eXtended Tape Facility

How many cartridge units **do you need** to write e.g. 10 tape files **simultaneously** ?

Situation today

- o Cartridges with high capacity are rarely completely filled
- o Highcapacity cartridges are expensive
- o many cartridges need expensive storage space
- o tape mounts require operator actions or a robot system
- o per tape data set a unit is needed
- o there are restrictions in multiprogramming if there are not sufficient cartridge units
- o the batch-window for processing gets shorter and shorter
- o blocking of tape data sets if often very low
- o many tape I/Os cost a lot of cpu time

Mode of Operation

...

XTF ist started after VSE IPL in a partition in which it will be permanently running. Different options, such as blocksize of container files (=physical files written by XTF), number of blocks, number of virtual cartridge units, etc. my be specified at startup time.

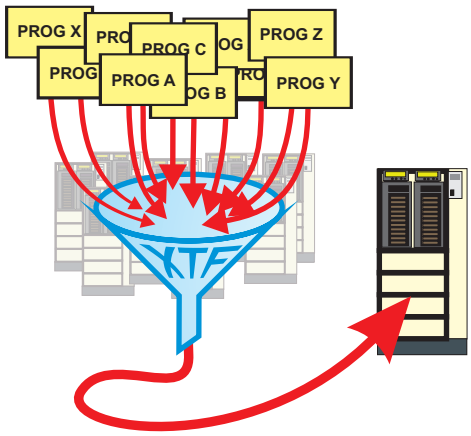
When XTF recognizes an I/O to a virtual tape unit, the I/O is simulated by XTF.

While writing to a virtual cartridge (on a virtual unit), those blocks are captured - possibly together with blocks, written to other virtual units - and are written to a real cartridge, without storing the data to an intermediate dasd.

When a virtual cartridge is to be read, XTF opens the real file (container file), in which the virtual cartridge was stored and passes only the appropriate data.

Additionally, there is an option to store a virtual cartridge - either while writing or explicitly after the creation - in a data space (a dataspace is virtual storage maintained by the operating system). At the time the virtual cartridge is to be read, XTF can present the data directly out of the data space. This allows also the same cartridge to be read by multiple programms simultaneously.

The number of cartridges to be held concurrently in the data space is only limited by the size of the data space. This size may be defined by the user.



...*one* and XTF !

Solution by XTF

- o XTF simulates under VSE up to 64 cartridge units
- o XTF simulates any number of virtual cartridges on one or multiple cartridges
- o XTF combines the data output of multiple virtual cartridge units to one or more real units
- o XTF can put output data sets additionally into a data space. This allows the data set later to be read without the need of a real unit.
- o XTF can read virtual cartridges into a data space in advance. This allows the virtual cartridge to be read by many programs - even simultaneously.
- o XTF has an own buffer management with best blocking and multiple buffers to avoid I/Os.
- o XTF administrates a central catalog, that holds information about which virtual cartridge is stored in which real tape dataset (container file)
- o XTF allows the simultaneously catalog access by any

Pfeilschifter GmbH Softwareentwicklung Tel. 09128-4852

Synergie with Tape Management Systems

XTF only knows mount and demount commands for virtual cartridges. Thus, it is in effect a "virtual" tape robot system.

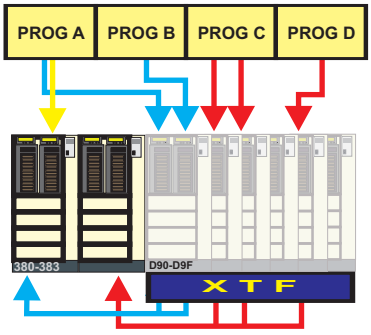
In our own tape management system BVS support for XTF is already integrated. Support for other tape management systems is achieved by their interfaces (exits) for open, close and end of job step. XTF supplies the appropriate exit.

To write an **output tape file** under control of XTF and BVS the volser field in the TLBL statement must be coded with VAULTx or AUTOx. For other tape management systems XTF becomes active by their special definitions of the tape datasets.

The first time XTF recognizes a mount for an ouput file XTF opens a real file (container file) to write the data. If applicable a real robot system is engaged.

For **input tape files** tape management systems usually look into their catalogs to determine the virtual cartridge to be used. Depending on the tape management system the mount is either given directly to XTF or the open exit becomes active. If there it is recognized, that a virtual cartridge is to be mounted, the appropriate mount command is passed to XTF.

When XTF gets control for an input mount, it checks if the virtual cartridge is hold in a data space. If not, XTF opens the real file (container file) holding the virtual cartridge to be mounted. Opening the container file is from the view of a tape management system like opening a regular tape file - if applicable a real obot system is engaged.



logical view of XTF

Advantages of XTF

Gain of a robot system or virtual tape server (VTS) as the capacity of the cartridges may be completely used

Gain of real cartridge units by combining output data sets to one unit

Gain of cartridges
- lower prime-costs
- smaller file

increased multiprogramming by relieving the bottleneck of cartridge units

Gain in performance by shortened mount processing

dramatic reduction of real tape I/O's by
- buffering the data up to 64K
- writing multiple blocks per I/O

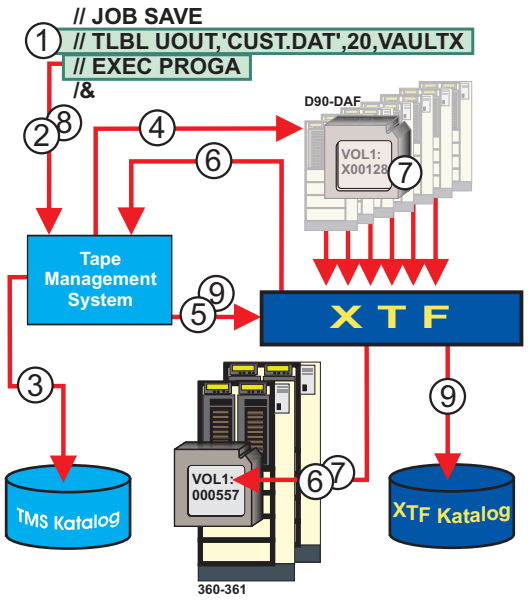
utilization of data in memory (without any program changes) by spooling of tape data sets into data spaces

multiple read of virtual cartridges via input data spaces at the same time by different programs

effortless implementation under a tape management system, such as BVS, since XTF appears as virtual robot system

user friendly, as XTF only needs 4 system commands

Output of a Tape File (under Control of BVS)



❶ BVS regards XTF as a robot system. Therefore in the TLBL statement e.g. VAULTX must be specified, to indicate that the robot system with its virtual cartridge is to be used

❷ At open of the tape file BVS gains control

❸ BVS looks for a free (virtual) cartridge in its catalog

❹ BVS looks for a free (virtual) cartridge unit

❺ the BVS robot interface sets up an appropriate VMOUNT command to XTF

❻ Only the first time XTF must open a real tape output file

❼ all write commands to the virtual unit areintercepted by XTF and written (pooled) to the real unit

❽ at close BVS gains again control

❾ the BVS robot interface sends the VDEMOUNT command to XTF. The virtual cartridge is entered into the XTF catalog. (However, the data set written to the virtual cartridge is written at normal end of job step into the BVS catalog)

Note: the XTF container file remains mounted for further pooled output and must be demounted explicitly

Pfeilschifter GmbH
Softwareentwicklung
Josef-Schlosser-Weg 2
90537 Feucht
Tel. 09128-48 52
Fax: 09128-91 64 51
email: info@pfeilschifter-gmbh.de
www.pfeilschifter-gmbh.de