



USAISEC

*US Army Information Systems Engineering Command
Fort Huachuca, AZ 85613-5300*

AD-A268 361



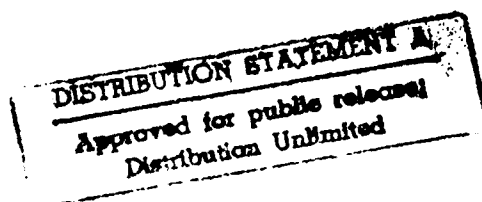
2

U.S. ARMY INSTITUTE FOR RESEARCH
IN MANAGEMENT INFORMATION,
COMMUNICATIONS, AND COMPUTER SCIENCES

SAMeDL: Technical Report Appendix E - Installation Guide

ASQB-GI-92-018

September 1992



42C
261



AIRMICS
115 O'Keefe Building
Georgia Institute of Technology
Atlanta, GA 30332-0800



REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188
Exp. Date: Jun 30, 1986

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS NONE		
2a. SECURITY CLASSIFICATION AUTHORITY N/A			3. DISTRIBUTION/AVAILABILITY OF REPORT N/A		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE N/A					
4. PERFORMING ORGANIZATION REPORT NUMBER(S)			5. MONITORING ORGANIZATION REPORT NUMBER(S) N/A		
6a. NAME OF PERFORMING ORGANIZATION	6b. OFFICE SYMBOL (If applicable)		7a. NAME OF MONITORING ORGANIZATION N/A		
6c. ADDRESS (City, State, and Zip Code)			7b. ADDRESS (City, State, and ZIP Code) N/A		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Software Technology Branch, ARL	8b. OFFICE SYMBOL (If applicable) AMSRL-CI-CD		9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c. ADDRESS (City, State, and ZIP Code) 115 O'Keefe Bldg. Georgia Institute of Technology Atlanta, GA 30332-0800			10. SOURCE OF FUNDING NUMBERS		
			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
			WORK UNIT ACCESSION NO.		
11. TITLE (Include Security Classification) SAMEDL: Technical Report Appendix E - Installation Guide					
12. PERSONAL AUTHOR(S) MS. Deb Waterman					
13a. TYPE OF REPORT Technical Paper	13b. TIME COVERED FROM Apr 91 TO Sept 92	14. DATE OF REPORT (Year, Month, Day) Sept 15, 1992		15. PAGE COUNT 35	
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUBGROUP	Ada Database Access, SAMeDL, Ada extension module, SQL		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)					
This report details the research efforts into the SQL Ada Module Database Description Language (SAMEDL). Four compilers are presented (Oracle, Informix, XDB, and Sybase) that allow Ada application programs to access database using a standard SQL query language. Copies of the compiler can be obtained from the DoD Ada Joint Program Office 703/614-0209.					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL LTC David S. Stevens			22b. TELEPHONE (Include Area Code) (404) 894-3110	22c. OFFICE SYMBOL AMSRL-CI-CD	

This research was performed by Statistica Inc., contract number DAKF11-91-C-0035, for the Army Institute for Research in Management Information, Communications, and Computer Sciences (AIRMICS), the RDTE organization of the U. S. Army Information Systems Engineering Command (USAISEC). This final report discusses a set of SAMeDL compilers and work environment that were developed during the contract. Request for copies of the compiler can be obtained from the DoD Ada Joint Program Office, 703/614/0209. This research report is not to construed as an official Army or DoD Position, unless so designated by other authorized documents. Material included herein is approved for public release, distribution unlimited. Not protected by copyright laws.

THIS REPORT HAS BEEN REVIEWED AND IS APPROVED

Glenn E. Racine

Glenn E. Racine, Chief
Computer and Information
Systems Division

James D. Gantt

James D. Gantt, Ph.D.
Director
AIRMICS

DTIC QUALITY INSPECTED 3

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

SAMeDL.TR.10.15 Sep 92

APPENDIX E
SAMeDL Installation Guides

SAMeDL Development Environment Installation Guide

Intermetrics, Inc.

Document	IR-VA-026-2
Date	01-September-1992

Published by
Intermetrics, Inc.
733 Concord Avenue, Cambridge, Massachusetts 02138

Copyright (c) 1992 by Intermetrics, Inc.

This material may be reproduced by or for the U.S. Government pursuant to the copyright license under the clause at DFARS 252.227-7013 (Oct. 1988).

Table of Contents

Installing the SAMeDL Development Environment (SDE)	5
Locating the SAMeDL Compiler and Support Packages	7
Additional Informix Setup Requirements.....	9
Additional Oracle Setup Requirements	9
Additional XDB Setup Requirements	10
Verifying the Success of the SAMeDL Installation	15
Building the SAMeDL Demo Program using the Informix RDBMS	15
Building the SAMeDL Demo Program using the Oracle RDBMS	15
Building the SAMeDL Demo Program using the Sybase RDBMS	16
Building the SAMeDL Demo Program using the XDB RDBMS	17
Running the SAMeDL Acceptance Tests	19
Running the SAMeDL Acceptance Tests using the Oracle RDBMS	22
Running the SAMeDL Acceptance Tests using the Sybase RDBMS	24
Running the SAMeDL Acceptance Tests using the XDB RDBMS	26
Release Notes for the SAMeDL Development Environments	29
Version 2.8 (Informix/PC-386/Interactive Unix/Alsys)	29
System Configuration	29
Implementation Dependent Features	29
Version 2.8 (Oracle/PC-386/Interactive Unix/Alsys)	31
System Configuration	31
Implementation Dependent Features	31
Version 2.8 (Sybase/Sun-Sparc/SunOS/Verdix)	33
System Configuration	33
Implementation Dependent Features	33
Version 2.8 (XDB/PC-386/Interactive Unix/Alsys).....	35
System Configuration	35
Implementation Dependent Features	35
Using NFS Mounted Disks for the SAMeDL Library	35

Installing the SAMeDL Development Environment (SDE)

The SDE installation procedure for Unix Environments depends on the distribution media you possess. Please follow the instructions listed for your media to ensure proper installation:

Installing the SDE from 3 and 1/2" Floppy Disks:

- (i) **su** *log on as a super-user*
- (ii) **mkdir /usr/lib/same** *create the SAMeDL installation directory*
- (iii) **cd /usr/lib/same** *cd to the new directory*
- (iv) Insert the diskette labeled **SAMeDL Compiler & Environment , Disk 1 of 3** into the the first floppy drive.
- (v) **tar xvf /dev/dsk/f0q18dt** *copy SAMeDL files into the target directory*
- (vi) Repeat steps (iv) and (v) with disks 2 and 3.
- (vii) **/install** *complete the installation*
- (viii) **chmod 555 /usr/lib/same** *make sure the SAMeDL home directory is readable*
- (ix) **exit** *exit from the super-user shell*

The example directory used in this installation guide is **/usr/lib/same**. You do not have to install SAMeDL into this directory. To install SAMeDL into another directory, simply execute the instructions above using your chosen directory where the guide refers to **/usr/lib/same**.

Installing the SDE from 1/4" Tape Cartridge:

- (i) **su** *log on as a super-user*
- (ii) **mkdir /usr/lib/same** *create the SAMeDL installation directory*
- (iii) **cd /usr/lib/same** *cd to the new directory*
- (iv) Insert the tape labeled **SAMeDL Compiler & Environment, Tape 1 of 1** into the tape drive.
- (v) **tar xvf /dev/rst0** *copy SAMeDL files into the target directory*
- (vi) **/install** *complete the installation*
- (viii) **chmod 555 /usr/lib/same** *make sure the SAMeDL home directory is readable*
- (ix) **exit** *exit from the super-user shell*

The example directory used in this installation guide is **/usr/lib/same**. You do not have to install SAMeDL into this directory. To install SAMeDL into another directory, simply execute the instructions above using your chosen directory where the guide refers to **/usr/lib/same**.

Locating the SAMeDL Compiler and Support Packages

After the installation procedure is complete, the **SDEPATH** environment variable should be set to point to the SAMeDL installation directory. Frequent users of SAMeDL may want to add it to their Unix search paths (consult your system administrator on how to do this). The SAMeDL installation directory should contain the following files:

DEMO	subdirectory containing the SAMeDL installation demo
STD_PKGS	subdirectory containing the SAMeDL standard packages, including samedl_std.sme and samedl_sys.sme which contain the SAMeDL packages SAMeDL_Standard and SAMeDL_System respectively.
TESTS	subdirectory containing the SAMeDL acceptance tests.
comp_std_pkgs	Unix C-shell script for compiling the standard packages
install	SAMeDL installation script
samedl	SAMeDL Compiler Executable
sde.cleanlib	SDE Module Manager Executable
sde.creatar	SDE Module Manager Executable (Informix, Oracle, and Sybase-targeted compilers only)
sde.creatlib	SDE Module Manager Executable
sde.ls	SDE Module Manager Executable
sde.mkscript	SDE Module Manager Executable
sde.purge	SDE Module Manager Executable
sde.rm	SDE Module Manager Executable
sde.rmlib	SDE Module Manager Executable

Performing Additional Setup Procedures

For ease of use in accessing the SAMeDL tools, you may want to add the path to the SAMeDL installation directory to your application path, or you may want to set the environment variable **SDEPATH** to point to the installation directory. Neither of these conditions is required for normal use of your SAMeDL toolset, but setting of the **SDEPATH** variable is required for building the SAMeDL demonstration program and running the SAMeDL acceptance tests.

Depending on which DBMS your SAMeDL compiler is compatible with, there may be additional procedures which must be performed in order to prepare the DBMS for SAMeDL compatibility. Please check below in case your version of SAMeDL requires additional installation-time steps.

Additional Informix Setup Requirements

As part of the Informix installation procedure, certain environment variables will need to be set. The use of SAMeDL **requires** that the **INFORMIXDIR** environment variable be set to point to the Informix installation directory (e.g. /usr/Informix). However, setting this one environment variable may not be enough to ensure the proper use of the DBMS. It is important that all necessary environment variables for Informix be set according to the Informix documentation recommendations prior to using your SAMeDL tools.

Additional Oracle Setup Requirements

The Oracle database should be installed according to the manual. In addition, in preparation for running the SAMeDL acceptance tests, an Oracle user **test** with password **test** should be created by the Oracle database administrator. Access to the **test** account should be enabled for the account from which the acceptance tests will be run and the necessary permissions should be granted to the **test** user to allow table creation. As a simple exercise, invoke the Oracle **sqlplus** command and enter user name **test** and password **test** when prompted. If Oracle rejects your login, then the installation demonstration and SAMeDL acceptance tests will not run properly from your account.

As part of the Oracle installation procedure, certain environment variables will need to be set. The use of SAMeDL **requires** that the **ORACLE_HOME** environment variable be set to point to the Oracle installation directory (e.g. /usr/Oracle). However, setting this one environment variable may not be enough to ensure the proper use of the DBMS. It is important that all necessary environment variables for Oracle be set according to the Oracle documentation recommendations prior to using your SAMeDL tools.

Additional Sybase Setup Requirements

Sybase should be installed according to the manual except that during the SQL Server configuration, case sensitivity must be turned-off. Select "**Dictionary order, case insensitive**" as the collating order. SAMeDL will not work correctly if case sensitivity is left on.

In addition, in preparation for running the SAMeDL acceptance tests, a Sybase user **test** with password **test** should be created by the Sybase database administrator. Access to the **test** account should be enabled for the account from which the acceptance tests will be run and the necessary permissions should be granted to the **test** user to allow table creation. As a simple exercise, invoke the Sybase **isql** command and enter user name **test** and password **test** when prompted. If

Sybase rejects your login, then the installation demonstration and SAMeDL acceptance tests will not run properly from your account.

As part of the Sybase installation procedure, certain environment variables will need to be set. The use of SAMeDL **requires** that the SYBASE environment variable be set to point to the Sybase installation directory (e.g. /usr/Sybase). However, setting this one environment variable may not be enough to ensure the proper use of the DBMS. It is important that all necessary environment variables for Sybase be set according to the Sybase documentation recommendations prior to using your SAMeDL tools.

Additional XDB Setup Requirements

A knowledgeable XDB user can configure the database in minutes. All that is required is that:

- Recovery is turned on.
- Backward logging is turned on.
- Autocommit is turned off.

Even if you are unfamiliar with XDB, the following steps will configure a database that can be used to run the acceptance tests.

- | | | |
|-------|----------------------------------|--|
| (i) | cd /usr/lib/same | <i>Any directory will suffice.. You must have write permission for this directory.</i> |
| (ii) | setenv XDBCFCG `pwd` | <i>Set the configuration environment variable.</i> |
| (iii) | Make sure XDBPATH is set. | <i>(usually setenv XDBPATH /usr/xdm is correct).</i> |
| (iv) | xdmconfig | <i>running this command will create the XDB configuration file</i> |

At this point you may be prompted for the XDB TERMINAL type.

- (v) Select the first terminal choice (1).

*Now you will be asked where to place the database and the configuration file.
Since you are in the correct directory*

- (vi) Press y and Enter *And now wait.*

Finally the XDB CONFIGURATION screen appears.

- (vii) Press function key F3 *Turn Recovery on.*

- (viii) Press function key F6

XDB USER PROFILE screen appears.

- (ix) Press function key F8 *Turn on Backward log.*

- (x) Press function key F9

XDB MULTIUSER OPTIONS screen appears.

- (xi) Press function key F2 *Turn off Autocommit.*

You've just configured the XDB to the proper settings. To exit **xdbconfig** press the **Esc** key 6 times (a single escape is not enough, two are required before the escape is recognized). Pressing the escape key a couple of extra times won't cause any problems.

Verifying the Success of the SAMeDL Installation

Included with the SAMeDL Development Environment is a demonstration program which is easy to build and can provide an example of SAMeDL as well as being a verification of correct system installation. The demonstration files are located in the **DEMO** subdirectory of the SAMeDL installation directory. Directions on how to build this demo are given below. Since these instructions are database-specific, please be sure to follow the directions pertaining to the appropriate SAMeDL configuration.

The scripts used to build the demo program rely on access to the DBMS and Ada compiler you are using. You should have installed the Ada compilation system and added the path to the Ada tools to your unix path prior to attempting to build the SAMeDL demonstration program, since the scripts will not work if they cannot find the ada compiler and linker. Also, you should have installed the DBMS and set any necessary environment variables, since the SAMeDL demonstration cannot be built without access to the DBMS server and tools. In particular, please make sure that your DBMS server is up and running prior to building the demonstration program.

The scripts used to build the demonstration programs require that the **SDEPATH** environment variable be set to point to the SAMeDL installation directory. Use the Unix **setenv** command to set this variable prior to running the scripts.

Building the SAMeDL Demo Program using the Informix RDBMS

- (1) Make sure the Informix **INFORMIXDIR** environment variable and the SAMeDL **SDEPATH** environment variable have been set to point to the Informix and SAMeDL installation directories (respectively).
- (2) Copy the contents of the **DEMO** subdirectory of the SAMeDL installation directory to your working directory.
- (3) From inside your working directory, invoke the Unix C-shell command file **build_recdb**. This file contains commands that will create a new Informix database called **recdb** and one table called **members** using the Informix **isql** utility. If you do not have access to this utility or else do not have permission to create databases and tables in Informix, then this script will not work and you should consult your database administrator to be granted the proper Informix permissions.
- (4) From inside your working directory, invoke the Unix C-shell command file **build_demo**. This file contains commands that will compile, link and invoke the Ada-Informix application **main**, which inserts and retrieves records from the Informix database **recdb**, created by **build_recdb**.

Building the SAMeDL Demo Program using the Oracle RDBMS

- (1) Make sure the Oracle **ORACLE_HOME** environment variable and the SAMeDL **SDEPATH** environment variable have been set to point to the Oracle and SAMeDL installation directories (respectively).
- (2) Copy the contents of the **DEMO** subdirectory of the SAMeDL installation directory to your working directory.

- (3) From inside your working directory, invoke the Unix C-shell command file **build_recdb**. This file contains commands that will create the Oracle table called **members** in account **test** using the Oracle **sqlplus** utility. If you do not have access to this utility or else do not have permission to create tables in Oracle, then this script will not work and you should consult your database administrator to be granted the proper Oracle permissions. (see the section on Additional Oracle Setup Requirements)
- (4) From inside your working directory, invoke the Unix C-shell command file **build_demo**. This file contains commands that will compile, link and invoke the Ada-Oracle application **main**, which inserts and retrieves records from the Oracle database created by **build_demo**.

Building the SAMeDL Demo Program using the Sybase RDBMS

To build the SAMeDL demo, follow these steps:

- (1) Make sure the Sybase **SYBASE** environment variable and the SAMeDL **SDEPATH** environment variable have been set to point to the Sybase and SAMeDL installation directories (respectively).
- (2) Copy the contents of the **DEMO** subdirectory of the SAMeDL installation directory to your working directory.
- (3) Create an Ada library using **a.mklib -i**.
- (4) Provide **WITH** directives to your Ada library to enable the proper binding of your executable. To do this, use the **a.info -i** command provided with your VADS environment. You must provide a **WITH** directive for each of the following items, in the given order:
 - (a) The object file generated by the call **samedl** in the **build_demo** script (for this example, **samedl.lib/E_1.o**)
 - (b) The Sybase library **\$_SYBASE/lib/libsybesql.a**
 - (c) The Sybase library **\$_SYBASE/lib/libsybdb.a**
 - (d) The Unix C math library **/usr/lib/libm.a** (full path name may be required).

Note: for (b) and (c), VADS may require that the full path name be given instead of the reference to the environment variable **SYBASE**.
- (5) From inside your working directory, invoke the Unix C-shell command file **build_recdb**. This file contains commands that will create the Sybase table called **members** in account **test** using the Sybase **isql** utility. If you do not have access to this utility or else user **test** does not have permission to create tables in Sybase, then this script will not work and you should consult your database administrator to be granted the proper Sybase permissions. (see the section on Additional Sybase Setup Requirements)
- (6) From inside your working directory, invoke the Unix C-shell command file **build_demo**. This file contains commands that will compile and link the Ada-Sybase

application **main**, which inserts and retrieves records from the Sybase database **recdb**.

- (7) Execute **main** to test the demo program.

Building the SAMeDL Demo Program using the XDB RDBMS

- (1) Make sure the XDB **XDBPATH** environment variable and the SAMeDL **SDEPATH** environment variable have been set to point to the XDB and SAMeDL installation directories (respectively).
- (2) Copy the contents of the **DEMO** subdirectory of the SAMeDL installation directory to your working directory.
- (3) From inside your working directory, invoke the Unix C-shell command file **build_recdb**. This file contains commands that will create the XDB table called **members** in the database pointed to by the **XDBCFCG** environment variable using the **xdbsql** utility. If you do not have access to this utility or else do not have permission to create tables in XDB, then this script will not work and you should consult your database administrator to be granted the proper XDB permissions. (see the section on Additional XDB Setup Requirements)
- (4) From inside your working directory, invoke the Unix C-shell command file **build_demo**. This file contains commands that will compile, link and invoke the Ada-XDB application **main**, which inserts and retrieves records from the XDB database created by **build_demo**.

Note: XDB requires a configuration file in order to operate. (See the section on Additional XDB Setup Requirements)

Running the SAMeDL Acceptance Tests

If you followed the steps for installing SAMeDL for Interactive Unix you will have also installed the SAMeDL acceptance tests. The tests reside in the subdirectory **TESTS**.

The scripts used to build the acceptance tests rely on access to the DBMS and Ada compiler you are using. You should have installed the Ada compilation system and added the path to the Ada tools to your unix path prior to attempting to run the SAMeDL acceptance tests, since the scripts will not work if they cannot find the ada compiler and linker. Also, you should have installed the DBMS and set any necessary environment variables, since the SAMeDL demonstration cannot be built without access to the DBMS server and tools. In particular, please make sure that your DBMS server is up and running prior to building the demonstration program.

The scripts used to run the acceptance tests require that the **SDEPATH** environment variable be set to point to the SAMeDL installation directory. Use the Unix **setenv** command to set this variable prior to running the scripts.

In the **\$SDEPATH/TESTS** subdirectory, 4 subdirectories will be found. The **T1**, **T2**, and **T3** directories contain acceptance test code. The **SRC** directory contains the source code for the SDE Module Manager. For more information on this code, consult the file **readme.txt** in the **SRC** directory.

The procedures for running the SAMeDL Acceptance Tests are database-specific. To run the tests, please consult the appropriate section on the following pages.

Running the SAMeDL Acceptance Tests using the Informix RDBMS

You should not try to run the SAMeDL acceptance tests until both the Informix RDBMS and the Alsys Ada compilation system have been properly installed, the Informix **INFORMIXDIR** environment variable has been properly set according to the instructions in the Informix User's Manuals, and the environment variable **SDEPATH** has been set to point to the SAMeDL installation directory.

To run the acceptance tests, which are found in the **TESTS/T1**, **TESTS/T2**, and **TESTS/T3** subdirectories of the SAMeDL installation directory, follow these simple steps:

Running the T1 tests:

The T1 test procedure consists of running 25 successful tests and 12 error tests through the SAMeDL compiler and then compiling the result interfaces into an Ada library. The T1 acceptance test procedure creates a large ada library. Before attempting to run the T1 tests, you should make sure that the Interactive Unix **ULIMIT** value is set to a value high enough to accommodate this library (a **ulimit** value of 18000 should be sufficient).

- (1) Copy the contents of the **TESTS/T1** subdirectory of the SAMeDL installation directory to your working directory.
- (2) From inside your working directory, invoke the Unix C-shell command file **build_t1**. This file contains commands that will run 37 tests through the SAMeDL compiler and then compile the resulting interface files into the ada library **testlib**, created by the **build_t1** command file.
- (3) To verify the success of the T1 test, examine the files in your working directory with the extensions **.lis** and **.err**.

Running the T2 tests:

The T2 test procedure consists of compiling, linking, and running the Ada-Informix application **test_drive**, which populates the Informix database **bank** and then performs and verifies a series of SQL queries on the database.

- (1) Copy the contents of the **TESTS/T2** subdirectory of the SAMeDL installation directory to your working directory.
- (2) From inside your working directory, invoke the Unix C-shell command file **build_bank**. This file contains commands that will create a new Informix database called **bank** and five tables using the Informix **isql** utility. If you do not have access to this utility or else do not have permission to create databases and tables in Informix, then this script will not work and you should consult your database administrator to be granted the proper Informix permissions.
- (3) From inside your working directory, invoke the Unix C-shell command file **build_t2**. This file contains commands that will compile, link, and run the **test_drive** application.

Note: After running the **test_drive** application once, you will not be able to run it again until the Informix **bank** database has been reset to its initial configuration (all tables empty).

Running the T3 tests:

The T3 test procedure consists of running 16 successful tests and 17 error tests through the SAMeDL compiler and then compiling the result interfaces into an Ada library. The T3 acceptance test procedure creates a large ada library. Before attempting to run the T3 tests, you should make sure that the Interactive Unix `ULIMIT` value is set to a value high enough to accommodate this library (a `ulimit` value of 18000 should be sufficient).

- (1) Copy the contents of the **TESTS/T3** subdirectory of the SAMeDL installation directory to your working directory.
- (2) From inside your working directory, invoke the Unix C-shell command file **build_t3**. This file contains commands that will run 33 tests through the SAMeDL compiler and then compile the resulting interface files into the ada library **testlib**, created by the **build_t3** command file.
- (3) To verify the success of the T3 test, examine the files in your working directory with the extensions **.lis** and **.err**.

Running the SAMeDL Acceptance Tests using the Oracle RDBMS

You should not try to run the SAMeDL acceptance tests until both the Oracle RDBMS and the Alsys Ada compilation system have been properly installed, the **ORACLE_HOME** (usually /usr/oracle) environment variable has been properly set according to the instructions in the Oracle User's Guide and the environment variable **SDEPATH** has been set to point to the SAMeDL installation directory.

To run the acceptance tests, which are found in the **TESTS/T1**, **TESTS/T2**, and **TESTS/T3** subdirectories of the SAMeDL installation directory, follow these simple steps:

Running the T1 tests:

The T1 test procedure consists of running 25 successful tests and 12 error tests through the SAMeDL compiler and then compiling the result interfaces into an Ada library -- no executable is produced. The T1 acceptance test procedure creates a large ada library. Before attempting to run the T1 tests, you should make sure that the Interactive Unix **ULIMIT** value is set to a value high enough to accommodate this library (a ulimit value of 18000 should be sufficient).

- (1) Copy the contents of the **TESTS/T1** subdirectory of the SAMeDL installation directory to your working directory.
- (2) From inside your working directory, invoke the C-shell command file **build_t1**. This file contains commands that will run 37 tests through the SAMeDL compiler and then compile the resulting interface files into the Ada library.
- (4) To verify the success of the T1 test, examine the files in your working directory with the extensions **.lis** and **.err**.

Running the T2 tests:

The T2 test procedure consists of compiling, linking, and running the Ada-Oracle application **test_drive**, which populates the Oracle database **bank** and then performs and verifies a series of **SQL** queries on the database.

- (1) Copy the contents of the **TESTS/T2** subdirectory of the SAMeDL installation directory to your working directory.
- (2) From inside your working directory, invoke the Unix C-shell command file **build_bank**. This file contains commands that will create the Oracle tables **cust**, **savings**, **branch**, **loan**, and **cheque** in account **test** using the Oracle **sqlplus** utility. If you do not have access to this utility or else do not have permission to create tables in Oracle, then this script will not work and you should consult your database administrator to be granted the proper Oracle permissions. (see the section on Additional Oracle Setup Requirements)
- (3) From inside your working directory, invoke the Unix C-shell command file **build_t2**. This file contains commands that will compile, link and invoke the Ada-Oracle application **test_drive**, which inserts and retrieves records from the Oracle database created by **build_bank**.

Note: After running the `test_drive` application once, you will not be able to run it again until the Oracle **bank** database has been reset to its initial configuration (all tables empty).

Running the T3 tests:

The T3 test procedure consists of running 16 successful tests and 17 error tests through the SAMeDL compiler and then compiling the result interfaces into an Ada library. The T3 acceptance test procedure creates a large ada library. Before attempting to run the T3 tests, you should make sure that the Interactive Unix `ULIMIT` value is set to a value high enough to accommodate this library (a `ulimit` value of 18000 should be sufficient).

- (1) Copy the contents of the **TESTS/T3** subdirectory of the SAMeDL installation directory to your working directory.
- (2) From inside your working directory, invoke the C-shell command file **build_t3**. This file contains commands that will run 33 tests through the SAMeDL compiler and then compile the resulting interface files into the Ada library.
- (3) To verify the success of the T3 test, examine the files in your working directory with the extensions `.lis` and `.err`.

Running the SAMeDL Acceptance Tests using the Sybase RDBMS

You should not try to run the SAMeDL acceptance tests until both the Sybase RDBMS and the Verdex Ada compilation system have been properly installed, the **SYBASE** (usually `/usr/sybase`) environment variable has been properly set according to the instructions in the Sybase User's Guide and the environment variable **SDEPATH** has been set to point to the SAMeDL installation directory.

To run the acceptance tests, which are found in the **TESTS/T1**, **TESTS/T2**, and **TESTS/T3** subdirectories of the SAMeDL installation directory, follow these simple steps:

Running the T1 tests:

The T1 test procedure consists of running 25 successful tests and 12 error tests through the SAMeDL compiler and then compiling the result interfaces into an Ada library -- no executable is produced.

- (1) Copy the contents of the **TESTS/T1** subdirectory of the SAMeDL installation directory to your working directory.
- (2) Create an Ada library using **a.mklib -i**.
- (3) From inside your working directory, invoke the C-shell command file **build_t1**. This file contains commands that will run 37 tests through the SAMeDL compiler and then compile the resulting interface files into the Ada library.
- (4) To verify the success of the T1 test, examine the files in your working directory with the extensions **.lis** and **.err**.

Running the T2 tests:

The T2 test procedure consists of compiling, linking, and running the Ada-Sybase application **test_drive**, which populates the Sybase database **bank** and then performs and verifies a series of SQL queries on the database.

- (1) Copy the contents of the **TESTS/T2** subdirectory of the SAMeDL installation directory to your working directory.
- (2) Create an Ada library using **a.mklib -i**.
- (3) Provide **WITH** directives to your Ada library to enable the proper binding of your executable. To do this, use the **a.info -i** command provided with your VADS environment. You must provide a **WITH** directive for each of the following items, in the given order:
 - (a) The object file generated by the call **samedl** in the **build_t2** script (for this example, **samedl.lib/E_1.o**)
 - (b) The Sybase library **\$SYBASE/lib/libsybcsql.a**
 - (c) The Sybase library **\$SYBASE/lib/libsybdb.a**

- (d) The Unix C math library `/usr/lib/libm.a` (full path name may be required).

Note: for (b) and (c), VADS may require that the full path name be given instead of the reference to the environment variable `SYBASE`.

- (4) From inside your working directory, invoke the Unix C-shell command file **build_bank**. This file contains commands that will create the Sybase database called `bank` in account `test` using the Sybase `isql` utility. If you do not have access to this utility or else user `test` does not have permission to create tables in Sybase, then this script will not work and you should consult your database administrator to be granted the proper Sybase permissions. (see the section on Additional Sybase Setup Requirements)
- (5) From inside your working directory, invoke the Unix C-shell command file **build_t2**. This file contains commands that will compile, link the Ada-Sybase application `test_drive`, which inserts and retrieves records from the Sybase database 'recdb'.

Note: After running the `test_drive` application once, you will not be able to run it again until the Sybase `bank` database has been reset to its initial configuration (all tables empty).

Running the T3 tests:

The T3 test procedure consists of running 16 successful tests and 17 error tests through the SAMeDL compiler and then compiling the result interfaces into an Ada library.

- (1) Copy the contents of the **TESTS/T3** subdirectory of the SAMeDL installation directory to your working directory.
- (2) Create an Ada library using **a.mklib -i**.
- (3) From inside your working directory, invoke the SunOS 4.1.1 C-shell command file **build_t3**. This file contains commands that will run 33 tests through the SAMeDL compiler and then compile the resulting interface files into the Ada library.
- (4) To verify the success of the T3 test, examine the files in your working directory with the extensions `.lis` and `.err`.

Running the SAMeDL Acceptance Tests using the XDB RDBMS

You should not try to run the SAMeDL acceptance tests until both the XDB RDBMS and the Alsys Ada compilation system have been properly installed, the XDB `XDBPATH` (usually `/usr/xdm`) environment variable has been properly set according to the instructions in the **XDB User Manuals**, and the environment variable `SDEPATH` has been set to point to the SAMeDL installation directory.

To run the acceptance tests, which are found in the **TESTS/T1**, **TESTS/T2**, and **TESTS/T3** subdirectories of the SAMeDL installation directory, follow these simple steps:

Running the T1 tests:

The T1 test procedure consists of running 25 successful tests and 12 error tests through the SAMeDL compiler and then compiling the result interfaces into an Ada library. The T1 acceptance test procedure creates a large ada library. Before attempting to run the T1 tests, you should make sure that the Interactive Unix `ULIMIT` value is set to a value high enough to accommodate this library (a `ulimit` value of 18000 should be sufficient).

- (1) Copy the contents of the **TESTS/T1** subdirectory of the SAMeDL installation directory to your working directory.
- (2) From inside your working directory, invoke the Unix C-shell command file **build_t1**. This file contains commands that will run 37 tests through the SAMeDL compiler and then compile the resulting interface files into the Ada library **testlib**, created by the **build_t1** command file.
- (3) To verify the success of the T1 test, examine the files in your working directory with the extensions **.lis** and **.err**.

Running the T2 tests:

The T2 test procedure consists of compiling, linking, and running the Ada-XDB application **test_drive**, which populates the XDB database and then performs and verifies a series of SQL queries on the database.

- (1) Copy the contents of the **TESTS/T2** subdirectory of the SAMeDL installation directory to your working directory.
- (2) From inside your working directory, invoke the Unix C-shell command file **build_bank**. This file contains commands that will create XDB tables called **cust**, **cheque**, **branch**, **loan**, and **savings** in the database pointed to by the `XDBCFCFG` environment variable using the `xdbsql` utility. If you do not have access to this utility or else do not have permission to create tables in XDB, then this script will not work and you should consult your database administrator to be granted the proper XDB permissions. (see the section on Additional XDB Setup Requirements)
- (2) From inside your working directory, invoke the Unix C-shell command file **build_t2**. This file contains commands that will compile, link, and run the **test_drive** application.

Note: After running the `test_drive` application once, you will not be able to run it again until the XDB database has been reset to its initial configuration (all tables empty).

Running the T3 tests:

The T3 test procedure consists of running 16 successful tests and 17 error tests through the SAMeDL compiler and then compiling the result interfaces into an Ada library. The T3 acceptance test procedure creates a large ada library. Before attempting to run the T3 tests, you should make sure that the Interactive Unix `ULIMIT` value is set to a value high enough to accommodate this library (a `ulimit` value of 18000 should be sufficient).

- (1) Copy the contents of the **TESTS/T3** subdirectory of the SAMeDL installation directory to your working directory.
- (2) From inside your working directory, invoke the Unix C-shell command file **build_t3**. This file contains commands that will run 33 tests through the SAMeDL compiler and then compile the resulting interface files into the ada library **testlib**, created by the **build_t3** command file.
- (3) To verify the success of the T3 test, examine the files in your working directory with the extensions **.lis** and **.err**.

Release Notes for the SAMeDL Development Environments

The release notes for the SAMeDL Development Environments, version 2.8, are included on the following pages. The release notes contain information specific to the SDE Release regarding the appropriate system configuration for the SDE and any implementation-dependent features included in the release.

Version 2.8 (Informix/PC-386/Interactive Unix/Alsys)

System Configuration

The Informix-targeted version 2.8 of the SAMeDL Development Environment is designed to run in conjunction with the following hardware and software configuration:

Machine:	PC 386
Operating System:	Interactive™ Unix, Version 3.2.2
Ada Compiler:	Aisys Ada, Version 5.1.1
Database Management:	Informix-OnLine Database Engine, Version 4.10
	Informix ESQL/™ Host Language Interface, Version 4.10
	Informix-SQL, Version 4.10

In addition, Intermetrics recommends the following Interactive Unix kernel parameter settings:

	<u>DEFAULT</u>	<u>MIN</u>	<u>MAX</u>
MAXUMEM	8192	2560	8192
ULIMIT	18000	2048	4194303

Implementation Dependent Features

The Informix-targeted version 2.8 of the SAMeDL Development Environment currently supports the following implementation dependent features of the SAMeDL language suggested in the noted sections of the SDE LRM:

<u>LRM Section</u>	<u>Language Feature</u>
[2.6]	reserved word ::= connect
[4.1.1.3]	word_list ::= context clause null value null_bearing assign not_null_bearing assign

The semantics of the implementation defined base domain options provided via the word_list grammar are explained in the SDE LRM, section 4.1.1.3.

[5.3] extended_statement ::=
 connect database_name |

 database_name ::=
 character_literal |
 constant_reference |
 input_param_ref

The semantics of the extended statements are described in section 6.2 of the SDE User's Guide.

Version 2.8 (Oracle/PC-386/Interactive Unix/Alsys)

System Configuration

The Oracle-targeted version 2.8 of the SAMeDL Development Environment is designed to run in conjunction with the following hardware and software configuration:

Machine:	PC-386
Operating System:	Interactive Unix, Version 2.1
Ada Compiler:	Alsys Ada, Version 5.1.1
Database Management:	Oracle Database Engine, Version 6.0
	Oracle Pro*C, Version 60
	Oracle SQLPlus (Interactive SQL interface), Version 6.0

In addition, Intermetrics recommends the following Interactive Unix kernel parameter settings:

	<u>DEFAULT</u>	<u>MIN</u>	<u>MAX</u>
MAXUMEM	8192	2560	8192
ULIMIT	18000	2048	4194303

Implementation Dependent Features

The Oracle-targeted version 2.8 of the SAMeDL Development Environment currently supports the following implementation dependent features of the SAMeDL language suggested in the noted sections of the SDE LRM:

<u>LRM Section</u>	<u>Language Feature</u>
[2.6]	reserved word ::= connect
[4.1.1.3]	word_list ::= context clause null value null_bearing assign not_null_bearing assign

The semantics of the implementation defined base domain options provided via the word_list grammar are explained in the SDE LRM, section 4.1.1.3.

[5.3]	extended statement ::= connect server user_id password [using database_name] database_name ::= limited_value_spec
-------	--

```
user_id ::=  
    limited_value_spec
```

```
password ::=  
    limited_value_spec
```

```
limited_value_spec ::=  
    character_literal      |  
    constant_reference     |  
    input_param_ref
```

The semantics of the extended statements are described in section 6.2 of the SDE User's Guide.

Version 2.8 (Sybase/Sun-Sparc/SunOS/Verdix)

System Configuration

The Sybase-targeted version 2.8 of the SAMeDL Development Environment is designed to run in conjunction with the following hardware and software configuration:

Machine:	Sun4
Operating System:	SunOS, Version 4.1.1
Ada Compiler:	Verdix Ada, Version 6.0.3c
Database Management:	Sybase Database Engine, Version 4.8
	Sybase Embedded SQL/C Host Language Interface, Version 4.8
	Sybase SQL (Interactive SQL interface), Version 4.8

Implementation Dependent Features

The Sybase-targeted version 2.8 of the SAMeDL Development Environment currently supports the following implementation dependent features of the SAMeDL language suggested in the noted sections of the SDE LRM:

<u>LRM Section</u>	<u>Language Feature</u>
[2.6]	reserved word ::= connect
[4.1.1.3]	word_list ::= context clause null value null_bearing assign not_null_bearing assign

The semantics of the implementation defined base domain options provided via the word_list grammar are explained in the SDE LRM, section 4.1.1.3.

[5.3]	extended statement ::= connect database_name connect server user_id password release all
	database_name ::= character_literal constant_reference
	user_id ::= limited_value_spec
	password ::=

limited_value_spec

limited_value_spec ::=
character_literal |
constant_reference |
input_param_ref

The semantics of the extended statements are described in section 6.2 of the SDE User's Guide.

Version 2.8 (XDB/PC-386/Interactive Unix/Alsys)

System Configuration

The XDB-targeted version 2.8 of the SAMeDL Development Environment is designed to run in conjunction with the following hardware and software configuration:

Machine:	PC 386
Operating System:	Interactive™ Unix, Version 3.2.2
Ada Compiler:	Alsys Ada, Version 5.1.1
Database Management:	XDB 2.3+ (This is not a commercial version. It has been upgraded by CA)

In addition, Intermetrics recommends the following Interactive Unix kernel parameter settings:

	<u>DEFAULT</u>	<u>MIN</u>	<u>MAX</u>
MAXUMEM	8192	2560	8192
ULIMIT	18000	2048	4194303

Implementation Dependent Features

The XDB-targeted version 2.8 of the SAMeDL Development Environment currently supports the following implementation dependent features of the SAMeDL language suggested in the noted sections of the SDE LRM:

<u>LRM Section</u>	<u>Language Feature</u>
[2.6]	reserved word ::= connect
[4.1.1.3]	word_list ::= context clause null value null_bearing assign not_null_bearing assign

The semantics of the implementation defined base domain options provided via the word_list grammar are explained in the SDE LRM, section 4.1.1.3.

[5.3]	extended statement ::= connect
-------	--------------------------------

The semantics of the extended statements are described in section 6.2 of the SDE User's Guide.

Using NFS Mounted Disks for the SAMeDL Library

The SAMeDL Development Environment works well with NFS mounted disks with a singular exception: deletion of a library via sde.rmlib. The contents of the library are removed, however,

the directory remains. The directory is not removed due to temporary files created by NFS. The solution is to use `rmdir` at command prompt after issuing the `sde.rmlib` command.