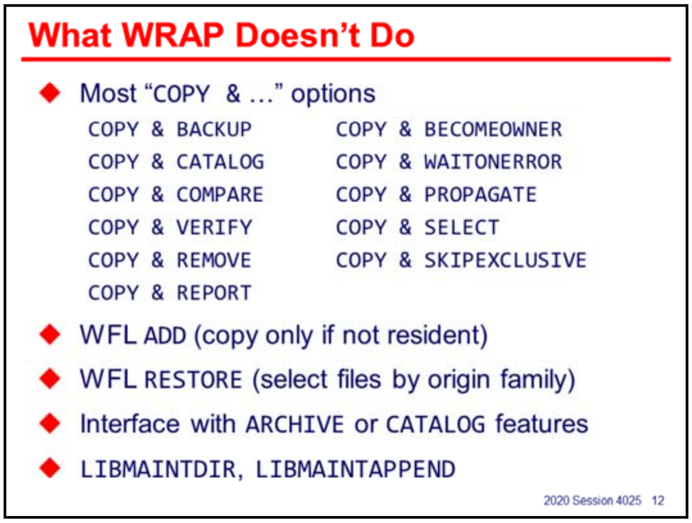
Wrapped container files often pass through other systems, and since they are just a byte-stream, can easily be modified. To protect MCP systems, certain types of files are by default marked as RESTRICTED when they are unwrapped and stored in an MCP file system. The file data for these is intact, but the files cannot be read or written while they have this restricted status.

The types of files that can be marked as restricted as they are unwrapped include:

- Certain system files
- Object code files
- Printer backup (BD) files
- DMSII DESCRIPTION files.

There are two ways these files can be unwrapped and made usable for an MCP system.

- After the files are unwrapped, they can be unrestricted by means of the ODT command "RESTRICT -FILE <file title>." This command must be executed separately for each file.
- The files can be marked as unrestricted as they are copied to the MCP system by UNWRAP if the attribute RESTRICTED=FALSE is specified in parentheses after the destination family name in the command. Doing this requires that UNWRAP be run under a privileged usercode.



What WRAP Doesn't Do

- Most "COPY & …" options

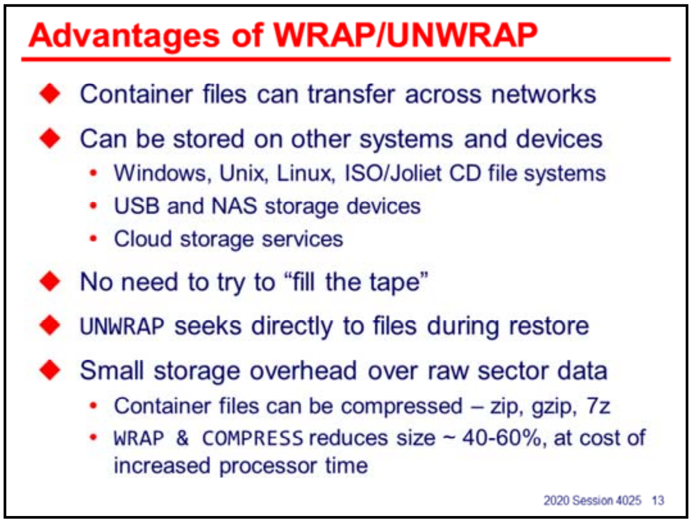
  COPY & BACKUP
  COPY & CATALOG
  COPY & COMPARE
  COPY & VERIFY
  COPY & REMOVE
  COPY & REPORT
  COPY & BECOMEOWNER
  COPY & WAITONERROR
  COPY & PROPAGATE
  COPY & SELECT
  COPY & SKIPEXCLUSIVE

- WFL ADD (copy only if not resident)
- WFL RESTORE (select files by origin family)
- Interface with ARCHIVE or CATALOG features
- LIBMAINTDIR, LIBMAINTAPPEND

2020 Session 4025 12

As capable as WRAP/UNWRAP is, there are a number of things it doesn't do compared to Library/Maintenance with tape media.

- As shown on the slide, WRAP and UNWRAP support only a few of the "&" command options compared to COPY. In particular, it does not support "& REMOVE" and "& REPORT." Container files include a per-file checksum, so the "& VERIFY" option is essentially baked in.
- UNWRAP cannot do the equivalent of a WFL ADD command, which restores only those files that are non-resident. UNWRAP will overwrite existing files of the same name.
- WRAP does not record the origin family name the way that COPY to tape does, so you cannot automatically restore files to the disk family that they came from. There is no equivalent for the WFL RESTORE command or the ORIGIN clause in COPY file lists. This is a potential problem if a container file holds files from multiple disk families and some of the files on different families have the same name. You can, however, restore files by their ordinal position in the container, by selecting files using "#<file number>" instead of the file name. Container directory listings generated by SYSTEM/FILEDATA show these file numbers.
- WRAP and UNWRAP do not have any interface to the ARCHIVE or CATALOG subsystems, so no separate record of where files were backed up is maintained by the MCP.
- Finally, there is no equivalent to the LIBMAINTDIR or LIBMAINTAPPEND features of COPY to tape, although the name directory and fast-seek capabilities of UNWRAP, which I will discuss next, provide much the same benefit as LIBMAINTDIR.



Advantages of WRAP/UNWRAP

- Container files can transfer across networks
- Can be stored on other systems and devices
  - Windows, Unix, Linux, ISO/Joliet CD file systems
  - USB and NAS storage devices
  - Cloud storage services
- No need to try to "fill the tape"
- `UNWRAP` seeks directly to files during restore
- Small storage overhead over raw sector data
  - Container files can be compressed – zip, gzip, 7z
  - `WRAP & COMPRESS` reduces size ~ 40-60%, at cost of increased processor time

2020 Session 4025 13

There are a number of advantages to using WRAP/UNWRAP for file backup:

- Container files, being byte streams, can be transferred transparently across networks. Once transferred back to an MCP system, UNWRAP can restore the files to the MCP file system with all data and attribute values intact.
- Container file byte streams can be stored on other systems and devices. These include:
  - Windows, Unix, Linux, and ISO/Joliet CD file systems.
  - USB and NAS storage devices, including generic USB "thumb drives," USB backup disks, and network-connected storage systems.
  - Cloud storage services.

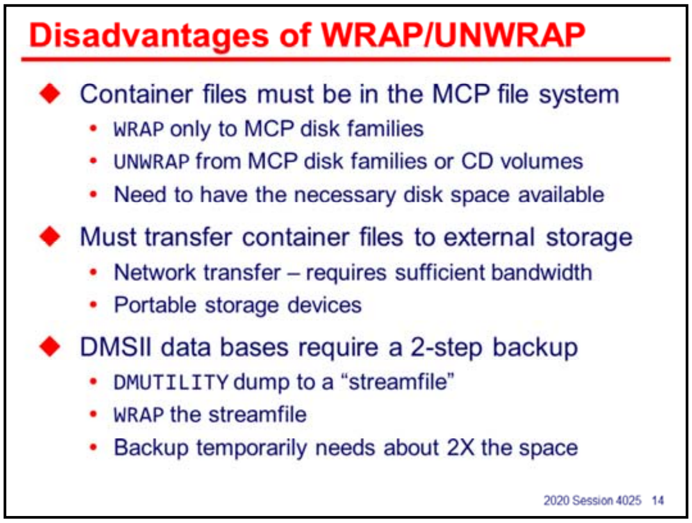
Unlike backup to high-capacity tape media, there is no need to structure your backups to "fill the tape." Container files are only as large as they need to be.

When selectively restoring files using UNWRAP, the restoration process does not need to sequentially scan the container file to find the files. It can seek directly to the locations within the container to where those files reside. Anyone who remembers the tedium of copying files from QIC tape will appreciate this.

Container files store data efficiently. Aside from the space required for the raw sector data of the files, the process adds only the space required for a small container header, the file name directory, and the disk headers, plus a few dozen bytes of control data per file. If a digital signature is used, this adds a small additional amount of data.

Container files can be compressed after creation using any of the popular compression methods, including zip, gzip, and 7z.

`WRAP & COMPRESS` reduces the size of a container file as it is being written. Some simple and unscientific tests I've run indicate compression reduces the container file size by 40-60%, but this will obviously vary with the nature of the data being compressed. Note that this compression comes at the cost of significantly higher MCP processor time.



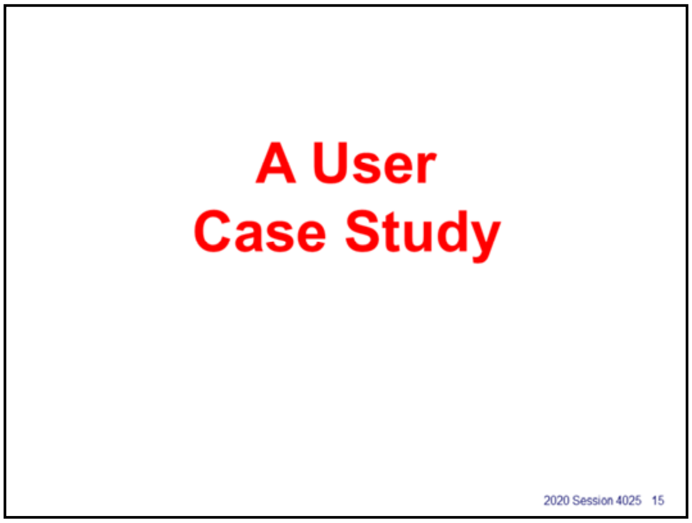
Disadvantages of WRAP/UNWRAP

- Container files must be in the MCP file system
  - `WRAP` only to MCP disk families
  - `UNWRAP` from MCP disk families or CD volumes
  - Need to have the necessary disk space available
- Must transfer container files to external storage
  - Network transfer – requires sufficient bandwidth
  - Portable storage devices
- DMSII data bases require a 2-step backup
  - `DMUTILITY` dump to a “streamfile”
  - `WRAP` the streamfile
  - Backup temporarily needs about 2X the space

2020 Session 4025 14

WRAP/UNWRAP has a number of disadvantages compared to COPY to tape:

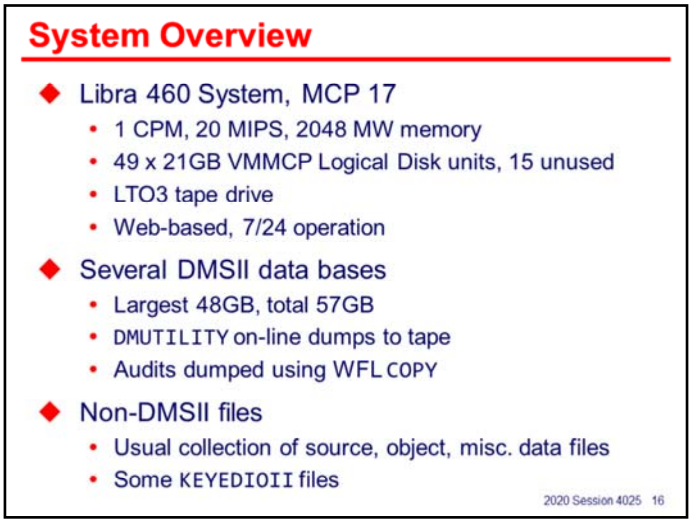
- At present, WRAP must create container files in the MCP file system and UNWRAP must read container files from the MCP file system or an ISO/Joliet CD volume. This means you must have sufficient space on some MCP disk family to hold a container file while it is being created by WRAP or being read by UNWRAP. This could be an issue for very large systems.
- Once you back up files with WRAP, you need to transfer them to some form of external storage for safety. If you are planning to transfer the files over a network to some other location, you will need to make sure you have sufficient bandwidth to transfer the container files in a reasonable amount of time. If you plan to transfer the container files directly to portable storage (e.g., USB devices), you will need to make sure the devices have sufficient capacity and have a means to transport them off site. You will also need a naming convention or other administrative scheme to identify and keep track of the files stored externally.
- DMSII data bases cannot be backed up directly using WRAP, or at least not in a way that they can be reliably restored. The proper procedure is to use `SYSTEM/DMUTILITY` to back up the data base as a “streamfile” dump. That streamfile dump can then be copied using WRAP. This means that the data base backup will need, temporarily, at least two times the disk space required for the data base files. DMSII audit trails can be copied with WRAP directly, but it is best to first be sure the audit trail files are closed. I will discuss DMSII backups in more detail during the case study, which comes next.



A User Case Study

2020 Session 4025 15

In this next portion of the presentation, I will talk about the experience I had in helping a customer migrate their MCP system backups from tape to WRAP/UNWRAP.



System Overview

- Libra 460 System, MCP 17
  - 1 CPM, 20 MIPS, 2048 MW memory
  - 49 x 21GB VMMCP Logical Disk units, 15 unused
  - LTO3 tape drive
  - Web-based, 7/24 operation
- Several DMSII data bases
  - Largest 48GB, total 57GB
  - `DMUTILITY` on-line dumps to tape
  - Audits dumped using WFL `COPY`
- Non-DMSII files
  - Usual collection of source, object, misc. data files
  - Some `KEYEDIOII` files

2020 Session 4025 16

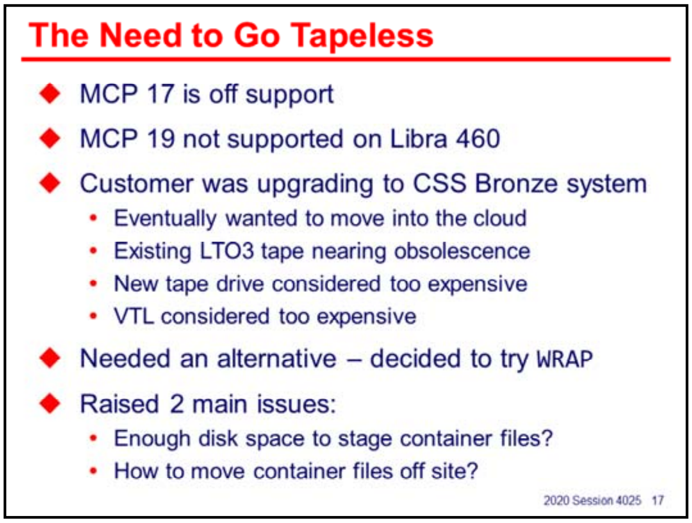
The customer in question had a Libra 460 running MCP 17. This was a relatively low-end system – 20 MIPS, 2048 mega-words of memory. It did have a generous amount of disk for a system that size, however, a total of 49 VMMCP Logical Disk units of 21GB each. Fifteen of these units had been set up for a project that never materialized, so were unused and available.

Backups were being done to LTO3 tape.

The site had significant on-line and web activity from the customer's clients, and ran 7/24 with as minimal a level of system interruptions as possible.

There were several DMSII data bases. The largest of these was about 47GB, with all data bases totaling about 57GB. The data bases were backed up two or three times a week using on-line dumps, with audit trails backed up daily using ordinary Library/Maintenance, i.e., no `SYSTEM/COPYAUDIT`.

Non-DMSII files were backed up about once a week, each family on specific days. These covered the usual collection of source, object, data, and miscellaneous files. There were some `KEYEDIOII` files. The applications using these were disabled during backups to assure the files were closed and could be copied safely.



The Need to Go Tapeless

- MCP 17 is off support
- MCP 19 not supported on Libra 460
- Customer was upgrading to CSS Bronze system
  - Eventually wanted to move into the cloud
  - Existing LTO3 tape nearing obsolescence
  - New tape drive considered too expensive
  - VTL considered too expensive
- Needed an alternative – decided to try WRAP
- Raised 2 main issues:
  - Enough disk space to stage container files?
  - How to move container files off site?

2020 Session 4025 17

The customer’s decision to move away from tape was essentially an economic one.

The Libra 460 was at end of life. MCP 17 has been off support for months, and MCP 19 will not run on a 460.

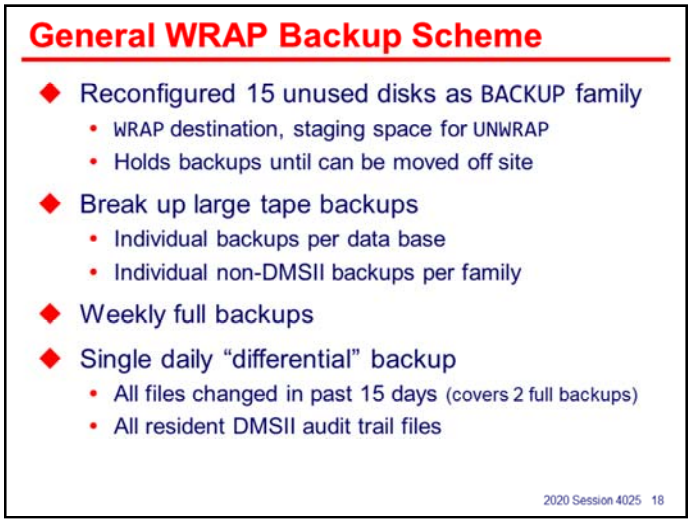
The customer decided to upgrade to a CSS Bronze system, and eventually wanted to reduce their office footprint and move their applications into the Cloud. The existing LTO3 tape drive was nearing obsolescence. Both a new LTO drive and a VTL were considered to be too expensive, so they needed an alternative to tape.

After discussing their requirements with them, we decided to try using WRAP for backup.

This decision raised two main issues:

- Did they have enough spare disk space to create the necessary container files?
- How would the container files be moved out of the MCP file system and transported to external storage?

The following slides discuss the backup scheme we developed and how we addressed these questions.



General WRAP Backup Scheme

- Reconfigured 15 unused disks as BACKUP family
  - WRAP destination, staging space for UNWRAP
  - Holds backups until can be moved off site
- Break up large tape backups
  - Individual backups per data base
  - Individual non-DMSII backups per family
- Weekly full backups
- Single daily “differential” backup
  - All files changed in past 15 days (covers 2 full backups)
  - All resident DMSII audit trail files

2020 Session 4025 18

To get the disk space we would need to hold the container files, we reconfigured the 15 unused Logical Disk units as one large family, creatively named BACKUP. This would be used both as the destination family for WRAP operations, and as staging space for container files when they needed to be read by UNWRAP.

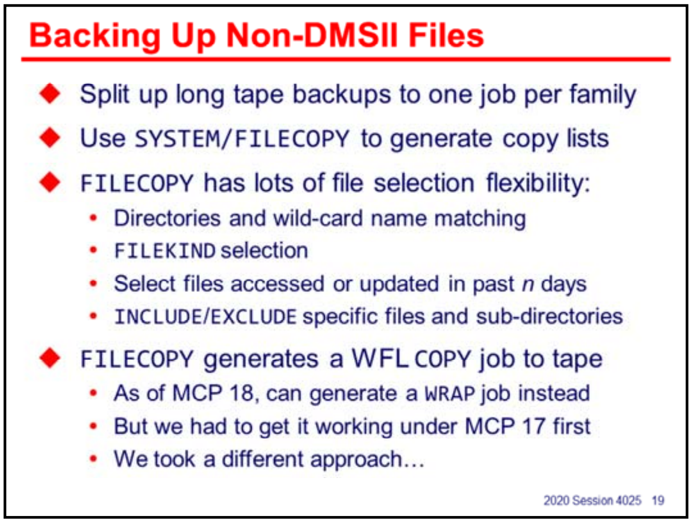
The next thing we did was look at the structure of their current backup jobs. At least one of these backed up files from multiple families and appeared to have been designed to maximize use of an LTO tape’s high capacity. That job was going to challenge the amount of space we had on the BACKUP family, but more importantly, the amount of time it would take to transfer the container file from the MCP file system.

The solution was to break up the large backup jobs so there was basically one job per disk family for non-DMSII backups, plus one job per data base. There were a half-dozen small DMSII data bases, so we decided to handle those with a single job for all of them.

In addition, we decided to move to a somewhat “differential” mode of backup. This meant that full backups would be done once per week. These full backups were distributed throughout the week to spread out the backup load. Then, on a daily basis, we would run one “differential” backup job that would capture all files on the system that were new or had been modified within the past 15 days (thus covering the past two full backups), plus all resident DMSII audit trails for all data bases.

Finally, we found that their tape backups were copying a lot of files they didn’t need. MCP systems tend to accumulate certain types of files, especially system logs and DMSII audit trail files. There were also a number of “temporary” files that were generated by batch jobs but never purged. This useless data was bloating the backups, and the manual procedures for cleaning up these files were not working very well.

After some analysis to determine what files they had and which ones they really needed, we implemented an automated file purging process to search directories where files tended to accumulate and remove them after they had reached a certain age. The purge process was simply a batch job that was run one a week or so, using a utility program that would do the necessary directory searching and file removal.



Backing Up Non-DMSII Files

- Split up long tape backups to one job per family
- Use `SYSTEM/FILECOPY` to generate copy lists
- `FILECOPY` has lots of file selection flexibility:
  - Directories and wild-card name matching
  - `FILEKIND` selection
  - Select files accessed or updated in past *n* days
  - `INCLUDE/EXCLUDE` specific files and sub-directories
- `FILECOPY` generates a WFL `COPY` job to tape
  - As of MCP 18, can generate a `WRAP` job instead
  - But we had to get it working under MCP 17 first
  - We took a different approach…

2020 Session 4025 19

The first part of the new backup scheme to discuss in detail is the one we developed for non-DMSII files.

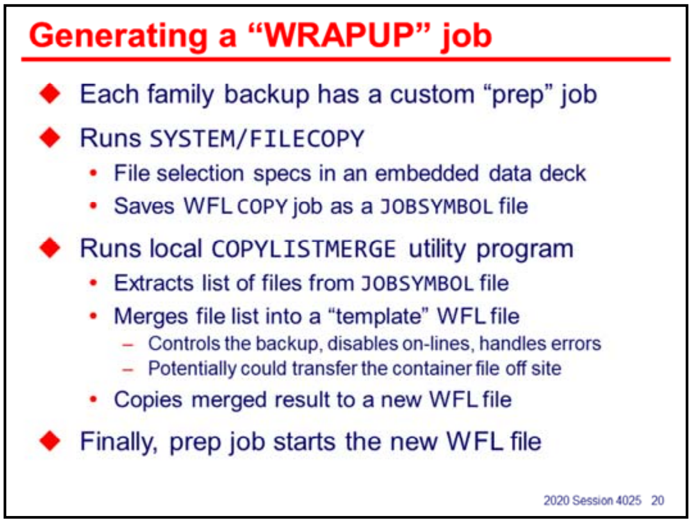
The general idea for backing up these files was to split up the rather long tape backup jobs to have one job per family, plus a common daily "differential" dump. The customer wanted only certain disk directories on each family to be backed up, and to exclude certain sub-directories as well.

We chose to use the MCP's standard `SYSTEM/FILECOPY` utility to generate the lists of files to be backed up. `FILECOPY` is designed to generate WFL backup jobs. It has a lot of flexibility in how it selects files for inclusion in the COPY list, including:

- Selecting whole directories.
- `SYSTEM/PDIR`-like wild-card name matching.
- Inclusion or exclusion of files by their `FILEKIND`.
- Selecting only files accessed or updated in the past *n* days or since a specified date/time.
- Selection by exception – specifically including or excluding certain files and sub-directories.

`FILECOPY` generates a WFL job that can be either started immediately or saved in a `JOBSYMBOL` file. It was originally designed to produce WFL `COPY` statements, but as of MCP 18 it has an option to generate `WRAP` statements instead – see its `WRAPLABEL` option.

The customer was still on MCP 17, however, and we had to get the new mechanism working on MCP 17 first, so having `FILECOPY` generate WRAP jobs was not possible. We had to take a different approach.



Generating a "WRAPUP" job

- Each family backup has a custom "prep" job
- Runs `SYSTEM/FILECOPY`
  - File selection specs in an embedded data deck
  - Saves WFL `COPY` job as a `JOBSYMBOL` file
- Runs local `COPYLISTMERGE` utility program
  - Extracts list of files from `JOBSYMBOL` file
  - Merges file list into a "template" WFL file
    - Controls the backup, disables on-lines, handles errors
    - Potentially could transfer the container file off site
  - Copies merged result to a new WFL file
- Finally, prep job starts the new WFL file

2020 Session 4025 20

For each disk family, we created a custom "prep" job to initiate the backup. This job runs `SYSTEM/FILECOPY` with the necessary selection specifications and other `FILECOPY` options in an embedded data deck. `FILECOPY` then generates a WFL job with the list of selected files as a `JOBSYMBOL` file.

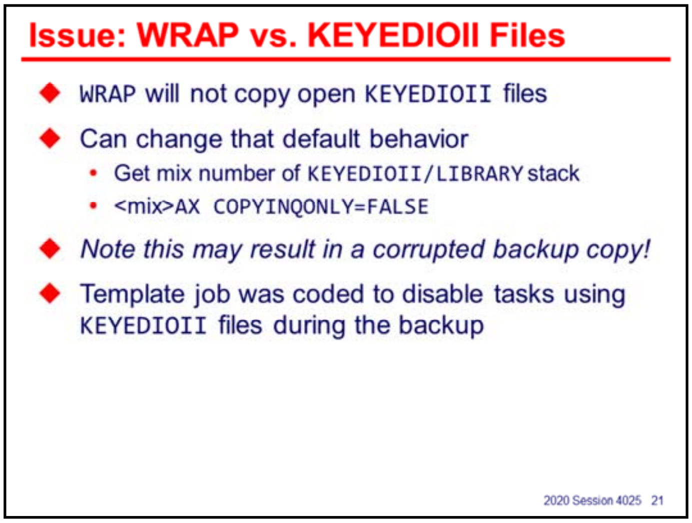
I wrote a simple COBOL utility program, `COPYLISTMERGE`, that extracts the list of files from the `FILECOPY`-generated output and inserts that list into a WFL "template job" file. The template file had everything necessary to do the backup, except that the list of files was represented by a marker record. `COPYLISTMERGE` simply copied the template file to a new file, inserting the list of files from `FILECOPY` in place of the maker record. We called the result a "WRAPUP" job.

The prep job then started the newly-created WFL file to run the backup. We consider the WRAPUP file to be temporary and it is overwritten with each backup run.

The template job file has all of the code necessary to control the backup, disable on-line programs if necessary, and handle errors. Although we did not implement it, the template job has the potential to include code to automatically transfer the resulting wrapped container file over a network after the WRAP portion of the backup completes.

The same mechanism was used for both the full-family backups and the daily "differential" backup. The only significant differences are that the differential prep job had essentially the union of all of the file selection criteria from the individual family prep jobs, plus was configured to select only those files that had been modified in the past 15 days, including DMSII audit trial files.

`FILECOPY` has a more sophisticated method for doing incremental and differential backups, using so-called "index files." See the `ADDED SINCE` and `ALLFILES SINCE` task requests. We chose not to use this, as it did not fit well with the way we wanted our differential backup to work.



Issue: WRAP vs. KEYEDIOII Files

- WRAP will not copy open KEYEDIOII files
- Can change that default behavior
  - Get mix number of KEYEDIOII/LIBRARY stack
  - <mix>AX COPYINQONLY=FALSE
- *Note this may result in a corrupted backup copy!*
- Template job was coded to disable tasks using KEYEDIOII files during the backup

2020 Session 4025 21

One issue with WRAP is that it will not copy KEYEDIOII files that are currently open for update. Library/Maintenance COPY has the same issue. WRAP will hang when it reaches the open KEYEDIOII file and wait for it to be closed.

You can change this behavior by changing the KEYEDIOII library's `COPYINQONLY` option, which by default is set to `TRUE`. To do that, find the mix number of the `KEYEDIOII/LIBRARY` stack. The ODT command `LIBS NAME =KEYEDIOII=` is a good way to do this. Then use that mix number in an ODT command similar to this:

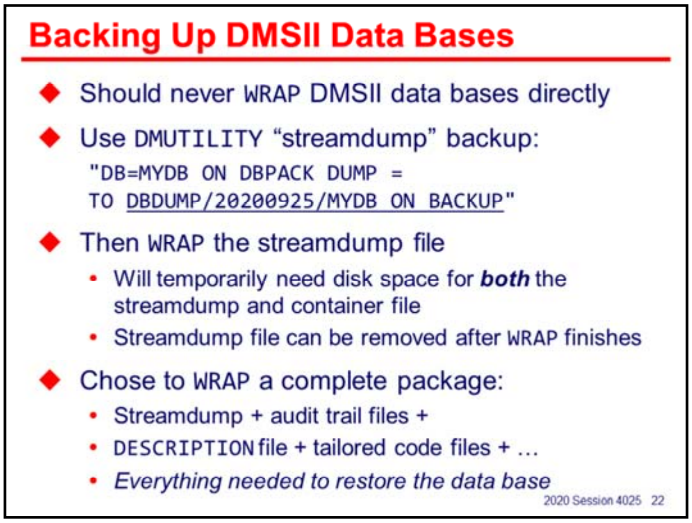
```
<mix>AX COPYINQONLY=FALSE
```

That will allow WRAP (and other Library/Maintenance commands) to back up a KEYEDIOII file even if it is currently open for update.

Note, however, that this copies the KEYEDIOII data and index files in whatever state they are presently in. There may be buffers in memory that have not been flushed to the files, so it is possible that the files as backed up will be corrupt.

We chose to set this option anyway, thinking that having a corrupt KEYEDIOII file on the backup was a lot better than having no file at all.

As a practical matter, this was not an issue for the customer. The backup job issues ODT commands to disable those on-line programs using KEYEDIOII files before initiating WRAP, and re-enables those programs after the WRAP completes. Thus, the files backed up would not have been in use, and would not be subject to corruption on the backup copy.



Backing Up DMSII Data Bases

- Should never `WRAP` DMSII data bases directly
- Use `DMUTILITY` "streamdump" backup:

```
"DB=MYDB ON DBPACK DUMP =
TO DBDUMP/20200925/MYDB ON BACKUP"
```

- Then `WRAP` the streamdump file
  - Will temporarily need disk space for ***both*** the streamdump and container file
  - Streamdump file can be removed after `WRAP` finishes
- Chose to `WRAP` a complete package:
  - Streamdump + audit trail files +
  - `DESCRIPTION` file + tailored code files + …
  - *Everything needed to restore the data base*

2020 Session 4025 22

Backing up DMSII data bases using WRAP requires extra consideration. Just as you should not use WFL COPY to back up DMSII data bases, you should not use WRAP to back them up. There are two main reasons for this:

1. Neither COPY nor WRAP provide any protection against copying a data base that is currently being updated. Doing so would very likely produce a backup with corrupted data base files.
2. Library/Maintenance tapes and wrapped container files cannot be used with DMSII recovery to restore a data base and apply transactions from audit trail files.

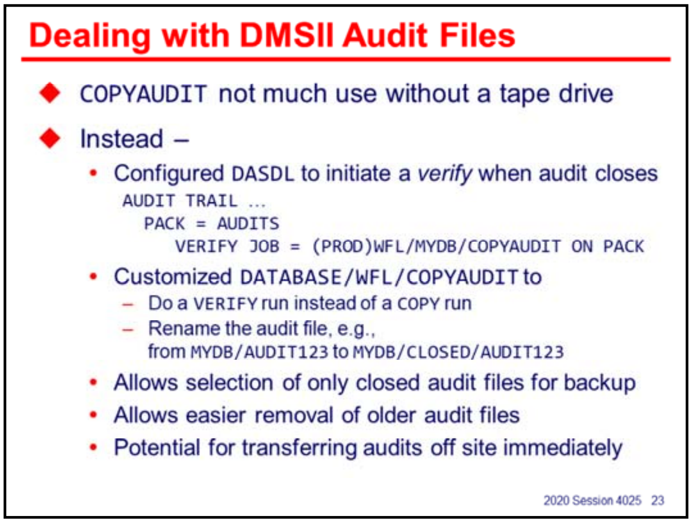
The proper way to back up DMSII data bases is with the `SYSTEM/DMUTILITY` program. `DMUTILITY` can dump a data base directly to tape, but it does not have the ability to dump a data base to a wrapped container file. What it does have, however, is the ability to dump a data base to a "streamfile." This is just an MCP file that contains the data base data blocks and backup metadata much the same as they would have been written to a tape. It supports on-line dumps.

Backing up to a streamfile instead of tape is easy – in the `TO` clause of the dump command, you simply specify a full title for the streamdump file instead of a tape name. You may optionally specify `FILES` and `BLOCKSIZE` options after the file title. The `FILES` option partitions the dump into multiple files. The `BLOCKSIZE` option specifies the streamfile `BLOCKSIZE` attribute, which defaults to 19,200 words.

The streamfile dump can then be wrapped like any other MCP file, possibly along with other files into the same container. Note that creating a streamfile and then wrapping it means that, temporarily, you will need available disk space amounting to at least twice the size of the stream file. The streamfile and wrapped container can be on separate disk families, however, and the streamfile can be removed once the container file has been created and the WRAP run finishes.

In our design, we created a separate backup job for each data base, except that we combined several very small data bases into one job and created one wrapped container for them.

We also decided to WRAP what might be termed a complete restore package for each data base. This included not only the streamfile dump, but all of the resident audit trail files, the `DESCRIPTION` file, the tailored code files (`DMSUPPORT`, etc.), the DASDL source, and critical WFL files associated with the data base. The goal was to have everything we needed to restore the data base in one wrapped container file.



Dealing with DMSII Audit Files

- COPYAUDIT not much use without a tape drive
- Instead –
  - Configured DASDL to initiate a *verify* when audit closes

```
AUDIT TRAIL ...
  PACK = AUDITS
     VERIFY JOB = (PROD)WFL/MYDB/COPYAUDIT ON PACK
```

  - Customized DATABASE/WFL/COPYAUDIT to
    - Do a VERIFY run instead of a COPY run
    - Rename the audit file, e.g., from MYDB/AUDIT123 to MYDB/CLOSED/AUDIT123
  - Allows selection of only closed audit files for backup
  - Allows easier removal of older audit files
  - Potential for transferring audits off site immediately

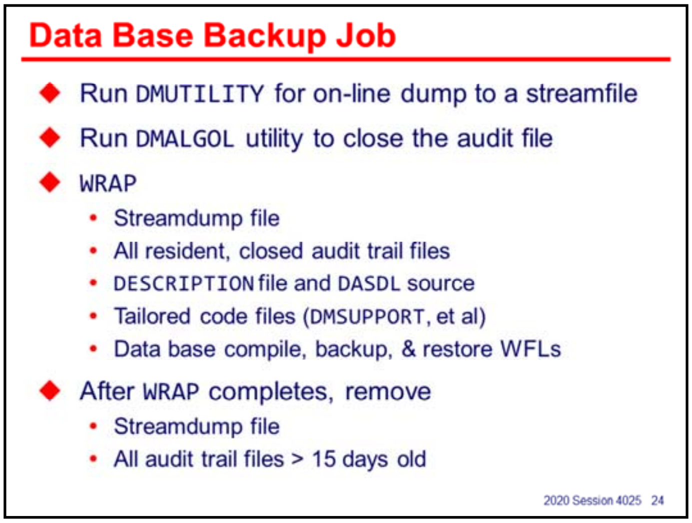
2020 Session 4025 23

DMSII audit trial files also require some special consideration when backing them up with WRAP. `SYSTEM/COPYAUDIT` is not much use for backing up audit trails when you don't have a tape drive.

Instead, we configured the data base DASDL to specify a `COPYAUDIT` job to be started when an audit trail file was closed by DMSII. Instead of copying the file, however, the `VERIFY` option was specified in DASDL. This causes DMSII to pass a `VERIFY` command to `COPYAUDIT`, which simply reads the file, and validates its checksums and inter-block consistency.

We also modified the standard `DATABASE/WFL/COPYAUDIT` job to change the name of the audit trail file after `COPYAUDIT` had verified it. If an audit trail file was named `DBNAME/AUDIT123`, this job changed it to `DBNAME/CLOSED/AUDIT123`. Adding the intermediate node allows us to select only closed audit files for backup, and to more easily remove older, obsolete audit files.

This approach also has the potential to include code to transfer audit trail files off site immediately after they are closed. We have not implemented this as yet, but are considering it for the future.



Data Base Backup Job

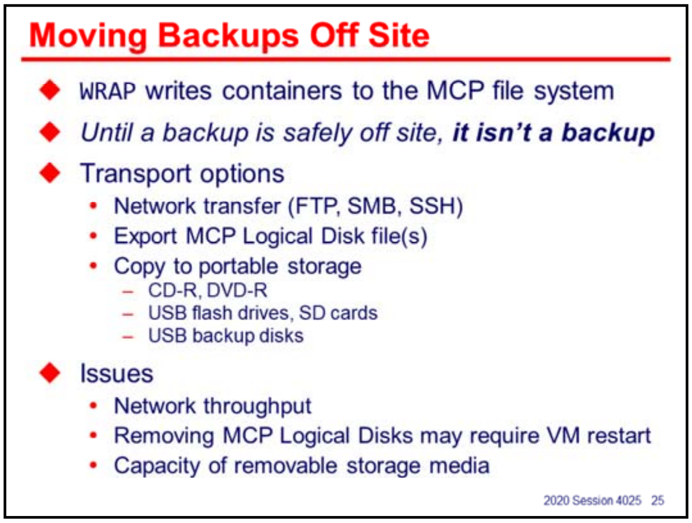
- Run DMUTILITY for on-line dump to a streamfile
- Run DMALGOL utility to close the audit file
- WRAP
  - Streamdump file
  - All resident, closed audit trail files
  - DESCRIPTION file and DASDL source
  - Tailored code files (DMSUPPORT, et al)
  - Data base compile, backup, & restore WFLs
- After WRAP completes, remove
  - Streamdump file
  - All audit trail files > 15 days old

2020 Session 4025 24

The data base backup jobs all had the same general pattern:

1. Run `SYSTEM/DMUTILITY` to generate an on-line dump to a streamfile.
2. Run a home-grown utility that uses the DMALGOL `DMINQ` API to close the audit file.
3. Wrap into one container file the steamdump file, all resident, closed audit files, the `DESCRIPTION` file, the DASDL source file, all tailored code files, and the WFLs used for data base compile, update, backup, and restore.
4. After the WRAP completes, remove the streamdump file and any closed audit trail files more than 15 days old.

The backup job for the several very small data bases worked the same way, except they had multiple runs of `DMUTILITY` and the audit-close utility program, followed by a single WRAP run to back up the files for all of the data bases into a single container file.



Moving Backups Off Site

- WRAP writes containers to the MCP file system
- *Until a backup is safely off site,* ***it isn't a backup***
- Transport options
  - Network transfer (FTP, SMB, SSH)
  - Export MCP Logical Disk file(s)
  - Copy to portable storage
    - CD-R, DVD-R
    - USB flash drives, SD cards
    - USB backup disks
- Issues
  - Network throughput
  - Removing MCP Logical Disks may require VM restart
  - Capacity of removable storage media

2020 Session 4025 25

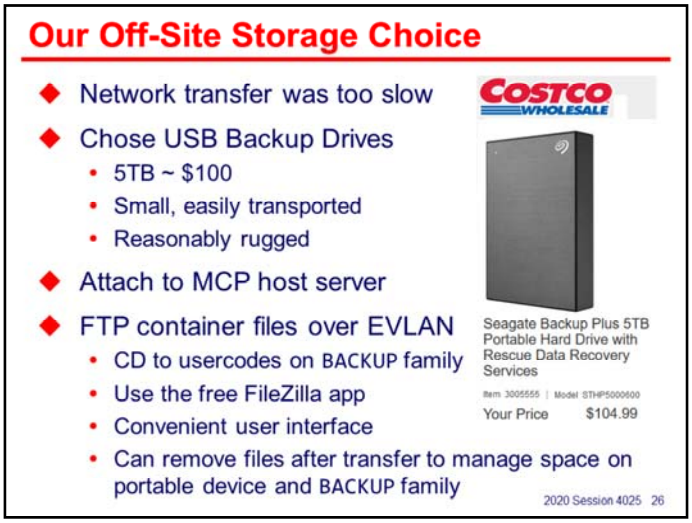
Now that we have some wrapped container files for our system, what are we going to do with them? Remember, until you have a backup safely stored off site, it isn't really a backup yet.

WRAP writes container files to the MCP file system, so an additional step is necessary to get those files off site. With tape backups, you simply dismount the tape and carry it away, but with WRAP, it's not that convenient. There are several possibilities for dealing with the problem, of which the main ones are:

- Transfer the container files over a network to a system or device at a remote location. Once transferred, the container file can be removed from the MCP host.
- Write the container files to an MCP disk family stored on Logical Disk devices, then remove the Logical Disk from the system configuration and transport it off site.
- Copy the wrapped container files to some sort of portable storage device, then disconnect the storage device and transport it off site. Examples would be CD-R/DVD-R devices, USB flash drives and SD cards, and USB backup disks. The MCP will not talk directly to USB devices, but on the smaller emulated MCP systems, you can plug a USB device into the Windows side of the system and transfer files from the MCP over the EVLAN.

There are potential issues with all of these options.

- The big issues with network transfer are the speed of your network and the speed of the MCP's network interface. This is especially true if you are transferring files over the Internet to a remote site. If you have a 50GB container file and a 10 Mb Internet connection, the transfer will probably take in excess of 12 hours. Older MCP systems running classic TCP/IP are not capable of very high transfer speeds, so even if your Internet connection speed is sufficient, your MCP system may not be able to pump the bits fast enough. The MCP's new TCPv3 implementation considerably eases that speed constraint.
- The idea of backing up to a Logical Disk unit and then removing the physical disk on which that Logical Disk is stored has some promise, but at present it's difficult at the Windows level to get the VM to let go of a physical device that holds a Logical Disk file.
- If you are using removable media, you will need to make sure that the container files you are creating will fit on the device. Inexpensive USB backup drives with capacities up to 5TB are now common. WRAP cannot segment container files the way that Library/Maintenance COPY to tape can switch volumes, so the size of the container file is something you need to limit by the way you select files for backup.



Our Off-Site Storage Choice

- Network transfer was too slow
- Chose USB Backup Drives
  - 5TB ~ $100
  - Small, easily transported
  - Reasonably rugged
- Attach to MCP host server
- FTP container files over EVLAN
  - CD to usercodes on BACKUP family
  - Use the free FileZilla app
  - Convenient user interface
  - Can remove files after transfer to manage space on portable device and BACKUP family

Seagate Backup Plus 5TB Portable Hard Drive with Rescue Data Recovery Services

Item 3005555 | Model STHP5000600

Your Price $104.99

2020 Session 4025 26

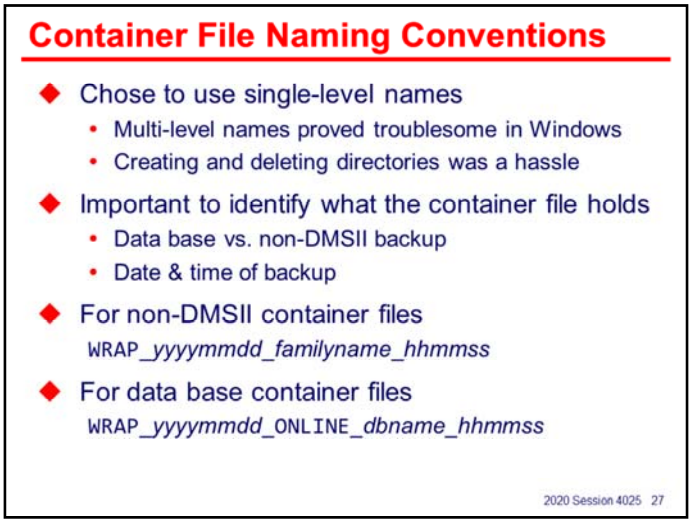
In my customer’s situation, we determined that network transfer was going to be too slow, and we were stuck with running classic TCP/IP until after the new Software Series system would be installed. We therefore decided to try using inexpensive USB backup drives. 5TB Seagate drives are currently available at Costco stores for about $100 USD each. We found that 3TB drives would be more than sufficient for current backup sizes. These drives are a little larger than a deck of playing cards and rugged enough that they can be transported with some care. They are slightly smaller than an LTO cartridge.

On our first attempt to use these backup drives, we connected one to a PC on the same LAN segment as the MCP system and transferred the files using simple Windows Explorer click-and-drag. That worked, but the MCP’s network interface was too slow to make it practical for most backups.

On our second attempt, we attached the USB drive to the Windows side of the Libra 460 server and transferred the files from the MCP over the internal EVLAN using FTP, which is more efficient than transferring files using Windows Explorer. We saw an almost 10X improvement in transfer time with this approach.

For the operators, we installed the free FileZilla application on the Windows side of the Libra 460. That allowed us to set up pre-configured site profiles for each of the production usercodes with a default source for that usercode on the MCP BACKUP family and an unspecified destination. After plugging in the USB drive, the system operator could open FileZilla, select the appropriate site profile, use FileZilla’s Explorer-like interface to set the destination device, and then initiate the file transfer.

FileZilla worked well for us. Its FTP engine is efficient, and it provides a convenient and easy-to-learn interface for the operator. A further advantage is that the operator could stay within FileZilla to inspect the files on both the backup drive and the BACKUP family, removing old files to make room for new backups.



Container File Naming Conventions

- Chose to use single-level names
  - Multi-level names proved troublesome in Windows
  - Creating and deleting directories was a hassle
- Important to identify what the container file holds
  - Data base vs. non-DMSII backup
  - Date & time of backup
- For non-DMSII container files

  WRAP_*yyyymmdd_familyname_hhmmss*
- For data base container files

  WRAP_*yyyymmdd*_ONLINE_*dbname_hhmmss*

2020 Session 4025 27

We gave some careful thought to the naming of the backup container files so that it would be easy to identify what the containers held and when they were created. We started out using multi-level file names, but found that the process of creating and deleting directories on the Windows file system of the backup drives was more trouble than it was worth. Instead, we decided to use long, single-level file names.

The three things to us that were most important to convey in the container file names were:

1. Whether it was a DMSII data base backup or a non-DMSII file backup.
2. The data base or disk family to which the backup was related.
3. The date and time of the backup.

Therefore, for the non-DMSII container files, the convention was:

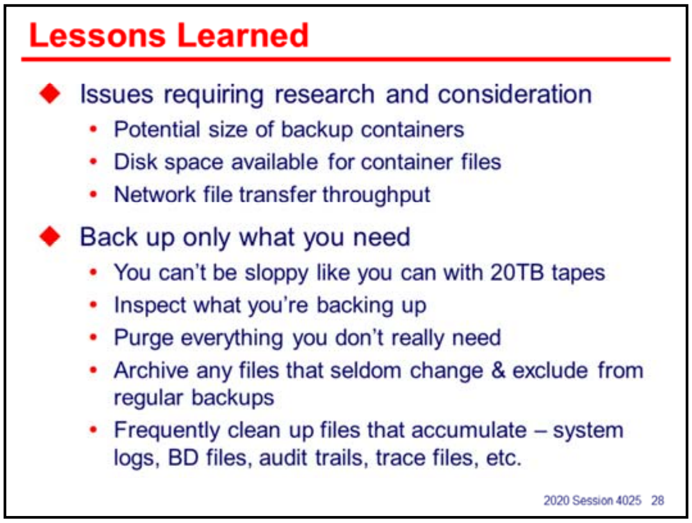
WRAP_*yyyymmdd_familyname_hhmmss*

And for the data base backup container files, the convention was:

WRAP_*yyyymmdd*_ONLINE_*dbname_hhmmss*

This ordered the files first by date, then by type and family/data base, and finally by creation time. We decided this would allow us to find the files we needed most quickly, and also aid in disposing of old files once they were no longer needed.

The physical backup drives also need to be labeled. Our high-tech solution was a slip of paper wrapped around the drive and held in place by a rubber band.



Lessons Learned

- Issues requiring research and consideration
  - Potential size of backup containers
  - Disk space available for container files
  - Network file transfer throughput
- Back up only what you need
  - You can't be sloppy like you can with 20TB tapes
  - Inspect what you're backing up
  - Purge everything you don't really need
  - Archive any files that seldom change & exclude from regular backups
  - Frequently clean up files that accumulate – system logs, BD files, audit trails, trace files, etc.

2020 Session 4025 28

We learned a number of lessons during the project, which I will try to summarize over the next two slides.

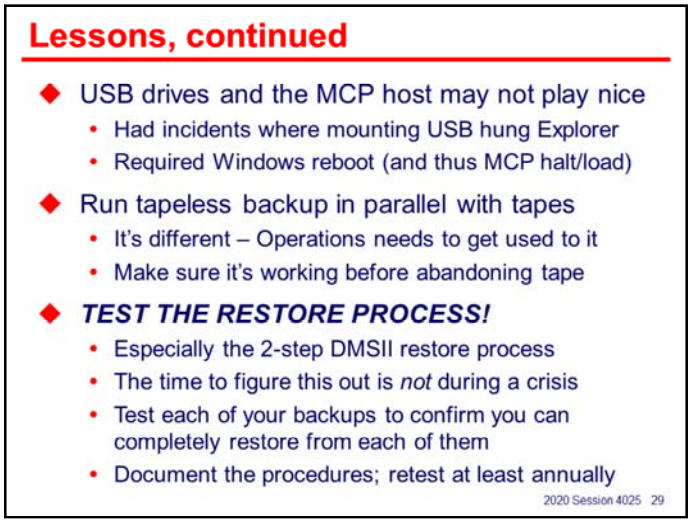
First, and most importantly, you need you understand your environment and how much you need to be backing up. This will probably require some research and analysis, but from knowing which files and directories you will be backing up, you can compute the total number of sectors and convert that to bytes, allowing perhaps 10% overhead for disk headers and other WRAP metadata, to estimate the size of your container files. From that you can evaluate whether you have the disk space available to create the necessary container files. This calculation will also help you estimate network transfer time, if that is the method you choose to transport the backups off site.

Second, you need to understand what files in your environment need to be backed up, and back up only those. The customer for this project had gotten a little sloppy over the years in cleaning up old, dead files. With the large capacities of modern disk drives, the large capacities of LTO media, and the practice of running unattended backups at night, they didn't realize how much file cruft they had accumulated. When converting to WRAP, we had to embark on a rather large file cleanup campaign in order to reduce both the size of the container files and the network transfer times to something that was workable.

This campaign raised issues with which the customer had to struggle a bit. If you have files that you are not backing up, then why are you keeping those files? If you have large numbers of files that you are keeping on the system for archival purposes and that never change, do you need to be backing them up all of the time? A more appropriate solution may be to archive these files to some stable medium, keep a couple of copies in different off-site locations, and exclude those files from the regular backups.

Most of the cruft we had to clean up was the result of files that tend to accumulate on MCP systems – system logs, printer backup BD files, DMSII audit trail files, and miscellaneous trace files. Their applications were also generating some files with date-stamped file names, but there was no established mechanism for getting rid of these files after their usefulness had ended.

One byproduct of this project was to establish such a mechanism for automatically scanning specified directories and removing files that had not been accessed in a specified number of days. We wrote a utility program to do this, and simply ran it from a WFL job about once a week. The data base backup jobs also ran this utility to get rid of old audit trail files.



Lessons, continued

- USB drives and the MCP host may not play nice
  - Had incidents where mounting USB hung Explorer
  - Required Windows reboot (and thus MCP halt/load)
- Run tapeless backup in parallel with tapes
  - It's different – Operations needs to get used to it
  - Make sure it's working before abandoning tape
- ***TEST THE RESTORE PROCESS!***
  - Especially the 2-step DMSII restore process
  - The time to figure this out is *not* during a crisis
  - Test each of your backups to confirm you can completely restore from each of them
  - Document the procedures; retest at least annually

2020 Session 4025 29

Another lesson was that USB drives and the MCP host system did not always play nice. We had a couple of incidents where mounting one of the USB backup drives on the Windows side of the Libra 460 caused Windows Explorer to hang. The customer eventually had to reboot Windows, which of course also forced an MCP halt/load. It's not clear what was causing this, other than the server was running Windows 2008R2, which is now quite old and may not properly support some modern USB devices.

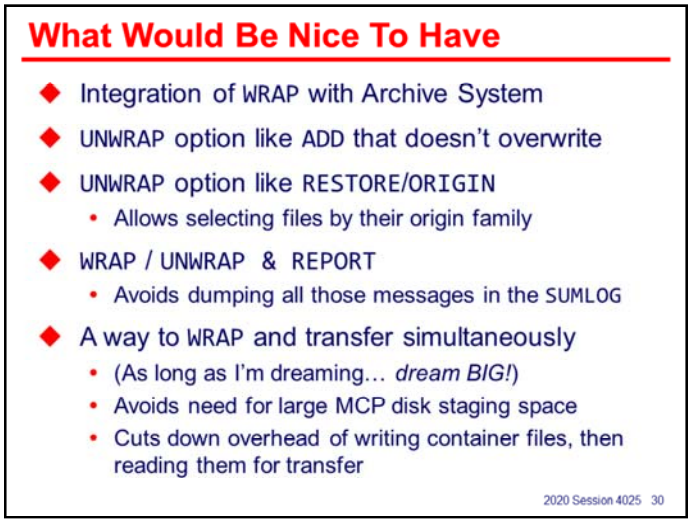
One very important thing to do when changing your backup scheme is to not get rid of the old backup mechanism until you are really sure that the new one is working properly and reliably. This is especially important when transitioning from magnetic tape backups to WRAP. It's a completely different animal, and your operations staff will need some time to get used to it. Make sure the jobs are working properly, and that the container files are getting transported properly to safe off-site storage.

Finally, and I cannot emphasize this enough, **test the restore process for the new mechanism**. Remember that backing up your files is not the goal – the goal is to be able to restore your files, your data bases, and perhaps your complete system.

Restoring from container files always requires at least one additional step – moving the container files from whatever off-site storage scheme you are using to the MCP file system. You need to do that before you can run UNWRAP to restore the necessary files.

Restoring DMSII data bases requires at least one additional step beyond that – you need to extract the DMUTILITY streamfile dump and audit trail files out of the backup container file before you can run a DMSII rebuild task. Just as with backing up DMSII data bases, you are going to need to have sufficient disk space available to stage the container and streamdump files before you begin the restore process.

The time to be figuring out how to do all of this is not when you are in the middle of a crisis. You need to have the procedures written and the necessary WFL jobs developed in advance. Then you need to test those to make sure they work properly and that operation staff are sufficiently familiar with them. Those tests should be repeated at least annually, and after upgrading the MCP version or moving to a new system.



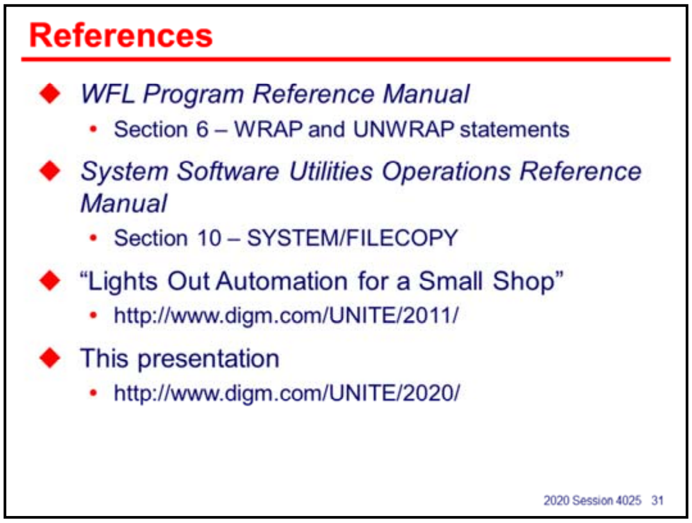
What Would Be Nice To Have

- Integration of WRAP with Archive System
- UNWRAP option like ADD that doesn't overwrite
- UNWRAP option like RESTORE/ORIGIN
  - Allows selecting files by their origin family
- WRAP / UNWRAP & REPORT
  - Avoids dumping all those messages in the SUMLOG
- A way to WRAP and transfer simultaneously
  - (As long as I'm dreaming… *dream BIG!*)
  - Avoids need for large MCP disk staging space
  - Cuts down overhead of writing container files, then reading them for transfer

2020 Session 4025 30

Using WRAP for MCP system backups has its plusses and minuses. It is definitely a viable method for backing up a system if you have sufficient disk space available and a reasonable way to get the backups off site. While working on my customer's project, I thought of a few areas where WRAP could be improved.

1. WRAP should be integrated with the Archive System. This would allow a much better and more reliable way to do differential and incremental backups.
2. UNWRAP really needs the equivalent of the Library/Maintenance ADD command – something that will restore missing files but not overwrite resident ones of the same name.
3. UNWRAP also needs the equivalent of the Library/Maintenance RESTORE command and the ORIGIN clause in file lists. It is easy to combine files from multiple families into one container file. It is not so easy to extract those files from such a container and restore them to their original families, especially if some of those families have files and directories with the same names.
4. WRAP and UNWRAP should both support the equivalent of the "`& REPORT`" option for COPY commands. This generates the list of files copied as a printer-backup file instead of dumping lots of messages into the system SUMLOG. [Note: I learned during the conference that this feature is currently scheduled for release 62.0]
5. Finally, as long as I'm dreaming, I'm gonna *Dream **BIG!*** It would be very nice to be able to have WRAP and UNWRAP work directly with container files on remote systems or storage devices. This would avoid the current problem of requiring extra disk space on the MCP system to stage the container files, as well as the extra overhead involved in writing the container file to the MCP file system and then having to read it back for transfer it for off-site storage. Ideally, we should be able to WRAP and transfer out at the same time, as well as transfer in and UNWRAP at the same time. This is a non-trivial problem, and different solutions may be needed depending on the type of remote system, device, or service that will store the container files.



References

- *WFL Program Reference Manual*
  - Section 6 – WRAP and UNWRAP statements
- *System Software Utilities Operations Reference Manual*
  - Section 10 – SYSTEM/FILECOPY
- “Lights Out Automation for a Small Shop”
  - http://www.digm.com/UNITE/2011/
- This presentation
  - http://www.digm.com/UNITE/2020/

2020 Session 4025 31

There are a few references you should be aware of when considering WRAP for your backup mechanism.

- The WRAP and UNWRAP commands are in Section 6 of the *Work Flow Language (WFL) Program Reference Manual*.
- The `SYSTEM/FILECOPY` utility is documented in Section 10 of the *System Software Utilities Operations Reference Manual*.
- Portions of this talk are based on a UNITE conference presentation I gave in 2011, which described a tapeless backup scheme I implemented for a different customer.
- At the link for this presentation you can find sample source code for the utility programs and WFL jobs used with this project.

END

Going Tapeless