INTEGRATED INFORMATION SUPPORT SYSTEM (I ISS)
Volume V - Common Data Model Subsystem

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

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

DATE

FOR THE COMMANDER:

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

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.
This document is to be used by the Common Data Model Administrator (CDMA) to determine the impact a software change might have upon other software modules within the CDM subsystem. The CDM Impact Analysis Utility is used to identify and report which software modules and external schemas are affected by a change to the CDM.
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:

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

<table>
<thead>
<tr>
<th>SECTION</th>
<th>INTRODUCTION</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>INTRODUCTION</td>
<td>1-1</td>
</tr>
<tr>
<td>2.0</td>
<td>PREREQUISITES</td>
<td>2-1</td>
</tr>
<tr>
<td>3.0</td>
<td>DEFINE THE IMPACT PRECOMPILE GROUP</td>
<td>3-1</td>
</tr>
<tr>
<td>4.0</td>
<td>PROCEDURES TO BUILD THE CDM IMPACT EXECUTABLE</td>
<td>4-1</td>
</tr>
</tbody>
</table>
SECTION 1
INTRODUCTION

The construction of the CDM Impact Tool requires precompiling of 38 routines for the purpose of discovering the impacts, storing them in the CDM and reporting them on a screen or a printer device. These routines have been grouped into a single logical unit of work. These modules are listed in a test file. After precompilation, the following steps must be executed in order to construct the CDM Impact executable:

- Generate the CDM Impact Request Processor Main Program
- Compile and insert into the object library (GENOLB) the generated CDM Impact Request Processor Main Program
- Create the CDM Impact executable

Section 2 lists the prerequisites of the Impact Environment. Section 3 of this document lists the group to be precompiled. Section 4 contains the step by step instructions of building CDM Impact.

SECTION 2
PREREQUISITES

The Prerequisites to creating the IMPACT ANALYSIS environment are:


2. Existence of an object library GENOLB in the directory cdmdir:[tools.Impa] 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 IMPAOLB according to normal Integration and Testing procedures.

5. The NDDL and NDML executables must be available.
SECTION 3

DEFINE THE IMPACT PRECOMPILE GROUP

This section contains the list of routines to be precompiled as a single logical unit of work. They are contained in the files Impact.tst and Impact2.tst.

Application Name - Impact.tst/IMPACT2.tst

ALGX
ALLDTS
ALLTAGS
AUCX
CATMEM
CIINIT
DBAREA
DBMSDB
DOMAUC
DOMS
ECAUC
ECX
FNDHSDB
GETILUW
GETTAG
HOSTDB
ICHKUN
IGTPKC
IMPACTX
INSAPP
INSCOM
INSOBJ
ISELHP
IVERARA
IVERATT
IVERAUC
IVERDB
IVERDBM
IVERDFD
IVERDI
IVERDOM
IVEROT
IVERENT
IVERHST
IVERKC
IVERLUW
IVEROAC
IVERPSB
IVERRC
IVERRCC
IVERRST
IVERRT
IVERSMD
IVERVEW
KCX
LUWAPP
MODUX
PSBDB
RC
TRACEX
TRAUC
TRDB
TRDF
TRDI
TRRC
TRRT
TRSET
TRUV
SECTION 4

PROCEDURES TO BUILD THE CDM IMPACT EXECUTABLE

The following steps must be executed to construct the CDM Impact executable:

1. Create the Oracle Impact tables in the CDM. Proceed as follows:

   $ UFI CDM/CDM
   UFI>START ORAIMP.DAT
   UFI>EXIT

2. Using NDDL, run the Impact meta data into the CDM. Proceed as follows:

   $ NDDL NDDLIMP.DAT

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

3. Precompile and compile the NDML-embedded source code. Proceed as follows:

   $@BLDIMP
   PRECOMPILE AND COMPILATE A GROUP OF PRC'S
   -----------------------------
   NAME OF THE APPLICATION>:IMPACT
   File: ALGX
   File: AUCX
   ...
   ...
   ...
   ...
   ...
   ...
   File: TRUV

   NDML PRECOMPILE SUCCESSFULLY COMPLETED
   BEGIN COMPILATING GENERATED CODE
   RESULTS OF COMPILATE CAN BE FOUND ON IMPACT.MSG

4. Repeat step 3 with IMPACT2 as the name of the application.

5. Execute the procedure file LNKIMP.COM to generate the RP-MAIN, compile the RP-MAIN, and place in GENOLB, generate the opt files and link the application.

   $ @LNKIMP
5. Run the flan executable using the file CDMIMP.FDL. Enter

```bash
$FLAN CDMIMP.FDL
```

The form definition files will be:

- CDMIMP.FD
- IMPACT1.FD
- IMPACT2.FD
- TRACE1.FD
- TRACE2.FD

6. Define the CDM Impact Function using the user interface utility SYSGEN. The name of the function is CDMIMPACT. The application is GRCDMIMPZZ, and the username, password and role is CDM.

The following pages contain listings for:

- BLDIMP.COM
- LNKIMP.COM
BLDNDDL.COM

RECEIVE A TEST FILE OF PRC'S AND PRECOMPILE AND COMPILCE THEM

WS:= WRITE SYS$OUTPUT
DEFINE IISSGLIB "CDMDIR:[TOOLS.IMPACT]GENOLB.OLB"
WS "PRECOMPILE AND COMPILCE A GROUP OF PRC'S"
WS "-------------------"
read an input file containing names of modules to be precompiled

INQUIRE AP " NAME OF THE APPLICATION>
CREATE 'AP'.DAT
OPEN/WRITE NDMLIN 'AP'.IN
OPEN/READ NDDLIN CDMDIR:[TOOLS.IMPACT]'AP'.TST
NEXT:
READ/END OF FILE=ENDONE NDDLIN FILE
WS "FILE:'File'
APPEND 'FILE'.PRC 'AP'.DAT
GOTO NEXT
ENDONE:
WRITE NDMLIN "CDMIMP VAX"," ',AP',.DAT ','AP',.ERR ','CDM/CDM"," FD=N"
CLOSE NDMLIN
CLOSE NDDLIN

INPUTS TO PRECOMPLIER ARE NOW SET UP
GO AHEAD AND RUN IT:
ASSIGN/USER_MODE SYS$COMMAND SYS$INPUT
ASSIGN 'AP'.IN NDML
ASSIGN 'AP'.OUT SYS$OUTPUT
RUN CMDIR:[RUNAREA]NDML.EXE

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
LEN = '$LENGTH(ERECC)'
UN = '$LOCATE("UNSUC","ERECC)'
IF 'UN' .EQS. 'LEN' THEN GOTO NERRLOOP
UN1 = 'UN' - 13
UN2 = '$EXTRACT(UN1,1,ERECC)'
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:

4-3
$WS " "
$WS "NDML PRECOMPILE SUCCESSFULLY COMPLETED"
$WS "BEGIN COMPILING GENERATED CODE"
$NDMLGDRD:
$ READ/END OF FILE = COMPERR EFLE EREC
$ LENG = 'F$LENGTH(EREC)'
$ UN = 'F$LOCATE("COMPILE ALL CODE", EREC)'
$ IF 'UN' .EQS. 'LEN' THEN GOTO NDMLGDRD
$!
$ASSIGN 'AP'.MSG SYS$OUTPUT
$UN1 = 'UN' - 34
$NNAM:='F$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"
! LNKIMP.COM
! THIS USES ORACLE VERSION 5.1
! CLONED WITH CHANGES 1/20/88 - Richard Stewart
! ADD ADDING TO NTM TABLES...NEED TO REMOVE FOR STANDALONE VERSION
$DEFINE/NOLOG TOOLOLB CDMDIR:[TOOLS.IMPA]IMPAOLB
$DEFINE/NOLOG IISGLIB "CDMDIR:[TOOLS.IMPA]GENOLB.OLB"
$if pl .eqs. "N" then goto linkimp
$WS="WRITE SYS$OUTPUT"
$ LINK:
$DEASSIGN SYS$OUTPUT
$WS ""
$WS "Beginning Generation of Rp-Main"
$LUW="CDMIMP"
$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
$CLOSE GENRPD.DAT
$OPEN/WRITE FDLIN FIX.FDL
$ WRITE FDLIN "IDENT ""23-FEB-1988 09:49:43 VAX-11 FDL Editor"
$ WRITE FDLIN "SYSTEM"
$ WRITE FDLIN "SOURCE VAX/VMS"
$ WRITE FDLIN "FILE"
$ WRITE FDLIN "ALLOCATION 3"
$ WRITE FDLIN "BEST_TRY_CONTIGUOUS yes"
$ WRITE FDLIN "EXTENSION 39"
$ WRITE FDLIN "ORGANIZATION sequential"
$ 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

4-5
$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 DBMSNAM .EQS. "ORACLE" THEN GOTO MAINPCC
$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
$LIB/REPLACE IISSGLIB 'PL'.OBJ
$!DELETE 'PL'.OBJ;*
$DELETE 'PL'.COB;*
$DELETE 'PL'.ERR;*

Compile the rpmain.c

vcc/debug/NOLIST/show=(include)/standard=portable -
/noopt/OBJECT=CDMDIR:[TOOLS. IMPA]RPMAIN.OBJ/DEFINE=VAX RPMAIN.C
$DELETE RPMAIN.C;*
$GOTO RDLOOP

done generating rp-main

ENDMAIN:
$CLOSE EFILE
$DEASSIGN SYS$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:
$WRITE SYS$OUTPUT " - LINKING CDMIMP.EXE "
$ASSIGN CDMIMP.LINK SYS$OUTPUT

inquire p5 "ENTER TWO LETTER NTM DIRECTORY PREFIX "
inquire p6 "ENTER NTM CLUSTER FOR THIS RP (T1V OR U1V) "
p5 = "GR"
p6 = "U1V"
p1 = "CDMIMP"
NTMTAB.COM
UPDATE NTM TABLES APITBL.DAT, APTTBL.DAT, ACTTBL.DAT

$!
$!!  M. DENMAN
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!
$!