Chapter 3

ITS Programming Examples

This chapter shows examples of **itsibmtape**, **itstar**, and **itsdd** in use.

3.1 Using itsibmtape

itsibmtape is likely to be the ITS application that you use most often. This section describes its use for transferring IBM data sets from the SDA to tape and from tape to the SDA.

3.1.1 SDA to Tape Transfers

In this example, a data set named **prospects.vacation.travel** resides in an SDA database named **/prospects**. This data set will be copied to four tape volumes. The tape drive system includes two stackers, both of which will be used.

The command to move this data set to the tapes is as follows:

```
% itsibmtape if=/prospects/vacation/travel of=/dev/rst0
    -fromsda label=osl,wsl dsn=prospects.vacation.travel
    vsn=PVT0001,PVT0002,PVT0003,PVT0004
    rformat=fb:31744:1024 bs=1904640 stackers=2
```

The various arguments are explained below.

Copyright © 1994 Thinking Machines Corporation

```
% itsibmtape if=/prospects/vacation/travel of=/dev/rst0
-fromsda label=osl,wsl dsn=prospects.vacation.travel
    vsn=PVT0001,PVT0002,PVT0003,PVT0004
    rformat=fb:31744:1024 bs=1904640 stackers=2
```

These arguments specify the input and output files.

```
% itsibmtape if=/prospects/vacation/travel of=/dev/rst0
    -fromsda label=osl,wsl dsn=prospects.vacation.travel
    vsn=PVT0001,PVT0002,PVT0003,PVT0004
    rformat=fb:31744:1024 bs=1904640 stackers=2
```

This specifies the direction of data movement — from the SDA to the tape drives.

```
% itsibmtape if=/prospects/vacation/travel of=/dev/rst0
    -fromsda label=osl,wsl dsn=prospects.vacation.travel
    vsn=PVT0001,PVT0002,PVT0003,PVT0004
    rformat=fb:31744:1024 bs=1904640 stackers=2
```

This option controls the type of label processing that is performed. The **wsl** entry specifies that, if no label exists, one will be created. If the label data matches, the tape will be used. If there is a label, but it does not match, the operator will be allowed to overwrite it. The **o** in **osl** indicates that the data set is being written to tape (output).

```
% itsibmtape if=/prospects/vacation/travel of=/dev/rst0
    -fromsda label=osl,wsl dsn=prospects.vacation.travel
    vsn=PVT0001,PVT0002,PVT0003,PVT0004
    rformat=fb:31744:1024 bs=1904640 stackers=2
```

The **dsn=** entry supplies the data set name, and **vsn=** specifies the serial numbers of the volumes that will receive the data set. Only the last 17 characters of the data set are recorded on the label. The volume serial numbers must be separated by commas without spaces.

```
% itsibmtape if=/prospects/vacation/travel of=/dev/rst0
    -fromsda label=osl,wsl dsn=prospects.vacation.travel
    vsn=PVT0001,PVT0002,PVT0003,PVT0004
    rformat=fb:31744:1024 bs=1904640 stackers=2
```

This option specifies that the data set is to have a fixed-block format on the tape. It will have a physical record length of 31744 and a logical

Version 7.3, April 1994
Copyright © 1994 Thinking Machines Corporation

record length of 1024. The physical record length is a constant multiple of the logical record length.

rformat must be specified for writing to tape. If this option is omitted when reading from tape, the format information is taken from the tape label.

```
% itsibmtape if=/prospects/vacation/travel of=/dev/rst0
    -fromsda label=osl,wsl dsn=prospects.vacation.travel
    vsn=PVT0001,PVT0002,PVT0003,PVT0004
    rformat=fb:31744:1024 bs=1904640 stackers=2
```

bs= specifies the buffer size that is to be allocated. To maximize performance, you should allocate as close to 50% of the available buffer memory as possible. Because **itsibmtape** double-buffers the data it transfers, no more than 50% can be allocated. The ITS tape controller board provides 4 Mbytes from its 8-Mbyte I/O buffer for **itsibmtape** transfers; the other 4 Mbytes are used for parity and other buffering.

The buffer size must also be a constant multiple of the physical record size. In this example, the multiple is 60. The buffer size should also be evenly divisible by 16 to simplify SDA-imposed alignment considerations.

```
% itsibmtape if=/prospects/vacation/travel of=/dev/rst0
    -fromsda label=osl,wsl dsn=prospects.vacation.travel
    vsn=PVT0001,PVT0002,PVT0003,PVT0004
    rformat=fb:31744:1024 bs=1904640 stackers=2
```

The **stackers=** option permits the use of multiple loaders on tape systems that are appropriately equipped. This feature enables **itsibmtape** to enhance overall performance by overlapping the rewind and unload phases of one tape with the load, seek, and read/write phases of another tape. This example uses two stackers, one of which will hold the even-numbered tapes and the other the odd-numbered tapes.

NOTE: itsibmtape assumes that, if there are additional stackers, they are sequential. For example, a second stacker is **/dev/rst1**, a third is **/dev/rst2**, etc.

Copyright © 1994 Thinking Machines Corporation

3.1.2 Tape to SDA Transfers

In this example, a database is stored on the set of tapes in variable-length records. These records can be loaded onto the SDA. The data will be read from device **/dev/rst0** and written to a file on the SDA called **/mfund/bond/xlist**. The command for performing this transfer might take the following form:

```
% itsibmtape if=/dev/rst0 of=/mfund/bond/xlist
    -tosda label=ial dsn=/bond/xlist
    vsn=BXL123,BXL888,BXL236,BXL....,BXL999
    rformat=vbs:32760:32760 bs=1965600 stackers=2
```

How do the **itsibmtape** options differ for this transfer from the previous instance?

```
% itsibmtape if=/dev/rst0 of=/mfund/bond/xlist
    -tosda label=ial dsn=/bond/xlist
    vsn=BXL123,BXL888,BXL236,BXL....,BXL999
    rformat=vbs:32760:32760 bs=1965600 stackers=2
```

This time, the tape device is specified as the source and the SDA file is specified for the output.

```
% itsibmtape if=/dev/rst0 of=/mfund/bond/xlist
    -tosda label=ial dsn=/bond/xlist
    vsn=BXL123,BXL888,BXL236,BXL....,BXL999
    rformat=vbs:32760:32760 bs=1965600 stackers=2
```

The direction option now specifies the SDA as the destination instead of the source.

```
% itsibmtape if=/dev/rst0 of=/mfund/bond.xlist
    -tosda label=ial dsn=/bond.xlist
    vsn=BXL123,BXL888,BXL236,BXL....,BXL999
    rformat=vbs:32760:32760 bs=1965600 stackers=2
```

This time ANSI labels are to be processed.

Version 7.3, April 1994
Copyright © 1994 Thinking Machines Corporation

```
% itsibmtape if=/dev/rst0 of=/mfund/bond/xlist
    -tosda label=ial dsn=/bond/xlist
    vsn=BXL123,BXL888,BXL236,BXL....,BXL999
    rformat=vbs:32760:32760 bs=1965600 stackers=2
```

These arguments have the appropriate data set name and volume serial numbers.

```
% itsibmtape if=/dev/rst0 of=/mfund/bond/xlist
    -tosda label=ial dsn=/bond/xlist
    vsn=BXL123,BXL888,BXL236,BXL....,BXL999
    rformat=vbs:32760:32760 bs=1965600 stackers=2
```

This time the data set is organized into variable-block spanned records. These records are (generally) 32760 bytes long. Because their length is variable, the logical record length is, by convention, the same as the physical record length.

rformat is optional when reading from tape. If it is omitted, the format information is taken from the tape label.

```
% itsibmtape if=/dev/rst0 of=/mfund/bond/xlist
    -tosda label=ial dsn=/bond/xlist
    vsn=BXL123,BXL888,BXL236,BXL....,BXL999
    rformat=vbs:32760:32760 bs=1965600 stackers=2
```

Here, as with the previous example, the buffer allocation request is 60 times the physical record size and is divisible by 16.

```
% itsibmtape if=/dev/rst0 of=/mfund/bond.xlist
    -tosda label=ial dsn=/bond.xlist
    vsn=BXL123,BXL888,BXL236,BXL....,BXL999
    rformat=vbs:32760:32760 bs=1965600 stackers=2
```

This transfer will also make use of two stackers.

The variable-length records employed in this example include meta-data in the form of block descriptor words (BDWs) and segment descriptor words (SDWs). See Appendix A for a discussion of variable-length record characteristics.

This meta-data is stored on the SDA along with the data of interest. The user application is responsible for locating the data of interest and by-passing the BDW and SDW content. This is done by reading a control file, which contains index information pointing to the primary data. **itsibmtape**

Copyright © 1994 Thinking Machines Corporation

creates the control file in the course of reading the data from the tape, but before it writes it to disk. The format of the control file is described by the header file, **cmt_ctl_file.h**.

3.2 Using itstar

This section describes the use of **itstar** for backing up SDA files and for loading files from tape onto the SDA.

3.2.1 SDA to Tape Transfers

In this example, two large SDA files are backed up on multiple tape volumes, using two tape devices (similar to specifying two stackers for **itsibmtape**).

```
% itstar -cv -f /dev/rst0 -f/dev/rst1 -M -V oct_bu
    /proj/sample.sets.all /proj/report.src
```

The **-c** option specifies create, while **-v** specifies verbose mode. These are standard **tar** options.

The two instances of **-f** cause the data to be written to two tape devices. An arbitrary number of **-f** instances can be specified. This multiple **-f** use is a **itstar** enhancement.

The **-M** option specifies that the data will be stored across multiple volumes.

The **-V** option allows the operator to give the backup a name, much like the **itsibmtape** label.

This example created its tape archive using absolute pathnames. It is more frequently done with relative pathnames so that they can later be extracted in any location. To create a tape archive with relative pathnames, simply create the archive from within the source directory:

```
% cd /proj
% itstar -cv -f /dev/rst0 -f/dev/rst1 -M -V oct_bu
    /proj/sample.sets.all /proj/report.src
```


Copyright © 1994 Thinking Machines Corporation

By default, **itstar** assumes its destination is a 3480-class tape device and blocks its output accordingly. Because it uses its own internal format, it is generally not necessary to reblock the data manually.

3.2.2 Tape to SDA Transfers

Extracting a tape archive with **itstar** follows the same rules as writing to tape, but in the reverse order.

```
% itstar -xv -f /dev/rst0 -f/dev/rst1 -M -V oct_bu
```

This command will extract (**-x** option) the entire backup that was created in Section 3.2.1 and restore it to its original location. Because absolute pathnames were used when the archive was created, the same pathnames will be used for the extraction process. Hence, this command does not require destination pathnames.

3.3 Using itsdd

The following example shows how to use **itsdd** for loading files from tape onto the SDA:

```
% mt -f /dev/nrst0 fsf
% itsdd if=/dev/nrst0 of=/prospects/vacation/travel
    bs=1904640 ifbs=31744 -tosda
```

The **mt** command casues **itsdd** to skip over the tape label that is at the start of the tape.

The device name, **/dev/nrst0**, is a synonym for **/dev/st0**. The **nr** portion specifies that the device will not rewind the tape when it is closed.

The **bs** option instructs **itsdd** to allocate approximately 2 Mbytes of buffer space. **ifbs** specifies an input fixed-block size of 31744 bytes.

To read a foreign tape with **itsdd** you only need to know the tape's block size. If you do not know the input block size, you can omit **ifbs**. **itsdd** will still be able to read the tape, but with some loss in performance.

-tosda specifies that the dataflow direction is from the tape to the SDA.

Copyright © 1994 Thinking Machines Corporation