Welcome to Speed Demon

Welcome to version 4.4 of Speed Demon. For improved performance, call Speed Demon in your programs instead of DBGET mode-2 to read serially through a large dataset.

There are a few restrictions to using Speed Demon:

1. The database must not be open in mode-3 or mode-7.
2. You must have read access to all of the fields in the dataset.
3. The field list cannot be the "same" list (i.e., no *-list).
4. The database must be local (i.e., not on a remote machine).
5. Entries cannot be updated or deleted.
6. On MPE V (or compatibility mode on MPE/iX), the calling program must not use DB-14 and DB-32 of the data stack.

MPE V

On MPE V, you install Speed Demon in the System SL (see *Installing Speed Demon* for instructions). Speed Demon/V should be about five times faster than DBGET mode-2.

MPE/iX

On MPE/iX, you run your native mode programs with an XL file: DemonXL.Pub.Robelle. Speed Demon/iX can be slightly slower than DBGET mode-2 or it can run as much as eight times faster. We are not certain why there is such a range of performance values. We would be happy to see performance results from users.

Conditions of Use

Speed Demon is included as part of Suprtool. Any licensed Suprtool users can use Speed Demon in their own application software. Speed Demon cannot be included in software being distributed to any sites that are not licensed to use Suprtool. Application software developers who would be interested in integrating Speed Demon into their products should contact Robelle Solutions Technology Inc.

Speed Demon vs. Suprtool

Suprtool is a program while Speed Demon is a set of callable routines. Application programs can *invoke* Suprtool to extract, select, and sort records from an IMAGE dataset, producing a disc file as output. Application programs can *call* Speed Demon to serially return the individual records of the same IMAGE dataset, one record at a time.

While Speed Demon lacks Suprtool's ability to extract fields or select and sort records, it has the advantage that it delivers the records directly into your program. Speed Demon/V is about 50% slower than Suprtool, but there is no need to read an output file, as with Suprtool. Speed Demon/iX runs at the same speed as Suprtool. With Speed Demon, your application program is responsible for selecting records and sorting them (e.g., using the SORT verb in COBOL).

Application programs should invoke Suprtool when they expect to select a *small percentage* of the records (i.e., when the overhead of reading the OUTPUT file will be minimized). Application programs should call Speed Demon when they expect to select a *large percentage* of the records.

Fourth-Generation Languages (4GLs)

Speed Demon will be more difficult to use from 4GLs than from a regular programming language (e.g., COBOL or Pascal). Remember: each call to Speed Demon returns a single record and you must build a loop to read and test them. Some 4GLs do a LOADPROC for every call to the Speed Demon intrinsics (e.g., Reactor from Speedware). Because LOADPROC is so slow, this eliminates any benefit from Speed Demon. We would like to hear from any users who have success using Speed Demon from a 4GL.

Documentation

The user documentation for Speed Demon assumes you have some previous experience with IMAGE, MPE, and programming. To print copies of the user manual, run the Printdoc program:

```
:run printdoc.pub.robelle;info="demon.doc.robelle"
```

Printdoc is menu driven, and is very easy to use. Printdoc asks you for information, and if you are not sure of your answer, you can ask for help by typing a question mark (?). Printdoc supports all types of LaserJet printers and regular line printers.

Caveats for Privileged Mode Users

Speed Demon/V uses split-stack mode. This should not concern any normal user, but there are two consequences for privileged mode users:

1. You cannot call Speed Demon/V from split-stack mode.

2. Do not enable a privileged mode Control-Y trap handler while calling Speed Demon/V.

SPDEPREFETCH JCW

Speed Demon can read data directly from disc into memory using Multi-Rec NOBUF reads. However, Speed Demon is often slowed down on MPE/iX while waiting for the file system to satisfy its read requests. Using prefetch on MPE/iX, Speed Demon is able to increase its throughput by instructing MPE/iX's memory manager to read the data from disc to memory ahead of time. This way, when Speed Demon needs the data it is already in memory and Speed Demon doesn't have to wait.

The SPDEPREFETCH JCW tells the memory manager how far ahead of Speed Demon it should fetch data. Setting this number too low won't give the memory manager sufficient time to get the data into memory before Speed Demon needs it. Setting the number too high may mean that on a busy system the data will be overwritten by something else before Speed Demon gets a chance to use it.

When reading an input dataset, Speed Demon/iX prefetches twice the buffer length of data from disc to memory. This default value of 2 works well, even on small machines. If Speed Demon is running stand-alone on a fast CPU with a lot of memory, you may wish to increase the prefetch amount. The maximum value for SPDEPREFETCH is 5 (i.e., five times the buffer length). If you don't want Speed Demon to prefetch, you can specify SPDEPREFETCH 0. This may be required when Speed Demon is operating with third-party software tools that intercept all file system calls (e.g., Netbase from Quest Software).

The SPDEPREFETCH JCW is checked by Speed Demon/V, but its value has no effect.

Installing Speed Demon

There are two versions of Speed Demon. The Speed Demon/V version is for MPE/V systems and for compatibility-mode programs on MPE/iX. Speed Demon/V runs in compatibility mode and must be installed in the system SL. The Speed Demon/iX version is for native-mode programs on MPE/iX. Speed Demon/iX runs in native mode and is accessed via an XL file.

Depending on what operating system you are on (MPE/V or MPE/iX) and what kind of programs will be calling Speed Demon (compatibility mode, native mode or both), you may end up installing either, both, or neither version. Read all the instructions carefully.

Before using either version of Speed Demon, you must build or upgrade the Robelle account and restore all files as described in the *Installing Suprtool* chapter of the Suprtool User Manual.

System SL Installation

This procedure is required for MPE/V systems, and for compatibility mode programs on MPE/iX.

The Speed Demon/V routines are distributed in the file DemonUSL.Pub.Robelle. The job stream Demon.SuprJob.Robelle installs the Speed Demon/V intrinsics in the System SL. This is the only method for installing Speed Demon/V. You also use this job stream to update Speed Demon/V when you receive a new version, or to re-install the interface after a MIT update from HP. You will need a small tape for a new cold load tape to contain the Speed Demon/V segments.

Warning: You must have created the Robelle account and restored all files as described in the installation chapter of the Suprtool User Manual. To install Speed Demon/V into the System SL, follow these steps:

Steps

1. Ensure that no one will use Speed Demon/V until the installation is complete. No one can be running a program which uses Speed Demon/V. Stop all jobs and send an operator warning.

```
:showjob
:warn @;please stop for 20 minutes
:abortjob #snnn
```

2. Stream the installation job. If MPE prompts for passwords, supply them.

```
:stream demon.suprjob.robelle
```

3. Speed Demon/V uses the SEGMENTER to add the segments into SL.Pub.Sys. It then requests a tape ("COLDLOAD") to create a new cold load tape containing MPE plus the Speed Demon/V intrinsics. Mount a tape with a write ring and :REPLY. Save this tape and use it for any future cold loads.

   If you're installing to an MPE/iX machine, the job does not create a cold load tape. You must create a system load tape manually.

4. If everything goes well, Speed Demon/V prints a final message on the console.

5. Please save the job listing for future reference.

Speed Demon/V is now installed and you should be able to use it in your application programs.

User XL Installation

This procedure is for native mode programs on MPE/iX, and is optional.

You normally access Speed Demon/iX by specifying XL="DemonXL.Pub.Robelle" when running your program. Therefore, there may be no need to perform any installation steps for Speed Demon/iX.

The advantage of leaving Speed Demon/iX in the Robelle account and always pointing to the XL when you run your programs, is that when you receive an upgrade of the Robelle account files, Speed Demon/iX is automatically upgraded with no effort on your part. The disadvantage is that you must remember to always put the XL= statement on all Run commands for your programs that use Speed Demon/iX. If you don't want to change your Run commands, you can copy Speed Demon/iX into your own XL that is already being searched.

Installing Speed Demon/iX in your own XL requires that both the account and the group where the XL resides have privileged mode capability. Calling programs do not have to have privileged mode capability. The XL can reside anywhere on your system, it does not have to be in the same account or the same group as your XL file. Here are the commands to install Speed Demon/iX in your own XL:

```
:hello user.acct
:linkedit
>xl xl                {we assume the XL already exists}
>copyxl from=demonxl.pub.robelle; replace
>exit
```

The Replace option is not in all versions of Linkedit. If your version does not allow Replace, and you already have Speed Demon in your XL file, you will need to manually purge the existing modules before copying the new one.

Demonxl can successfully be installed into the System XL file, but this is not recommended by HP. Demonxl can be combined into an XL file with other Robelle XL files, except Qcompxl (a part of Qedit).

Alternatively you can specify the XL for a program to use with the Altprog and Link commands in Linkedit. To add an XL to a program that you are building with the Link command, you can specify the following from within Linkedit:

```
:linkedit
HP Link Editor/iX (HP30315A.06.14) Copyright Hewlett-Packard Co
LinkEd> link from=main; to=myprog; xl=demonxl.pub.robelle
```

To alter an existing program to use the Demonxl, you can use the Altprog command from within Linkedit:

```
LinkEd>altprog myprog; xl=demonxl.pub.robelle
```

Please see the Linkedit manual for details about the Link and Altprog commands in Linkedit.

Group or Pub SL Installation

This procedure is for MPE/V systems and for compatibility mode programs on MPE/iX. It is not the recommended or supported procedure. See the Speed Demon/V instructions in the previous section.

We only support Speed Demon/V when it is installed in the System SL. These are the reasons for this policy:

1. Speed Demon/V requires privileged mode. When Speed Demon/V is installed in the System SL, it can obtain privileged mode for itself while guaranteeing that privileged mode is not available to the calling program. If Speed Demon/V is installed in a Pub or Group SL, the calling program is installed in a group and account with privileged mode.

2. Some third-party software intercepts all IMAGE and file system calls (e.g., DBOPEN and FOPEN). Writing intercept calls is straightforward from a Pub or Group SL, but very difficult inside the System SL. Emulating all IMAGE and file system calls correctly is hard, and can introduce problems for Speed Demon/V. By installing Speed Demon/V in the System SL, we ensure that Speed Demon/V is calling the real IMAGE and file system intrinsics.

Netbase
Pub SL
Group SL

Netbase

Netbase is a product from Quest software that intercepts IMAGE and File System calls. Netbase allows any database or file to be automatically directed to a remote machine. Netbase does not support System SL installation. For this reason, the only way to make Speed Demon/V and Netbase work together is to install Speed Demon/V in a Pub SL along with the Netbase software.

Pub SL

Installing Speed Demon/V in a Pub SL requires that both the account and the Pub group have privileged mode capability. The calling program must be installed in the same account (it does not have to be in the Pub group) and it must be run with Lib=P. Here are the commands to modify an existing account called Dev.

Example
Capabilities

```
:hello manager.sys
 :altacct dev;cap=...,pm
 :hello mgr.dev,pub
 :altgroup pub;cap=...,pm
 :segmenter
 -buildusl $newpass,400,8
 -auxusl demonusl.pub.robelle
 -copy segment,suprrobelle0
 -copy segment,suprrobelle1
 -buildsl sl,400,8
 -addsl suprrobelle0
 -addsl suprrobelle1
 -exit
```

The previous example shows the capability list as "cap=...,pm". You must fill in the "..." by obtaining the existing capabilities for the Dev account and the Pub group. On MPE/iX, you can use the :listacct and :listgroup commands. On other versions of MPE, you will need to :run Listdir5.Pub.Sys.

Group SL

The steps for installing Speed Demon/V in a group SL are the same as for the Pub SL, except that you use the actual group name instead of the Pub group. Your user programs must be installed in the same group as the SL and they must be run with Lib=G.

Accessing Speed Demon

Speed Demon is a set of routines, or intrinsics, that you call from your program. The Speed Demon/V intrinsics are installed in the system SL. The Speed Demon/iX intrinsics are installed in an XL. You access Speed Demon by calling these intrinsics, just as you would the IMAGE intrinsics. There are three primary intrinsics:

SPDEDBINIT. Initializes the Speed Demon environment and decides what dataset to read.

SPDEDBSCAN. Replaces calls to DBGET mode-2 and returns the same condition word values (e.g., 0 for okay, 11 for end-of-file).

SPDEDBSHUT. Cleans up the Speed Demon environment. Before you can call SPDEDBINIT for another dataset, you must call SPDEDBSHUT.

Dbget
Speed Demon
Spdedbinit
Errors
MPE/iX

Original Program Using DBGET

Suppose that your program scans the d-sales dataset to produce a report of all d-sales records with a sales-total greater than $10,000.00. Currently, your COBOL program looks like this:

Example

```
05-report-control          section.

     move false-value to end-of-d-sales-flag.

     perform 10-read-and-report
        thru 10-read-and-report-exit
       until end-of-d-sales.

 05-report-control-exit.  exit.

 10-read-and-report         section.

     perform 10-10-get-d-sales.

     if not end-of-d-sales then
         if dsa-sales-total > 10000 then
             perform 20-report-d-sales
                thru 20-report-d-sales-exit.

     go to 10-read-and-report-exit.

 10-10-get-d-sales.
     call "DBGET" using db-base
                        db-set-d-sales
                        db-mode2
                        db-status-area
                        db-all-list
                        db-buffer-d-sales
                        db-dummy-arg.
     if db-end-of-file then
         move true-value to end-of-d-sales-flag
     else
     if not db-stat-ok then
         perform 99-image-fatal-error.

 10-read-and-report-exit.  exit.
```

Original Program Using Speed Demon

You would modify the original code to use Speed Demon to serially read through the d-sales dataset. The original report-control section had no initializing paragraph because we assumed that the d-sales dataset was rewound to the beginning. The modified code must call SPDEDBINIT before reading the dataset. When we have finished reading the dataset, we finish off with a call to SPDEDBSHUT.

Example
How To Use

```
05-report-control          section.

     perform 05-05-init-d-sales.

     move false-value to end-of-d-sales-flag.

     perform 10-read-and-report
        thru 10-read-and-report-exit
       until end-of-d-sales.

     perform 05-20-close-d-sales.

     go to 05-report-control-exit.

 05-05-init-d-sales.
     call "SPDEDBINIT" using db-base
                             db-set-d-sales
                             db-mode1
                             db-status-area
                             spde-db-control
                             db-dummy-arg.
     if not db-stat-ok then
         perform 98-spde-error.

 05-20-close-d-sales.
     call "SPDEDBSHUT" using db-base
                             db-set-d-sales
                             db-mode1
                             db-status-area
                             db-dummy-arg.
     if not db-stat-ok then
         perform 98-spde-fatal-error.

 05-report-control-exit.  exit.
```

Read-And-Report
Field Lists

The read-and-report section has one change. Calls to DBGET are replaced with calls to SPDEDBSCAN. If SPDEDBSCAN fails, we perform a different error section. Note that the parameter list for SPDEDBSCAN is not the same as for DBGET. Our original program used the @-list, so we don't need to make any changes to the buffer-d-sales declaration.

```
10-read-and-report         section.

    perform 10-10-read-d-sales
    if not end-of-d-sales then
        if dsa-sales-total > 10000 then
            perform 20-report-d-sales
               thru 20-report-d-sales-exit.

    go to 10-read-and-report-exit.

10-10-read-d-sales.
    call "SPDEDBSCAN" using db-base
                            db-status-area
                            db-buffer-d-sales
                            db-dummy-arg.
    if db-end-of-file then
        move true-value to end-of-d-sales-flag
    else
    if not db-stat-ok then
        perform 98-spde-error.

10-read-and-report-exit.  exit.
```

If the original program used partial field lists, we would call SPDEDBINIT with mode-2 and pass the field list as the last parameter. For example,

```
move "CUST-ACCOUNT,PRODUCT-NO,PRODUCT-PRICE,PURCH-DATE;"
   to db-list-d-sales.
call "SPDEDBINIT" using db-base
                        db-set-d-sales
                        db-mode2
                        db-status-area
                        spde-db-control
                        db-list-d-sales.
if not db-stat-ok then
    perform 98-spde-error.
```

How to Use Speed Demon

To replace the DBGET mode-2 calls with calls to SPDEDBSCAN we did the following:

1. Called SPDEDBINIT to initialize the d-sales dataset.

2. Replaced calls to DBGET with calls to SPDEDBSCAN and modified the parameters passed to SPDEDBSCAN.

3. Performed a different error section for Speed Demon errors.

4. Called SPDEDBSHUT to close the dataset.

SPDEDBINIT and the Control Record

SPDEDBINIT requires a special control record. For COBOL, we suggest that this control record be placed in the COPYLIB and copied into programs that use Speed Demon. A common error in using Speed Demon is typing the control record incorrectly. Use the file Cobol2.Qlibsrc.Robelle instead. You can copy the control record directly into your COBOL program with the following Qedit commands (use /JOIN with EDIT/3000):

```
/add 50.1=cobol2.qlibsrc.robelle 1/20
```

Definition

The definition of the control record, with the proper initializing values is:

```
01  spde-db-control.
    05  spde-version     pic s9(4) comp value 0.
    05  spde-buffer-size pic s9(9) comp value zeros.
    05  spde-future      pic x(20) value spaces.
```

Error Handling

Programs that call IMAGE intrinsics usually have an error section for fatal IMAGE errors. These sections typically call DBEXPLAIN and abort the program. When a fatal Speed Demon error occurs, SPDEEXPLAIN should be called instead of DBEXPLAIN.

```
98-spde-fatal-error        section.

    call "SPDEEXPLAIN" using db-status-area.

    call "QUIT" using \db-status-area\.

98-spde-fatal-error-exit.  exit.
```

MPE/iX

To access Speed Demon/iX from a native mode program, you must modify the :Run command for your program to include an XL file. The following example shows how you would compile, link, and run a native mode COBOL program that calls Speed Demon/iX:

```
:cob74xl testsrc.demon
:link
:run $oldpass;xl='demonxl.pub.robelle'
```

Speed Demon Intrinsics

In this chapter we describe the Speed Demon intrinsics in alphabetic order. The parameters to each intrinsic are word arrays (just like IMAGE). You must include every parameter or pass a dummy variable as a place holder. The condition code is not returned by any Speed Demon intrinsic.

The Status Array

The status array is the same communications area that you use with the IMAGE intrinsics. If an error occurs, you should call the SPDEEXPLAIN intrinsic. The Speed Demon intrinsic does not return exactly the same status array as DBGET.

SPDEEXPLAIN Intrinsic

Prints a three-line message on $stdlist which describes the results of a Speed Demon intrinsic call based on information in the status array.

SPDEEXPLAIN *status*

Status

The twenty-byte array used to call any Speed Demon intrinsic which SPDEEXPLAIN will describe. The database and dataset parameters must not change between the call to the Speed Demon intrinsic and the call to SPDEEXPLAIN.

The Speed Demon intrinsics can return many of the standard IMAGE error numbers, but Speed Demon has many of its own. SPDEEXPLAIN will print the correct error message in either case. Because the dataset parameter is not passed to the Speed Demon intrinsic, SPDEEXPLAIN may not be able to print the name of the dataset in the second line of the message.

SPDEDBINIT Intrinsic

Initialize Speed Demon, specify what dataset to read, and what field list to use.

SPDEDBINIT *Database, Dataset, Mode, Status, Control, List*

Database

Database name as returned from DBOPEN (first two characters must be non-blank).

Dataset

Dataset name or dataset number to read. You must have full-read access to this dataset.

Mode

SPDEDBINIT can be called in any of four modes:

0: Return the version number of SPDEDBINIT. All the other parameters are ignored and Speed Demon is not initialized.

1: Initialize Speed Demon to sequentially access the dataset of the specified database.

2: Same as mode-1, but allows partial field lists.

3: Same as mode-2, but Speed Demon will read to the physical end of the dataset. Use this mode if your program expects changes to the dataset at the same time Speed Demon is reading it.

Status

If SPDEDBINIT succeeds with mode-0, the following values are returned in the status array.

1. Zero.

2. Major version number.

3. Minor version number.

4. Pre-release version number.

5-10. Zeros.

If SPDEDBINIT succeeds with mode-1, mode-2, or mode-3, the following values are returned in the status array.

1. Zero.

2. Entry length of the dataset.

3-4. Buffer size used to read the dataset.

4-5. Number of records that Speed Demon will read. For mode-3, this is always the capacity of the dataset. For mode-1 and mode-2, this is the number of entries for master datasets and the highwater mark for detail datasets.

7-10. Zeros.

Control

Special control record that must be declared and initialized as follows:

```
01  spde-db-control.
    05  spde-version     pic s9(4) comp value 0.
    05  spde-buffer-size pic s9(9) comp value zeros.
    05  spde-future      pic x(20) value spaces.
```

Normally you won't be changing this control buffer. You might want to change the spde-buffer-size. This sets the size of the file system reads and the size of the Speed Demon extra data segment. The default size is 14,336 words, but you may wish to reduce it.

List

For mode-0 or mode-1 this is a dummy parameter. For mode-2 or mode-3, this is the field list that corresponds to the buffer that will be returned by SPDEDBSCAN.

Highwater Mark

When you specify a detail dataset, Suprtool and Speed Demon read records until the highwater mark. If there is a big difference between the number of entries in the dataset and the highwater mark, Suprtool and Speed Demon could take much longer to read the dataset (this is true for all programs, not just Suprtool and Speed Demon). Because mode-1 and mode-2 calls to SPDEDBINIT return the highwater mark for detail datasets in the status area (this is the record limit), you can write your own programs to automate checking for excessive highwater marks.

Notes

You must call DBOPEN before calling SPDEDBINIT. This is just like IMAGE -- you cannot call DBGET unless you have first called DBOPEN.

When you call SPDEDBINIT, Speed Demon determines how many records to read. For master datasets, mode-1 and mode-2 of SPDEDBINIT uses DBINFO to obtain the number of entries in the dataset. For detail datasets, mode-1 and mode-2 of SPDEDBINIT uses the highwater mark for the dataset. SPDEDBSCAN stops reading the dataset when it has read that number of entries. If the dataset is changing dynamically, new records might be missed. Use mode-3 to force Speed Demon to read to the physical end of the dataset. This mode makes Speed Demon more reliable, but slower when the dataset is mostly empty.

SPDEDBSCAN Intrinsic

Read the next entry in the dataset.

SPDEDBSCAN *Database, Status, Buffer, Dummy*

Database

Database name as passed to DBOPEN.

Status

If SPDEDBSCAN succeeds, the following values are returned in the status array. The forward and backward pointers are not returned.

1. Zero.

2. Entry length of the record returned in buffer.

3-4. Record number of the record read.

5-10. Zeros.

Buffer

Array where the values of the record read are placed. This buffer must correspond to the field list specified in SPDEDBINIT.

Dummy

This is a dummy parameter and it may contain anything.

SPDEDBSHUT Intrinsic

Close the Speed Demon environment. This must be done if another dataset will be read, or to restart at the beginning of the same set. SPDEDBSHUT will optionally print statistics about the read operation.

SPDEDBSHUT *Database, Dataset, Mode, Status, Dummy*

Database
Dataset
Mode
Status
Dummy

Database

Database name as passed to DBOPEN.

Dataset

Dataset name or dataset number to shut. This must be the same dataset name or dataset number passed to SPDEDBINIT.

Mode

SPDEDBSHUT can be called in any of three modes:

0: Return the version number of SPDEDBSHUT. All the other parameters are ignored and Speed Demon is not closed.

1: Close the Speed Demon environment.

2: Close the Speed Demon environment and print a summary of the read operation on $stdlist.

Status

If SPDEDBSHUT succeeds with mode-0, the following values are returned in the status array.

1. Zero.

2. Major version number.

3. Minor version number.

4. Pre-release version number.

5-10. Zeros.

If SPDEDBSHUT succeeds with mode-1 or mode-2, zeros are returned in the status array.

Dummy

This is a dummy parameter. It may contain anything.

Demon Program

Speed Demon includes a program for demonstration and verification. This program is called Demon.Pub.Robelle. The Demon program requests a field list to process, and accepts all valid IMAGE field lists (except for *-list "same list"). You can use this feature to compare the speed of DBGET to Speed Demon. There are several ways of accessing this program.

Speed Demon
Dbget
Image Blocks
Version Numbers

Reading with Speed Demon (Parm=0)

If you run the Demon program with no parm-value and no entry-point, you are prompted for a database and a password. To terminate the program, hit Return at the database prompt. Demon attempts to open the database with mode-5. If this succeeds, you are prompted for a dataset name. After prompting for the dataset name, Demon prompts for a field list. Demon uses SPDEDBINIT to verify that the dataset name and field list are valid and to initialize Speed Demon. Speed Demon accepts all valid IMAGE field lists, including partial lists, except for the same list "*". Demon then prints the entry length and buffer values returned from SPDEDBINIT.

Next you will be prompted for an output file name. You may specify any valid MPE file name, or Return for no output. The Demon program uses buffered I/O to write to the output file. Do not compare the cpu or wall time of the Demon program with Suprtool, because Suprtool will always be faster. After the Demon program has read the dataset, the SPDEDBSHUT statistics are printed on $stdlist.

Reading with DBGET (Parm=1)

If you run the Demon program with Parm=1, you will be prompted for a database, password, dataset, field list, and output file. The difference from Parm=0 is that the Demon program uses DBGET mode-2 to read the dataset instead of using Speed Demon. Use Parm=1 to compare the speed of DBGET versus that of Speed Demon.

Reading All IMAGE Blocks (Parm=3)

Running the Demon program with Parm=3 is the same as Parm=0, but SPDEDBINIT is called with mode-3. This forces Speed Demon to read to the physical end of the dataset. Use this feature to compare the speed of SPDEDBINIT mode-2 versus mode-3.

Verifying Version Numbers

The Demon program includes a special entry point for verifying the Speed Demon version numbers. There are two Speed Demon version numbers: one for SPDEDBINIT and one for SPDEDBSHUT. The Demon program prints two sets of these version numbers: one set for the Demon program routines and one set for the System SL. The Demon program will notify you if Speed Demon has not been installed in the System SL. For example:

Example

```
:run demon.pub.robelle,version

  Speed Demon/Copyright Robelle Solutions Technology Inc.
  (Version 4.4)

 Demon program Internal Version Numbers:
   SPDEDBINIT (Version 4.4)
   SPDEDBSHUT (Version 4.4)

 Speed Demon System SL Version Numbers:
   SPDEDBINIT (Version 4.4)
   SPDEDBSHUT (Version 4.4)
```

Demon/iX does not print the System SL version numbers. If you are on MPE/iX and have installed Speed Demon/V into the System SL, you can get the version numbers by running the compatibility mode version of Demon:

```
:run demonCM.pub.robelle,version
```

Examples of Calling Speed Demon

This chapter contains working examples of COBOL and Pascal source code that call Speed Demon. You can copy the examples from the manual, but typing them from scratch would be tedious and error-prone. The best way to copy the examples is to take them from the on-line Speed Demon documentation file. The file is stored in Robelle Qedit format. If you have Qedit, just Text a copy of Demon.Doc.Robelle and extract the portions that you are interested in.

If you don't have Qedit, use the Qcopy program to copy the documentation file to a new file in non-Qedit format, then use your favorite text editor to edit this file.

```
:run qcopy.qlib.robelle
>from=demon.doc.robelle;to=mydemon.source;new
>exit
```

Cobol Example
Pascal Example

COBOL Example

The following is an example COBOL program that shows how to call Speed Demon. This program prompts for a database, database password, dataset, and field list to process. Because upshifting is so messy in COBOL, you must enter the dataset name and field list in uppercase. If you want to use the @-field-list, just specify @ when asked by the sample program.

Source

```
$control nolist
$control source,errors=10
 identification division.
 program-id.    testread.
 author.        David Greer, Robelle Solutions Technology Inc.
 date-written.  May 11, 1990.
 security.      Copyright Robelle Solutions Technology Inc.
 remarks.

*************************************************************
*                                                           *
*                testread - test spde intrinsics.           *
*                                                           *
*  version:  1.0                                            *
*  purpose:                                                 *
*                                                           *
*  General-purpose program to test Speed Demon.             *
*                                                           *
*  Demonstrates that you must call DBOPEN before you call   *
*  the Speed Demon procedures.                              *
*                                                           *
*************************************************************

 environment division.
 configuration section.
 source-computer.  hp-3000 series 37.
 object-computer.  hp-3000 series 37.
 special-names.
     top is new-page.

 input-output section.
 file-control.
     select line-printer assign to "LINEPRT".

 data division.
 file section.
 fd  line-printer
     data record is line-record.
 01  line-record            pic x(132).

$page "CONSTANTS"
 working-storage section.
 01  true-value             pic x value "T".
 01  false-value            pic x value "F".

$page "VARIABLES"
 01  line-count             pic s9(4) comp.
 01  page-no                pic s9(4) comp.

 01  input-line                  pic x(80).
     88  answer-spaces           value spaces.

 01  end-of-db-set-flag          pic x.
     88  end-of-db-set           value "T".

 01  in-record-count             pic s9(9) comp.
```

```
 01  display-number             pic -zzz,zz9.
$page "image area"

 01  image-area.
     05  db-all-list                pic x(2) value "@ ".
     05  db-same-list               pic x(2) value "* ".
     05  db-null-list               pic s9(4) comp value 0.
     05  db-dummy-arg               pic s9(4).
     05  db-password                pic x(8).
     05  db-mode0                   pic s9(4) comp value 0.
     05  db-mode1                   pic s9(4) comp value 1.
     05  db-mode2                   pic s9(4) comp value 2.
     05  db-get-serial  redefines db-mode2 pic s9(4) comp.
     05  db-mode3                   pic s9(4) comp value 3.
     05  db-rewind-set  redefines db-mode3 pic s9(4) comp.
     05  db-get-backwards redefines db-mode3 pic s9(4) comp.
     05  db-mode4                   pic s9(4) comp value 4.
     05  db-get-direct  redefines db-mode4 pic s9(4) comp.
     05  db-mode5                   pic s9(4) comp value 5.
     05  db-get-chained redefines db-mode5 pic s9(4) comp.
     05  db-mode6                   pic s9(4) comp value 6.
     05  db-get-previous redefines db-mode6 pic s9(4) comp.
     05  db-mode7                   pic s9(4) comp value 7.
     05  db-get-keyed   redefines db-mode7 pic s9(4) comp.
     05  db-status-area.
         10  db-cond-word           pic s9(4) comp.
             88  db-stat-ok             value zeros.
             88  db-end-of-chain        value 15.
             88  db-begin-of-chain      value 14.
             88  db-no-entry            value 17.
             88  db-end-file            value 11.
             88  db-begin-file          value 10.
         10  db-stat2               pic s9(4) comp.
         10  db-stat3-4             pic s9(9) comp.
         10  db-chain-length        pic s9(9) comp.
             88  db-empty-chain     value zeros.
         10  db-stat7-8             pic s9(9) comp.
         10  db-stat9-10            pic s9(9) comp.

$page "db- variables"

 01  db-base.
     05  filler                 pic x(2)  value spaces.
     05  db-name                pic x(26).

 01  db-set                     pic x(16).

 01  db-list                    pic x(80).

 01  db-set-200                 pic s9(4) comp value 200.

 01  spde-control.
     05  spde-version           pic s9(4) comp value 0.
     05  spde-buffer-size       pic s9(9) comp value zeros.
     05  filler                 pic x(20) value spaces.

 01  db-buffer                  pic x(4096).
```

```
$page "[00] MAINLINE"
 procedure division.
 00-main                         section.

    perform 05-get-parameters
       thru 05-get-parameters-exit.

    if not answer-spaces then
        perform 10-init-testread
           thru 10-init-testread-exit
         move false-value        to end-of-db-set-flag
         move zeros              to in-record-count
         perform 20-read-set
            thru 20-read-set-exit
           until end-of-db-set
         move in-record-count to display-number
         display "In=", display-number.

    perform 90-close-files
       thru 90-close-files-exit.

 00-main-exit.  goback.
$page "[05] get-parameters"
*
* Prompt for the database, password, dataset and field list.
*
 05-get-parameters               section.

     perform 05-10-get-database.

     if not answer-spaces then
         perform 05-20-get-password
         if not answer-spaces then
             perform 05-30-get-dataset
             if not answer-spaces then
                 perform 05-40-get-field-list.

     go to 05-get-parameters-exit.

 05-10-get-database.
     move spaces                 to input-line.
     display "Enter Database Name".
     accept input-line.
     move input-line             to db-name.

 05-20-get-password.
     move spaces                 to input-line.
     display "Enter Database Password".
     accept input-line.
     move input-line             to db-password.

 05-30-get-dataset.
     move spaces                 to input-line.
     display "Enter Dataset Name (must be uppercase)".
     accept input-line.
     move input-line             to db-set.
```

```
 05-40-get-field-list.
     move spaces                 to input-line.
     display "Enter @ or Field-List (must be uppercase)".
     accept input-line.
     move input-line             to db-list.

 05-get-parameters-exit.  exit.
$page "[10] init-testread"
*
*  Open the database and initialize Speed Demon.
*

 10-init-testread                section.

     perform 10-10-open-base.

     perform 10-20-init-spde.

     perform 10-30-display-suprinit.

     go to 10-init-testread-exit.

 10-10-open-base.
     call "DBOPEN" using db-base
                         db-password
                         db-mode5
                         db-status-area.
     if not db-stat-ok then
         perform 99-fatal-error.

 10-20-init-spde.
     call "SPDEDBINIT" using db-base
                             db-set
                             db-mode2
                             db-status-area
                             spde-control
                             db-list.
     if not db-stat-ok then
         perform 98-supr-error.

 10-30-display-suprinit.
     move db-stat2               to display-number.
     display "Entry Length: ", display-number.
     move db-stat3-4             to display-number.
     display "Buffer Size:  ", display-number.

 10-init-testread-exit.  exit.
$page "[20] read-set"
*
*  Read all the records in the dataset.
*

 20-read-set                     section.

     perform 20-10-get-next.
```

```
     if not end-of-db-set then
         add 1 to in-record-count
         perform 20-20-display-next.

     go to 20-read-set-exit.

 20-10-get-next.
     call "SPDEDBSCAN" using db-base
                             db-status-area
                             db-buffer
                             db-dummy-arg.
     if db-end-file then
         move true-value               to end-of-db-set-flag
     else
     if not db-stat-ok then
         perform 98-supr-error.

 20-20-display-next.
*     move db-stat3-4             to display-number.
*     display "Record# ", display-number.

 20-read-set-exit.  exit.
$page "[90] close-files"
*
*  Cleanup by closing the dataset and the database.
*

 90-close-files                   section.

     perform 90-10-close-dataset.

     perform 90-20-close-base.

     go to 90-close-files-exit.

 90-10-close-dataset.
     call "SPDEDBSHUT" using db-base
                             db-set
                             db-mode2
                             db-status-area
                             db-dummy-arg.
     if not db-stat-ok then
         perform 98-supr-error.

 90-20-close-base.
     call "DBCLOSE" using db-base
                          db-dummy-arg
                          db-mode1
                          db-status-area.
     if not db-stat-ok then
         perform 99-fatal-error.

 90-close-files-exit.  exit.
$page "[98] supr-error"

 98-supr-error                   section.
```

```
     call "SPDEEXPLAIN" using db-status-area.

     stop run.

 98-supr-error-exit.  goback.
$page "[99] fatal-error"

 99-fatal-error                   section.

     call "DBEXPLAIN" using db-status-area.

     stop run.

 99-fatal-error-exit.  goback.
```

Pascal Example

Calling Speed Demon from Pascal is more difficult than from other languages due to Pascal's tighter type-checking on procedure parameters. The following is a complete example program written in Pascal/V that reads the `process` dataset of the `menu.qlib` database. This example program compiles and runs in native mode under MPE/iX, but you must make the two changes indicated. Because the buffer length parameter is not 32-bit aligned, Pascal/iX produces a warning when compiling the spde_db_control type. Note that the $check_actual_parm$ setting is left at 2 for the entire program. This is required to prevent parameter type mismatches between Speed Demon and Pascal.

Source

```
$list off$
{ $HP3000_16$  <-- Require this compiler option for MPE/iX}
program driver(input,output);
type
   longint  = integer;
   shortint = -32768 ..  +32767;{Remove this line on MPE/iX}

   type_db_base =
      record
         db_id   : packed array[1..2] of char;
         db_name : packed array[1..14] of char;
      end;

   type_db_set =
      record
         name : packed array[1..16] of char;
      end;

   type_db_status_area = array[1..10] of shortint;

   type_db_mode = array[1..1] of shortint;

   type_db_buffer = array[1..46] of shortint;

   type_spde_db_control =
      record
         spde_version         : shortint;
         spde_buffer_size     : integer;
         spde_future          : packed array[1..20] of char;
      end;

   type_db_dummy_arg = array[1..2] of shortint;

var
   spde_db_control : type_spde_db_control;

   db_status_area : type_db_status_area;

   db_dummy_arg   : type_db_dummy_arg;

   db_mode1       : type_db_mode;
   db_mode2       : type_db_mode;

   db_base : type_db_base;

procedure dbopen; intrinsic;
procedure dbexplain; intrinsic;

function proctime : longint; intrinsic;
function timer : longint; intrinsic;

$check_formal_parm 2$
$check_actual_parm 2$
procedure spdedbinit(var db_base : type_db_base;
                     var db_set  : type_db_set;
                     var db_mode : type_db_mode;
                     var db_stat : type_db_status_area;
```

```
                     var control : type_spde_db_control;
                     var dummy   : type_db_dummy_arg
                    ); external spl;
procedure spdedbscan(var db_base   : type_db_base;
                     var db_status : type_db_status_area;
                     var db_buffer : type_db_buffer;
                     var dummy     : type_db_dummy_arg
                    ); external spl;
procedure spdedbshut(var db_base : type_db_base;
                     var db_set  : type_db_set;
                     var db_mode : type_db_mode;
                     var db_stat : type_db_status_area;
                     var dummy   : type_db_dummy_arg
                    ); external spl;
procedure spdeexplain(var status : type_db_status_area
                     ); external spl;
$check_formal_parm 3$


function open_base : boolean;
var
   db_password : packed array[1..8] of char;
   mode1       : shortint;
begin
   open_base := false;
   db_base.db_id    := '  ';
   db_base.db_name := 'menu.qlib;    ';
   db_password     := ';;      ';
   mode1           := 1;
   dbopen(db_base
         ,db_password
         ,mode1
         ,db_status_area
         );
   if db_status_area[1] <> 0 then
      dbexplain(db_status_area)
   else
      open_base := true;
end (*open_base*);

function init_spde : boolean;
var
  db_set : type_db_set;
begin
   init_spde := false;
   with spde_db_control do
      begin
         spde_version    := 0;
         spde_buffer_size:= 0;
         spde_future     := '                    ';
      end;
   db_set.name := 'PROCESS;        ';
   spdedbinit(db_base
             ,db_set
             ,db_mode1
             ,db_status_area
             ,spde_db_control
```

```
            ,db_dummy_arg
            );
   if db_status_area[1] <> 0 then
      spdeexplain(db_status_area)
   else
      init_spde := true;
end (*init_spde*);

function read_set : boolean;
var
   db_buffer : type_db_buffer;
begin
   read_set := false;
   spdedbscan(db_base
            ,db_status_area
            ,db_buffer
            ,db_dummy_arg
            );
   if db_status_area[1] = 0 then
      read_set := true
   else
   if db_status_area[1] <> 11 then
      spdeexplain(db_status_area);
end (*read_set*);

procedure test_spde;
var
   start_cpu : longint;
   start_wall: longint;
   in_count  : longint;
begin
   start_cpu := proctime;
   start_wall:= timer;
   in_count  := 0;
   while read_set do
      in_count := in_count + 1;
   start_cpu := proctime - start_cpu;
   start_wall:= timer - start_wall;
   writeln('In=',in_count:1);
   write('Cpu-time=',start_cpu:1,' ');
   write('Wall-time=',start_wall:1);
   writeln;
end (*test_spde*);

procedure complete_driver;
var
  db_set : type_db_set;
begin
   db_set.name := 'PROCESS;        ';
   spdedbshut(db_base
            ,db_set
            ,db_mode2
            ,db_status_area
            ,db_dummy_arg
            );
   if db_status_area[1] <> 0 then
      spdeexplain(db_status_area);
```

```
end (*complete_driver*);

begin (*driver*)

   db_mode1[1] := 1;
   db_mode2[1] := 2;
   if open_base then
      if init_spde then
      begin
         test_spde;
         complete_driver;
      end;
end.
```

Speed Demon Error Messages

Speed Demon returns errors similar to IMAGE. Each error is associated with an error number. The following are the error numbers, descriptions, and causes that can be returned by any Speed Demon intrinsic.

-538: SPDEPREFETCH JCW VALUE *n* IS LESS THAN ZERO OR GREATER THAN 5

The SPDEPREFETCH JCW determines if Speed Demon should prefetch data into memory and if so, how far ahead it should read. A value of zero tells Speed Demon not to prefetch data. Values greater than five are not allowed. Check the value of the SPDEPREFETCH JCW and set it to an appropriate number.

-537: FINDJCW FAILED WITH ERROR# *n*

A call to the FINDJCW intrinsic failed. The value of *n* is the error number returned from FINDJCW. This error should never occur. If it does, please report it to Robelle Solutions Technology Inc.

-536: ROBELLE PREFETCH FAILURE ERROR# *n*

The internal Robelle prefetch module has returned an error. The value of *n* is the prefetch error number. This message is followed by a prefetch error message. You should report this error to Robelle.

-535: SPEED DEMON HAS EXPIRED

Your version of Speed Demon has expired. If you have received a nonexpiring version of Speed Demon/V, be sure to install it in the System SL. For Speed Demon/iX, make sure that you have restored DemonXL.Pub.Robelle. If you are using Speed Demon/iX from your own XL file, be sure to reinstall Speed Demon/iX after restoring the nonexpiring version.

-534: SPDEDBINIT NOT CALLED

Before calling SPDEDBSCAN or SPDEDBSHUT, you must call SPDEDBINIT successfully. Make sure that SPDEDBINIT was actually called and that the condition word returned from SPDEDBINIT was zero.

In Speed Demon/V, this error can also be caused by corrupting the stack. The Speed Demon intrinsics use two words of the DL-area (DB-32 and DB-14) to store global information. Setting the first word (DB-32) to zero, possibly because of a bug, will cause this error.

-533: FLIMIT OF ACTUAL MPE FILE IS INVALID

SPDEDBINIT verifies that the physical MPE file that corresponds to the dataset matches the information returned by DBINFO. SPDEDBINIT computes the number of records in the MPE file as follows:

```
#records = (capacity - 1) / blockfactor + 1
```

where capacity and blockfactor are reported by the FORM command of Suprtool or QUERY. This number must match the EOF of the MPE file (you can check this with :listf,2).

This error is usually caused by restoring a earlier version of the dataset from a backup tape.

-532: INVALID BLOCK SIZE FOR DATASET: xx

The MPE file corresponding to the dataset has a block length that is less than the one reported from DBINFO. This can only happen if the MPE file is corrupted.

-531: INVALID BLOCKING FACTOR FOR DATASET: xx

The MPE file corresponding to the dataset has a blocking factor other than one. This can only happen if the MPE file is corrupted.

-530: INVALID FILECODE FOR DATASET: xx

This error should never occur. If it does, please report it to Robelle Solutions Technology Inc.

-529: SPEED DEMON NOT ALLOWED ON REMOTE DATABASES

Speed Demon only works with databases on the local machine. Check for a :File command specifying the database on a remote machine.

-528: SPEED DEMON REQUIRES FULL-READ ACCESS TO THE DATASET

The password used in DBOPEN must grant read access to the entire dataset passed to SPDEDBINIT. The creator password always grants full-read access to every dataset.

-527: THE DATASET CHANGED SINCE CALLING SPDEDBINIT

The dataset name or dataset number passed to SPDEDBINIT must be the same one passed to SPDEDBSHUT.

-526: INVALID EXTRA DATA SEGMENT IDENTIFIER

This error should never occur. If it does, please report it to Robelle Solutions Technology Inc.

-525: DATASET IS ALREADY ACTIVE

You can only use Speed Demon to read one dataset at a time. Two calls were made to SPDEDBINIT. Make sure that SPDEDBSHUT is called before calling SPDEDBINIT again.

-524: INVALID DATA SEGMENT INDEX

This error can only happen in Speed Demon/V. The DL-area of the stack has been corrupted. Make sure that DB-32 and DB-14 are not used by the calling program.

-523: UNABLE TO OBTAIN EXTRA DATA SEGMENT

Speed Demon needs an extra data segment the length of the buffer size plus a few hundred bytes (the default buffer size is 28,672 bytes). SPDEDBINIT was unable to allocate this extra data segment. Either the system is out of resources (e.g., DST table is full) or the maximum extra data segment size is configured too small.

-522: INVALID FUTURE PARAMETER

The future part of the control record does not contain twenty blanks. Make sure that the control record was declared correctly, and that the correct name was passed to SPDEDBINIT. Verify that the control record is the fifth parameter of SPDEDBINIT, and be sure that either a dummy argument or a field list was passed as the last parameter to SPDEDBINIT.

-521: INVALID BUFFER LENGTH: xxxx

The buffer length of the control record did not contain a valid value. The minimum buffer size is 4096 and the maximum is 14,336, unless zero was specified. Make sure that the control record was declared correctly, the correct name was passed to SPDEDBINIT, verify that the control record is the fourth parameter of SPDEDBINIT, and be sure that a dummy argument was passed as the last parameter to SPDEDBINIT.

-520: INVALID VERSION NUMBER: x

The version number of the control record did not contain a zero. Make sure that the control record was declared correctly, the correct name was passed to SPDEDBINIT, verify that the control record is the fifth parameter of SPDEDBINIT, and be sure that a dummy argument was passed as the last parameter to SPDEDBINIT.

Assertion Errors

It is possible for Speed Demon to abort because of an internal inconsistency. These are called assertion errors and they should be reported to Robelle Solutions Technology Inc. There is one assertion error that can be caused by problems in the application software.

Each Speed Demon/V intrinsic checks for ten words of room in the status area. If there is not enough room, Speed Demon/V is aborted with assertion error number one. If this happens, you should check that the declaration of the status area is correct. This problem could also be caused by a large portion of the application stack being wiped out.

How to Contact Robelle

In the United States, in Canada, and in places not listed below, contact us at the following address:

Robelle Solutions Technology Inc.
Suite 201, 15399-102A Ave.
Surrey, B.C. Canada V3R 7K1

Toll-free: 1.888.robelle
: (1.888.762.3553)
Phone : 604.582.1700
Fax : 604.582.1799

E-mail : solutions@robelle.com
E-mail : support@robelle.com
Web : www.robelle.com

For our international distributors listing, note that the phone and fax numbers shown are for out-of-country dialing.

Europe
Africa
Asia and Australia
North America

Europe

France, Belgium

ARES
Attention: Renee Belegou
Phone: 33 1 69 86 60 24
Fax: 33 1 69 28 19 18
E-mail: rbelegou@ares.fr
Web: www.ares.fr

Germany

SWS SoftWare Systems GmbH
Attention: Renate Pfund
Phone: 49 7621 689 190
Fax: 41 31 981 32 63
E-mail: info@sws.ch
Web: www.sws.ch

The Netherlands, Belgium

Samco Automation b.v.
Attention: Marius Schild
Phone: 31 13 5215655
Fax: 31 13 5288815
E-mail: marius@samco.nl
Web: www.samco.nl

Nordic Countries

Ole Nord AB
Attention: Ole Nord
Phone: 46 8 623 00 50
Fax: 46 8 35 42 45
E-mail: info@olenordab.se
Web: www.olenordab.se

Switzerland, Austria

SWS SoftWare Systems AG
Attention: Renate Pfund
Phone: 41 31 981 06 66
Fax: 41 31 981 32 63
E-mail: info@sws.ch
Web: www.sws.ch

United Kingdom, Ireland

Robelle Consulting
Attention: Clive Oldfield
Phone: +44 20 7473 2558
Fax: +44 20 7473 2558
E-mail: robelle_oldfield@email.msn.com

Africa

South Africa

Synergy Computing (Pty) Ltd
Attention: Paul Howard
Phone: 27 21 685 7809
Fax: 27 21 685 7927
E-mail: synergy@synergy.co.za

Asia and Australia

Australia, New Zealand

MRFM Pty. Ltd.
Attention: Michael Redmond
Phone: +61 3 9629 8633
Fax: +61 3 9629 8062
E-mail: mredmond@mrfm.com.au
Web: www.mrfm.com.au

Hong Kong

SCS Computer Systems Ltd.
Attention: Steven Lai
Phone: 852 2609 1338
Fax: 852 2607 3042

Singapore, Malaysia

Singapore Computer Systems Ltd.
Attention: Toh Tiau Hong
Phone: 65 441 2688
Fax: 65 441 2811
E-mail: tohth@scs.com.sg
Web: www.scs.com.sg

North America

Mexico

Infosistemas Financieros SA de CV
Attention: Anita De Urquijo
Phone: 52 5 813 1325
Fax: 52 5 813 3026
E-mail: adeurquijo@if.com.mx
Web: www.if.com.mx