Special Report
CMU/SEI-89-SR-5
May 1989

SAME Standard Package Installation Guide

Marc H. Graham

May 1989

Special Report

CMU/SEI-89-SR-5

May 1989

SAME Standard Package Installation Guide

Marc H. Graham

Binding of Ada and SQL Project

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

This report was prepared for the SEI Joint Program Office HQ ESC/AXS

5 Eglin Street

Hanscom AFB, MA 01731-2116

The ideas and findings in this report should not be construed as an official DoD position. It is published in the interest of scientific and technical information exchange.

FOR THE COMMANDER

(signature on file)

Thomas R. Miller, Lt Col, USAF, SEI Joint Program Office

This work is sponsored by the U.S. Department of Defense.

This document is available through Asset Source for Software Engineering Technology (ASSET) / 1350 Earl L. Core Road ; P.O. Box 3305 / Morgantown, West Virginia 26505 / Phone: (304) 284-9000 / Fax: (304) 284-9001 / e-mail: sei@asset.com / WWW: http://www.asset.com/sei.html

Copies of this document are available through the National Technical Information Service (NTIS). For information on ordering, please contact NTIS directly: National Technical Information Service / U.S. Department of Commerce / Springfield, VA 22161. Phone: (703) 487-4600.

This document is also available through the Defense Technical Information Center (DTIC). DTIC provides acess to and transfer of scientific and technical information for DoD personnel, DoD contractors and potential con tractors, and other U.S. Government agency personnel and their contractors. To obtain a copy, please contact DTIC directly: Defense Technical Information Center / 8725 John J. Kingman Road / Suite 0944 / Ft. Belvoir, VA 22060-6218. Phone: 1-800-225-3842 or 703-767-8222.

Use of any trademarks in this report is not intended in any way to infringe on the rights of the trademark holder.

1. Introduction

This document outlines the procedures for installing the SQL Ada Module Extensions (SAME) standard packages. It assumes that the source of these packages has been off-loaded from the distribution medium to a machine with the desired Ada compiler. (A description of the distribution medium accompanies the medium.)

Installing the SAME standard packages is a two-step operation. First, a subset of the source must modified; second, the source is compiled into the proper Ada library or libraries.

2. Modifying the Source

There are six files on the distribution medium which need to be modified before compilation. These files are:

1. SQL_Standard (a package specification)
2. SQL_System (a package specification)
3. SQL_Communications_Pkg_Body (a package body)
4. SQL_Database_Error_Pkg_Body (a package body)
5. To_String (a subunit body)
6. To_SQL_Char_Not_Null (a subunit body)

2.1. SQL_Standard and SQL_System

The packages SQL_Standard and SQL_System (see Figures 1 and 2) contain type definitions and constants that describe the DBMS to the SAME. Strictly speaking, the package SQL_Standard is part of the ANSI SQL standard. The type and subtype definitions in SQL_Standard describe the standard ANSI database types INT, SMALLINT, REAL, DOUBLE PRECISION, and CHAR.

The information needed to describe the numeric types can be found in the documentation supplied by the DBMS vendor. If, for example, the DBMS data type INTEGER (or INT) references 32-bit integers, and SMALLINT references 16-bit integers, these type definitions should be:

```
type Int is range -(2**31) .. (2**31) - 1;
type Smallint is range -(2**15) .. (2**15) - 1;
```

The treatment of character data in SQL_Standard is meant to support the character sets defined by the implementor. If the DBMS character set is ASCII, then the definitions in SQL_Standard should read:

```
package Character_Set renames Standard;
subtype Character_Type is Character_Set.Character;
```

Otherwise, it is assumed there exists an Ada package that supports the DBMS character set for Ada programs. Assuming that that package is called Host_Character_Support_Pkg, and that the character enumeration type definition within that package is named *Character*, the above definitions should read:

```
package Character_Set renames Host_Character_Support_Pkg;
subtype Character_Type is Character_Set.Character;
```

The package SQL_System declares two constants whose values cannot be deduced from the information in SQL_Standard. The constant MAXCHRLEN is the length of the longest string the DBMS will store. This value must be in the range of the Ada predefined type Standard.Integer for the target compiler. The constant MAXERRLEN is the length of the longest error message the DBMS error message routine will produce. This error message routine or function, which is not controlled by the ANSI standard, varies from DBMS to DBMS. The value of MAXERRLEN must not exceed MAXCHRLEN. Setting MAXERRLEN to MAXCHRLEN will usually work, but may waste space.

Note Well: The packages SQL_Standard and SQL_System are basic to every SAME application. Recompilation of these packages necessitates recompilation of every SAME standard package and application. It is imperative that the values in these packages be correctly set.

```
  package Sql_Standard is
  package Character_Set renames csp;
    subtype Character_Type is Character_Set.cst;
    type Char is array (positive range <>)
        of Character_Type;
    type Smallint is range bs..ts;
    type Int is range bi..ti;
    type Real is digits dr;
    type Double_Precision is digits dd;
--  type Decimal is to be determined;
    type Sqlcode_Type is range bsc..tsc;
    subtype Sql_Error is Sqlcode_Type
        range Sqlcode_Type'FIRST .. -1;
    subtype Not_Found is Sqlcode_Type
        range 100..100;
    subtype Indicator_Type is t;

--    csp is an implementor-defined package and cst is an
--    implementor-defined character type. bs, ts, bi, ti, dr,
--    dd, bsc, and tsc are implementor-defined integral values.
--    t is int or smallint corresponding to an
--    implementor-defined <exact numeric type> of indicator
--    parameters.

end sql_standard;
```

Figure 1: The Package SQL_Standard

2.2. SQL_Communications_Pkg and SQL_Database_Error_Pkg

The packages SQL_Communications_Pkg and SQL_Database_Error_Pkg are used for SAME standard error processing. (See the document *Guidelines for the Use of the SAME,* CMU/SEI-89-TR-16, for details.) The installation of these packages may require modifying their bodies. Modification of the specifications of these packages is not generally required. The body of SQL_Communications_Pkg will depend on the platform, that is, the Ada compiler and DBMS. The body of SQL_Database_Error_Pkg is independent of the platform but dependent on decisions to be made at installation time.

```
-- SQL_System is a "platform-specific" package
--   within the SAME
package SQL_System is
    -- MAXCHRLEN is the upper bound of the SQL_Char_Pkg
    --   subtypes SQL_Char_Length and SQL_Unpadded_Length
    -- SQL_Char_Length is a subtype of Natural with a lower
    --   bound of 1
    -- SQL_Unpadded_Length is a subtype of Natural with a lower
    --   bound of 0

    MAXCHRLEN : constant integer := str_leng;

    -- MAXERRLEN is the maximum length of the error message
    --   string returned from the DBMS error message function

    MAXERRLEN : constant integer := msg_leng;

end SQL_System;
```

Figure 2: The Package SQL_System

The package SQL_Database_Error_Pkg declares the procedure Process_Database_Error. This procedure is called by abstract module procedures when an unexpected DBMS exceptional condition (non-zero SQLCODE value) has occurred. The procedure should take whatever action is mandated by local policy for error reporting. The exception SQL_Database_Error, defined in SQL_Communications_Pkg, will be raised upon termination of Process_Database_Error.

The distributed version of Process_Database_Error calls the function SQL_Database_Error_Message in package SQL_Communications_Pkg, which returns an explanatory error message acquired from the DBMS. It then prints the message with text_io.put. The installer should modify this body as appropriate for his or her installation.

2.3. To_String and To_SQL_Char_Not_Null

The subunits To_String and To_SQL_Char_Not_Null are used for converting character representations. If the DBMS character set is ASCII, the distributed versions of these subunits can be used without modification.

Assume that is not the case. Assume further the package Host_Character_Support_Pkg mentioned earlier contains routines for converting between its string type, Host_String, a one dimensional array over the type Host_Character_Support_Pkg.Character, and the Ada predefined type Standard.String. Specifically, suppose that within the support package are to be found functions with the following declarations:

```
function To_String (Value : Host_String) return String;
function To_Host_String (Value : String) return Host_String;
```

In that case, these subunits could be recoded as follows:

```
with Host_Character_Support_Pkg;
separate (SQL_Char_Pkg)
function To_String (Value : SQL_Char_Not_Null)
    return String is
package HCSP renames Host_Character_Support_Pkg;
begin
    return (HCSP.To_String(HCSP.Host_String(Value)));
    -- notice explicit Ada type conversion
end To_String;

with Host_Character_Support_Pkg;
separate (SQL_Char_Pkg)
function To_SQL_Char_Not_Null (Value : String)
    return SQL_Char_Not_Null is
package HCSP renames Host_Character_Support_Pkg;
begin
    return (SQL_Char_Not_Null(HCSP.To_Host_String(Value)));
    -- notice inverse Ada type conversion
end To_SQL_Char_Not_Null;
```

Note: Unlike the package specifications SQL_Standard and SQL_System, the remaining user modifications are all to bodies. Whereas modifications to the package specifications must be made before any compilations can take place, the remaining modifications can be done later, and redone, without recompilation effects.

3. Compilation Order

The SAME standard packages are divided into groups. Packages within a group may be compiled in any order. All packages in one group must be compiled before compiling packages in the next group. Notice that the bodies of packages are distributed in separate files from those containing the specifications of pacages. These files are named <package name>_Body.

Group I.

1. SQL_Standard
2. SQL_System
3. SQL_Exceptions

Group II.

1. SQL_Boolean_Pkg

Group III.

1. SQL_Char_Pkg
2. SQL_Real_Pkg
3. SQL_Double_Precision_Pkg

Group IV.

1. SQL_Communications_Pkg
2. SQL_Int_Pkg
3. SQL_Smallint_Pkg
4. SQL_Enumeration_Pkg

Group V.

1. SQL_Base_Types_Pkg
2. To_SQL_Char_Not_Null
3. To_String
4. SQL_Decimal_Pkg

Group VI.

1. SQL_Database_Error_Pkg

Table of Contents

List of Figures