What's New in DATA-MINER Version 7.1

DATA-MINER version 7.1 is currently available for z/VSE and z/OS.

DB2 Support

DATA-MINER can access DB2 for z/OS and DB2 Server for z/VSE. DATA-MINER uses a dynamic SQL interface, so your script does not require a precompile or BIND. All DATA-MINER input functions are supported; for example, you can create reports or extract data from your DB2 tables.

Arrays and Indexes

Record fields and user variables now support arrays. This is accomplished with the optional `OCCURS` argument when you define record field names and user variables. There are multiple ways to access these arrays by index or subscripts.

System DATE variables

Several DATE variables have been added to provide access the current date. Refer to system variables `SYSDATE`, `SYSDATE-LONG`, and `SYSDATE-JULIAN`.

System PARM variables

Information can now be passed into DATA-MINER by the `PARM` parameter of the `EXEC` statement. Refer to system variables `PARM-DATA`, `PARM-LENGTH` and `PARM-REGISTER`.

REPORT titles

The DATA-MINER Report Writer now allows titles 132 characters long. Field names and/or user variables can also be placed in the title lines.

SORT Command

DATA-MINER can sort a file into any order and use any field types recognized by your SORT program. The sort can use multiple keys in any combination of ascending and descending. All fields and files used by the DATA-MINER `SORT` command are defined the same way as any other field and can be used in any other DATA-MINER commands executed at the same time as the sort. Besides simply sorting a file, you can add extra processing to your sorting script by using the timing commands.

Timing Commands

DATA-MINER allows you to say when you want a command or set of commands to be executed using the timing commands. Timing commands are meant to be used mainly in conjunction with the `SORT` command but can also be used with any DATA-MINER script. They save you setting up switches, creating “first time” code, and so on:-

BEFORE INPUT Timing Command

Commands in this timing are executed before the main file is read. The most common use is to execute “first time” commands. This may be useful, for example, for reading a control record from a file before processing the main file.

DURING INPUT Timing Command

Commands in this timing are executed for every record read from the main file. This can be used to accumulate totals, to make decisions on whether to keep or skip records, and so on. `DURING INPUT` and `DURING OUTPUT` commands are executed at the same time in a non-SORT script. In a SORT script, `DURING INPUT` commands are run on the records from the unsorted input file. If you are familiar with SORT exits, this is similar to E15 processing or the COBOL `INPUT-PROCEDURE`.clause.

DURING OUTPUT Timing Command

Commands in this timing are executed for every record read from the main file. This is the default timing and is used to specify commands to be executed during the output stage of a script. In a SORT script, `DURING OUTPUT` commands are run on the records being written to the sorted output file. If you are familiar with SORT exits, this is similar to E35 processing or the COBOL `OUTPUT-PROCEDURE`.clause. If you have deleted a record `DURING INPUT`, that record is not processed by `DURING OUTPUT`.

AFTER OUTPUT Timing Command

This is the timing where you set up end-of-job commands. For example, you could print totals, update a control file, and so on.

TRACE Command

`TRACE` lets you print trace messages to show what commands your script has executed. `TRACE` can be turned on and off during the execution of your script, allowing you, for example, to trace only for record 12345678.

Procedures

Procedures or subroutines let you create sections of your script that can be executed from several places in the script without having to enter the commands multiple times. They also let you write a script where the general flow is at the top of the script—read a record, do some calculations, print a report—and the details are in procedures at the bottom of the script.

Pre-Start Procedure

A new parameter, `BEGIN`, is available on the `AUTO` command to name a `PROC` that is to be performed before the first record of the main input file is read. If there is no main input file, this procedure is run at the start of the script.

End-of-File Procedure

A new parameter, EOF, is available on the AUTO command to name a PROC that is to be performed when end-of-file is reached on the main input file.

Support for More Field Types

The VAR field type allows the use of variable length character fields of up to 32,767 bytes. DATA-MINER automatically adjusts the length of the field when a new value is placed in the field.

STATS Command and Statistics

STATS tells DATA-MINER where you want the execution statistics printed if at all.

The format is

```
     STATS { Printer | Console | Both | None }
```

Only the first character of the parameter need be entered. “Printer” turns off “Console,” and vice versa.

File I/O statistics displayed at end-of-job now include extra information such as the file ID from the DLBL or DD statement (for example, ACCOUNT) and the file name (for example XYZCO.ACCOUNTS.MASTER). Statistics are broken down by the kind of I/O done for each file

German Support

Most of the DATA-MINER commands can now be written in German. German and English can be freely mixed.

HIDE Command

HIDE gives you a useful way to hide the contents of a field that you do not want to be in an output file. For example, you may want to create a test data file from production data but hide customer addresses and balances. HIDE fills character fields with lowercase xxx and numeric fields with zero.

CLOAK Command

CLOAK replaces the contents of a field with a random value of the same type. CLOAK does not try to make sure that the cloaked value makes sense, but it does make sure that any blanks in the original field are blanks in the cloaked version. So “JOE BLOGGS” may end up as “XRT FBRKKW.” A different cloaking algorithm is used every time a field is cloaked, but CLOAK should not be regarded as being as secure as fully encrypting a record. Numeric fields are replaced with a random number of the same field type.

CALL Command

`CALL` lets you call a phase or load module from your script, perhaps to do something that DATA-MINER is not equipped to do or to do some processing using a program that you have already written in another language. Any number of parameters can be passed to the called program. Data can be sent to the called program, and the called program can return values in the data fields passed to it.

VSAM Performance

VSAM performance has been improved, especially for ESDS output

CPU Performance

CPU utilization has been improved in most commands