

COBOL for OS/ ~~360~~ 327S)390 & V

Contents

| | |
|---------------------------------------|-----------|
| About this book | xi |
| How this book will help you. | xi |
| Abbreviated terms | xi |
| How to read syntax diagrams | xii |
| How exampams | |

| | |
|---|----|
| Loading a table dynamically. | 58 |
| Initializing a table (INITIALIZE) | 58 |

| | |
|---|-----|
| Example: batch compilation | 226 |
| Specifying compiler options in a batch compilation | 227 |
| Example: precedence of options in a batch compilation | 228 |
| Example: LANGUAGE option in a batch compilation | 229 |



If choosing one of the items is optional, the entire stack appears below the main path.



This requires changes to the JCL to specify appropriate COBOL compiler options for DLL support; the prelink step must specify (at least) the DLL

Part 1. Coding your program

Chapter 1. Structuring your program

| | |
|---|----|
| Loading a table dynamically. | 58 |
| Initializing a table (INITIALIZE) | 58 |
| Assigning values when you define a table (VALUE) | 58 |

| | |
|--|-----|
| Parameters for creating QSAM files | 123 |
| Retrieving QSAM files | 123 |
| Parameters for retrieving QSAM files | 124 |
| Ensuring file attributes match your program | 124 |
| Processing existing files | 125 |
| Defining variable-length (format-V) records | 125 |
| Defining format-U records | 125 |
| Defining fixed-length records | 125 |
| Processing new files | 125 |
| Processing files dynamically created by COBOL | 126 |
| Using striped extended-format QSAM data sets | 126 |
| Allocation of buffers for QSAM files. | 127 |
| Accessing HFS files using QSAM. | 127 |
| Restrictions and usage | 127 |
| Identifying QSAM files to CMS | 128 |

Defining files to the operating system

For all files that you process in your COBOL program, you need to define the files to the operating system with an appropriate system data definition:

- DD statement for OS/390 JCL
- ALLOCATE


```
DATA DIVISION.  
Working-Storage Section.  
01 numb pic 9(4) value 5.  
01 fact pic 9(8) value 0.  
Local-Storage Section.  
01 num pic 9(4).  
PROCEDURE DIVISION.  
    move numb to num  
  
    if numb = 0  
        move 1 to fact  
    else  
        subtract 1 from numb  
        call 'factorial'  
        multiply num by fact  
    end-if.  
  
    display num '!' = ' fact.  
    goback.  
End Program factorial.
```

Sharing data in nested programs

Some applications consist of nested programs—

Conditional statements

A conditional statement is either a simple conditional statement (IF, EVALUATE, SEARCH

RELATED TASKS

“Selecting program actions”

05 Customer-Name.

Initializing a right-justified field:

The **MVVE** statement carries out the assignment with truncation. You can, however, specify the **DIAGTRUNC** compiler option to request that the compiler issue a warning diagnostic for **MVVE**.

Set to stderr

Output is routed to stderr (file descriptor 2).

05 Initial-count Pic S9(4) Usage Binary Value 1000.

Regardless of which USAGE clause you use to control the internal representation of a value, you use the same PICTURE

You can specify scaling (that is, decimal positions or implied integer positions) in the



A loss of precision is possible in conversions between fixed-point data formats and floating-point data formats (short floating point, long floating point, or external floating point). These conversions happen during arithmetic evaluations that have a mixture of both fixed-point and floating-point operands.

Compute Ave-Tax = Function Mean (Tax-S Tax-T Tax-W Tax-B)
Compute Median-Tax = Function Median (Tax-S Tax-T Tax-W Tax-B)
Compute Tax-Range = Function Range (Tax-S Tax-T Tax-W Tax-B)

```
compute report-matrix-col = (emp-count ** .5) + 1
compute report-matrix-col = function sqrt(emp-count) + 1
if report-matrix-tot < function sqrt(emp-count) + 1
```

Chapter 4. Handling tables

A *table* is a collection of data items that have the same description, such as account totals or monthly averages. A table is the COBOL equivalent of an array of elements. It consists of a table name and subordinate items called *table elements*.

RELATED TASKS

- “Creating variable-length tables (DEPENDING ON)” on page 62
- “Nesting tables”
- “Putting values into a table” on page 58
- “Referring to an item in a table” on page 55
- “Searching a table” on page 65

RELATED REFERENCES

OCCURS clause (*IBM COBOL Language Reference*).907490m..40.20.907490m..40.20147.02490m.963.20

Define static values in WORKING-STORAGE in one of these ways:

- Initialize each table n0ms idividually.:
- Initialize table thf
- Initialize ccurr of table of thf valu.:.

Creating variable-length tables (DEPENDING ON)

DATA DIVISION.
FILE SECTION.
FD LOCATION-FILE
RECORDING MODE F
BLOCK 0 RECORDS
RECORD 80 CHARACTERS
LABEL RECORD STANDARD.
01 LOCATION-/29h-LE

WORKRDI-STORAGILE SECTION.

Example: binary search



When control reaches the PERFORM statement, the code for the paragraph
010-PROCESS-ONE-MONTH



05 BILL-INFO.
10 INV-NO PIC X(6).
10 INV-AMT PIC \$\$, \$\$\$.99.
10 AMT-PAID PIC \$\$, \$\$\$.99.


```
ACCEPT REFMD-TIME-ITEM FROM TIME.  
DISPLAY "CURRENT TIME IS: "  
* Retrieve the portion of the time value that corresponds to  
*   the number of hours:  
    REFMD-TIME-ITEM (1:2)  
    ":"  
* Retrieve the portion of the time value that corresponds to  
*   the number of minutes:  
    REFMD-TIME-ITEM (3:2)  
    ":"  
* Retrieve the portion of the time value that corresponds to  
*   the number of seconds:  
    REFMD-TIME-ITEM (5:2)  
"  
"
```


IGZCA2D syntax

Finding the date of compilation

FILE-CONTROL.

In this information, *QSAM logical record* refers to the QSAM definition, and refers to the QSAM definition.

RELATED CONCEPTS

The

RELATED REFERENCES

“Layout of format-F records” [↳ Chapter 10: formatVF records](#)

Adding records to QSAM files

Sp.i8yBy the number of lines the page is advanced with an integer (or anwith a) following. If you o

Remember that information in the JCL or environment variable overrides information in the data set label.

Processing existing files

When your program processes an existing file, code the description of the file in

RELATED REFERENCES

“

OPTCD=*Q*

Q

you divide the key by a value equal to the number of slots in the data set to

Defining fixed-length records

To define the records to be fixed length, use one of the following coding options:

| | | | |
|--|--|--|--|
| | | | |
| | | | |
| | | | |

Coding input and output statements for VSAM files

Use these COBOL statements for processing VSAM files:

OPEN

Initially loading records sequentially into a file: Initially loading a file means writing records into the file for the first time. This is not the same as writing records into a file fr332.9(writing)]TJFich9(a)-332.ll9(a)-332p(ecor)17vious9(r)17.9(ecor)17.9(ds)-332.9(i)

Use the access method services

```
DEFINE CLUSTER NUMBERED  
RECORDSIZE(avg, m)
```

In your COBOL program the alternate index is identified solely by the ALTERNATE RECORD KEY clause of the FILE CONTROL paragraph. The ALTERNATE RECORD KEY definitions must match the definitions that you have made in the

When you use alternate indexes in your COBOL program, you must specify not only a system name (using a DD statement or environment variable) for the base cluster, but also one for each alternate index path. No language mechanism exists to explicitly declare system names for alternate index paths within the program.

| Aspect of VSAM | What you can do | Rationale and comments |
|----------------|-----------------|------------------------|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

READ To read a record from the file.

With sequential processing, your program reads one record after another in the same order in which they were entered when the file was created.

WRITE To create a record in the file.

Your program writes new records to the end of the file.

CLOSE To release the connection between the file and your program.

Chapter 11. Sorting and merging files

You can arrange records in a particular sequence by using the SORT

The entry-332293 is-332293 te

The RETURN statement makes each sorted record available to your output

Input and output

Define input and output data sets, if any.

SORTLIB (DEFSORT library)

Define the library containing the sort modules, for example, SYS1. SORTLIB.

To improve sort performance on variable-length files, specify the most frequently

There is no maximum number of keys, but the keys must be located in the first 4092 bytes of the record description.

The total length of the keys cannot exceed 4092 bytes unless the EQUALS

PROCEDURE DIVISION.

run time, however, any parameters on control statements in the sort control

Handling errors in arithmetic operations

The results of arithmetic operations might be larger than the fixed-point field that

- **ERROR declaratives**
- **FILE STATUS**

abnormal. Establish a file status key by using the FILE STATUS clause in the FILE-CONTROL paragraph and data definitions in the

WORKING-STORAGE SECTION.
01 MASTER-CHECK-KEY PIC X(2).

PROCEDURE DIVISION.

```
OPEN INPUT MASTERFILE  
IF MASTER-CHECK-KEY NOT = "00"  
    DISPLAY "Nonzero file status returned from OPEN " MASTER-CHECK-KEY
```

This example also illustrates how output from this program might look if the file being processed contains six records.

```
IDENTIFICATION DIVISION
PROGRAM-ID. EXAMPLE.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
FILE-CONTROL.
  SELECT VSAMFILE ASSIGN TO VSAMFILE
  ORGANIZATION IS INDEXED
  ACCESS DYNAMIC
  RECORD KEY IS VSAMFILE-KEY
  FILE STATUS IS FS-CODE VSAM-CODE.
DATA DIVISION.
FILE SECTION.
FD  VSAMFILE
  RECORD 30.
01  VSAMFILE-REC.
    10 VSAMFILE-KEY          PIC X(6).
    10 FILLER                PIC X(24).
WORKING-STORAG8XgbmbbbbX?kt#)WORKING-SI2h3RETURN-Zg9hh)LE-REC.Z@?*t8Xgbmbbbb059hh)ISZg9hh)FS-6hICZ
```


Language Environment condition handling you can write your own error-handling programs in COBOL. They can report, analyze, or even fix up and allow your program to resume running.

To have Language Environment pass control to your user-written error program, you must first identify and register its entry point to Language Environment. PROCEDURE-POINTER data items allow you to pass the entry address of procedure

The simplest way to compile your program under OS/390 is to code JCL that uses

Additional details on invoking cataloged procedures, overriding and adding to
EXEC


```
/*
/*  PARAMETER  DEFAULT VALUE   USAGE
/*  LNGPRFX   IGY.V2R2M    PREFIX FOR LANGUAGE DATA SET NAMES
```




For fixed-length records (RECFM=F or RECFM=FB), LRECL is the logical record length, and BLKSIZE equals LRECL multiplied by *n*, where *n* is equal to the blocking factor.

//S_DUNCH DD /S_OUT=B

Specifying compiler options under OS/390

The compiler is installed and set up with default compiler options. While installing

| | | |
|--|--|--|
| | | |
| | | |
| | | |
| | | |

3. The values that you specify on PR0CESS or CBL statements within your COBOL source programs
4. The values that you specify in the cob2 command's -q option string
- 5.

If you use the COPY

Chapter 15. Compiling under CMS

Under CMS, the COBOL compiler can compile any COBOL source program it can

filename

RELATED TASKS

- “Naming generated files under CMS”
- “Correcting errors” on page 252





When conflicting options are specified at the same level in the hierarchy, the option specified aken

When you specify the ADATA option with this option, you can create a SYSADATA file for later analysis by program understanding tools.

This option can be set as the installation default option or as a compiler invocation option, but cannot be set on a CBL or PROCESS statement.

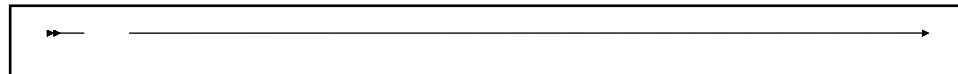
The specification of the ANALYZE option forces the handling of the following character strings as reserved words:

- CICS
- EXEC
- END-EXEC
- SQL

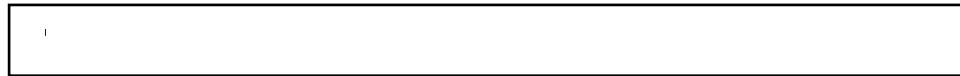
RELATED REFERENCES

“Conflicting compiler options” on page 259

ADVDefault50 .36108e9f f41e02208Tm [(Default507e f48 7.115579r f2500 f29 a5.5re5



Use BUFSIZE to allocate an amount of main storage to the buffer for each compiler

DBCS

- READ . . . INTO
- RELEASE . . . FROM
- RETURN . . . INTO
- REWRITE . . . FROM
- WRITE . . . FROM

RELATED REFERENCES

“Conflicting compiler options” on page 259

DUMP



RELATED REFERENCES

“Conflicting compiler options” on page 259

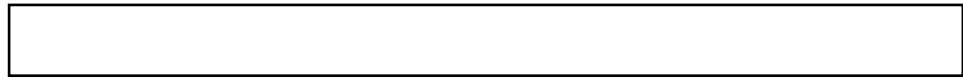
DYNAM



Use EXPORTALL to instruct the compiler to automatically export the following symbols when the object deck is link-edited to form a DLL:

-

FLAGMIG



IDLGEN

The mapping of COBOL to IDL is designed to balance two conflicting objectives,

`INTDATE(LILIAN)` instructs the compiler to use the Language Environment Lilian starting date for integer dates used with date intrinsic functions. Day 1 is Oct 15, 1582.

With `INTDATE(LILIAN)`, the date intrinsic functions return results compatible with the Language Environment date callable services. These results will be different than in COBOL/370 Release 1.

Usage notes:

1. When `INTDATE(LILIAN)`

NUMPROC(PFD)

the error message information and DISPLAY output directed to the system logical output device are routed to the same destination.

RELATED TASKS

“Displaying values on a screen or in a file (DISPLAY)” on page 29

PGMNAME(LONGUPPER)

CICS: You must use RENT for programs to be run under CICS.

OS/390 UNIX: You must use RENT for programs to be run in the OS/390 UNIX environment.

When a reentrant program is to be run with extended addressing, you can use the DATA(24|31) option to control whether dynamic data areas are allocated in unrestricted storage or in storage obtained from below 16 MB. Compile programs with RENT or RMDE(ANY) if they will be run with extended addressing in virtual

MAX requests the largest available block of storage in the user region for use during compilation.

Use **SIZE** to indicate the amount of main storage available for compilation (where

The compiler passes the following suboption string to the DB2 coprocessor:
“string1 string2 string3”

The concatenated strings are delimited with single spaces as shown. If multiple

IBM COBOL for OS/390 & VM uses the Language Environment-provided dump services to produce dumps that are consistent in content and format to those produced by other Language Environment-conforming member languages. Whether Language Environment produces a dump for unhandled conditions depends on the setting of the run-time option `*Tuw/XbX?f8hm66h3XhRMTHDACT`

1. Use the



TYPECHK



1. Group names used in a MOVE CORRESPONDING statement are in the XREF listing. In



- Hyphens that are not the first or last character are transed(e)-332.9o(e)-332.ze(ar)17.o(e)

- When *library-name* is a user ns

RELATED CONCEPTS

“Scope terminators” on page 20

RELATED REFERENCES

| Use | Listing | Contents | Compiler option |
|---|---------|----------|-----------------|
| 1. To eliminate messages, turn off the options (such as | | | |

| Symbol | Definition |
|---------------------|---------------------------|
| WHEN=n ¹ | Evaluate WHEN cell number |

Example: TGT memory map

The following example shows LIST output for the task global table (TGT) with

| | | |
|--------|--------|---------------------------------|
| 0052EC | 00018C | FCB CELLS |
| 005300 | 0001A0 | ALL PARAMETER BLOCK |
| 005364 | 000204 | INTERNAL PROGRAM CONTROL BLOCKS |

- (1) Hexadecimal offset of the TGT field from the start of the COBOL program
(not shown for programs compiled with the

Example: XREF output - data-name cross-references

The following example shows a sorted cross-reference of data names, produced by the XREF compiler option.

An "M" p

Example: XREF output - program-name cross-references

The following example shows a sorted cross-reference of program names,

Example: VBREF compiler output

RELATED REFERENCES

“TEST

Part 3. Targeting COBOL programs for certain environments

- REPLACE statements that contain EXEC commands must occur after the PROCEDURE DIVISION

CICS LINK instead. If you are calling a COBOL program that has been translated, you must pass DFHEIBLK and DFHCOMMAREA as the first two parameters in the CALL /F3 1 Tfca358 0 TDus

even if your COBOL program terminates normally after successfully using the external CICS interface, the job step could end with an undefined return code.

To resolve this issue, you can add a RETURN CICS command to your COBOL program. This command will ensure that the job step ends with a defined return code, even if the program terminates normally.

| Required compiler option | Condition |
|--------------------------|---|
| NODYNAM | The program is translated by the CICS translator. |
| LIB | The program contains a COPY or BASIS statement. |

In addition, you might want to use the following recommended options under certain circumstances:

RELATED REFERENCES
“WORD” on page 301

Handling errors by using CICS HANDLE

The setting of the CBLPSHP0P run-time option affects the state of the HANDLE

- Use the

You can use standard JCL procedural statements to compile your program with the

The ISPF shell can direct stdout and stderr only to an HFS file, not to your

Environment variables of interest for COBOL programs

_CEE_RUNOPTS

Run-time options.

LIBPAT

Chapter 24. Writing object-oriented programs

Subclasses

In the mail-order catalog example, Orders is a general class. One of the first things

IDENTIFICATION DIVISION.

01 action PIC X(10).
01 item PIC X(5).

PROCEDURE DIVISION USING item action.

END METHOD WriteUserOutput.

END CLASS UserInterface.

| |
|--|
| |
| |
| |
| |
| |

LOCAL-STORAGE SECTION.
01 backorder.

```
77 item PIC X(5).
01 item-table.
  02 item-entry OCCURS 10 TIMES.
    03 item-element PIC X(5).
```

```
PROCEDURE DIVISION.
```

```
*
```

```
*
```

```
*
```

```
*
```


The method

MOVE 0 TO status-count.

```
01 backorder.  
02 backorder-number PIC 9(5).  
02 backorder-date PIC X(8).  
02 backorder-count PIC 99.  
02 backorder-table.
```


USING order-number
END-PERFORM

SOM provides a predefined interface repository data set called *somhlq*. SGOSIR, which contains the interfaces of all the SOM system classes. (*somhlq*

SOMIR

Specifies the location of the interface repository files.

For example:

```
[somir]
SOMIR='/' somhIq.SGOSIR'; //' prefix.LOCAL.SOMIR';
```

```
/* Compile the client program with type checking, prelink and link
/* the client program importing the class DLL and the SOM kernel DLL
/*-----
```

Then you would write the COBOL code to map the operation, adding the color blue to an object, as follows:

```
1 color binary pic 9(9).
```




The unsigned form of binary data is mapped as for the SOM IDL long type, except that the PICTURE clause does not specify the character S

If you have to supply i nout arguments of any of the complex types, you would do well to allocate the storage dynamically using **OMMAllocate**, and declare the COBOL equivalent type in the **LINKAGE SECTION**

| IDL type | in | inout/out | Return value |
|-----------------|-----------|------------------|---------------------|
| | | | |

Passing environment variables

```
*****
* Declare the environment variable itself: *
*****  
1 ev.  
2 major binary pic 9(9).  
88 no-exception value 0.  
88 any-exception value 1 thru 999999999.  
88 user-exception value 1.  
88 system-exception value 2.  
22 E/00(E00(ES)]00(0/00(ES])JT*[(2 -1.0)Sj1.52-1.1111 TD[(22)5-/)]00
```


filestem = spred;

Example: COBOL variable-length string class

Performing arithmetic on date fields. 525

called program processing is completed, the program can either transfer control back to the calling program or end the run unit.

The called COBOL program starts running at the top of the PROCEDURE DIVISION.

RELATED TASKS

“Ending and reentering main programs or subprograms”

“Making recursive calls” on page 472

“Transferring control to another program” on page 461

RELATED REFERENCES

Language Environment Programming Guide

Ending and reentering main programs or subprograms

You can use any of three termination statements in a main program or subprogram, but they have different effects, as shown in the table below:

| |
|--|
| |
| |
| |
| |
| |

- Data defined in the LOCAL-STORAGE SECTION will be in the initial state each time the outermost containing program is called. (For nested programs, LOCAL-STORAGE

- You want to call subprograms in their unused state, and you cannot use the INITIAL attribute.

When you cannot use the INITIAL attribute to ensure that a subprogram is placed in its unused state each time it is called, you can set the unused state by using a combination of dynamic CALL and CANCEL statements. When you cancel the subprogram that was first called by a COBOL program, the next call will cause the subprogram to be reinitialized to its unused state.

Using the CANCEL statement to explicitly cancel a subprogram that was dynamically loaded and branched to by a non-COBOL program does not result in any action being taken to release the subprogram's storage or to delete9(can not)-332.9(m8xp10765
0T

Dynamic calls to alternate entry points require:

- NAME or ALIAS



Describing arguments in the calling program

In the calling program, describe arguments in the

The data passed from a calling program might contain header information that you want to ignore. Because pointer data items are not numeric, you cannot directly

02 NAME PIC X(20).
02 DEPT PIC 9(4).
02 SALARY PIC 9(6).

PROCEDURE DIVISION USING DEPT-X.

SET ADDRESS OF SALARY-REC TO ADDRESS OF REAL-SALARY-REC

The address of SALARY-REC is now based on the address of REAL-SALARY-REC, or

CALL . . .

Stop Run

Chapter 30. Creating a DLL or a DLL application

“Prelinking certain DLLs”

Procedure division.

- If the target DLL is a PDS or PDSE member, then the DLL member name must

//CHEK DD DSNAME=CHECK2,

Millennium language extensions (MLE)

`COMPUTE` statement to store the date, with the `ON SIZE ERROR` phrase to detect whether or not the date is within the century window.

“Example: internal bridging

RELATED CONCEPTS**RELATED TASKS**

“Example: UNDATE”

Example: DATEVAL

Assume that a program contains a field Date-Copied and that this field is referenced many times in the program, but that most of these references move it

Using structured programming

Using structured programming statements (such as EVALUATE

More efficient
$$\begin{aligned} V1 * V2 * V3 * (C1 * C2 * C3) \\ C1 + C2 + C3 + V1 + V2 + V3 \end{aligned}$$
Less efficient
$$\begin{aligned} V1 * V2 * V3 * C1 * C2 * C3 \\ V1 + C1 + V2 + C2 + V3 + C3 \end{aligned}$$

Often, in production programming, there is a tendency to place constant factors on

computations with binary operands if the precision is eight digits or fewer. Above 18 digits, the compiler always uses decimal arithmetic. With a precision of nine to 18 digits, the compiler uses either form.

To produce the most efficient code for a

A floating-point exponent causes floating-point arithmetic to be used to compute the exponentiation.

RELATED CONCEPTS

“Formats for numeric data” on page 36

- Eliminate constant computations by performing them when the program is compiled.
- Eliminate constant conditional expressions.
- Aggregate moves of contiguous items (such as those that often occur with the use of MOVE CORRESPONDING) into a single move. Both the source and target must

You might have a customized system that requires certain options for optimum performance.

1. To see what your system defaults are, get a short listing for any program and review the listed option settings.
- 2.

OS/390 UNIX

Use the SYSLIB environment variable to define multiple paths to your

paragraph-name.
COPY DOWORK.

The statements included in the

RELATED CONCEPTS

- “Formats for numeric data” on page 36
- “Fixed-point versus floating-point arithmetic” on page 49

RELATED REFERENCES

- “Fixed-point data and intermediate results” on page 561
- “Floating-point data and intermediate results” on page 566
- “ON SIZE ERROR and intermediate results” on page 567
- “Arithmetic expressions in nonarithmetic statements” on page 568
- “ARITH” on page 262

| | | |
|--|--|--|
| | | |
| | | |
| | | |
| | | |

Intrinsic functions evaluated in fixed-point arithmetic

Appendix B. Complex OCCURS DEPENDING ON

Complex OCCURS DEPENDING ON (ODO) is supported as an extension to the COBOL 85 Standard.

The basic forms of complex ODO permitted by the compiler are as follows:

- **Variably located item or group:** A data item described by an OCCURS clause with the DEPENDING ON option is followed by a nonsubordinate data item or group.
- **Variably located table:** A data item described by an OCCURS

4. Restore the variably located data items from the data area where you saved them.

NoteVARY-FIELD-1did-332.s 2not-332.s 2overlay-332.s 2lhVARY-FIELD-2.**RELATED REFERENCES**"

The suboptions *str1*, *str2*, *str3*

The compiler does not allow recursive calls to *text-name*. That is, a COPY member can be named only once in a set of nested COPY statements until the end-of-data for that copy member is reached.

The processing of PRTEXIT is as follows:

RELATED REFERENCES

“Appendix C. EXIT compiler option

Program-ID. Skelinx.

* S Y S I N u x < i r X I T S I N u x < i P R O C E S S O R #) * Z g b 9 *
G o

3. Home code
4. Work code
5. Commuter name
6. Home address
7. Home phone
8. Work phone


```
DEFINE MASTERCATALOG      -
        (NAME(MASTCAT)      -
         CYL(1)              -
```


C06Stay Active
C07Sport Shop
C08Stay Sporty
C09Hot Sports
C10The Sportsman
C11Playing Ball
C12Sports Play

. . .

- Transaction errors
- Sales by product and area
- Individual sales performance and commissions
- Response time between the sale date and the date the sold products are shipped

Your output might vary slightly in appearance, depending on your system.

“Example: IGYTSALE transaction errors”

“Example: IGYTSALE sales analysis by product by area” on page 600

“Example: IGYTSALE sales and commissions” on page 601

“Example: IGYTSALE response time from sale to ship” on page 602

Example: IGYTSALE transaction errors

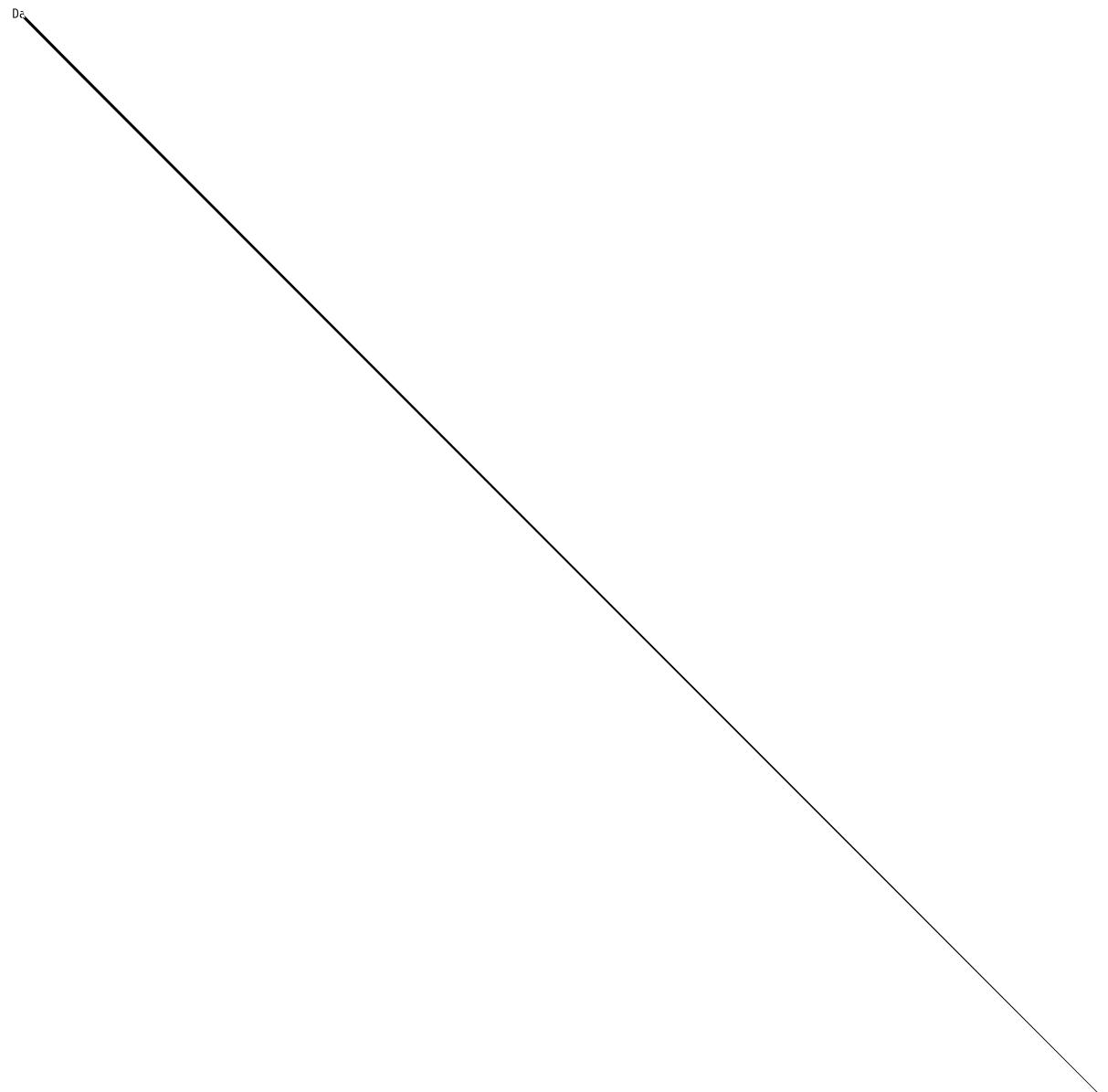
The following sample of IGYTSALE output shows transaction errors in the last column.

D4

Example: IGYTSALE sales analysis by product by area

The following sample of IGYTSALE output shows sales by product and area.

D_c



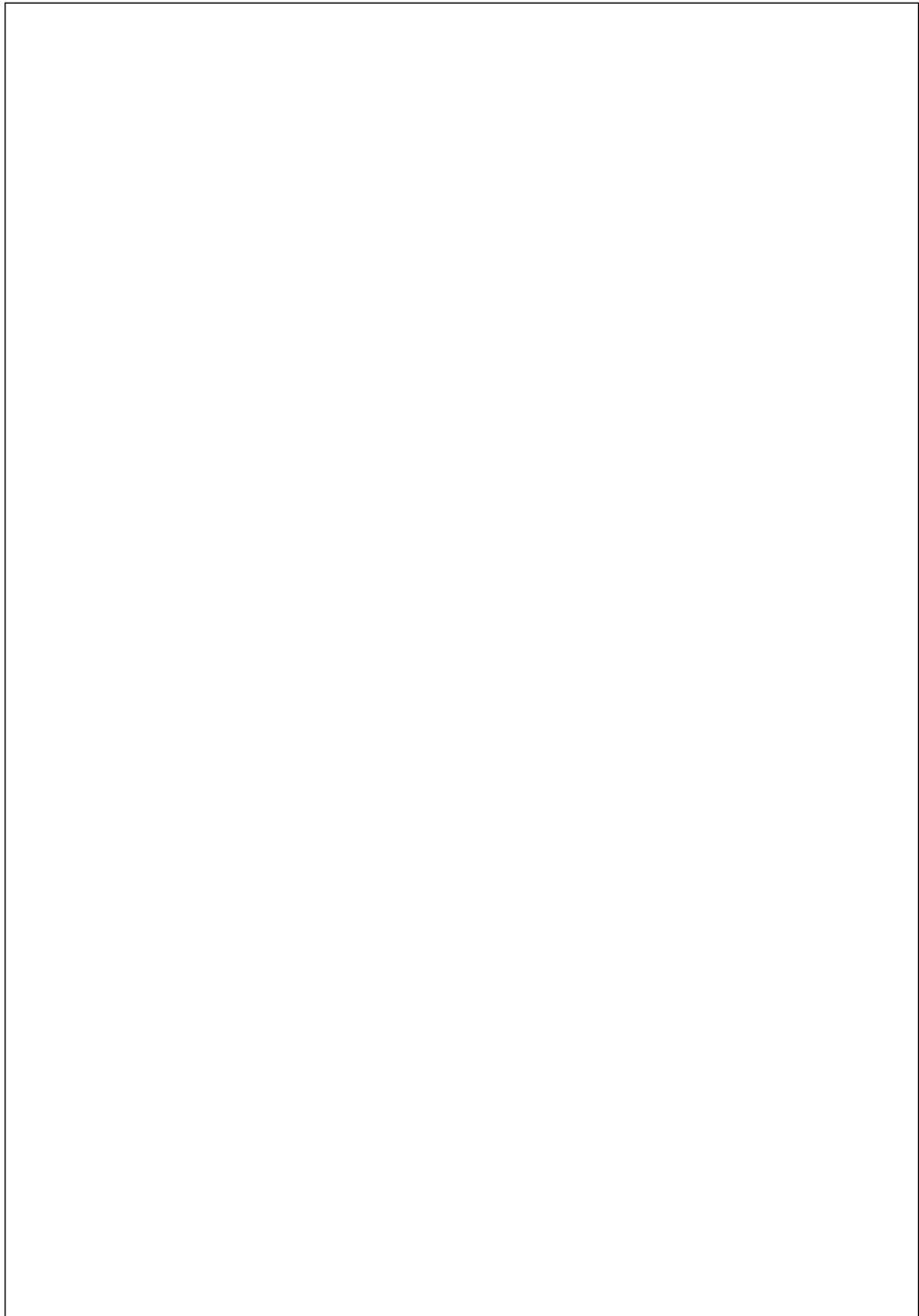
Example: IGYTSALE sales and commissions

The following sample of IGYTSALE output shows sales performance and commissions by salesperson.

D


```
FILEDEF SYSOUT   DISK IGYTSALE SYSOUT   A
FILEDEF SYSIN    DISK IGYTSALE SYSIN    *
FILEDEF IGYTABLE DISK IGYTABLE INPUT    *
FILEDEF IGYTRANS DISK IGYTRANA INPUT    *
FILEDEF IGYPRINT DISK IGYPRINT OUTPUT   *
FILEDEF IGPRT2   DISK IGPRT2  OUTPUT   *
```

5. Run the program by using the LOAD and START commands:



Notices

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, or other countries, or both:

array. In Language Environment, an aggregate that

century window. A century window is a 100-year interval within which any two-digit year is unique. Several types of century window are available to COBOL programmers:

- For windowed date fields, you use the YEARWINDOW comp.ds, fuEion.
- For the windowing intrinsic funcEions DATE-TO-YYYYMMDD, DAY-TO-YYYYDDD, and YEAR-TO-YYYY, you specify the century window with *argument-2*.
- For Language Environment callable services, you specify the century window in CEESCEN.

* **character.** The basic indivisible unit of the language.

character posiEion.

* current volume pointer.

Distributed Debugger.

G

* **global name.** A name that is declared in only one

transmission and handling of data during execution of

file and before the execution of a CLOSE statement without the REEL or UNIT phase for that file.

* **I-O status.** A conceptual entity that contains the

* **LOCAL-STORAGE SECTION.** The section of the

selection structure. A program processing logic in

*Application Programming: EXEC DLI Commands for
CICS and IMS*, SC26-8018

*Application Programming: Transaction Manager
Summary*, SC26-8038

Application Programming: Transaction Manager,
SC26-8017

OS/390 ISPF

User's Guide, SC34-4791 & SC34-4792

compiler options (*continued*)

NODYNAM 467

NOFASTSRT 183

file access mode (*continued*)

job resubmission 507

READ statement 117 (*continued*)

SIZE compiler option 291
skip a block of records 115

Readers' Comments — We'd Like to Hear from You

