

AD-A250 477



WRDC-TR-90-8007
Volume V
Part 46



INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume V - Common Data Model Subsystem
Part 46 - CDM Compare Utility Build Instructions User's Manual

M. Apicella, S. Singh

Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH 45324-6209



September 1990

Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited

92-13583



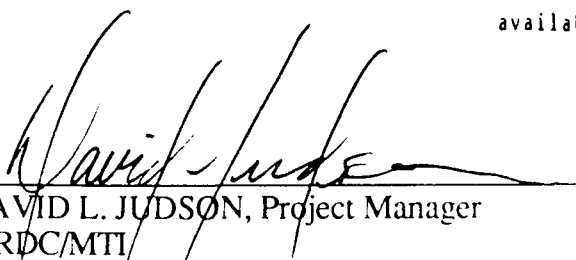
MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations


DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

FOR THE COMMANDER:


BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

25 July 91
DATE

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UM 620341431		5. MONITORING ORGANIZATION REPORT NUMBER(S) WRDC-TR-90-8007 Vol. V, Part 46	
6a. NAME OF PERFORMING ORGANIZATION Control Data Corporation; Integration Technology Services	6b. OFFICE SYMBOL (if applicable) WRDC/MTI	7a. NAME OF MONITORING ORGANIZATION WRDC/MTI	
6c. ADDRESS (City, State, and ZIP Code) 2970 Presidential Drive Fairborn, OH 45324-6209		7b. ADDRESS (City, State, and ZIP Code) WPAFB, OH 45433-6533	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Wright Research and Development Center, Air Force Systems Command, USAF	8b. OFFICE SYMBOL (if applicable) WRDC/MTI	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM. F33600-87-C-0464	
8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, Ohio 45433-6533		10. SOURCE OF FUNDING NOS.	
11. TITLE (Include See block 19) See block 19		PROGRAM ELEMENT NO. 78011F	PROJECT NO. 595600
		TASK NO. F95600	WORK UNIT NO. 20950607
12. PERSONAL AUTHOR(S) Control Data Corporation: Apicella, M. L., Singh, S.			
13a. TYPE OF REPORT Final Report	13b. TIME COVERED 4 / 1 / 87 - 12 / 31 / 90	14. DATE OF REPORT (Yr., Mo., Day) 1990 September 30	15. PAGE COUNT 30
16. SUPPLEMENTARY NOTES WRDC/MTI Proj. Priority 6203			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify block no.)	
FIELD	GROUP	SUB GR.	
1308	0905		
19. ABSTRACT (Continue on reverse if necessary and identify block number) This User Manual (UM) provides instructions on how to build the Common Data Model (CDM) Compare Utility. BLOCK 11: INTEGRATED INFORMATION SUPPORT SYSTEM Vol V - Common Data Model Subsystem Part 46 - CDM Compare Utility Build Instructions User's Manual			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED x SAME AS RPT. DTIC USERS		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson	22b. TELEPHONE NO. (Include Area Code) (513) 255-7371	22c. OFFICE SYMBOL WRDC MTI	

FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0464, DAPro Project. This contract is sponsored by the Manufacturing Technology Directorate, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Bruce A. Rasmussen, Branch Chief, Integration Technology Division, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. Jimmy P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the Control Data Corporation subcontractors and their contributing activities:

<u>SUBCONTRACTOR</u>	<u>ROLE</u>
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.
Simpact Corporation	Responsible for Communication development.

Structural Dynamics
Research Corporation

Responsible for User Interfaces,
Virtual Terminal Interface, and Network
Transaction Manager design,
development, implementation, and
support.

Arizona State University

Responsible for test bed operations
and support.

TABLE OF CONTENTS

	<u>Page</u>
SECTION 1. INTRODUCTION.....	1-1
SECTION 2. PREREQUISITES.....	2-1
SECTION 3. PROCEDURES TO BUILD THE CDM COMPARE EXECUTABLES.....	3-1

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

SECTION 1

INTRODUCTION

The CDM Compare Utility is composed of three executables. CDM Compare is performed in three phases - CDM EXTRACT, CDM COMPARE and CDM REPORT. The creation of these executables for the three phases is achieved by precompiling the database access programs, flanning the form definition files, compiling the source code and linking the EXTRACT, COMPARE and REPORT executables.

Section 2 lists the prerequisites of the CDM Compare Utilitycd Environment. Section 3 contains the step by step instructions for building the CDM Compare Utility executables.

SECTION 2
PREREQUISITES

The Prerequisites to creating the CDM COMPARE environment are:

1. Existence of an object library TOOLOLB in the directory cdmdir:[tools.comp] for the Compare software.
2. Existence of an object library GENOLB in the directory cdmdir:[tools.comp] for the generated code.
3. Existence of a a FORMS directory; this is the directory pointed to by the logical IISSULIB.
4. All the software must be compiled and placed in TOOLOLB according to normal Integration and Testing procedures.
5. The NDDL and NDML executables must be available.

SECTION 3

PROCEDURES TO BUILD THE CDM COMPARE EXECUTABLES

The following steps must be executed in order to construct the CDM Compare Executables.

CDM EXTRACT PHASE

1. The following files must have been compiled according to normal Configuration Management Procedures and placed in the TOOLOLB.OLB object library:

CXMAIN.COB (Main Program)
CXFPPIO.COB
CXFFPIO.COB
2. The CXMAIN.COB must be compiled and left as a local object. PRECOMPILE and COMPILE the following NDML embedded source code by executing the procedure file BLDXCMP.COM:

CXISKM.PRC
CXCSKM.PRC
CXESKM.PRC
CXCISM.PRC
CXCESM.PRC
CXCMP.PRC

Proceed as follows:

```
$ @BLDXCMP
PRECOMPILE AND COMPILE A GROUP OF PRC's
-----
NAME OF THE APPLICATION>: CDMXCMP
NAME OF THE PRC FILE (C/R TO STOP, LEAVE.PRC OFF): CXISKM
NAME OF THE PRC FILE (C/R TO STOP, LEAVE.PRC OFF): CXCSKM
NAME OF THE PRC FILE (C/R TO STOP, LEAVE.PRC OFF): CXESKM
NAME OF THE PRC FILE (C/R TO STOP, LEAVE.PRC OFF): CXCISM
NAME OF THE PRC FILE (C/R TO STOP, LEAVE.PRC OFF): CXCESM
NAME OF THE PRC FILE (C/R TO STOP, LEAVE.PRC OFF): CXCMP

NAME OF THE PRC FILE (C/R TO STOP, LEAVE.PRC OFF): <CR>
NDML PRECOMPILE SUCCESSFULLY COMPLETED
BEGIN COMPILING GENERATED CODE
RESULTS OF COMPILE CAN BE FOUND ON CDMXCMP.MSG
```

3. Execute the procedure file LNKXCMP.COM to generate the RP-MAIN, compile the RP-MAIN and link the CDM Extract executable. This procedure file automatically updates the two NTM tables - APITBL.DAT and APTIBL.DAT. Proceed as follows:

\$ @LNKXCMP
4. Plan the file CXSCRN.FDL. The form definition files will be:

SDCXSCREEN.FD

SDCXSCHEMA.FD

5. Define the CDM Extract Function using the user interface utility SYSGEN. The name of the function is CDMXTRCT, the application is GRCDMXTRCT, the user name, password and role is CDM.

CDM COMPARE PHASE:

1. The following files must have been compiled according to the normal configuration management procedures and placed in the TOOLOBL.OLB object library:

CCMAIN.COB (Main Program)
CCSORT.COB
CCFPIO.COB
CCSFIO.COB
CCCOMP.COB

2. The CCMAIN.COB must be compiled and left as a local object. Using NDDL, run the Compare meta data into the CDM. Proceed as follows:

\$NDDL CDMCCDM.DAT

Examine CDMCCDM.OUT to assure that all NDDL commands completed successfully.

3. Create the Oracle Compare tables in the CDM. Proceed as follows:

\$UFI CDM/CDM

UFI>START CDMCORA.DAT
UFI>EXIT

4. Precompile and compile the following NDML embedded source code by executing the procedure file BLDCCMP.COM:

Proceed as follows:

\$@BLDCCMP

PRECOMPILE AND COMPILE A GROUP OF PRC's

NAME OF THE APPLICATION>: CDMCCMP
NAME OF THE PRC FILE (C/R TO STOP, LEAVE.PRC OFF): CCNDML
NAME OF THE PRC FILE (C/R TO STOP, LEAVE.PRC OFF): <CR>
NDML PRECOMPILE SUCCESSFULLY COMPLETED
BEGIN COMPILING GENERATED CODE
RESULTS OF COMPILING CAN BE FOUND ON CDMCCMP.MSG

5. Execute the procedure file LNKCCMP.COM to generate the RP-MAIN, compile the RP-MAIN and link the CDM Compare executable. This procedure file automatically updates the two NTM tables - APITBL.DAT and APTIBL.DAT. Proceed as follows:

\$ @LNKCCMP

6. Define the CDM Compare Function using the user interface utility SYSGEN. The name of the function is CDMCCMPZ. The name of the application is GRCDMCCMPZ, the user name, password and role is CDM.
7. Plan the file CCSCRN.FDL. The form definition file will be:

SDCCSCREEN.FD

CDM REPORT PHASE

1. The file CDMRRP.FDL must be processed by the Report Writer using the procedure file GENAP.COM. The following files are created:

CDMRRP.C	CDMRRP.PRC	SDCREPRT.FD
SCL.FD	IR.FD	

The NDML embedded source code generated is then precompiled and compiled automatically in the same procedure if no errors are encountered.

Proceed as follows:

\$ @BLDRCMP

GAP AND PRECOMPILE

ENTER NAME OF THE .FDL FILE (LEAVE .FDL OFF) CDMRRP
ENTER YOUR CDM USERNAME/PASSWORD

NDML PRECOMPILE SUCCESSFULLY COMPLETED
BEGIN COMPILING GENERATED CODE
RESULTS OF COMPILING CAN BE FOUND ON CDMRCMP.MSG

2. Execute the procedure file LNKRCMP.COM to generate the RP-MAIN, compile the RP-MAIN and link the CDM Report executable. This procedure automatically updates the two NTM tables - APITBL.DAT and APTIBL.DAT. Proceed as follows:

\$ @LNKRCMP

3. Define the CDM Compare Function using the user interface utility SYSGEN. The name of the function is CDMREPRT, the name of the application is GRCDMREPRT, the user name, password and role is CDM.

The following pages contain listings for:

BLDXCMP.COM
LNKXCMP.COM
BLDCCMP.COM
LNKCCMP.COM
BLDRCMP.COM
LNKRCMP.COM

```

$!
$!
$!      BLDXCMP.COM
$!
$!
$WS:= WRITE SYS$OUTPUT
$DEFINE IISSGLIB "CDMDIR:[TOOLS.COMP]GENOLB.OLB"
$DEFINE CDMTEMPS "CDMDIR:[TOOLS.COMP.TEMPS]"
$WS "PRECOMPILE AND COMPILE A GROUP OF PRC'S"
$WS "-----"
$!
$INQUIRE AP " NAME OF THE APPLICATION>"
$ CREATE 'AP'.DAT
$ OPEN/WRITE NDMLIN 'AP'.IN
$NEXT:
$ INQUIRE PRC "NAME OF PRC FILE (C/R TO STOP, LEAVE .PRC OFF)"
$ IF PRC .EQS. "" THEN GOTO INDONE
$ APPEND CDMDIR:[TOOLS.COMP]'PRC'.PRC 'AP'.DAT
$ GOTO NEXT
$ INDONE:
$!
$!      5/24/88: FDL stuff added in because GENRPD now requires fixed
$!              length .DAT files since conversion to FIOPS.
$!
$ OPEN/WRITE FDLIN CDMDIR:[COM]FIX.FDL
$ WRITE FDLIN "IDENT      ""23-FEB-1988 09:49:43      VAX-V1 FDL
Editor""
$ WRITE FDLIN "      "
$ WRITE FDLIN "SYSTEM"
$ WRITE FDLIN "      SOURCE      VAX/VMS"
$ WRITE FDLIN "      "
$ WRITE FDLIN "FILE"
$ WRITE FDLIN "      ALLOCATION      391"
$ WRITE FDLIN "      BEST TRY CONTIGUOUS  yes"
$ WRITE FDLIN "      EXTENSION      39"
$ WRITE FDLIN "      ORGANIZATION      sequential"
$ WRITE FDLIN "      "
$ WRITE FDLIN "RECORD"
$ WRITE FDLIN "      BLOCK_SPAN      yes"
$ WRITE FDLIN "      CARRIAGE_CONTROL  carriage_return"
$ WRITE FDLIN "      FORMAT      fixed"
$ WRITE FDLIN "      SIZE      80"
$ CLOSE FDLIN
$ CONVERT/PAD=%040/FDL=CDMDIR:[COM]FIX.FDL 'AP'.DAT 'AP'.DAT
$ WRITE NDMLIN "CDMXTRCT VAX VAX COBOL NDML COBOL ",AP,".DAT
",AP,".ERR ", "CDM/CDM", " FD=N"
$ CLOSE NDMLIN
$ OPEN/WRITE EDIPRC CDMDIR:[COM]EDIT.PROC
$ WRITE EDIPRC "S/80/200/WH"
$ WRITE EDIPRC "EXIT"
$ CLOSE EDIPRC
$ EDIT/COMMAND=CDMDIR:[COM]EDIT.PROC CDMDIR:[COM]FIX.FDL
$ DEASSIGN SYS$OUTPUT
$ CONVERT/PAD=%040/FDL=CDMDIR:[COM]FIX.FDL 'AP'.IN 'AP'.IN
$ DELETE CDMDIR:[COM]FIX.FDL;*, CDMDIR:[COM]EDIT.PROC;*
$!
$!      INPUTS TO PRECOMPILER ARE NOW SET UP
$!      GO AHEAD AND RUN IT:

```

```
$!  
$ ASSIGN/USER MODE SYS$COMMAND SYS$INPUT  
$ RENAME 'AP'.IN NDML.DAT  
$ ASSIGN 'AP'.OUT SYS$OUTPUT  
$ RUNNDML  
$ RENAME NDML.DAT 'AP'.IN  
$!  
$ ALLDONE:  
$ DEASSIGN SYS$OUTPUT  
$!  
$! check the .out file for errors in precompiling  
$!  
$OPEN/READ EFLE 'AP'.OUT  
$ZR="0"  
$ NERRLOOP:  
$ READ/END_OF_FILE=COMPERR EFLE EREC  
$ LENG = 'F$LENGTH(EREC)'  
$ UN = 'F$LOCATE("UNSUCC",EREC)'  
$ IF 'UN'.EQS. 'LENG' THEN GOTO NERRLOOP  
$ UN1 = 'UN' - 13  
$ UN2 = 'F$EXTRACT(UN1,1,EREC)'  
$ IF UN2.EQS. ZR THEN GOTO NDMLGOOD  
$WS "THE PRECOMPILE OF 'AP' HAS 'UN2' UNSUCCESSFUL ROUTINES"  
$WS "CHECK THE 'AP'.ERR FILE FOR ERRORS"  
$GOTO EXIT  
$COMPERR:  
$WS "PRECOMPILE FAILED"  
$GOTO EXIT  
$!  
$! the precompile was successful, compile the code  
$!  
$ NDMLGOOD:  
$WS "  
$WS "NDML PRECOMPILE SUCCESSFULLY COMPLETED"  
$WS "BEGIN COMPILING GENERATED CODE"  
$ DELETE 'AP'.DAT;*  
$ NDMLGDRD:  
$ READ/END_OF_FILE = COMPERR EFLE EREC  
$ LENG = 'F$LENGTH(EREC)'  
$ UN = 'F$LOCATE("==> USE", EREC)'  
$ IF 'UN'.EQS. 'LENG' THEN GOTO NDMLGDRD  
$!  
$ASSIGN 'AP'.MSG SYS$OUTPUT  
$UN1 = 'UN' + 8  
$NNAM='F$EXTRACT(UN1,30,EREC)'  
$CLOSE EFLE  
$ @'NNAM'  
$DEASSIGN SYS$OUTPUT  
$WS "RESULTS OF COMPILE CAN BE FOUND ON 'AP'.MSG"  
$EXIT:  
$DEFINE IISGLIB "CDMDIR:[TEST]GENOLB.OLB"
```

```
$!  
$!          LNKXCMP.COM  
$!          THIS USES ORACLE VERSION 5.1  
$!  
$!  
$!          $DEFINE/NOLOG TOOLOLB   CDMDIR:[TOOLS.COMP]COMPOLB  
$!          $DEFINE/NOLOG IISSGLIB "CDMDIR:[TOOLS.COMP]GENOLB.OLB"  
$!          $DEFINE CDMTEMPS "CDMDIR:[TOOLS.COMP.TEMPS]"  
$!          $if p1 .eqs. "N" then goto linkXCMP  
$!          $WS="WRITE SYS$OUTPUT"  
$!          $ LINK:  
$!          $DEASSIGN SYS$OUTPUT  
$!          $WS " "  
$!          $WS "Beginning Generation of Rp-Main"  
$!          $LUW="CDMXTRCT"  
$!          $CDM="CDM/CDM"  
$!  
$!          generate the rp-main  
$!          NOTE:  this is done automatically if you link  
$!  
$!  
$!          set up .dat file to send to genrpd  
$!  
$!          $OPEN/WRITE GENRPD.DAT GENRPD.DAT  
$!          $WRITE GENRPD.DAT LUW, " ", CDM, " VAX"  
$!          $CLOSE GENRPD.DAT  
$!          $ OPEN/WRITE FDLIN FIX.FDL  
$!          $ WRITE FDLIN "IDENT    ""23-FEB-1988 09:49:43    VAX-11 FDL  
$!          Editor""  
$!          $ WRITE FDLIN "      "  
$!          $ WRITE FDLIN "SYSTEM"  
$!          $ WRITE FDLIN "          SOURCE                      VAX/VMS"  
$!          $ WRITE FDLIN "      "  
$!          $ WRITE FDLIN "FILE"  
$!          $ WRITE FDLIN "          ALLOCATION                      3"  
$!          $ WRITE FDLIN "          BEST TRY CONTIGUOUS          yes"  
$!          $ WRITE FDLIN "          EXTENSION                    39"  
$!          $ WRITE FDLIN "          ORGANIZATION                  sequential"  
$!          $ WRITE FDLIN "      "  
$!          $ WRITE FDLIN "RECORD"  
$!          $ WRITE FDLIN "          BLOCK_SPAN                      yes"  
$!          $ WRITE FDLIN "          CARRIAGE_CONTROL              carriage_return"  
$!          $ WRITE FDLIN "          FORMAT                        fixed"  
$!          $ WRITE FDLIN "          SIZE                          80"  
$!          $ CLOSE FDLIN  
$!          $ CONVERT/PAD=%040/FDL=FIX GENRPD.DAT GENRPD.DAT  
$!  
$!          now run genrpd  
$!  
$!          $ASSIGN/USER_MODE SYS$COMMAND SYS$INPUT  
$!          $ASSIGN 'LUW'.RPD SYS$OUTPUT  
$!          $RUNGENRPD  
$!          $DEASSIGN SYS$OUTPUT  
$!          $DELETE GENRPD.DAT;*, FIX.FDL;*  
$!  
$!          now get the needed information to compile the rp-main(s)  
$!
```

```

$ASSIGN 'LUW'.RDCOMP SYS$OUTPUT
$GENRPFDFLAG = 0
$OPEN/READ EFLE 'LUW'.RPD
$ RDLOOP:
$READ/END OF FILE=ENDMAIN EFLE EREC
$ LENG = 'F$LENGTH(EREC)'
$ DBMS = 'F$LOCATE("FOR DBMS", EREC)'
$ UN = 'F$LOCATE("STORED ON", EREC)'
$ MN = 'F$LOCATE("MODULE", EREC)'
$ DB = 'F$LOCATE("DATA BASE", EREC)'
$ RM = 'F$LOCATE("REMOTE/", EREC)'
$ HST = 'F$LOCATE("RUN AT", EREC)'
$ IF 'MN' .NES. 'LENG' THEN GOTO SAVMODNM
$ IF 'DB' .NES. 'LENG' THEN GOTO SAVDBN
$ IF 'RM' .NES. 'LENG' THEN GOTO RMLC
$ IF 'HST' .NES. 'LENG' THEN GOTO SAVEHST
$ IF 'DBMS' .NES. 'LENG' THEN GOTO SAVEDBMS
$ IF 'UN' .EQS. 'LENG' THEN GOTO RDLOOP
$GENRPFDFLAG = 1
$UN1 = 'UN' + 16
$UNEND = 'F$LOCATE(".", EREC) - UN1
$PL := 'F$EXTRACT(UN1, UNEND, EREC)'
$IF DBMSNM .EQS. "ORACLE" THEN GOTO MAINPCC
$WS "A NEW DBMS TYPE MUST BE ADDED TO THE MAIN COMPILE PART OF THIS
PROCEDURE"
$GOTO EXIT
$!
$! get the rp-main mod name
$!
$ SAVMODNM:
$MN1 = 'MN' + 7
$RPMN1 := 'F$EXTRACT(MN1, 9, EREC)'
$LENG = 'F$LENGTH(RPMN1)'
$MN2 = 'F$LOCATE("ZZZ", RPMN1)'
$IF 'MN2' .EQS. 'LENG' THEN MODLOC = 0
$IF 'MN2' .NES. 'LENG' THEN MODLOC = 2
$RPMN := 'F$EXTRACT(MODLOC, 5, RPMN1)'
$GOTO RDLOOP
$!
$! get the remote/local status
$!
$ RMLC:
$RM1 = 'RM' + 13
$RMSW := 'F$EXTRACT(RM1, 1, EREC)'
$GOTO RDLOOP
$!
$! get the database name
$!
$ SAVDBN:
$DB1 = 'DB' + 10
$DBN := 'F$EXTRACT(DB1, 30, EREC)'
$GOTO RDLOOP
$!
$! get the host name
$!
$ SAVEHST:
$HST1 = 'HST' + 7
$HSTNM := 'F$EXTRACT(HST1, 3, EREC)'
$GOTO RDLOOP

```

```
$!  
$!  get the dbms name  
$!  
$  SAVEDBMS:  
$DBMS1 = 'DBMS' + 9  
$DBMSNM := 'F$EXTRACT(DBMS1, 30, EREC)  
$GOTO RDLOOP  
$!  
$!  
$!  oracle precompile the rp-main (if needed)  
$!  
$  MAINPCC:  
$PCC INAME='PL'.TMP LNAME='PL'.ERR USERID='CDM' -  
  ONAME='PL'.COB INCLUDE=SYS$ORACLE: HOST=COB74 MAXLITERAL=160  
REBIND=YES  
$ON ERROR THEN WS "ORACLE ERROR IN RP-MAIN ''PL'.TMP"  
$ON ERROR THEN GOTO EXIT  
$COBOL/ANSI_FORMAT/CHECK=ALL/COPY_LIST/CROSS_REFERENCE/OBJECT='PL'.  
OBJ -  
  /FIPS=74/NOLIST/CHECK=ALL/STANDARD=(SYNTAX)/DEBUG=ALL 'PL'.COB  
$ON ERROR THEN WS "COBOL ERROR IN RP-MAIN ''PL'.TMP"  
$ON ERROR THEN GOTO EXIT  
$DELETE 'PL'.COB;*  
$DELETE 'PL'.ERR;*  
$!  
$!  Compile the rpmain.c  
$!  
$vcc/debug/NOLIST/show=(include)/standard=portable -  
  /noopt/OBJECT=CDMDIR:[TOOLS.COMP]RPMAIN.OBJ/DEFINE=VAX  
RPMAIN.C  
$delete rpmain.c;*  
$GOTO RDLOOP  
$!  
$!  done generating rp-main  
$!  
$  ENDMAIN:  
$CLOSE EFLE  
$DEASSIGN SYS$OUTPUT  
$IF GENRPDFLAG .EQ. 0 THEN GOTO MAINERR  
$WS " "  
$WS "GENERATION OF REQUEST PROCESSOR MAIN COMPLETE"  
$GOTO STARTLINK  
$!  
$!  there was an error in generating the rp-main  
$!  
$  MAINERR:  
$WS "THE GENRPD HAD ERRORS.  EXAMINE ''LUW'.RPD"  
$GOTO EXIT  
$STARTLINK:  
$DELETE 'LUW'.RPD;*  
$DELETE 'LUW'.RDCOMP;*  
$DELETE CMDIR:[RUNAREA]CDMXTRCT.EXE;*  
$WRITE SYS$OUTPUT " - LINKING CDMXTRCT.EXE "  
$ASSIGN CDMXTRCT.LINK SYS$OUTPUT  
$!  
$! inquire p5 "ENTER TWO LETTER NTM DIRECTORY PREFIX "  
$! inquire p6 "ENTER NTM CLUSTER FOR THIS RP (TIV OR UIV) "  
$!  
$P5 = "GR"
```



```

$P6 = "UIV"
$P1 = "CDMXTRCT"
$!
$! NTMTAB.COM
$!
$! 15-APR-87
$! M. DENMAN
$!
$! UPDATE NTM TABLES APITBL.DAT, APTTBL.DAT
$!
$!
$ FLAG=0
$ P7:=P5+P1
$ OPEN/READ APITBL.DAT CMDIR:[RUNAREA]APITBL.DAT
$ RD1:
$ READ/END OF FILE=CHK APITBL.DAT ENTRY
$ RPND=F$EXTRACT(0,10,ENTRY)
$ IF RPND .NES. P7 THEN GOTO RD1
$ FLAG=1
$ CHK:
$ CLOSE APITBL.DAT
$ IF FLAG .NES. 0 THEN GOTO NOUPD
$ GOTO UPD
$ NOUPD:
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "RP MAIN ALREADY IN NTM TABLES"
$ WRITE SYS$OUTPUT " "
$ GOTO LINKXCMP
$ UPD:
$ RPAPI=P7+P6+"1"
$ RPAPT=P1+"9999010120001130N0"
$ OPEN/APPEND APITBL.DAT CMDIR:[RUNAREA]APITBL.DAT
$ OPEN/APPEND APTTBL.DAT CMDIR:[RUNAREA]APTTBL.DAT
$ WRITE APITBL.DAT RPAPI
$ WRITE APTTBL.DAT RPAPT
$ CLOSE APITBL.DAT
$ CLOSE APTTBL.DAT
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "NTM TABLES UPDATED WITH RP ",P1
$ WRITE SYS$OUTPUT " "
$!
$!DEFINE CDMROLB "CDMDIR:[CDMR]CDMROLB"
$LINKXCMP:
$@SYS$ORACLE:LFOR CMDIR:[RUNAREA]CDMXTRCT -
CDMDIR:[TOOLS.COMP]CXMAIN.OBJ,-
CDMDIR:[TOOLS.COMP]RPMAIN.OBJ,-
'PL'.OBJ,-
CDMDIR:[COM]CDMI/OPTIONS,-
SYS$ORACLE:SQLLIB/LIB,-
CDMDIR:[COM]CDMUI.OPT/OPT,-
CDMDIR:[COM]CDMNTM.OPT/OPTIONS N
$DEASSIGN SYS$OUTPUT
$DELETE CDMXTRCT.LINK;*, RPMAIN.OBJ;*, 'PL'.OBJ;*
$WRITE SYS$OUTPUT "LINKING COMPLETED"
$EXIT:
$DEFINE/NOLOG IISSGLIB "CDMDIR:[TEST]GENOLB.OLB"
$DEASSIGN TOOLLB
$!

```

```

$!
$!
$!      BLDCCMP.COM
$!
$!
$WS:= WRITE SYS$OUTPUT
$DEFINE IISSGLIB "CDMDIR:[TOOLS.COMP]GENOLB.OLB"
$DEFINE CDMTEMPS "CDMDIR:[TOOLS.COMP.TEMPS]"
$WS "PRECOMPILE AND COMPILE A GROUP OF PRC'S"
$WS "-----"
$!
$INQUIRE AP " NAME OF THE APPLICATION>"
$ CREATE 'AP'.DAT
$ OPEN/WRITE NDMLIN 'AP'.IN
$NEXT:
$ INQUIRE PRC "NAME OF PRC FILE (C/R TO STOP, LEAVE .PRC OFF)"
$ IF PRC .EQS. "" THEN GOTO INDONE
$ APPEND CDMDIR:[TOOLS.COMP]'PRC'.PRC 'AP'.DAT
$ GOTO NEXT
$ INDONE:
$!
$! 5/24/88: FDL stuff added in because GENRPD now requires fixed
$! length .DAT files since conversion to FIOPS.
$!
$ OPEN/WRITE FDLIN CDMDIR:[COM]FIX.FDL
$ WRITE FDLIN "IDENT "23-FEB-1988 09:49:43 VAX-11 FDL
Editor""
$ WRITE FDLIN " "
$ WRITE FDLIN "SYSTEM"
$ WRITE FDLIN " SOURCE VAX/VMS"
$ WRITE FDLIN " "
$ WRITE FDLIN "FILE"
$ WRITE FDLIN " ALLOCATION 391"
$ WRITE FDLIN " BEST_TRY_CONTIGUOUS yes"
$ WRITE FDLIN " EXTENSION 39"
$ WRITE FDLIN " ORGANIZATION sequential"
$ WRITE FDLIN " "
$ WRITE FDLIN "RECORD"
$ WRITE FDLIN " BLOCK_SPAN yes"
$ WRITE FDLIN " CARRIAGE_CONTROL carriage_return"
$ WRITE FDLIN " FORMAT fixed"
$ WRITE FDLIN " SIZE 80"
$ CLOSE FDLIN
$ CONVERT/PAD=%040/FDL=CDMDIR:[COM]FIX.FDL 'AP'.DAT 'AP'.DAT
$ WRITE NDMLIN "CDMCMPPAR VAX VAX COBOL NDML COBOL"," ",AP,".DAT
",AP,".ERR ","CDM/CDM"," FD=Y"
$ OPEN/WRITE EDIPRC CDMDIR:[COM]EDIT.PROC
$ WRITE EDIPRC "S/80/200/WH"
$ WRITE EDIPRC "EXIT"
$ CLOSE EDIPRC
$ CLOSE NDMLIN
$ EDIT/COMMAND=CDMDIR:[COM]EDIT.PROC CDMDIR:[COM]FIX.FDL
$ DEASSIGN SYS$OUTPUT
$ CONVERT/PAD=%040/FDL=CDMDIR:[COM]FIX.FDL 'AP'.IN 'AP'.IN
$ DELETE CDMDIR:[COM]FIX.FDL;*, CDMDIR:[COM]EDIT.PROC;*
$!
$! INPUTS TO PRECOMPIER ARE NOW SET UP

```

```
$! GO AHEAD AND RUN IT:
$!
$ ASSIGN/USER MODE SYS$COMMAND SYS$INPUT
$ RENAME 'AP'.IN NDML.DAT
$ ASSIGN 'AP'.OUT SYS$OUTPUT
$ RUN CMDIR:[RUNAREA]NDML
$ RENAME NDML.DAT 'AP'.IN
$!
$ ALLDONE:
$ DEASSIGN SYS$OUTPUT
$!
$! check the .out file for errors in precompiling
$!
$OPEN/READ EFLE 'AP'.OUT
$ZR:="0"
$ NERRLOOP:
$ READ/END OF FILE=COMPERR EFLE EREC
$ LENG = 'F$LENGTH(EREC)'
$ UN = 'F$LOCATE("UNSUCC",EREC)'
$ IF 'UN' .EQS. 'LENG' THEN GOTO NERRLOOP
$ UN1 = 'UN' - 13
$ UN2 = 'F$EXTRACT(UN1,1,EREC)'
$ IF UN2 .EQS. ZR THEN GOTO NDMLGOOD
$WS "THE PRECOMPILE OF 'AP' HAS 'UN2' UNSUCCESSFUL ROUTINES"
$WS "CHECK THE 'AP'.ERR FILE FOR ERRORS"
$GOTO EXIT
$COMPERR:
$WS "PRECOMPILE FAILED"
$GOTO EXIT
$!
$! the precompile was successful, compile the code
$!
$ NDMLGOOD:
$WS " "
$WS "NDML PRECOMPILE SUCCESSFULLY COMPLETED"
$WS "BEGIN COMPILING GENERATED CODE"
$DELETE 'AP'.DAT;*
$ NDMLGDRD:
$ READ/END OF FILE = COMPERR EFLE EREC
$ LENG = 'F$LENGTH(EREC)'
$ UN = 'F$LOCATE("==> USE", EREC)'
$ IF 'UN' .EQS. 'LENG' THEN GOTO NDMLGDRD
$!
$ASSIGN 'AP'.MSG SYS$OUTPUT
$UN1 = 'UN' + 8
$NNAM:='F$EXTRACT(UN1,30,EREC)'
$CLOSE EFLE
$ @'NNAM'
$DEASSIGN SYS$OUTPUT
$WS "RESULTS OF COMPILE CAN BE FOUND ON 'AP'.MSG"
$EXIT:
$DEFINE IISSGLIB "CDMDIR:[TEST]GENOLB.OLB"
```

```
$!  
$! LNKCCMP.COM  
$! THIS USES ORACLE VERSION 5.1  
$!  
$!  
$!  
$!  
$DEFINE/NOLOG TOOLOLB CDMDIR:[TOOLS.COMP]COMPOLB  
$DEFINE/NOLOG IISSGLIB "CDMDIR:[TOOLS.COMP]GENOLB.OLB"  
$DEFINE CDMTEMPS "CDMDIR:[TOOLS.COMP.TEMPS]"  
$if pl .eqs. "N" then goto linkCCMP  
$WS="WRITE SYS$OUTPUT"  
$ LINK:  
$DEASSIGN SYS$OUTPUT  
$WS " "  
$WS "Beginning Generation of Rp-Main"  
$LUW="CDMCMPAR"  
$CDM="CDM/CDM"  
$!  
$! generate the rp-main  
$! NOTE: this is done automatically if you link  
$!  
$!  
$! set up .dat file to send to genrpd  
$!  
$OPEN/WRITE GENRPD.DAT GENRPD.DAT  
$WRITE GENRPD.DAT LUW, " ", CDM, " VAX"  
$CLOSE GENRPD.DAT  
$ OPEN/WRITE FDLIN FIX.FDL  
$ WRITE FDLIN "IDENT ""23-FEB-1988 09:49:43 VAX-11 FDL  
Editor""  
$ WRITE FDLIN " "  
$ WRITE FDLIN "SYSTEM"  
$ WRITE FDLIN " SOURCE VAX/VMS"  
$ WRITE FDLIN " "  
$ WRITE FDLIN "FILE"  
$ WRITE FDLIN " ALLOCATION 3"  
$ WRITE FDLIN " BEST_TRY CONTIGUOUS yes"  
$ WRITE FDLIN " EXTENSION 39"  
$ WRITE FDLIN " ORGANIZATION sequential"  
$ WRITE FDLIN " "  
$ WRITE FDLIN "RECORD"  
$ WRITE FDLIN " BLOCK_SPAN yes"  
$ WRITE FDLIN " CARRIAGE_CONTROL carriage_return"  
$ WRITE FDLIN " FORMAT fixed"  
$ WRITE FDLIN " SIZE 80"  
$ CLOSE FDLIN  
$ CONVERT/PAD=%O40/FDL=FIX GENRPD.DAT GENRPD.DAT  
$!  
$! now run genrpd  
$!  
$ASSIGN/USER MODE SYS$COMMAND SYS$INPUT  
$ASSIGN 'LUW'.RPD SYS$OUTPUT  
$RUNGENRPD  
$DEASSIGN SYS$OUTPUT  
$DELETE GENRPD.DAT;*, FIX.FDL;*  
$!  
$! now get the needed information to compile the rp-main(s)
```

```
$!  
$ASSIGN 'LUW'.RDCOMP SYS$OUTPUT  
$GENRPDFLAG = 0  
$OPEN/READ EFLE 'LUW'.RPD  
$ RDLOOP:  
$READ/END_OF_FILE=ENDMAIN EFLE EREC  
$ LENG = 'F$LENGTH(EREC)'  
$ DBMS = 'F$LOCATE("FOR DBMS", EREC)'  
$ UN = 'F$LOCATE("STORED ON", EREC)'  
$ MN = 'F$LOCATE("MODULE", EREC)'  
$ DB = 'F$LOCATE("DATA BASE", EREC)'  
$ RM = 'F$LOCATE("REMOTE/", EREC)'  
$ HST = 'F$LOCATE("RUN AT", EREC)'  
$ IF 'MN' .NES. 'LENG' THEN GOTO SAVMODNM  
$ IF 'DB' .NES. 'LENG' THEN GOTO SAVDBN  
$ IF 'RM' .NES. 'LENG' THEN GOTO RMLC  
$ IF 'HST' .NES. 'LENG' THEN GOTO SAVEHST  
$ IF 'DBMS' .NES. 'LENG' THEN GOTO SAVEDBMS  
$ IF 'UN' .EQS. 'LENG' THEN GOTO RDLOOP  
$GENRPDFLAG = 1  
$UN1 = 'UN' + 16  
$UNEND = 'F$LOCATE(".", EREC) - UN1  
$PL := 'F$EXTRACT(UN1, UNEND, EREC)'  
$IF DBMSNM .EQS. "ORACLE" THEN GOTO MAINPCC  
$WS "A NEW DBMS TYPE MUST BE ADDED TO THE MAIN COMPILE PART OF THIS  
PROCEDURE"  
$GOTO EXIT  
$!  
$! get the rp-main mod name  
$!  
$ SAVMODNM:  
$MN1 = 'MN' + 7  
$RPMN1 := 'F$EXTRACT(MN1, 10, EREC)'  
$LENG = 'F$LENGTH(RPMN1)'  
$MN2 = 'F$LOCATE("ZZZ", RPMN1)  
$IF 'MN2' .EQS. 'LENG' THEN MODLOC = 0  
$IF 'MN2' .NES. 'LENG' THEN MODLOC = 2  
$RPMN := 'F$EXTRACT(MODLOC, 5, RPMN1)  
$GOTO RDLOOP  
$!  
$! get the remote/local status  
$!  
$ RMLC:  
$RM1 = 'RM' + 13  
$RMSW := 'F$EXTRACT(RM1, 1, EREC)'  
$GOTO RDLOOP  
$!  
$! get the database name  
$!  
$ SAVDBN:  
$DB1 = 'DB' + 10  
$DBN := 'F$EXTRACT(DB1, 30, EREC)'  
$GOTO RDLOOP  
$!  
$! get the host name  
$!  
$ SAVEHST:  
$HST1 = 'HST' + 7  
$HSTNM := 'F$EXTRACT(HST1, 3, EREC)'
```

```
$GOTO RDLOOP
$!
$!  get the dbms name
$!
$  SAVEDBMS:
$DBMS1 = 'DBMS' + 9
$DBMSNM := 'F$EXTRACT(DBMS1, 30, EREC)
$GOTO RDLOOP
$!
$!
$!  oracle precompile the rp-main (if needed)
$!
$  MAINPCC:
$PCC INAME='PL'.TMP LNAME='PL'.ERR USERID='CDM' -
  ONAME='PL'.COB INCLUDE=SYS$ORACLE: HOST=COB74 MAXLITERAL=160
REBIND=YES
$ON ERROR THEN WS "ORACLE ERROR IN RP-MAIN ''PL'.TMP"
$ON ERROR THEN GOTO EXIT
$COBOL/ANSI_FORMAT/CHECK=ALL/COPY_LIST/CROSS_REFERENCE/OBJECT='PL'.
OBJ -
  /FIPS=74/NOLIST/CHECK=ALL/STANDARD=(SYNTAX)/DEBUG=ALL 'PL'.COB
$ON ERROR THEN WS "COBOL ERROR IN RP-MAIN ''PL'.TMP"
$ON ERROR THEN GOTO EXIT
$DELETE 'PL'.COB;*
$DELETE 'PL'.ERR;*
$!
$!  Compile the rpmain.c
$!
$vccl/debug/list=RPMAIN.lis/show=(include)/standard=portable -
  /noopt/OBJECT=CDMDIR:[TOOLS.COMP]RPMAIN.OBJ/DEFINE=VAX
RPMAIN.C
$delete rpmain.c;*, rpmain.lis;*
$GOTO RDLOOP
$!
$!  done generating rp-main
$!
$  ENDMAIN:
$CLOSE EFLE
$DEASSIGN SYS$OUTPUT
$IF GENRPDFLAG .EQ. 0 THEN GOTO MAINERR
$WS " "
$WS "GENERATION OF REQUEST PROCESSOR MAIN COMPLETE"
$GOTO STARTLINK
$!
$!  there was an error in generating the rp-main
$!
$  MAINERR:
$WS "THE GENRPD HAD ERRORS.  EXAMINE ''LUW'.RPD"
$GOTO EXIT
$STARTLINK:
$DELETE 'LUW'.RPD;*
$DELETE 'LUW'.RDCOMP;*
$DELETE CMDIR:[RUNAREA]CDMCCMPZ.EXE;*
$WRITE SYS$OUTPUT " - LINKING CDMCCMPZ.EXE "
$ASSIGN CDMCCMPZ.LINK SYS$OUTPUT
$!
$! inquire p5 "ENTER TWO LETTER NTM DIRECTORY PREFIX "
$! inquire p6 "ENTER NTM CLUSTER FOR THIS RP (TIV OR UIV) "
$!
```

```
$P5 = "GR"
$P6 = "UIV"
$P1 = "CDMCCMP"
$!
$!   NTMTAB.COM
$!
$!
$!   UPDATE NTM TABLES APITBL.DAT, APTTBL.DAT
$!
$!
$ FLAG=0
$ P7=P5+P1
$ OPEN/READ APITBL.DAT CDMDIR:[RUNAREA]APITBL.DAT
$ RD1:
$ READ/END OF FILE=CHK APITBL.DAT ENTRY
$ RPND=F$EXTRACT(0,10,ENTRY)
$ IF RPND .NES. P7 THEN GOTO RD1
$ FLAG=1
$ CHK:
$ CLOSE APITBL.DAT
$ IF FLAG .NES. 0 THEN GOTO NOUPD
$ GOTO UPD
$ NOUPD:
$ WRITE SYS$OUTPUT "  "
$ WRITE SYS$OUTPUT "RP MAIN ",P1," ALREADY IN NTM TABLES"
$ WRITE SYS$OUTPUT "  "
$ GOTO LINKCCMP
$ UPD:
$ RPAPI=P7+"Z"+P6+"1"
$ RPAPT=P1+"Z9999010120001130N0"
$ OPEN/APPEND APITBL.DAT CDMDIR:[RUNAREA]APITBL.DAT
$ OPEN/APPEND APTTBL.DAT CDMDIR:[RUNAREA]APTTBL.DAT
$ WRITE APITBL.DAT RPAPI
$ WRITE APTTBL.DAT RPAPT
$ CLOSE APITBL.DAT
$ CLOSE APTTBL.DAT
$ WRITE SYS$OUTPUT "  "
$ WRITE SYS$OUTPUT "NTM TABLES UPDATED WITH RP ",P1
$ WRITE SYS$OUTPUT "  "
$!
$!DEFINE CDMROLB "CDMDIR:[CDMR]CDMROLB"
$LINKCCMP:
$@SYS$ORACLE:LFOR CDMDIR:[RUNAREA]CDMCCMPZ -
CDMDIR:[TOOLS.COMP]CCMAIN.OBJ,-
CDMDIR:[TOOLS.COMP]RPMAIN.OBJ,-
'PL'.OBJ,-
CDMDIR:[COM]CDMI/OPTIONS,-
SYS$ORACLE:SQLLIB/LIB,-
CDMDIR:[COM]CDMUI.OPT/OPT,-
CDMDIR:[COM]CDMNTM.OPT/OPTIONS N
$DEASSIGN SYS$OUTPUT
$DELETE CDMCCMPZ.LINK;*, RPMAIN.OBJ;*, 'PL'.OBJ;*
$WRITE SYS$OUTPUT "LINKING COMPLETED"
$EXIT:
$DEFINE/NOLOG IISSGLIB "CDMDIR:[TEST]GENOLB.OLB"
$DEASSIGN TOOLOLB
$!
```

```
$!  
$!      BLDRCMP.COM  
$!  
$!  
$DEFINE IISSGLIB "CDMDIR:[TOOLS.COMP]GENOLB.OLB"  
$DEFINE CDMTEMPS "CDMDIR:[TOOLS.COMP.TEMPS]"  
$WS:= WRITE SYS$OUTPUT  
$!  
$!  
$WS "   GAP AND PRECOMPILE"  
$WS " "  
$ GP:  
$WS " "  
$!  
$INQUIRE FIL "Enter Name Of FDL File (Leave .FDL Off)"  
$INQUIRE CDM "Enter Your CDM Username/Password"  
$LUW = "CDMREPRT"  
$HOST = "VAX"  
$DEL = "Y"  
$!  
$CREATE 'LUW'.DAT  
$ GAP:  
$!  
$WS " "  
$WS "BEGINNING GAP"  
$!  
$!  
$!  
$!  Get the application name out of the file to be gapped.  
$!  
$OPEN/READ FILE CDMDIR:[TOOLS.COMP]'FIL'.FDL  
$  READFDL:  
$READ/END OF FILE=FDLERR FILE REC  
$REC = F$EDIT(REC, "UPCASE")  
$LEN = F$LENGTH(REC)  
$LOC1 = F$LOCATE("APPLICATION", REC)  
$FLAG = "A"  
$LOC = 'LOC1' + 10  
$IF 'LOC1' .NES. 'LEN' THEN GOTO ENDREAD  
$LOC1 = F$LOCATE("REPORT", REC)  
$FLAG = "R"  
$LOC = 'LOC1' + 5  
$ IF 'LOC1' .EQS. 'LEN' THEN GOTO READFDL  
$ENDREAD:  
$CLOSE FILE  
$FILELOOP:  
$LOC = 'LOC' + 1  
$TEMP = F$EXTRACT(LOC, 1, REC)  
$IF TEMP .EQS. " " THEN GOTO FILELOOP  
$APNAME = F$EXTRACT(LOC, LEN, REC)  
$IF FLAG .EQS. "A" THEN END = F$LOCATE(" ", APNAME)  
$IF FLAG .EQS. "R" THEN END = F$LOCATE("(", APNAME)  
$AP = F$EXTRACT(LOC, END, REC)  
$!  
$!  This trim is done in case the user put spaces between the  
report name  
$!  and the parameter form
```



```
$!  
$IF FLAG .EQS. "R" THEN AP = F$EDIT(AP, "TRIM")  
$!  
$! Create the .dat file to send to gap  
$!  
$OPEN/WRITE ADLIN GAP.DAT  
$WRITE ADLIN "CDMDIR:[TOOLS.COMP]", FIL, ".FDL"  
$WRITE ADLIN CDM  
$CLOSE ADLIN  
$!  
$! Run gap  
$!  
$ASSIGN 'AP'.OUT SYS$OUTPUT  
$ASSIGN GAP.DAT SYS$INPUT  
$ASSIGN 'AP'.ERR SYS$ERROR  
$GAP1  
$DEASSIGN SYS$INPUT  
$DEASSIGN SYS$ERROR  
$DEASSIGN SYS$OUTPUT  
$!  
$! Read the error file from gapping  
$!  
$OPEN/READ EFLE 'AP'.ERR  
$ERSW:="N"  
$ LOOPERR:  
$ READ/END OF FILE=ENDERR EFLE EREC  
$ LENG = 'F$LENGTH(EREC)'  
$ ERIN = 'F$LOCATE("ERR",EREC)'  
$ IF 'ERIN' .EQS. 'LENG' THEN GOTO LOOPERR  
$ ERSW:="Y"  
$ ENDERR:  
$CLOSE EFLE  
$IF ERSW .EQS. "Y" THEN GOTO GAPERR  
$!DELETE 'AP'.ERR;*  
$!DELETE 'AP'.OUT;*  
$!DELETE GAP.DAT;*  
$WS " "  
$WS "GAP SUCCESSFULLY COMPLETED"  
$!  
$! C compile  
$!  
$!  
$! TRUNCATE THE APNAME TO 6 CHARACTERS BECAUSE GAP DOES.  
$!  
$SHORTAP = F$EXTRACT(0, 6, AP)  
$ASSIGN 'SHORTAP'.OUT SYS$OUTPUT  
$vcc/debug/list='shortap'.lis/standard=portable/noopt 'SHORTAP'  
$DEASSIGN SYS$OUTPUT  
$ON ERROR THEN WS "ERROR IN GENERATED C PROGRAM"  
$ ON ERROR THEN GOTO EXIT  
$LIB/REPLACE/NOGLOBAL IISSGLIB 'SHORTAP'.OBJ  
$!DELETE 'SHORTAP'.OUT;*  
$DELETE 'SHORTAP'.OBJ;*  
$DELETE 'SHORTAP'.LIS;*  
$PURGE 'SHORTAP'.C  
$PURGE 'SHORTAP'.PRC  
$WS " "  
$WS "C PROGRAM SUCCESSFULLY COMPILED AND INSERTED INTO IISSOLIB"  
$!
```

```

$!
$!
$!
$! set up prc file to precompile
$!
$APPEND 'SHORTAP'.PRC 'LUW'.DAT
$DELETE 'SHORTAP'.PRC;*
$GOTO PRECOMPILE
$!
$ GAPERR:
$WS " YOU HAVE ERRORS IN GAP"
$WS " PLEASE CHECK 'AP'.ERR FOR MESSAGES"
$GOTO EXIT
$ FDLERR:
$WS " "
$WS "THE WORD 'APPLICATION'/'REPORT' WAS NOT FOUND"
$WS "IN YOUR FDL FILE. ONE OF THESE MUST EXIST IN THE FILE."
$GOTO EXIT
$!
$! This section of the proc is performed if you want to precompile
$!
$ PRECOMPILE:
$WS " "
$WS "Beginning Precompile "
$!
$! set up .dat file to send to the precompiler
$!
$!
$!
$! 5/24/88: FDL stuff added in because GENRPD now requires fixed
$! length .DAT files since conversion to FIOPS.
$!
$ OPEN/WRITE FDLIN CDMDIR:[COM]FIX.FDL
$ WRITE FDLIN "IDENT ""23-FEB-1988 09:49:43 VAX-11 FDL
Editor""
$ WRITE FDLIN " "
$ WRITE FDLIN "SYSTEM"
$ WRITE FDLIN " SOURCE VAX/VMS"
$ WRITE FDLIN " "
$ WRITE FDLIN "FILE"
$ WRITE FDLIN " ALLOCATION 391"
$ WRITE FDLIN " BEST TRY CONTIGUOUS yes"
$ WRITE FDLIN " EXTENSION 39"
$ WRITE FDLIN " ORGANIZATION sequential"
$ WRITE FDLIN " "
$ WRITE FDLIN "RECORD"
$ WRITE FDLIN " BLOCK SPAN yes"
$ WRITE FDLIN " CARRIAGE_CONTROL carriage_return"
$ WRITE FDLIN " FORMAT fixed"
$ WRITE FDLIN " SIZE 80"
$ CLOSE FDLIN
$ CONVERT/PAD=%040/FDL=CDMDIR:[COM]FIX.FDL 'LUW'.DAT 'LUW'.DAT
$ OPEN/WRITE NDMLIN NDML.DAT
$ WRITE NDMLIN LUW, " VAX VAX COBOL NDML COBOL"," ",LUW,".DAT
",LUW,".ERR ","CDM/CDM"," FD=Y"
$ CLOSE NDMLIN
$ OPEN/WRITE EDIPRC CDMDIR:[COM]EDIT.PROC
$ WRITE EDIPRC "S/80/200/WH"
$ WRITE EDIPRC "EXIT"
$ CLOSE EDIPRC

```

```
$ EDIT/COMMAND=CDMDIR:[COM]EDIT.PROC CDMDIR:[COM]FIX.FDL
$ DEASSIGN SYS$OUTPUT
$ CONVERT/PAD=%040/FDL=CDMDIR:[COM]FIX.FDL NDML.DAT NDML.DAT
$ DELETE CDMDIR:[COM]FIX.FDL;* , CDMDIR:[COM]EDIT.PROC;*
$!
$!   INPUTS TO PRECOMPILER ARE NOW SET UP
$!   GO AHEAD AND RUN IT:
$!
$   ASSIGN/USER_MODE SYS$COMMAND SYS$INPUT
$ ASSIGN 'LUW'.OUT SYS$OUTPUT
$ RUN CMDIR:[RUNAREA]NDML
$!
$ DEASSIGN SYS$OUTPUT
$!
$!   check the .out file for errors in precompiling
$!
$ OPEN/READ EFLE 'LUW'.OUT
$ ZR="0"
$ NERRLOOP:
$   READ/END OF FILE=COMPERR EFLE EREC
$   LENG = 'F$LENGTH(EREC)'
$   UN = 'F$LOCATE("UNSUCC",EREC)'
$   IF 'UN' .EQS. 'LENG' THEN GOTO NERRLOOP
$   UN1 = 'UN' - 13
$   UN2 = 'F$EXTRACT(UN1,1,EREC)'
$   IF UN2 .EQS. ZR THEN GOTO NDMLGOOD
$WS "THE PRECOMPILE OF 'LUW' HAS 'UN2' UNSUCCESSFUL ROUTINES"
$WS "CHECK THE 'LUW'.ERR FILE FOR ERRORS"
$GOTO EXIT
$!
$!   the precompile was successful, compile the code
$!
$   NDMLGOOD:
$WS " "
$WS "NDML PRECOMPILE SUCCESSFULLY COMPLETED"
$   NDMLGDRD:
$ READ/END OF FILE = COMPERR EFLE EREC
$ LENG = 'F$LENGTH(EREC)'
$ UN = 'F$LOCATE("==> USE", EREC)'
$ IF 'UN' .EQS. 'LENG' THEN GOTO NDMLGDRD
$!
$ASSIGN 'LUW'.MSG SYS$OUTPUT
$UN1 = 'UN' + 8
$NNAM='F$EXTRACT(UN1,30,EREC)'
$CLOSE EFLE
$ @'NNAM'
$!
$DEASSIGN SYS$OUTPUT
$WS "RESULTS OF COMPILE CAN BE FOUND ON 'LUW'.MSG"
$ EXIT:
$DEFINE IISGLIB "CDMDIR:[TEST]GENOLB.OLB
```

```

$!
$!          LNKRCMP.COM
$!          THIS USES ORACLE VERSION 5.1
$!
$!
$!
$!
$!
$DEFINE/NOLOG TOOLOLB   CDMDIR:[TOOLS.COMP]COMPOLB
$DEFINE/NOLOG IISSGLIB "CDMDIR:[TOOLS.COMP]GENOLB.OLB"
$DEFINE CDMTEMPS "CDMDIR:[TOOLS.COMP.TEMPS]"
$if pl .eqs. "N" then goto LINKRCMP
$WS="WRITE SYS$OUTPUT"
$ LINK:
$DEASSIGN SYS$OUTPUT
$WS " "
$WS "Beginning Generation of Rp-Main"
$LUW="CDMREPR"
$CDM="CDM/CDM"
$!
$! generate the rp-main
$! NOTE:  this is done automatically if you link
$!
$!
$! set up .dat file to send to genrpd
$!
$OPEN/WRITE GENRPD.DAT GENRPD.DAT
$WRITE GENRPD.DAT LUW, " ", CDM, " VAX"
$CLOSE GENRPD.DAT
$ OPEN/WRITE FDLIN FIX.FDL
$ WRITE FDLIN "IDENT      ""23-FEB-1988 09:49:43      VAX-11 FDL
Editor""
$ WRITE FDLIN "      "
$ WRITE FDLIN "SYSTEM"
$ WRITE FDLIN "      SOURCE                      VAX/VMS"
$ WRITE FDLIN "      "
$ WRITE FDLIN "FILE"
$ WRITE FDLIN "      ALLOCATION                      3"
$ WRITE FDLIN "      BEST_TRY_CONTIGUOUS          yes"
$ WRITE FDLIN "      EXTENSION                      39"
$ WRITE FDLIN "      ORGANIZATION                  sequential"
$ WRITE FDLIN "      "
$ WRITE FDLIN "RECORD"
$ WRITE FDLIN "      BLOCK_SPAN                      yes"
$ WRITE FDLIN "      CARRIAGE_CONTROL                  carriage_return"
$ WRITE FDLIN "      FORMAT                      fixed"
$ WRITE FDLIN "      SIZE                      80"
$ CLOSE FDLIN
$ CONVERT/PAD=%040/FDL=FIX GENRPD.DAT GENRPD.DAT
$!
$! now run genrpd
$!
$ASSIGN/USER MODE SYS$COMMAND SYS$INPUT
$ASSIGN 'LUW'.RPD SYS$OUTPUT
$RUNGENRPD
$DEASSIGN SYS$OUTPUT
$DELETE GENRPD.DAT;*, FIX.FDL;*
$!
$! now get the needed information to compile the rp-main(s)
$!

```

```

$ASSIGN 'LUW'.RDCOMP SYS$OUTPUT
$GENRPDFLAG = 0
$OPEN/READ EFLE 'LUW'.RPD
$ RDLOOP:
$READ/END_OF FILE=ENDMAIN EFLE EREC
$ LENG = 'F$LENGTH(EREC)'
$ DBMS = 'F$LOCATE("FOR DBMS", EREC)'
$ UN = 'F$LOCATE("STORED ON", EREC)'
$ MN = 'F$LOCATE("MODULE", EREC)'
$ DB = 'F$LOCATE("DATA BASE", EREC)'
$ RM = 'F$LOCATE("REMOTE/", EREC)'
$ HST = 'F$LOCATE("RUN AT", EREC)'
$ IF 'MN' .NES. 'LENG' THEN GOTO SAVMODNM
$ IF 'DB' .NES. 'LENG' THEN GOTO SAVDBN
$ IF 'RM' .NES. 'LENG' THEN GOTO RMLC
$ IF 'HST' .NES. 'LENG' THEN GOTO SAVEHST
$ IF 'DBMS' .NES. 'LENG' THEN GOTO SAVEDBMS
$ IF 'UN' .EQS. 'LENG' THEN GOTO RDLOOP
$GENRPDFLAG = 1
$UN1 = 'UN' + 16
$UNEND = 'F$LOCATE(".", EREC) - UN1
$PL := 'F$EXTRACT(UN1, UNEND, EREC)'
$IF DBMSNM .EQS. "ORACLE" THEN GOTO MAINPCC
$WS "A NEW DBMS TYPE MUST BE ADDED TO THE MAIN COMPILE PART OF THIS
PROCEDURE"
$GOTO EXIT
$!
$! get the rp-main mod name
$!
$ SAVMODNM:
$MN1 = 'MN' + 7
$RPMN1 := 'F$EXTRACT(MN1, 10, EREC)'
$LENG = 'F$LENGTH(RPMN1)'
$MN2 = 'F$LOCATE("ZZZ", RPMN1)'
$IF 'MN2' .EQS. 'LENG' THEN MODLOC = 0
$IF 'MN2' .NES. 'LENG' THEN MODLOC = 2
$RPMN := 'F$EXTRACT(MODLOC, 5, RPMN1)'
$GOTO RDLOOP
$!
$! get the remote/local status
$!
$ RMLC:
$RM1 = 'RM' + 13
$RMSW := 'F$EXTRACT(RM1, 1, EREC)'
$GOTO RDLOOP
$!
$! get the database name
$!
$ SAVDBN:
$DB1 = 'DB' + 10
$DBN := 'F$EXTRACT(DB1, 30, EREC)'
$GOTO RDLOOP
$!
$! get the host name
$!
$ SAVEHST:
$HST1 = 'HST' + 7
$HSTNM := 'F$EXTRACT(HST1, 3, EREC)'
$GOTO RDLOOP

```

```

$!
$!  get the dbms name
$!
$  SAVEDBMS:
$DBMS1 = 'DBMS' + 9
$DBMSNM := 'F$EXTRACT(DBMS1, 30, EREC)
$GOTO RDLOOP
$!
$!
$!  oracle precompile the rp-main (if needed)
$!
$  MAINPCC:
$PCC INAME='PL'.TMP LNAME='PL'.ERR USERID='CDM' -
  ONAME='PL'.COB INCLUDE=SYS$ORACLE: HOST=COB74 MAXLITERAL=160
REBIND=YES
$ON ERROR THEN WS "ORACLE ERROR IN RP-MAIN ''PL'.TMP"
$ON ERROR THEN GOTO EXIT
$COBOL/ANSI_FORMAT/CHECK=ALL/COPY_LIST/CROSS_REFERENCE/OBJECT='PL'.
OBJ -
  /FIPS=74/NOLIST/CHECK=ALL/STANDARD=(SYNTAX)/DEBUG=ALL 'PL'.COB
$ON ERROR THEN WS "COBOL ERROR IN RP-MAIN ''PL'.TMP"
$ON ERROR THEN GOTO EXIT
$DELETE 'PL'.COB;*
$DELETE 'PL'.ERR;*
$!
$!  Compile the rpmain.c
$!
$vccl/debug/NOLIST/show=(include)/standard=portable -
  /noopt/OBJECT=CDMDIR:[TOOLS.COMP]RPMMAIN.OBJ/DEFINE=VAX
RPMMAIN.C
$delete rpmain.c;*
$GOTO RDLOOP
$!
$!  done generating rp-main
$!
$  ENDMAIN:
$CLOSE EFLE
$DEASSIGN SYS$OUTPUT
$IF GENRPDFLAG .EQ. 0 THEN GOTO MAINERR
$WS " "
$WS "GENERATION OF REQUEST PROCESSOR MAIN COMPLETE"
$GOTO STARTLINK
$!
$!  there was an error in generating the rp-main
$!
$  MAINERR:
$WS "THE GENRPD HAD ERRORS.  EXAMINE ''LUW'.RPD"
$GOTO EXIT
$STARTLINK:
$DELETE 'LUW'.RPD;*
$DELETF 'LUW'.RDCOMP;*
$WRITE SYS$OUTPUT " - LINKING CDMREPRT.EXE "
$ ASSIGN CDMREPRT.LINK SYS$OUTPUT
$!
$DELETE CMDIR:[RUNAREA]CDMREPRT.EXE;*
$!
$!  inquire p5 "ENTER TWO LETTER NTM DIRECTORY PREFIX "
$!  inquire p6 "ENTER NTM CLUSTER FOR THIS RP (T1V OR U1V) "
$!

```

```
$P5 = "GR"
$P6 = "UIV"
$P1 = "CDMREPR"
$!
$!   NTMTAB.COM
$!
$!
$!   UPDATE NTM TABLES APITBL.DAT, APTTBL.DAT
$!
$!
$ FLAG=0
$ P7=P5+P1
$ OPEN/READ APITBL.DAT CDMDIR:[RUNAREA]APITBL.DAT
$ RD1:
$ READ/END_OF_FILE=CHK APITBL.DAT ENTRY
$ RPND=F$EXTRACT(0,10,ENTRY)
$ IF RPND .NES. P7 THEN GOTO RD1
$ FLAG=1
$ CHK:
$ CLOSE APITBL.DAT
$ IF FLAG .NES. 0 THEN GOTO NOUPD
$ GOTO UPD
$ NOUPD:
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "RP MAIN ",P1," ALREADY IN NTM TABLES"
$ WRITE SYS$OUTPUT " "
$ GOTO LINKRCMP
$ UPD:
$ RPAPI=P7+P6+"1"
$ RPAPT=P1+"9999010120001130N0"
$ OPEN/APPEND APITBL.DAT CDMDIR:[RUNAREA]APITBL.DAT
$ OPEN/APPEND APTTBL.DAT CDMDIR:[RUNAREA]APTTBL.DAT
$ WRITE APITBL.DAT RPAPI
$ WRITE APTTBL.DAT RPAPT
$ CLOSE APITBL.DAT
$ CLOSE APTTBL.DAT
$ WRITE SYS$OUTPUT " "
$ WRITE SYS$OUTPUT "NTM TABLES UPDATED WITH RP ",P1
$ WRITE SYS$OUTPUT " "
$!
$!DEFINE CDMROLB "CDMDIR:[CDMR]CDMROLB"
$LINKRCMP:
$@SYS$ORACLE:LFOR CDMDIR:[RUNAREA]CDMREPR -
iissglib/lib/include=(cdmrrp),-
CDMDIR:[TOOLS.COMP]RPMAN.OBJ,-
'PL'.OBJ,-
CDMDIR:[COM]CDMI/OPTIONS,-
SYS$ORACLE:SQLLIB/LIB,-
CDMDIR:[COM]CDMUI.OPT/OPT,-
CDMDIR:[COM]CDMNTM.OPT/OPTIONS N
$DEASSIGN SYS$OUTPUT
$DELETE CDMREPR.LINK;*, RPMAN.OBJ;*, 'PL'.OBJ;*
$WRITE SYS$OUTPUT "LINKING COMPLETED"
$EXIT:
$DEASSIGN TOOLLB
$!
```