

# Sequential Improvements



**Alex Kodat**  
**Sirius Software Inc.**



Sirius Software, Inc.

# Traditional Sequential File Access with User Language

- Oriented around images
  - Very convenient for fixed format records
- Open statement opens file for sequential access
- Read Image statement reads a record from the file
- Write Image statement writes a record to the file
- Close statement writes a record to the file



# Problems with Image-Oriented Sequential I/O API

- It's not O-O
  - So the cool kids will make fun of us
- No direct support for longstrings
  - Can be faked with `$lstr_get_image` and `$lstr_set_image`
  - But clunky and error prone because of blank stripping
- Clunky for character stream data
- Doesn't support 204 streams
- Can't dynamically set/retrieve file (DCB) attributes
- Impractical for multiple record type files
  - Just generally inflexible
- Can't be passed between requests (no globals)



# Introducing the Dataset Class

- An object-oriented API for accessing sequential datasets
- A Dataset object is really a cursor on a dataset
- Actually operates on Model 204 streams
  - Can be a sequential dataset (ALLOCATE, DD, DLBL or FILEDEF)
  - Can be defined via DEFINE STREAM
    - Good for having a play with streams
- Available in Sirius Mods 7.2



# Dataset Class Constructor

- Called New, of course
- Single parameter is DD or stream name
- Does not open the dataset
  - So never fails, even if DD name is undefined
- New dataset object always has state *Closed*



# DatasetState Enumeration

- Enumeration that describes the state of a dataset (duh)
- All dataset objects have a current, non-null state accessed by the State readOnly property
  - **Closed** – The initial state and state after a Close method
  - **Open** – Dataset has been opened and can be read from or written to
  - **AfterEnd** – Input dataset has read to end
  - **Full** – Output dataset has filled up
- Typically used for tests:
  - `if %dataset:state eq afterEnd then`



# RecordFormat Enumeration

- Enumeration that describes the format of a dataset (duh)
- All dataset objects have a non-null RecordFormat
  - **F** – Fixed length records, not blocked
  - **FB** – Fixed length records, not blocked
  - **V** – Variable length records, not blocked (a fiction, variable length records always blocked)
  - **VB** – Variable length records, blocked
  - **U** – Unformatted blocks (no difference between blocks and records)
- RecordFormat property be set before opening dataset
  - Usually set on open for input datasets
  - Can only be retrieved when dataset is open



# The Output Property

- Boolean
  - **True** indicates dataset is for output (writes)
  - **False** indicates dataset is for input (reads)
- Can only be set while dataset is closed
- Can be retrieved any time
- Default (initial) value is False
- Incorrect type of Read/Write operation results in request cancellation



# RecordLength and BlockSize properties

- RecordLength indicates maximum size of records to be read/written
- BlockSize indicates maximum size of blocks to be read/written
- Can be set while file is closed
- Typically set automatically for input files during Open
- Maximum values of 32767 (BSAM limitation)
  - Includes length of 4 byte BDW/RDW in V/VB files
  - So, maximum RecordLength is 32763 for V/VB files
  - And maximum length of data inrecords is 32759



# Typical Processing for Input Datasets

- Instantiate a Dataset object with New
- Open the object with Open method
  - Check State if error possible
- Read records until State becomes AfterEnd
- Close dataset with Close method
- Explicitly/implicitly discard object



# Example of Input Dataset Object

b

```
%ds    is object dataset  
%ls    is longstring
```

```
%ds = new('CCAIN')
```

```
%ds:open  
repeat forever  
    %ls = %ds:readRecord  
    if %ds:state eq afterEnd then  
        loop end  
    end if  
    printText {%ls:length} {%ls}  
end repeat
```

```
* %ds:close    - Unnecessary, end of request will close
```

```
* %ds:discard - Unnecessary, end of request will discard
```

```
end
```



# Input Dataset Miscellany

- Fixed length record returned with whatever padding they were stored with
  - Since there's no official pad character standard
  - So applications need to do their own stripping as needed
- Invalid blocks automatically detected and treated as EOF
  - Though bad blocks should be exceedingly rare
- Close/Open pair “rewinds” the dataset
- Carriage control characters simply returned as data
- Result of ReadRecord should usually be assigned to (longstring) variable
  - Because State should be checked **after** read
  - Another read when State=AfterEnd results in request cancellation



# Example of Output Dataset Object

b

```
%ds    is object dataset  
%i     is float
```

```
%ds = new('SEQTEST')  
%ds:output = true  
%ds:recordFormat = vb  
%ds:blocksize = 6184  
%ds:recordLength = 1024
```

```
%ds:open  
for %i from 1 to 500  
    %ds:writeRecord('All work and no play make Jack a dull boy...' -  
                    with %i)  
end for  
%ds:writeRecord('Red rum! Red rum! Red rum!')
```

\* The following is necessary or all the data won't be written

```
%ds:close
```

\* %ds:discard - Unnecessary, end of request will discard

```
end
```



# Output Dataset Miscellany

- Output records for fixed format are padded with blanks
- Attempt to write record longer than RecordLength results in request cancellation
- Discard is different from Close
  - Close ensures all records written out
  - Discard chucks records still in buffer and cancels active I/Os
- Attempt to write if State=Full results in request cancellation
  - So check State after every write if you're worried about this



# Reading Files FTPed to MVS (or CMS or VSE)

- Files usually (always) uploaded as RECFM V/VB with all records the same length, except the last one
- So just, concatenate the results of ReadRecord into a longstring

```
%ls    is longstring

%ds = new('PDF')

%ds:open
repeat forever
  %ls = %ls with %ds:readRecord
  if %ds:state eq afterEnd then
    loop end
  end if
end repeat
```

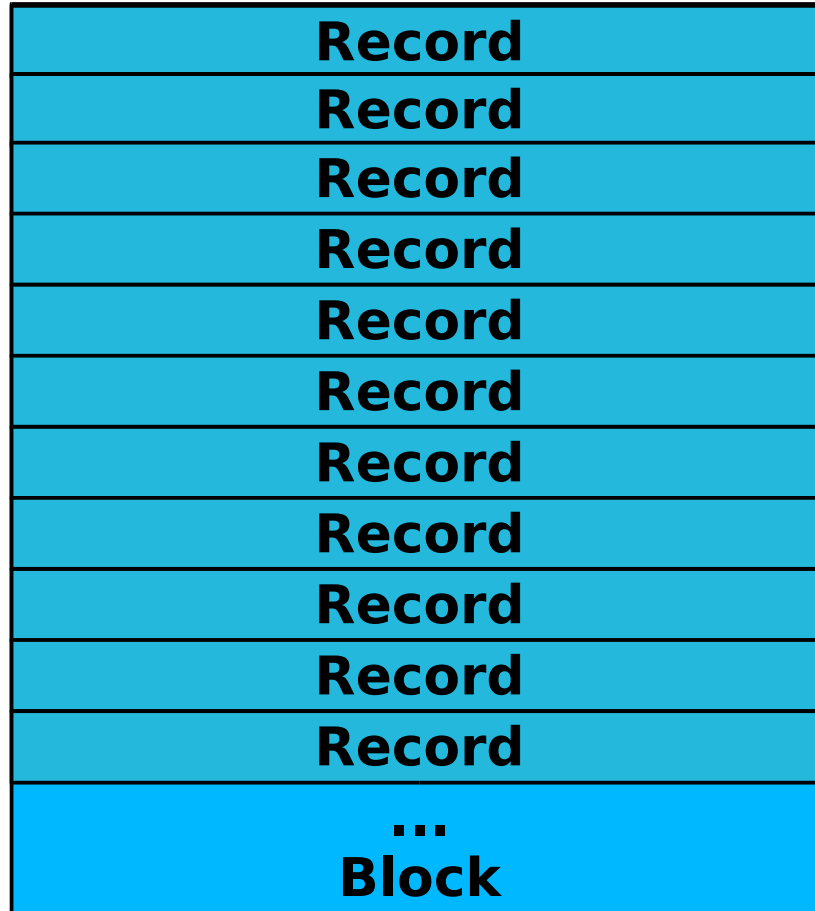


# ReadBlock and WriteBlock

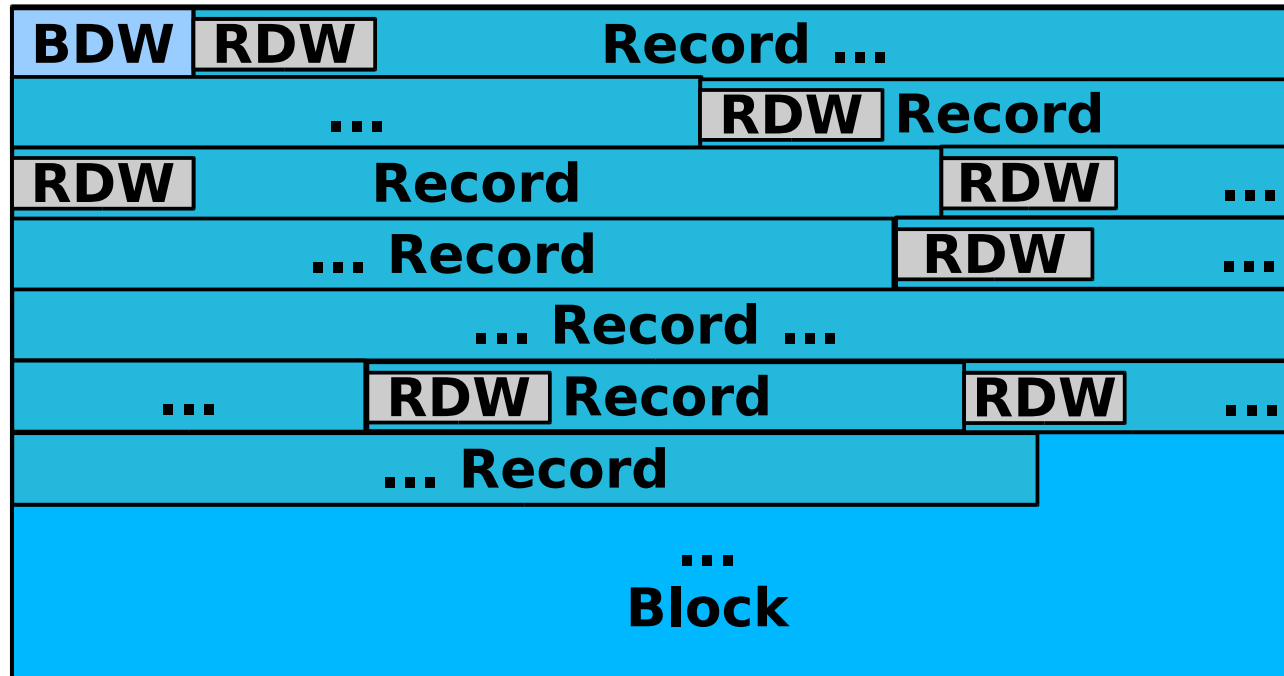
- Allows blocks of data to be written rather than records
  - Programmer responsible for concatenating fixed length strings into an FB block
  - Programmer responsible for building VB block
    - Requires binary BDWs and RDWs
- Format of blocks validated
  - Bad block format results in request cancellation
- Most likely to be useful for debugging or problem diagnosis
- For RecordFormat U, ReadBlock/WriteBlock identical to ReadRecord/WriteRecord



# Fixed Block Format



# Variable Block Format



## BDW and RDW format

**Byte 0-1** – Length of block/record, including BDW/RDW

**Byte 2-3** – Zero



# Hex Dump of Block with RecordFormat=F, RecordLength=48, and BlockSize=336

```
000000 : D396A4A2 A840A689 A38840A8 96A49940 |Lousy with your |
000010 : 839695A3 8595A340 40404040 40404040 |content          |
000020 : 40404040 40404040 40404040 40404040 |                 |
000030 : E689A388 40A68881 A340A388 85409481 |With what the ma|
000040 : 9185A2A3 89834083 81959596 A3408689 |jestic cannot fi|
000050 : 95844040 40404040 40404040 40404040 |nd              |
000060 : C9954082 A4A28995 85A2A240 968640A8 |In business of y|
000070 : 96A49940 9389A585 A2404040 40404040 |our lives       |
000080 : 40404040 40404040 40404040 40404040 |                 |
000090 : E3888540 97859983 8597A389 96956B40 |The perception,  |
0000A0 : 89A34089 A240A699 9695876B 40948993 |it is wrong, mil|
0000B0 : 85408186 A3859940 94899385 40404040 |e after mile     |
0000C0 : E3888540 97888195 A3969440 A381A2A3 |The phantom tast|
0000D0 : 85408499 89959289 958740A6 89958540 |e drinking wine  |
0000E0 : 86999694 40A896A4 99408885 8593A240 |from your heels  |
0000F0 : E6854088 81A58540 81999989 A5858440 |We have arrived  |
000100 : A3969640 9381A385 40A39640 979381A8 |too late to play |
000110 : 40404040 40404040 40404040 40404040 |                 |
000120 : A3888540 82938585 84899587 40888581 |the bleeding hea|
000130 : 99A340A2 8896A640 40404040 40404040 |rt show         |
000140 : 40404040 40404040 40404040 40404040 |                 |
```



# Hex Dump of Block with RecordFormat=VB, RecordLength=48, and BlockSize=336

```
000000 : 01040000 001B0000 D396A4A2 A840A689 |??  ?  Lousy wi|
000010 : A38840A8 96A49940 839695A3 8595A300 |th your content |
000020 : 260000E6 89A38840 A68881A3 40A38885 |?  With what the|
000030 : 40948191 85A2A389 83408381 959596A3 | majestic cannot|
000040 : 40868995 84001D00 00C99540 82A4A289 | find ?  In busi|
000050 : 9585A2A2 40968640 A896A499 409389A5 |ness of your liv|
000060 : 85A20030 0000E388 85409785 99838597 |es ?  The percep|
000070 : A3899695 6B4089A3 4089A240 A6999695 |tion, it is wron|
000080 : 876B4094 89938540 8186A385 99409489 |g, mile after mi|
000090 : 93850033 0000E388 85409788 8195A396 |le ?  The phanto|
0000A0 : 9440A381 A2A38540 84998995 92899587 |m taste drinking|
0000B0 : 40A68995 85408699 969440A8 96A49940 | wine from your |
0000C0 : 88858593 A2002400 00E68540 8881A585 |heels ?  We have|
0000D0 : 40819999 89A58584 40A39696 409381A3 | arrived too lat|
0000E0 : 8540A396 40979381 A8001B00 00A38885 |e to play ?  the|
0000F0 : 40829385 85848995 87408885 8199A340 | bleeding heart |
000100 : A28896A6 40404040 40404040 40404040 |show |
```



# Dataset Class Performance

- Do ReadRecord/WriteRecord rather than ReadBlock/WriteBlock to let assembler code manage records
- NumberOfBuffers property can be used to allow overlap of I/O and processing
  - Can only be set before dataset opened
  - NumberOfBuffers=1 → I/O thread-level synchronous
  - NumberOfBuffers=2 → I/O thread-level asynchronous
  - NumberOfBuffers>2 → I/O thread-level asynchronous, BSAM chained scheduling used for faster I/O
  - Of course, each buffer uses real memory



# Other Uses for Dataset Objects

- Passing to SirTune report writer (SirtuneReport method)
- Passing to SirAud report writer (Sirius Mods 7.3)
- Passing to SirScan (some day)
  - For using SirScan to look at old journals



# Conclusions

- Dataset class provides a nice new object-oriented way of accessing sequential datasets
  - Much more flexible than current image-based capability
  - Very easy to use
- Especially useful for sysprogs
- CMS users beware
  - Sequential dataset support for 204 under CMS is a bit flaky



# Any Comments or Questions?



Sirius Software, Inc.