Chapter 3
Report Driver

This chapter describes the following Report Driver Menu options, utilities, and elements:

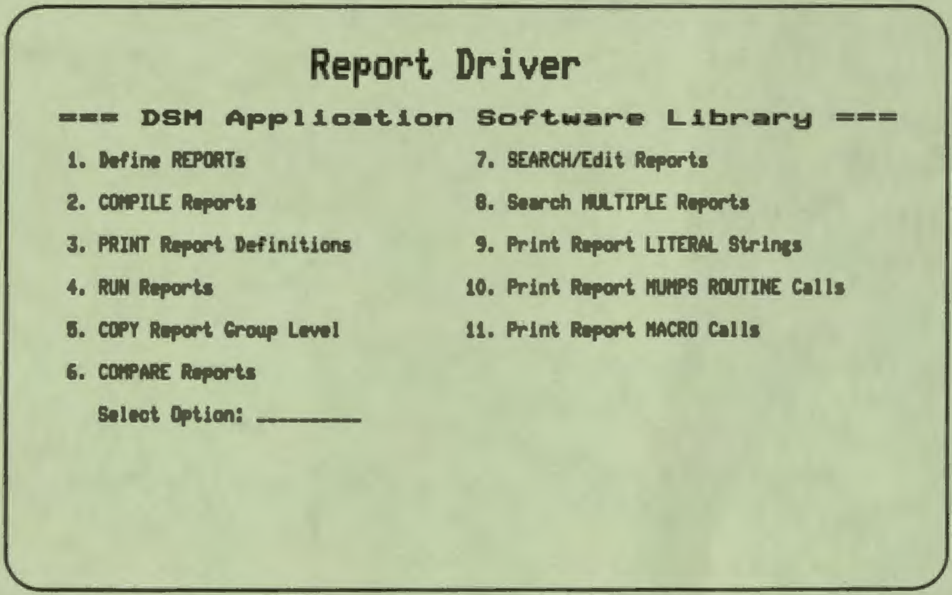
- Define Reports Option

  Collection Specifications
  Sort Lists
  Format Group Specifications
  Print Items
  Report Driver Functions
  Report Driver Macros
  Report Driver Variables
  Report Driver Comments
  Format Descriptors

- Compile Reports Option
- Run Reports Option
- Copy Report Group Level Option
- Calling Reports from Data Screens and Option Screens
- Report Driver Utilities

Figure 3–1 shows the Report Driver Menu and its options.

Figure 3–1 Report Driver Menu



```
                         Report Driver
      === DSM Application Software Library ===

  1. Define REPORTs                  7. SEARCH/Edit Reports
  2. COMPILE Reports                 8. Search MULTIPLE Reports
  3. PRINT Report Definitions        9. Print Report LITERAL Strings
  4. RUN Reports                    10. Print Report MUMPS ROUTINE Calls
  5. COPY Report Group Level        11. Print Report MACRO Calls
  6. COMPARE Reports
     Select Option: ________
```

This chapter describes each option in detail. See the *DASL Programmer's Guide* for instructions and examples on how to create reports for your application.

Define Reports Option

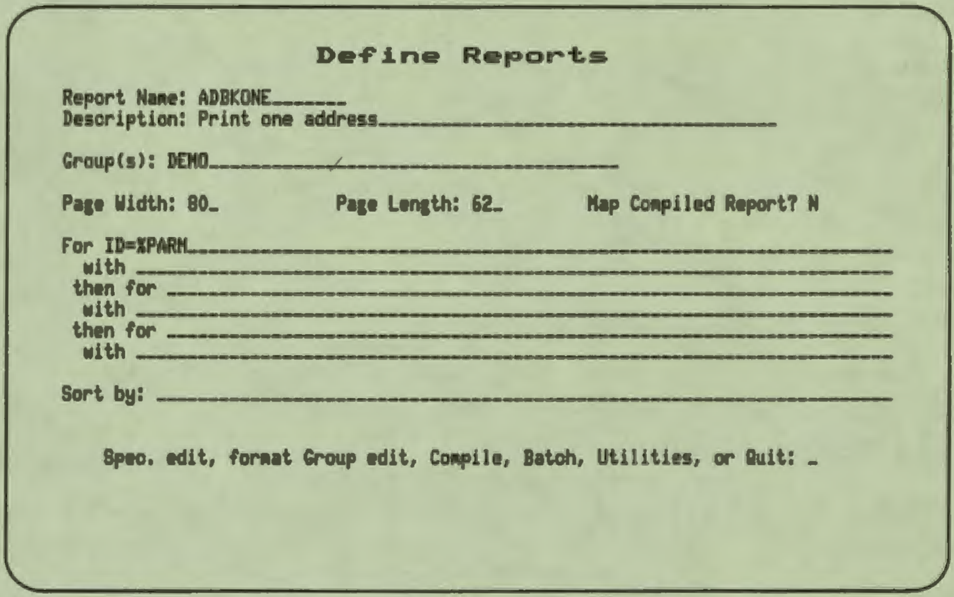
You use the Define Reports Option to create, edit, compile, copy, delete, print, or add documenting comments to reports. The Define Reports Option consists of two screens.

Collecting and Sorting Report Input

In the first screen of the Define Reports Option, you enter information to identify the report and also specify how to collect and sort report data. The latter specifications are called the *collection specifications* and the *sort list*.

To create a new report or edit an existing report, enter a report name (one letter followed by 0 to 13 alphanumeric characters) after the "Report Name:" prompt. Figure 3–2 shows the first screen of the Define Reports Option.

Figure 3–2 Define Reports Screen #1



```
                        Define Reports

   Report Name: ADBKONE_______
   Description: Print one address____________________________________

   Group(s): DEMO_____________________________________

   Page Width: 80_           Page Length: 62_       Map Compiled Report? N

   For ID=%PARM______________________________________________________
     with ___________________________________________________________
    then for ________________________________________________________
     with ___________________________________________________________
    then for ________________________________________________________
     with ___________________________________________________________

   Sort by: _________________________________________________________

       Spec. edit, format Group edit, Compile, Batch, Utilities, or Quit: _
```

Table 3–1 describes the information you enter to collect and sort data.

Table 3–1 Information for Collection and Sorting

Item	Type of Information	Examples
GENERAL REPORT INFORMATION		
Report Name	A name for the report containing a leading alphabetic character followed by 0 to 13 alphanumeric characters.	ADBKALL ADBKONE
Description	A free-text description of the report for documentation purposes that helps users make accurate selections from lookup lists.	Lists all addresses.
Group(s)	Required field that classifies reports by application area. Groups are used as selection criteria in the utilities.	DEMO PHYS
Page Width	The width of the report in characters. The default width is 80.	132
Page Length	The length of the report page in lines. A 0 entry produces the report as a formatted sequential file with no control characters or line feeds. The default length is 62.	55
Map Compiled Report?	A flag that is used by the Mapped Section Utility in the Development Environment. The Mapped Section Utility checks the value of the flag. If the flag value is Y, the DASL software maps the compiled report. Map frequently used reports for efficient disk access.	Y
COLLECTION SPECIFICATIONS		
For . . . with section	Up to three nested specification lines that define the data collections to include in the report. A specification line consists of a collection expression (For . . .) that includes a data name, including qualifiers, to be processed by the report, and a list of conditional expressions (with . . .) or truth-valued expressions that the collected data must satisfy to be included in the report.	For DDNM with @ D %VALID For SDNM=%PARM with SDTYPE="OPTION" For TOBDAT From %PARM("FDT") to %PARM("TDT")

Table 3–1 (Cont.) Information for Collection and Sorting

Item	Type of Information	Examples
SORT SPECIFICATIONS		
Sort by	A list of data names whose order specifies how the report is sorted. The first data name in the list corresponds to level 1, the next to level 2, and so on. Associated with each nested sort level is an identifying header and a trailer for summary information. You usually specify the report name and description at header level 0.	NAME,JOBDESC

After you complete the sort list, the DASL software displays the following menu prompt:

```
Spec. edit, format Group edit, Compile, Batch, Utilities, or Quit:
```

Editing Existing Reports

If the report you specify at the "Report Name:" prompt is an existing one, the DASL software fills in the screen and also displays the menu prompt.

Enter S in response to the menu prompt to edit the information on the screen. You can then change any of the data (except the report name) and save your changes.

Compiling the Report

Enter C to compile the report in your current process. The DASL software then compiles the report and returns to the menu prompt only when it completes the compile operation.

Enter B at the menu prompt to compile the report in a batch process. The DASL software then submits a batch job and returns to the menu prompt immediately.

Using the Utilities

Enter U at the menu prompt to delete, copy, print, declare user-defined variables, or add documenting comments to the report. The DASL software then displays the following utilities decision prompt:

```
Edit comments, Variables, Delete, Copy, Format, Print, or Search:
```

Adding Comments

To add comments to document the report, enter E. You then see a comments screen that you can use to specify the creator, editor, and version number of the report, and also to enter text in a scroll region to document the report.

Declaring Variables

Because data names and variables share the same syntax, the DASL software treats any DASL name in a report as a data name. Before you can use a local variable in the report, you must declare it as a local variable through the variable utility option.

Enter V at the menu prompt to define variables. The system displays a scroll region where you can list all local variable names used in the report. For each local variable, supply the following:

- Name
- Description
- Deletion flag

The value of the deletion flag determines the clean up of the variable. If the value of the deletion flag is Y, then the DASL software adds the variable to the KILL statement at the end of the report. If the value of the deletion flag is N, then the DASL software does not add the variable to the KILL statement at the end of the report.

Deleting Reports

To delete the report, enter D in response to the utilities decision prompt and verify your choice where prompted.

Copying Reports

To make a copy of the report under a different name, enter C and specify the new report name. You can later edit the new report to change any of the copied values.

Displaying Report Images

To display an image of the report on your terminal screen, enter F. The image you see is a rough outline of the report format. You can use this image to format the report header level group without running the report.

Printing Reports

To print the report, enter P in response to the utilities decision prompt. The DASL software displays a device selection screen that you can use to produce the report through a batch process or through your current process. If you select a batch process, you can print the report on any VMS device or write the report to a VMS file. If you produce the report in your current process, you can also display the report on your terminal.

Searching and Editing Report Text Strings

To search for and edit text strings in the report, enter S in response to the utilities decision prompt. The DASL software then calls the Search/Edit Reports Option.

Starting Over

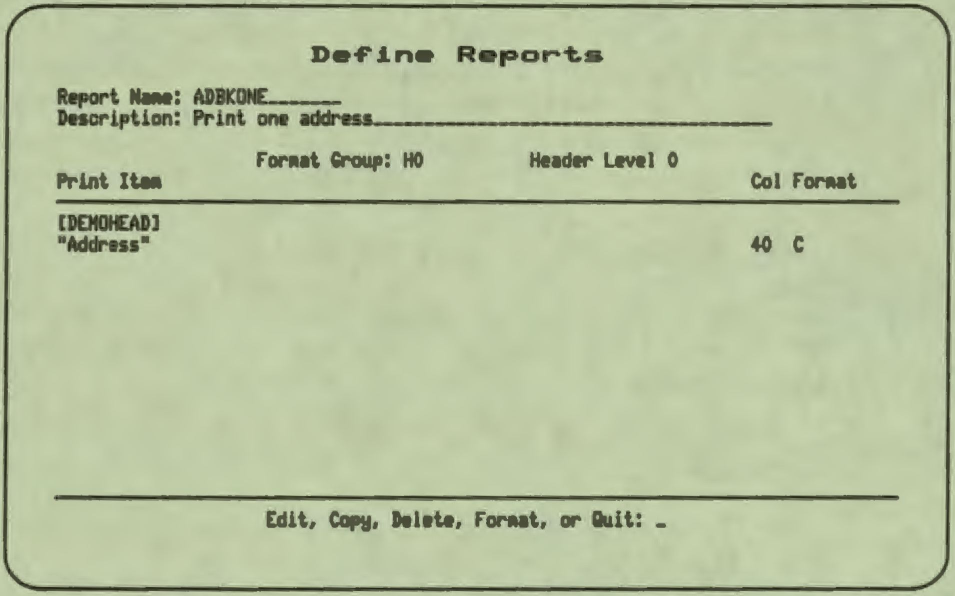
Enter Q at the menu prompt to start over. The DASL software moves the cursor back to the "Report Name:" prompt.

Specifying the Report Output

You can specify the form and content of report output in the second screen of the Define Reports Option. The specifications you enter in this screen are called the *format group specifications* and the *print items*.

Enter G (format Group) at the menu prompt of the first screen to move to the second screen of the Define Reports Option. You then see the screen shown in Figure 3–3.

Figure 3–3 Define Reports Screen #2



```
                           Define Reports

Report Name: ADBKONE_______
Description: Print one address___________________________________

                Format Group: H0           Header Level 0
Print Item                                                  Col Format
______________________________________________________________________
[DEMOHEAD]
"Address"                                                   40  C





______________________________________________________________________
                Edit, Copy, Delete, Format, or Quit: _
```

Table 3–2 describes the information you enter in this screen.

Table 3–2 Information for Defining Report Output

Item	Type of Information	Examples
Format Group	The name of a set of print specifications for the report. A format group name includes the name of a report section, which can be data (D), header (H), trailer (T), or footer (F), and the number of the level if this is a header or trailer section.	H1, H2 D T1, T2 F
Level	A data, header, trailer, or footer level. If at the header or trailer level, the number of the level is also displayed. This number corresponds to the data name's position within the sort list you specified on the first Define Reports Screen.	Data Level Header Level 1
Print Item	The name of any of the following that you want to print in this data or header level: a data name, a VAX DSM expression, a literal string in quotation marks, a Report Driver Function, or a macro.	D ^SUBR NAME [ADBK]
Col	The column where you position the print item.	2 15
Format	The position of the print item within the report field: left, center, or right justify.	L, C, or R.

To create a new format group or edit an existing format group, enter the format group name in response to the "Format Group:" prompt. If you enter an existing format group, the DASL software displays in the scroll region the list of print items for the format group. You can then choose to edit the existing print items or add new print items to the list. If the format group you select does not exist, the DASL software moves the cursor to the scroll region so that you can define the print items for the new format group.

When you have entered print items, the DASL software displays the following prompt:

```
Save, Edit, Copy, Delete, Format, or Quit:
```

At this prompt, you can:

- Enter S to save the specifications and any changes you have made.
- Enter E to edit the specifications.
- Enter C to copy the print items for the current report level to another report level within the same report.
- Enter D to delete the current report level.
- Enter F to display an image of the report on your terminal screen or a specified output device.
- Enter Q to quit the screen without saving changes.

Copying Report Group Levels

Enter C to copy print items for a report group level to another level within the same report. At the "From line:" and "To line:" prompts, the DASL software displays the beginning and ending lines of print items in the scroll region for the current report level. At these prompts, you can:

- Press Return to copy all print items from the current level.
- Enter line numbers to copy a section of print items.

The DASL software then prompts you for the report group level to which you want to copy print items in the current level. At the "Group to:" prompt, enter the name of another report level in the same report, or enter an asterisk (*) for a list.

If the report group level you specify is an existing one, the DASL software displays the line number of the last line of that level at the "Insert after line:" prompt. In this field, you can:

- Press Return to insert the copied print items after the last line of the report level print items.
- Enter a line number to insert the copied print items within the listing of print items.

Note: From this screen, you can only copy report group levels within the same report. To copy a report group level from one report to another report, use the Copy Report Group Levels Option of the Report Driver Menu.

Related Sections

Collection Specifications
Copy Report Group Level Option
Format Group Specifications
Print Items
Report Driver Variables
Search/Edit Reports Option
Sort Lists

Collection Specifications

Collection specifications identify the data that you want to collect for a report. Each collection specification can consist of a data name, an optional qualifier, and an optional conditional expression. You define collection specifications in the first screen of the Define Reports Option.

Collection Data Names

Collection specifications consist of up to three pairs of For..With statements with the following syntax:

```
For:   data name{qualifier}
 with:    {conditional expression}
```

where

data name is a data name that identifies the global whose values you want to collect (The data name must be a key data name or a dependent data name with a cross-reference.)

qualifier is an optional qualifier to further specify the values to collect

conditional expression is an expression that defines the circumstances of the collection

The For statements indicate DSM $ORDER loops. The data name you specify at the "For:" prompt functions as the data name you use as a $ORDER argument. For example, the statement For: DATA1 generates the following DSM code:

```
TAG       S DATA1=""
          S DATA1=$O(^DATA(DATA1)) I DATA1="" Q
          G TAG
```

The DASL software uses the data name to access the global and retrieve data values directly from the global node in which the data name is a subscript. For this reason, the data name you specify in a collection specification is usually the primary key of the global. If you use key data names, the DASL software can retrieve the values directly from the global in which the data names are subscripts.

You can, however, use a dependent data name that has a cross-reference global defined for it. If you use a dependent data name, the DASL software first obtains the global reference and the primary key of the cross-reference global from the Data Dictionary entry for the dependent data name. Then the DASL software redefines the cross-reference global node as a primary key and proceeds as with primary key data.

You must use either a key data name or a dependent data name with a cross-reference in the For statement.

You can also build a chain of alternate keys by using the binary OR operator with data names you use in collection expressions. To do so, you replace the single data name with a complex expression containing the binary OR operator. For example:

```
dataname1!dataname2!dataname3!...datanameN
```

Nested Collection Expressions

You can include up to three collection specifications in a report, by using the following syntax for the second and third specifications:

```
then for:   data name{qualifier}
with:   {conditional expression}
```

The DASL software nests any collection expression beyond the first in the collection expression that precedes it. Suppose you specified the data name NAME for the first collection expression, the data name VISIT for the second collection expression, and the data name RESULT for the third collection expression.

For each value of NAME that fits the first collection specification, the DASL software retrieves every value in the nodes represented by VISIT that fits the second collection specification. For every value of VISIT, the DASL software retrieves every value in the nodes represented by RESULT that fits the third collection expression.

Limiting the Range of Collection

Qualifiers and conditional expressions limit the data that you collect for the report. The DASL software first applies qualifiers to select a range of nodes (individual records) from the global subscripting level represented by the data name. The DASL software then applies conditional expressions to the values in those nodes to select only the values you want. For example, the following collection specification collects all records for a data name NAME:

```
For:  NAME
```

The following collection specification collects only the records for names between A and H:

```
For:  NAME FROM "A" TO "H"
```

The following collection specification collects records for names between A and H that also contain the state MA:

```
For:  NAME FROM "A" TO "H"
 with:  STATE="MA"
```

Qualifiers

You can use qualifiers to limit the range of sibling nodes from which your report collects data values. When you specify qualifiers, the DASL software returns only values from nodes that fall within the qualifier range.

Each qualifier can be made up of any valid expression and the character strings From . . . to, or Equal to (=). Table 3–3 shows the qualifier forms and meanings.

Table 3–3 Collection Qualifier Forms

Form	Meaning
For **name**	Collects all values of **name**
For **name** from **exprl** to **expr2**	Collects all values of **name** from **exprl** through **expr2**
For **name** from **expression**	Collects all values of **name** starting with the value **expression**
For **name** to **expression**	Collects all values of **name** through **expression**
For **name=expression**	Collects only the **expression** value of name

Conditional Expressions

You can use conditional expressions to exclude data from the report collection. If the conditional expression is true, the DASL software sets %FND to 1, and selects the associated record. If the conditional expression is false, the DASL software sets %FND to 0, and does not select the record.

A conditional expression can be any of the following:

- Dependent data
- Truth-valued expression
- VAX DSM statements preceded by an at sign (@)

Truth-Valued Conditional Expressions

You can use any expression that produces a truth value of 1 (true) or 0 (false). DASL truth-valued conditional expressions are more restrictive than truth-valued expressions described in the *VAX DSM Language Reference Manual*.

Each truth-valued conditional expression must contain at least one of the following:

- String literals
- Numeric literals
- Variables (can be either data names or local variable names)

Truth-valued conditional expressions can also contain any of the following:

- Numeric relational operators
- String relational operators
- Logical operators

You can also use parentheses in truth-valued conditional expressions to change the order of evaluation, in the same way you use parentheses when you write VAX DSM routines.

For example, to test whether DATE is before today and less than the value of %PARM (passed from a screen) or whether CODE is 5, you can write the following conditional expression:

```
with:    CODE=5!(DATE<"T"&(DATE>%PARM))
```

Using the Variable %PARM

You can use the DASL local variable or array %PARM to pass data from a screen to a report. Many reports are run as batch jobs. Because batch jobs use a VMS command procedure to execute a DSM routine, local variables defined in previous screens are not available to the report. You can use %PARM to pass data from a previous screen to a report through the command procedure.

For example, in the DASL demonstration system, users can specify a name to print in the Print Selected Addresses Option (screen ADBKSEL). The ADBKSEL screen sets the value of %PARM=ID, the primary key data name for the name that the user entered. The following example shows the definition for the CQ field of the ADBKSEL screen. The CQ field sets the value of %PARM, identifies the report to call as %RPNAME, and passes the value of %PARM to the report:

```
Field Name: CQ              Data Name:
Description: Continue or Quit.
Line: 8          Column: 40       Justify: C             Data Length: 1
Prompt: "Continue or Quit: "
Help Text: "Enter "C" to continue or "Q" to quit."
Attributes: DEMAND ; REQUIRED ; EXIT CLEAN
Validations: TABLE C,Q
    ------------------------------ Actions --------------------------
    NXTSCN:%RES="Q" ; SET %RPNAME="ADBKONE" ; SET %PARM=ID
    DONP %REPDEV ; NXTSCN
    -----------------------------------------------------------------
```

The following example shows the collection specification for the ADBKONE report:

```
For ID=%PARM
```

VAX DSM Statements

You can use a set of VAX DSM statements as a conditional expression. If you begin a ..with statement with an at sign (@), the DASL software executes the remaining characters in the statement as VAX DSM statements and returns the truth value in the variable %FND, for example:

```
with:   @ S %FND=$D(DATG($J,"U","D",DDNM))

with:   @ S %FND=1

with:   @ I '$D(DDNM) D INIT^DAINIT S %FND=1
```

When you use VAX DSM statements as a conditional expression, you must define the variable %FND explicitly, either within the VAX DSM statements or outside the statements.

Related Sections

Define Data Names Option, Chapter 1
Define Reports Option
Sort Lists

Sort Lists

You specify a sort list for each report through the Define Reports Option. The sort list is a list of sort keys that specifies how to sort the report data you collect.

Explanation

Generally, a sort list is made up of primary keys or dependent data names. You can also specify the following as sort keys:

- A pointer chain that the DASL software evaluates to retrieve values for a dependent data name
- A user-defined variable

The DASL software sorts by building a sort global using the sort list as subscripts. Therefore, the total length of values of sort data names cannot exceed 90 for any single record. To reduce the number of characters in the values of data names in the sort list, enter a colon and a length limit after the sort data name. The DASL software truncates the values for sorting purposes.

Pointer Chains in Sort Lists

Using pointer chains, you can access data in one global using data stored in another global. Each pointer in the chain is a dependent data name in one global that has the same value as a primary key data name (subscript) in another global. In the Report Driver, you can construct a pointer chain as follows:

result(pointer1{,pointer2,...,pointerN})

where:

result is a dependent data name to print

pointer1 is a dependent data name that points to a key data name

pointer2 is a second dependent data name that is pointed to by the pointer data name associated with pointer1 (Pointer2, in turn, points to a key data name.)

pointerN is the last dependent data name in the chain (PointerN points to a key data name that contains the desired data as a field in its global node.)

For example, the report AAUDIT contains the following collection specifications and sort list:

```
For   BADT FROM %PARM("FDT") TO %PARM("TDT")
  with
 then   for BATM
  with
 then   for BACNT
Sort by:   BADT,BATM,PNAME(BAACC,ICNEF),PIC,BAACC
```

In the sort list, the item PNAME(BAACC,ICNEF) represents a pointer chain. PNAME is a dependent data name that is not stored in the same global as the collection key data names BADT and BATM.

BADT and BATM are keys in the ^AUDIT("BILL",BADT,BATM) global. PNAME is a dependent data name in the ^PF(ICN,"DEMOG") global.

To access PNAME, the following process occurs in the pointer chain:

1. The pointer data name BAACC in the ^AUDIT("BILL",BADT,BATM) global points to the key data name ACC in the global ^EF(ACC,"INFO").
2. Then, the pointer ICNEF in the ^EF(ACC,"INFO") global points to the key data name ICN in the ^PF(ICN,"DEMOG") global.
3. Since PNAME is a dependent data name in ^PF(ICN,"DEMOG"), you can now access the values for PNAME.

Note that the Report Driver traverses the pointer chain from left to right. In the Report Driver, the DASL software evaluates pointer chains at print time rather than at collection time. In addition, the DASL software reevaluates pointer chains each time they are encountered.

User-Defined Variables in Sort Lists

When you specify a user-defined variable as a sort key, the DASL software does not know the maximum length of its value. You can limit the sort length by truncating these values on the report. The truncated format then becomes:

variable:length

Reverse Sorting

On occasion, you may want to sort certain items in reverse order. Suppose you want to sort data in a report on a DATE data name so that information associated with the latest dates prints first and information associated with the earliest dates prints last.

Because the DASL software keeps dates internally in $HOROLOG format, DATE data names are sorted in ascending (numeric collating) order. Therefore, you cannot write such a report unless you specify *reverse sorting order*.

To sort in reverse order, precede the data name for which you want a reverse sort with a minus sign, as follows:

```
Sort by:   -TESTDA
```

Then, when building the temporary sort global, the DASL software precedes sort subscripts for the data-name values with negative numbers.

> **Note: The data name you specify for reverse sorting must have a NUMERIC, INTEGER, DATE, or TIME validation.**

Related Sections

Collection Specifications
Format Group Specifications
Pointers, Chapter 1

Examples

Suppose you have a personnel database consisting of the following global:

```
^EMPFIL(ID)=NAME;DOB;PHONE;STREET;CITY;STATE;ZIP
```

You can create a simple report that lists all employees alphabetically by entering a collection specification of ID (to locate the global that contains the data you want to collect) and a sort list of NAME (to specify the collating order in which you want the values presented on the report).

You do not have to specify a sort list. Without a sort list, the DASL software sorts the values in the order of the data names in the collection expression.

Therefore, using the employee file global ^EMPFIL, described previously, you can create a simple report that lists all employees in order of employee identification by entering a collection specification of ID and no sort list.

You can also specify several data names in the sort list. The only limitation on the number of data names you list is the 90-character length of the subscript in the scratch global. If you specify more than one sort key, the DASL software nests each sort key name within the sort key that precedes it.

Using the employee file global ^EMPFIL, described previously, you can create a complex report that lists all employees by ZIP code and, within ZIP code, alphabetically by name by entering a collection specification of ID and sort keys of ZIP and NAME. See Figure 3–4 in the "Format Group Specifications" section for a description of a report using this sort list.

Format Group Specifications

Format group specifications provide the structure for each page in a DASL report. The DASL software divides all report pages into levels and format groups.

Levels

Each report contains at least one level, called level 0. Level 0 outlines each report page. That is, level 0 provides the banner headers at the top of each page and one trailer in the final report page. You can use header level 0 (H0) to specify the format for printing the date, time, report name and description, and other general information.

In addition, the report page can contain other levels nested within level 0. The number of levels depends on the number of items in your sort list.

Format Groups

Each level can contain format groups called headers and trailers. The DASL software prints headers at the beginning of each iteration of the level. Headers introduce a new value for the sort key at that level. You can specify any number of headers and trailers, depending on the number of data names in your sort list.

The DASL software places trailers at the end of each iteration of the level. Trailers contain summary information about the level.

DASL reports also can contain the following special format groups:

- Data
- Footer

The data format group is a level in itself, called the data level. The data level is nested within the lowest header and trailer level in the report. If you have one sort key, the data level is nested in level 1. If you have two sort keys, the data level is nested in level 2.

The footer is also a level in itself. The items you define in the footer are printed at the bottom of every report page (like a footnote). You can use footers to display fixed information such as installation names, titles, or any other information that you want to display as a footnote at the bottom of the page.

The format group specification is a definition of all the levels and format group sections that a page in the report contains. During report definition, you refer to levels and format groups by a shorthand *format group name*. The format group name for headers is H followed by the level number. The format group name for trailers is T followed by the level number.

For example, header 0 is H0. Header 2 is H2. Because reports only contain one data level, the format group name for the data level is D. The format group name for footers is F.

Table 3–4 lists some standard format group names.

Table 3–4 Standard Format Groups

Format Group/Level	Standard Name
Header 0	H0
Header 1	H1
Header 2	H2
Header 3	H3
Data	D
Footer	F
Trailer 3	T3
Trailer 2	T2
Trailer 1	T1
Trailer 0	T0

Report Structure

The format group specification determines the report structure, depending on the type of output device you select for the report. If you print the report on a hardcopy device, the DASL software places header 0 at the top of every report page and prints any defined nested header levels before printing the data level. If you display the report on a terminal, the DASL software places header 0 only at the top of the screen and scrolls the body of the report.

If trailer 0 is defined, the DASL software writes trailer 0 only at the end of the report. For this reason, trailer 0 is usually used for statistical information covering numeric data, such as the minimum values, maximum values, or mean value found.

As with headers, the way the DASL software treats footers depends on the device to which the report is sent. If you display the report on a terminal, the DASL software does not display a defined footer. If you print the report on a hardcopy device, the DASL software writes the footer print items at the bottom of every printed page on hardcopy reports.

During report compilation, the DASL software computes the length of the footer in lines expressed in terms of the $LINE Report Driver Function. The DASL software subtracts that number of lines in the footer from the page length you specified when you defined the report identification information. The DASL software then stores the result in %PL as the actual page length. When the report prints, the DASL software stops page printing and begins footer printing whenever the number of lines on the page reaches the number specified in %PL.

In many cases, some format groups remain undefined. For example, you may not want to use headers for all nested sort keys in the report. You also may not want to use any trailers, including trailer 0.

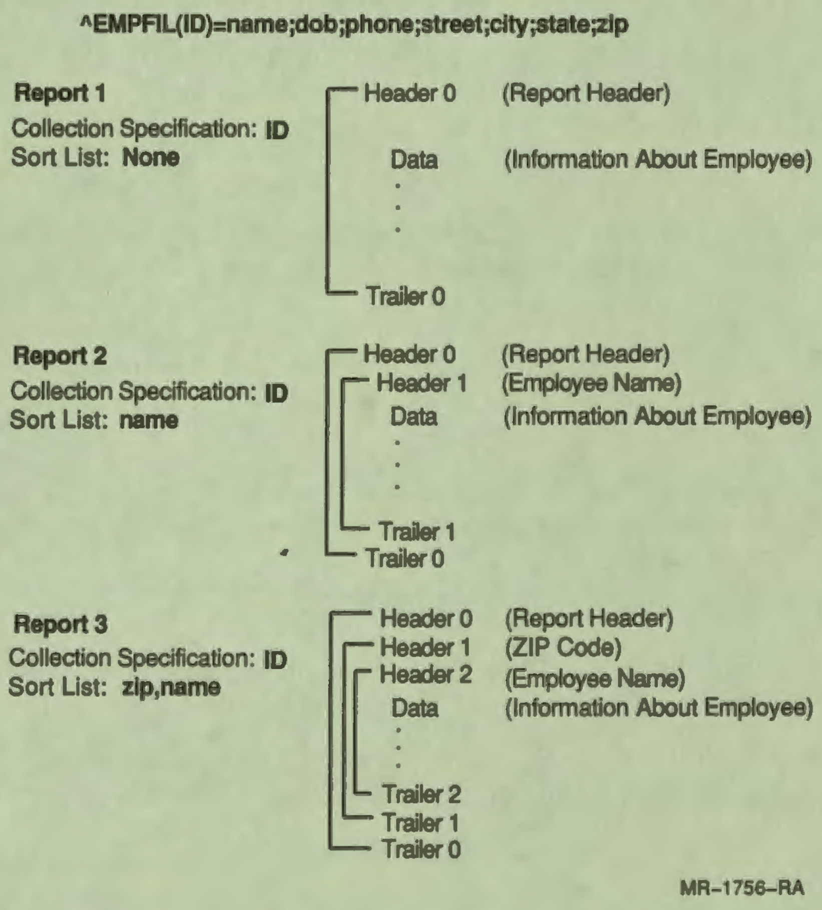
An Example of Report Structure

Suppose you have a simple personnel database consisting of the following global:

```
^EMPFIL(ID)=NAME;DOB;PHONE;STREET;CITY;STATE;ZIP
```

Figure 3–4 shows reports based on this database.

Figure 3–4 Report Structure



Report 1 in Figure 3–4 represents a simple report in which you list all employees by the primary key, ID, the employee identification number. You define the following format group specifications for this report:

- Header 0 (H0) to be printed at the top of every page with print items specifying the name of the report, the time, the date, and your organization
- Data (D) with print items specifying the data names that make up the employee name, address, and zip code, and how you want those elements to be formatted

Report 2 in Figure 3–4 creates a similar report sorted alphabetically by employee name. In this report, the sort list includes the data name NAME. You define the following format group specifications for this report:

- Header 0 (H0) to be printed at the top of every page with print items specifying the name of the report, the time, the date, and your organization
- Header 1 (H1) with print item specifying the employee name
- Data (D) with print items specifying the data names that make up the employee address, and zip code, and how you want those elements to be formatted

Report 3 in Figure 3–4 creates a report listing employees by zip code and within zip code, alphabetically. You define the following format group specifications for this report:

- Header 0 (H0) to be printed at the top of every page with print items specifying the name of the report, the time, the date, and your organization
- Header 1 (H1) with print items specifying the ZIP code
- Header 2 (H2) with print items (data names) specifying the employee name, and how they are to be formatted
- Data (D) with print items specifying the data names that make up the employee address and how you want those address elements to be formatted

Related Sections

Define Reports Option
Print Items
Sort Lists

Print Items

Print items are items you insert in a report header, data level, footer, or trailer level to describe the values you want to print. The print specification for each format group consists of a print item, a column position, and a format descriptor.

Explanation

You can use the following as print items:

- Expressions, including:
  - Data names
  - User-defined variables
  - Executable VAX DSM code
- Report Driver functions
- Macros
- Comments

Expressions

The expression you use as a print item must be a valid argument to a DSM WRITE statement. In addition, the expression can contain only one expression atom (as described in the *VAX DSM Language Reference Manual*).

The following exceptions apply:

- Local variable names must be data names or user-defined variables.
- Data-name qualifiers can be present.
- The at sign (@) designates executable VAX DSM code.

Data Names

Normally, the DASL software regards all names used in reports as data names. You can specify a single data name as a print item, or you can specify a pointer chain that the DASL software evaluates to retrieve values for dependent data names. See the "Sort Lists" section of this chapter for a description of the use of pointer chains in the Report Driver.

Data-Name Qualifiers

You can use qualifiers with data names to perform print operations. The syntax for data-name qualifiers is:

data name{/qualifier}{/qualifier}{:postcond}

Table 3–5 lists the data-name qualifiers and their meanings.

Table 3–5 Data-Name Qualifiers

Qualifier	Meaning
/PROMPT	Print the data name's default prompt from the Data Dictionary with the values.
/PROMPT/STRIP	Print the data name's default prompt with all punctuation stripped (no colon after prompt text). Use this qualifier for column headings.
/PROMPT/NODATA	Print the default prompt but do not print the data value.
/EVAL	Evaluate values for the data name but do not print them at this time. The qualifier /EVAL can be used only with dependent data names. No other qualifier is allowed. You can use just one /EVAL qualifier with multiple data names that share a common global reference as long as you do not use pointer syntax.

You can use multiple data names that share a common global reference with the /EVAL qualifier. Separate data names by commas. You cannot use pointer chain syntax when you specify multiple data names, and you cannot use other qualifiers. The syntax for multiple data names with /EVAL is:

data name,data name/EVAL

User-Defined Variables

You can also specify user-defined variables as print items. Unlike data names, the report compiler does not evaluate a user-defined variable. Before you can use a DASL name as a local variable, you must declare that name to be a local variable through the variables utility in the Define Reports Option. While you define a local variable at report definition time, you assign a value to the variable during run time through the Run Reports Option.

Any local variable you define in a report must be a standard DASL name of up to seven characters, the first of which must be alphabetic. You cannot define local variable names beginning with the percent sign (%) because the DASL software reserves names beginning with the percent character for internal variables.

At execution, the DASL software creates a local array containing all declared local variables in the report. Whenever the DASL software encounters a name in the report, it determines whether the name is a defined data name or a declared variable.

The local variables have precedence over data names. That means that if the DASL software finds a name that is both a local variable and a data name, it assumes that the name refers to the local variable and treats it accordingly.

If the name is not a defined data name, the DASL software checks the name against the entries in the local variable list array. If the name is in the local array, the DASL software proceeds with the report. If the name is not in the local array, the DASL software generates an error.

Executable VAX DSM Statements

The DASL software interprets an expression beginning with the at sign (@) as a line of VAX DSM statements to be executed. For example, the following expression calls the routine named DISX.

```
@ D ^DISX
```

You must include the space between the @ and the D. Otherwise, the DASL software assumes the D to be a line label and the rest of the line to be illegal syntax.

> **Note: Any time you call a VAX DSM routine to set local variables, you must also use the routine to kill those variables. Although the DASL software automatically kills local variables associated with data names, it does not automatically kill local variables you set, unless you declare the variables through the Variables option.**

When you use a VAX DSM routine within a report to print items, you must manually keep track of the line count on the report page. If you are not using the DASL Report Driver routines to print items, the DASL software does not automatically keep track of line count.

Report Driver Functions

Report Driver functions are print items that provide formatting instructions or statistical information within the report.

Some Report Driver functions perform such functions as causing page or line breaks, printing the current date or time, or resetting the report page number.

Other Report Driver functions cause the DASL software to print certain statistical calculations kept during report data collection. You can specify these Report Driver functions only in trailer levels.

See the following section of this chapter for a description of each Report Driver function.

Macros

You can use macros as print items to include a format group level of one report as part of a second report. Macros can provide standardized headers or any other format group information that you want to include in particular reports.

Macros have the following format:

[report name, format group]

where:

report name	is the name of the original report you want to include as a print item in a second report
format group	is a header level, data level, or trailer level in the original report

When you use a macro as a print item in your current report level, the DASL software includes in the current report level all print items in the report and level specified by the macro. If there are any other macros within the macro used as a print item, the DASL software also includes the other macros in the report. If you do not indicate a level in the macro, the DASL software assumes the data level of the specified report.

Macros cannot use a postconditional expression; you can, however, embed a macro within a conditional $SECTION.

Postconditional Expressions

You can append a postconditional expression to any print item except a macro. The format for the postconditional expression is:

print item{:postcond}

If you use a postconditional expression with a print item, the DASL software prints the print item only if the postconditional expression is true.

Comments

You can insert a documenting comment whenever you are defining a header, trailer, or data level. The comment marker is similar to the DIGITAL Command Language (DCL) comment marker, the exclamation point (!).

Related Sections

Define Reports Option
Expression Atoms (*VAX DSM Language Reference Manual*)
Pointers, Chapter 1
Report Driver Functions
Report Driver Macros
Sort Lists

Examples

The following examples show valid print items:

`NAME/PROMPT`	data name, qualifier
`[REPHEAD]`	macro
`"Page"_%PC`	literal string, variable
`AGE:BTHDAY'=""`	data name with postconditional
`$LINE("_",1,60)`	Report Driver function

Report Driver Functions

Report Driver functions are print items you can use in your reports. Each Report Driver function is an uppercase command preceded by a dollar sign ($).

Explanation

Some Report Driver functions provide formatting instructions: they mark the beginning and end of logical sections, control pagination, and print the current date and time. Some Report Driver functions provide statistical information: they cause the DASL software to print statistical calculations kept during report data collection.

Formatting Functions

Table 3–6 lists the Report Driver functions that perform formatting functions and their effects.

Table 3–6 Report Driver Formatting Functions

Function	Action
$DATE	Prints the date on which the report image was created with collected data.
$ENDREP	Ends a logical looping section begun with $REPEAT.
$ENDSEC	Ends a logical section begun with $SECTION.
$LINE	Ends the current line. The DASL software performs an implicit $LINE at the end of every format group specification.
$LINE(**integer**)	Terminates the current line and performs the number of new line operations specified by **integer**.
$LINE(**"char",left,right**)	Terminates the current print line and prints a new line consisting of repetitions of the quoted character argument. The repetitions begin at the column specified by **left** and end at the column specified by **right**. Both **left** and **right** are optional. If you do not specify them, the DASL software assigns 1 as the value of **left**, and the page width as the value of **right**.
$NOLINE	Suppresses the implicit $LINE Report Driver function at the end of a format group specification, permitting multiple format groups to print on the same physical line.
$PAGE	Ends the current print line and performs a new page operation. $PAGE ignores the current column-position and format-list values.

Table 3–6 (Cont.) Report Driver Formatting Functions

Function	Action
$PAGEC	Prints the current page count.
$PAGEN	Prints the current page number.
$QUITREP[:postcond]	Conditionally ends a logical looping section that begun with $REPEAT. $QUITREP does not replace $ENDREP.
$REPEAT(**data name**)	Begins a logical section of print items to be looped. The DASL software prints the section once for each value of the data name you specify in the $REPEAT argument. You end the section with $ENDREP.
$REQUIRE(**lines**)	Tests whether the number of lines specified in the argument remain at the bottom of the report page. The DASL software forces a new page if fewer lines than are specified in the **lines** argument remain to be printed.
$SECTION	Begins a logical section of print items without looping. You must end the section with $ENDSEC.
$SETPAGE	Resets the current page number so that the page that follows the current page has the page number 1.
$TIME	Prints the time that the report image was created with collected data.

Statistical Functions

Some Report Driver functions cause the DASL software to print statistical calculations kept during report data collection. You can specify these functions only in trailer levels.

Table 3–7 lists the Report Driver functions that cause the DASL software to print statistical calculations and their effects.

Table 3–7 Report Driver Statistical Functions

Function	Action
$AVE(**data name**)	Prints the average value of the specified data name within the current level.
$COUNT	Prints the number of records within the current level.
$MAX(**data name**)	Prints the maximum value of the specified data name within the current level.
$MED(**data name**)	Prints the median of values for the specified data name within the current level.
$MIN(**data name**)	Prints the minimum value of the specified data name within the current level.
$STD(**data name**)	Prints the standard deviation of values for the specified data name within the current level.
$TOTAL(**data name**)	Prints the total of all the values for the data name within the current level.

Data Types and Statistical Functions

DASL data names are one of five types: DATE, TIME, STRING, NAME, or NUMERIC. In many cases, you cannot perform a statistical function on all data types.

Table 3–8 shows the statistical functions that you can use with each data type. The vertical column on the left displays statistical functions. The horizontal column at the top displays DASL data types. A Y or an N at the conjunction of each column indicates whether you can use that statistical function with the data type.

Table 3–8 Data Types and Statistical Functions

	DATE	TIME	STRING	NAME	NUMERIC
$AVE	Y	Y	N	N	Y
$COUNT	Y	Y	Y	Y	Y
$MAX	Y	Y	N	N	Y
$MED	Y	Y	N	N	Y
$MIN	Y	Y	N	N	Y
$STD	N	N	N	N	Y
$TOTAL	N	N	N	N	Y

As shown in Table 3–8, you can use the $STD and $TOTAL functions only with NUMERIC data types. You can specify STRING or NAME data types only with the $COUNT function.

Related Sections

Define Reports Option
Print Items
Report Driver Comments
Report Driver Macros

$AVE

The $AVE function prints the average value of the specified data name in the current report level.

Format

$AVE(data name){:postcond}

where:

data name is the data name whose average value you want

postcond is an optional postconditional expression

Comments

Keep the following points in mind when you use the $AVE function:

- You can use the $AVE function only within trailer format group specifications. You cannot use $AVE within the T0 format group level.
- You cannot use the $AVE function with STRING or NAME type data names.
- If you add a postconditional expression to the $AVE function, the DASL software prints the average data-name value only if the postconditional expression is true.

Related Sections

Collection Specifications
$COUNT
Define Reports Option
$MAX
$MIN
Report Driver Functions
$STD
$TOTAL

$COUNT

The $COUNT function prints the number of data values written within the current report level.

Format

$COUNT{:postcond}

where:

postcond	is an optional postconditional expression

Comments

Keep the following points in mind when you use the $COUNT function:

- You can use the $COUNT function only within trailer format group specifications. You cannot use the $COUNT function in the T0 format group level.
- You can use the $COUNT function with all data name types, including STRING and NAME.
- The $COUNT function counts data values only within the current report level; it does not provide a running count for an entire report.
- If you add a postconditional expression to the $COUNT function, the DASL software prints the number of values written only if the postconditional expression is true. The postconditional applies only to the printing of the $COUNT function; it does not apply to the collection of data. For example, the following postconditional expression applies to the collection of data and is not valid:

  ```
  $COUNT:SALARY=>50000
  ```

  The following postconditional expression applies to the printing of the $COUNT function and is valid:

  ```
  $COUNT:SALARY
  ```

Related Sections

$AVE
Collection Specifications
$MAX
$MIN
Report Driver Functions
$STD
$TOTAL

$DATE

The $DATE function prints the date on which the report print image was created from collected data.

Format

$DATE{:postcond}

where:

postcond is an optional postconditional expression

Comment

If you add a postconditional expression to the $DATE function, the DASL software prints the date only if the postconditional expression is true.

Related Sections

Report Driver Functions
$TIME

$LINE

The $LINE function ends the current report line and begins a new line. Optionally, you can also use the $LINE function to repeatedly print a character on the new line or to perform a specified number of new-line operations.

Format

$LINE{:postcond}

$LINE(integer){:postcond}

$LINE("character"{,left,right}){:postcond}

where:

integer	is an integer specifying how many new line operations to perform
"character"	is a character you want repeatedly printed on the new line
left	is an integer specifying the column in which to start printing the character
right	is an integer specifying the column in which to stop printing the character
postcond	is an optional postconditional expression

Comments

Keep the following points in mind when you use the $LINE function:

- The $LINE function without an argument ends the current print line. The $LINE function ignores the current column position and format list values.
- The Report Driver performs an implicit $LINE function at the end of every format group specification. You can use the $NOLINE function to suppress this implicit $LINE operation.
- The $LINE function with an **integer** argument ends the current line and prints **integer** number of blank lines.
- The $LINE function with a single character (**"character"**) argument ends the current line and repeatedly prints the quoted character across the new line from column 1 to the page width. If you add the optional **left** and **right** arguments, $LINE repeatedly prints the quoted characters from the column specified in **left** to the column specified in **right**.

  Again, the $LINE function ignores the current column position and format list values.

- If you append a postconditional expression to the $LINE function, the DASL software executes the $LINE function only if the postconditional expression is true.

Related Sections

$NOLINE
Report Driver Functions

Examples

The following example terminates the current line and prints 4 blank lines.

```
$LINE(4)
```

The following example terminates the current line and prints a line of dashes (-) in the new line.

```
$LINE("-")
```

The following example terminates the current line and prints a line of dashes from column 15 to column 45 in the new line.

```
$LINE("-",15,45)
```

$MAX

The $MAX function prints the maximum value of the specified data name within the current report level.

Format

$MAX(data name){:postcond}

where:

data name	is the data name whose maximum value you want
postcond	is an optional postconditional expression

Comments

Keep the following points in mind when you use the $MAX function:

- You can only use the $MAX function within trailer format group specifications. You cannot use the $MAX function within the T0 format group.
- You cannot use the $MAX function with STRING or NAME type data names.
- If you add a postconditional expression to the $MAX function, the DASL software prints the maximum value only if the postconditional expression is true.

Related Sections

$AVE
$COUNT
$MIN
Report Driver Functions
$STD
$TOTAL

$MED

The $MED function prints the median of the values for the specified data name within the current report level.

Format

$MED(data name){:postcond}

where:

data name is the data name whose median value you want to print

postcond is an optional postconditional expression

Comments

Keep the following points in mind when you use the $MED function:

- You cannot use the $MED function with STRING or NAME type data names.
- You can use the $MED function only within trailer format group specifications. You cannot use the $MED function within the T0 format group level.
- If you add a postconditional expression to the $MED function, the DASL software prints the median value only if the postconditional expression is true.

Related Sections

$AVE
$COUNT
$MAX
$MIN
Report Driver Functions
$STD
$TOTAL

$MIN

The $MIN function prints the minimum value of the specified data name within the current report level.

Format

$MIN(data name){:postcond}

where:

data name is the data name whose minimum value you want

postcond is an optional postconditional expression

Comments

Keep the following points in mind when you use the $MIN function:

- You can only use the $MIN function within trailer format group specifications. You cannot use the $MIN function within the T0 format group level.
- You cannot use the $MIN function with STRING or NAME type data names.
- If you add a postconditional expression to the $MIN function, the DASL software prints the minimum value only if the postconditional expression is true.

Related Sections

$AVE
$COUNT
$MAX
Report Driver Functions
$STD
$TOTAL

$NOLINE

The $NOLINE function suppresses the implicit $LINE operation the DASL software performs at the end of a format group specification. The $NOLINE function permits multiple format groups to print on the same physical report line.

Format

$NOLINE

Related Sections

$LINE
Report Driver Functions

$PAGE

The $PAGE function ends the current print line and starts a new report page.

Format

$PAGE{:postcond}

where:

postcond	is an optional postconditional expression

Comments

Keep the following points in mind when you use the $PAGE function:

- When you use the $PAGE function to perform a new page operation, the DASL software ignores the current column position and format list values.
- If you append a postconditional expression to the $PAGE function, the DASL software executes the $PAGE function only if the postconditional expression is true.

Related Sections

$LINE
Report Driver Functions

$PAGEC

The $PAGEC function prints the number of pages in the current report.

Format

$PAGEC{:postcond}

where:

postcond is an optional postconditional expression

Comments

Keep the following points in mind when you use the $PAGEC function:

- The $PAGEC and $PAGEN functions perform slightly different functions. The $PAGEC function prints the number of physical pages in the report.

  The $PAGEN function prints the current report page number. If you use a $SETPAGE function to reset the page numbers (using variable %PG), the $PAGEN function does not reflect the total number of physical pages in the report.
- The Report Driver variable %PC contains the value of the current physical page count.
- If you append a postconditional expression to a $PAGEC function, the DASL software executes the $PAGEC function only if the postconditional expression is true.

Related Sections

$PAGEN
Report Driver Functions
$SETPAGE

$PAGEN

The $PAGEN function prints the current page number in the report.

Format

$PAGEN{:postcond}

where:

postcond is an optional postconditional expression

Comments

Keep the following points in mind when you use the $PAGEN function:

- The $PAGEN and $PAGEC functions perform slightly different functions. The $PAGEN function prints the current report page number. If you used a $SETPAGE function to reset the page numbers, the $PAGEN function does not reflect the total number of physical pages in the report.

  The $PAGEC function prints the actual count of physical pages created in the report.
- The Report Driver variable %PG contains the value of the running page count.
- If you append a postconditional expression to the $PAGEN function, the DASL software prints the page number only if the postconditional expression is true.

Related Sections

$PAGEC
Report Driver Functions
$SETPAGE

$QUITREP

The $QUITREP function applies conditions to end a logical looping section that began with the $REPEAT Report Driver function.

You must always use the $QUITREP function with a postconditional expression to apply conditions to the continued printing of the $REPEAT section.

The $QUITREP function does not replace a $ENDREP function. Whenever you specify a $QUITREP function, you must also specify an $ENDREP function to return to the start of the $REPEAT section.

Format

$QUITREP{:postcond}

where:

postcond is an optional postconditional expression

Related Sections

$REPEAT and $ENDREP
Report Driver Functions

Example

The following example shows how to use the $QUITREP function to stop printing a line of text after six repetitions and if a flag for a brief report has been set.

```
$REPEAT(TEXTCNT)
 TEXT
 $LINE
 $QUITREP:TEXTCNT=6&(FLAG="BRIEF")
$ENDREP
```

$REPEAT and $ENDREP

The $REPEAT function begins a section of print items containing looping. The DASL software prints the section for each value of the data name you specify in the $REPEAT argument. Repetition continues until the DASL software encounters a null value for the data name.

You use the $ENDREP function to end a logical looping section that began with the $REPEAT function.

Format

$REPEAT(data name{:n}){:postcond}

$REPEAT(local array{:n}){:postcond}

$REPEAT(global reference{:n}){:postcond}

$REPEAT({:n}){:postcond}

where:

data name is the key data name for whose values you want the logical section repeated

local array is a local array using declared variables

global reference is a DSM global reference using any combination of data names, local variables, numbers, or string literals (The last subscript of the global reference must be a data name or a local variable.)

n is a numeric value that specifies the maximum number of times to repeat

postcond is an optional postconditional expression

Comments

Keep the following points in mind when you use the $REPEAT function:

- The data name you specify must define a collection as described in the "Collection Specifications" section of this chapter. The data name must be a key data name.
- If you use the optional numeric qualifier, the DASL software repeats until it either encounters a null value for the specified data name, or repeats the number of times specified by **n**.
- The DASL software forces new data names used in the $REPEAT argument to evaluate at the beginning of the section.
- If you do not specify a data name, local array, or global reference, $REPEAT performs a simple loop. You can terminate the loop by using the $REPEAT({:n}) form of the function, or by using the $QUITREP function.

- If you append a postconditional expression to a $REPEAT function, the DASL software repeats the section only if the postconditional expression is true.
- You must end the logical section with the $ENDREP function. Use the following syntax for the $ENDREP function:

  $ENDREP

Related Sections

$QUITREP
Report Driver Functions

Example

The following example from the SAMPLE report shows the use of $REPEAT and $ENDREP to print the Print Schema Diagram report described in Chapter 4. The code contained within $REP and $ENDREP uses an array to construct a picture of the database tables in the schema diagram.

```
                   Format Group: D_           Data Level
Print Item                                                   Col Format

! Routine to build image in ^DATG
@ D GLOBAL^%BUILD
$REPEAT(^DATG($J,"S","%QSCHEMA",%N))
^DATG($J,"S","%QSCHEMA",%N,"GNDUM1")                         1
! This will test space at bottom of page every 6 lines
$REQUIRE(6):%N+1#6=0
$LINE
$ENDREP
```

The basic format you can use to write an array within $REPEAT and $ENDREP is:

```
$REPEAT(ARRAY(%N))
  ARRAY(%N)
  $REQUIRE(number of lines)
  $LINE
$ENDREP
```

$REQUIRE

The $REQUIRE function forces a page break.

Format

$REQUIRE(lines){:postcond}

where:

lines	is an integer number specifying the number of blank lines that must be left at the bottom of the current page
postcond	is an optional postconditional expression

Comments

Keep the following points in mind when you use the $REQUIRE function:

- The $REQUIRE function tests whether the specified number of blank lines are left on the current page. If the current page has fewer than **lines** number of lines left at the end of the page, the $REQUIRE function forces a page break. Any subsequent output is printed on a new page.
- Use the $REQUIRE function before printing lines that should not be interrupted by a page break. You can specify the necessary number of lines to be left blank, so that the page breaks either before or after the printed lines.
- If you append a postconditional expression to a $REQUIRE function, the DASL software performs the line test only if the postconditional expression is true.

Related Section

Report Driver Functions

$SECTION

You use the $SECTION function to mark the beginning of a logical section of print items, and the $ENDSEC function to end the logical section. You can use the $SECTION function to apply conditions to a section of print items.

Format

$SECTION{:postcond}

where:

postcond is an optional postconditional expression

Comments

Keep the following points in mind when you use the $SECTION and $ENDSEC functions:

- The DASL software evaluates a new data name at the beginning of the section in which it appears, if the data name has not been evaluated anywhere else. You can use the $SECTION function to force evaluation of independent data names.
- You end the logical section with the $ENDSEC function. The syntax for the $ENDSEC function is:

  $ENDSEC
- The DASL software ignores spaces preceding a print item. You can use spaces to visually differentiate items contained between a $SECTION and a $ENDSEC function from other print items.
- If you append a postconditional expression to a $SECTION function, the DASL software executes the $SECTION function only if the postconditional expression is true.

  If the postconditional expression is false, the DASL software does not print the associated logical section. In this case, the DASL software does not evaluate a new data name that appears within the section.

Related Section

Report Driver Functions

Examples

In this example, adapted from the ADBKONE report of the demonstration system, the $SECTION and $ENDSEC functions apply conditions to a logical section or print items to produce a full listing of the items enclosed in that section if the user specifies FULL.

```
Format Group D:                                  Data Level

NAME                                                              5
BTHDAY/PROMPT                                                     48
$LINE
$SECTION:LISTING="FULL"
    STREET                                                        10
    PHHOME/PROMPT                                                 46
    $LINE
    CITY_", ":CITY]""                                             10
    STATE_"  "
    ZIP
    PHWORK/PROMPT                                                 46
    $LINE
    PHOTH1/PROMPT:PHOTH1]""                                       45
    $LINE(2)
$ENDSEC
```

In this example, adapted from the ADBKONE report of the DASL demonstration system, the $SECTION function applies conditions to a logical section of print items to print a birthday if AGE is greater than 35.

```
Format Group: D           Data Level

Print Item                                         Col Format

NAME                                                5
$SECTION:AGE>35
 BTHDAY/PROMPT                                      48
$ENDSEC
$LINE
STREET                                              10
PHHOME/PROMPT                                       46
$LINE
CITY_",":CITY]""                                    10
STATE_""
ZIP
PHWORK/PROMPT                                       46
```

$SETPAGE

The $SETPAGE function reinitializes the running page counter, the variable %PG (sets %PG to 0). The next report page printed has the page number of 1.

Format

$SETPAGE{:postcond}

where:

postcond	is an optional postconditional expression

Comments

Keep the following points in mind when you use the $SETPAGE function:

- The $PAGEN function prints the value of %PG. If you use the $SETPAGE function to reset the page numbers, the $PAGEN function does not reflect the total number of physical pages in the report.
- If you append a postconditional expression to a $SETPAGE function, the DASL software resets the page numbering only if the postconditional expression is true.

Related Section

Report Driver Functions

$STD

The $STD function prints the standard deviation of the values for the specified data name within the current report level.

Format

$STD(data name){:postcond}

where:

data name	is the data name whose standard deviation you want to print
postcond	is an optional postconditional expression

Comments

Keep the following points in mind when you use the $STD function:

- You can only use the $STD function within trailer format group specifications. You cannot use the $STD function within the T0 format group.
- You can only use the $STD function with NUMERIC type data names.
- You cannot use the $STD function with some types of input and output transforms for NUMERIC type data names.
- If you add a postconditional expression to the $STD function, the DASL software prints the standard deviation only if the postconditional expression is true.

Related Sections

$AVE
$COUNT
$MAX
$MIN
Report Driver Functions
$TOTAL

$TIME

The $TIME function prints the time at which the report print image was created from collected data.

Format

$TIME{:postcond}

where:

postcond is an optional postconditional expression

Comments

Keep the following points in mind when you use the $TIME function:

- You can specify the format of the time by providing a time output format specification in the Site Parameters Option of the Development Environment Menu.
- If you append a postconditional expression to a $TIME function, the DASL software prints the time only if the postconditional expression is true.

Related Sections

$DATE
Report Driver Functions
Site Parameters Option, Chapter 5

$TOTAL

The $TOTAL function prints the total of all values for a specified data name within the current report level.

Format

$TOTAL(data name){:postcond}

where:

data name	is the data name whose total you want to print
postcond	is an optional postconditional expression

Comments

Keep the following points in mind when you use the $TOTAL function:

- You can use the $TOTAL function only within trailer format group specifications. You cannot use the $TOTAL function within the T0 format group level.
- You can use the $TOTAL function only with NUMERIC type data names.
- If you add a postconditional expression to the $TOTAL function, the DASL software prints the total only if the postconditional expression is true.

Related Sections

$AVE
$COUNT
$MAX
$MIN
Report Driver Functions
$STD

Report Driver Macros

Macros are a means of incorporating previously defined format group specifications, such as standard header levels, from one report into another report.

Format

[reportname{,format group}]

where:

reportname	is the name of a previously defined DASL report
format group	is a format group specification for a header, trailer, or data level within the report

Comments

Keep the following points in mind when you use Report Driver macros:

- You can use Report Driver macros in header levels, trailer levels, and data levels.
- The default level for a macro substitution is the data level. When you use **reportname** without an optional format group specifier, the DASL software includes all print items in the data level of the specified report (including any macro calls) in the current level of the report you are defining.

  If the report you name does not exist, the DASL software returns an error.
- When you use **reportname** with an optional format group specifier, the DASL software includes all print items in that format group of the specified report (including any macro calls) in the current level of the report you are defining.

  If the report exists but the format group specified does not exist, the DASL software returns an error.
- You can use the Print Report Macro Calls Option to create a report listing all macros used as print items in selected reports.

Related Sections

Format Group Specifications
Print Items
Report Driver Functions

Examples

The following example incorporates all print items for the data level of report EMPLIS into the current level of the report you are defining.

```
[EMPLIS]
```

The following example incorporates all print items from H0 (header 0) of the report STDHDR into the current level of the report you are defining.

```
[STDHDR,H0]
```

Report Driver Variables

DASL variables are system-defined local variables used by the Screen Driver and the Report Driver. Report Driver variables pass information between the DASL software and the application software.

Setting Report Driver Variables

Report Driver variables contain information about report processing. Each Report Driver variable consists of an uppercase name preceded by a percent sign (%). You examine or set these variables at report run time.

Table 3–9 lists the Report Driver variables and their contents. The table also specifies whether or not you can set the variable.

Table 3–9 DASL Report Driver Variables

Variables	Contents	Set
%CNT	Number of records printed	N
%DEV	Name of user's principal device in the Device Dictionary	N
%ECNT	Number of error records encountered in report collection	Y
%FND	A flag used in report collection indicating that the record be included in the report: 0 if report item rejected 1 if report item selected	Y
%IOD	Internal port name for user-selected output device, or name of a VMS sequential file	N
%PARM	Variable used to pass input from a data screen to a report	Y
%PDV	Physical port name of principal device of current process	N
%PG	Running page count (Reset by $SETPAGE.)	N
%PC	Current page count	N
%PL	Number specifying page length in lines	N
%PW	Number specifying page width in columns	N
%RPCOPY	Optional user-specified number of copies to print	Y
%RPDES	Optional report description to be displayed in Report Directory	Y
%RPDEV	User-specified device name in the DASL Device Dictionary for report output	Y

Table 3–9 (Cont.) DASL Report Driver Variables

Variables	Contents	Set
%RPFNC	User-specified report destination (Print, Report Directory, or both)	Y
%RPNAME	Name of report to run	Y
%RPPAGE	Optional report page on which to start printing	Y
%RPQUE	Batch queue for report (Defaults to value in logical name DSM$DASL_BATCH.)	Y
%RPTIME	Optional time and date to begin report data collection in $H format	Y
%SCP	Flag that indicates report is displayed to terminal	N

Comment

The Report Driver variables that begin with the characters %RP are related only to the Report Directory or the Run Report Utility (%DARPDRV).

Related Sections

Define Reports Option
Print Items
Report Directory Option, Chapter 6

Report Driver Comments

You can include comments in the print item list at any level of a report definition.

Format

!{!} text

where:

! is the comment character

text is the comment text

Comments

Keep the following points in mind when you use Report Driver comments:

- You can include documenting comments in print item lists at any level in a report definition. The comment mark is similar to the DIGITAL Command Language (DCL) comment character, the exclamation point (!).
- If you precede a comment with one exclamation point, the DASL software does not include the comment in the routine compiled from the report definition, for example:

  ```
  ! Print Employee Bank + Account No. for Auto Deposit
  ```

- If you precede a comment with two or more exclamation points, the DASL software does include the comment in the routine compiled from the report definition, for example:

  ```
  !! Print Employee Bank + Account No. for Auto Deposit
  ```

Related Section

Print Items

Format Descriptors

You use format descriptors to specify how you want to justify and display a print item in a report.

Explanation

To produce the format you want, enter one or more format descriptors for each print item. If you use more than one format descriptor, separate each format descriptor in the list with a comma.

Table 3–10 lists the format descriptors you can use with report print items.

Table 3–10 Format Descriptors

Descriptor	Use
B	Overstrike the print item to simulate boldface (does not work on laser printers).
C	Center the print item.
CR	Suppress the printing of a leading minus sign, but append the string "CR" if the value of the print item is negative.
L	Left justify the print item.
N*n*	Display the numerically-valued print item rounded to *n* digits.
P	Suppress the printing of a leading minus sign, but surround the print item with parentheses if the print item value is negative.
R	Right justify the print item.
T*n*	Truncate the print item as specified by *n* to leave *n* columns between this print item and the next print item on the line.
W	Display the print item in double-width format. (The DASL software still interprets the column-position value you specify in single-width format.)
+	Add a leading plus sign to the print item.
$	Add a leading dollar sign to the print item.

Related Section

Print Items

Examples

The following example displays the associated print item as a left-justified dollar amount.

```
L,$,N2
```

The following example displays the associated print item centered in double-width characters.

```
C,W
```

Compile Reports Option

You use the Compile Reports Option to convert reports into executable VAX DSM routines. You can use the Compile Reports Option to compile all reports, or only the reports you specify.

Compiling All Reports

To compile all reports, enter Y in response to the "Compile all reports:" prompt that you see when you first choose the option. Then, choose whether you want to compile in your current process or in batch process. If you are compiling all application reports, you can compile by batch when system processing is light.

Compiling Specified Reports

If you want to compile only certain reports, enter N in response to the "Compile all reports:" prompt. You then see a scroll region where you list the reports you want to compile.

Specify the reports you want by entering one or more of the following:

- Report names
- An asterisk (*) to compile all reports
- A partial name followed or preceded by an asterisk (*)
- An at sign (@) followed by a DASL group name from the Group Dictionary to compile all the reports in the group
- A range of report names in the form **name1-name2** or **-name** to compile all reports in the range

If you choose a range in the form **name1-name2**, the DASL software includes all reports that fall within the specified range, but excludes the reports you specify as the limits of the range. If you choose a range in the form **-name**, the DASL software deletes the report -name from the selection.

If you specify a range in the scroll region, you can verify all reports that fall in the range. Enter X in response to the "Continue, eXpand wildcards, Edit, or Quit:" decision prompt that the DASL software displays when you finish entering reports in the scroll region. The DASL software then lists all reports that fall within the range.

To exclude any of the listed reports, enter D in response to the "Continue, Delete, or Quit:" prompt that the DASL software displays after expanding the list. Then, move to the top of the scroll region. Use any of the scroll-region editing keys to change any entry.

Note: The other choices at this prompt are C to continue to the device selection screen where you specify where you want to print the report, and Q to start over again.

Compiling the Reports

When you finish specifying reports, the DASL software asks whether you want to compile the reports through a batch process or in your current process. If you are compiling more than one report, you can compile through a batch process.

Using the ^%DARPCMP Routine to Compile Reports

You can also use the ^%DARPCMP routine to compile reports. The routine prompts you for report names in the same manner as the Compile Reports Option Screen. You can use the ^%DARPCMP routine to compile reports when you are debugging reports. The DASL software displays system errors on the screen when you use this routine.

To display output at your terminal, set the variable DB to 1 before using the ^%DARPCMP routine.

Related Sections

Define Reports Option
Run Reports Option

Run Reports Option

You use the Run Reports Option to execute any compiled DASL report.

Explanation

When you use the Run Reports Option, you specify the name of the report to run. If the report you specify exists, the DASL software displays the report description as confirmation.

The cursor then moves to a scroll region. Because you are not calling the report from an option or data screen, you may need to set variables before running the report. In the scroll region, enter the names of any variables you need to define and the values you want to give them. When you finish, enter R in response to the "Edit, Run, or Quit:" decision prompt.

The DASL software displays the device selection screen that you use to produce the report in your current process or through a batch process. After you select the process to use, specify the report destination. You can print the report on any VMS device, or write the report to a VMS file. If you produce the report through your current process, you can also display the report on your current device.

Related Sections

Calling Reports from Data Screens and Option Screens
Define Reports Option

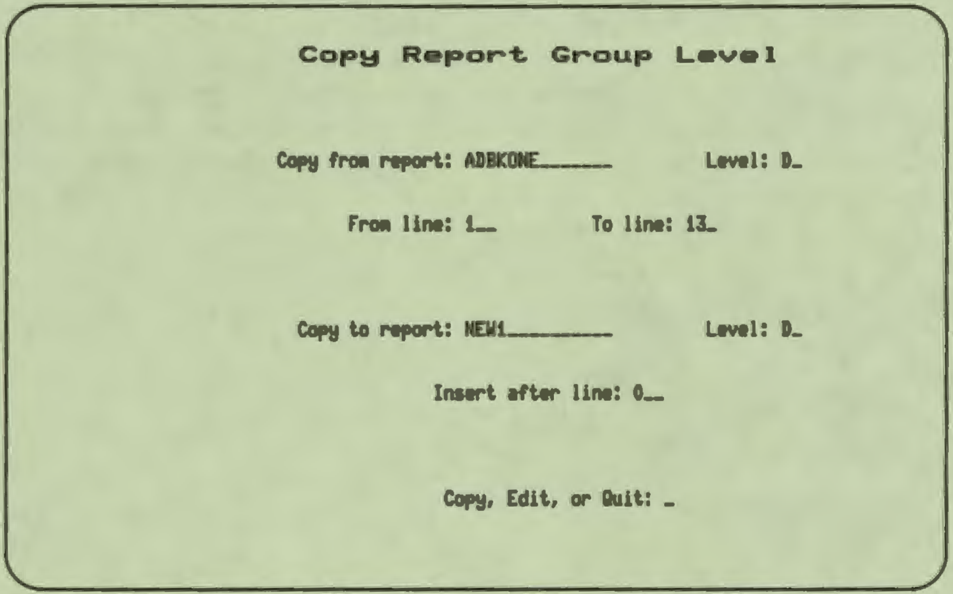
Copy Report Group Level Option

You use the Copy Report Group Level Option to copy print items for a report group level from one report to another report. You can copy all print items or a section of print items, and you can specify where to insert those items in the report.

Explanation

When you select the Copy Report Group Level Option, the DASL software displays the screen shown in Figure 3–5.

Figure 3–5 Copy Report Group Level Screen



```
                    Copy Report Group Level


            Copy from report: ADBKONE_______         Level: D_

                 From line: 1__         To line: 13_


             Copy to report: NEW1__________          Level: D_

                          Insert after line: 0__


                           Copy, Edit, or Quit: _
```

In this screen, the DASL software prompts you for the report name and the report level that you want to copy to another report. You must specify an existing report and report level that have been previously defined through the Define Reports Option.

In the "From line:" and "To line:" fields, the DASL software displays the beginning and ending lines of print items for the format group level. At these prompts, you can:

- Press Return to copy all print items for the report level.
- Copy a section of print items by entering beginning and ending lines for the print items you want to copy.

The DASL software then displays the prompts for the report and report level to which you want to copy a report level. At the "Copy to report:" prompt, enter the name of a report to which you want to copy the report level. At the "Level:" prompt, enter the report level. You can specify either a new report or an existing report in these fields.

If the report is an existing one, the DASL software displays the ending line of print items for that report and report level in the "Insert after line:" field. You can:

- Press Return to insert the copied print items at the end of the print item listing.
- Enter a line number to insert the copied print items within the print item listing.

At the "Copy, Edit, or Quit:" prompt, you can:

- Enter C to copy the report level.
- Enter E to edit the screen.
- Enter Q to quit the screen without making changes.

When you copy a report level to another report, the DASL software inserts lines of comments at the beginning and ending lines of the copied print items to indicate that you have copied those print items.

Related Sections

Define Reports Option
Format Group Specifications
Print Items

Calling Reports from Data Screens and Option Screens

You can call any DASL application report or utility report from an option screen or a data screen. The DASL software uses the Report Driver variable %RPNAME to call the specified report.

Defining a Front-End Data Screen

Before you can specify a report to call in an option screen, you must define a front-end data screen. The front-end data screen contains:

- Screen title, usually the same title as the report title for easy identification
- Report driver variables, such as %RPNAME, %PARM, and user-specific variables

The front-end data screen calls a DASL device selection screen. The DASL software provides the device selection screens %REPDEV and %REPDEVH for application reports, and the device selection screen %UDEV for DASL utility reports.

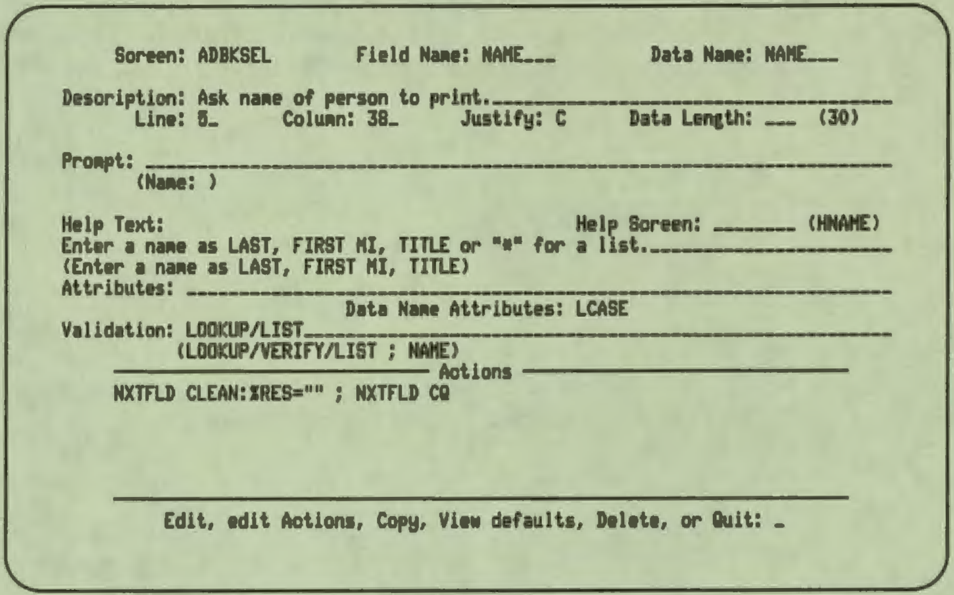
How to Call a Report from a Data Screen

To call an application report or a DASL utility report from a data screen, follow these steps in the data screen:

1. Set the value of %RPNAME to the name of the report to be called with a SET screen or field action.
2. Set values for %PARM, a variable that specifies parameters for the collection phase of the report, if necessary.
3. Set required variables for the Utility Device Selection Screen, if necessary. See Table 3–12 for a list of variables.
4. Call a device selection screen (%REPDEV, %REPDEVH, or %UDEV) with a DO or DONP screen or field action.

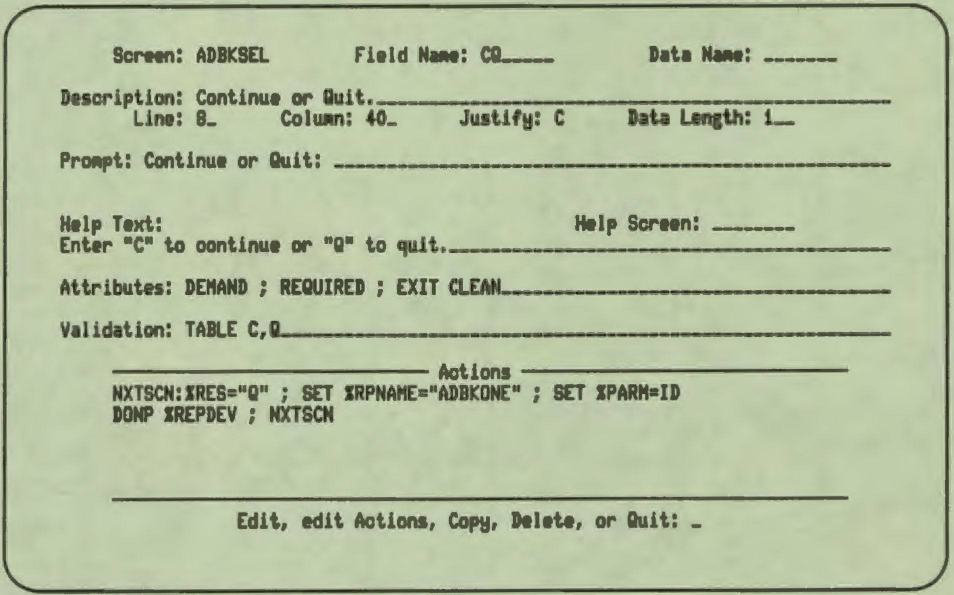
For example, the ADBKSEL screen in the demonstration system calls the Print Selected Addresses Report (ADBKONE). Figure 3–6 and Figure 3–7 show the completed screens for the NAME field and the CQ field in the ADBKSEL screen. The user enters a name in the NAME field and branches to the CQ field. Then, the screen sets %PARM=ID, and calls the ADBKONE report and the %REPDEV device selection screen in the CQ field.

Figure 3–6 NAME Field of the ADKSEL Screen



```
     Screen: ADBKSEL        Field Name: NAME___        Data Name: NAME___

  Description: Ask name of person to print.______________________________________
        Line: 5_      Column: 38_      Justify: C      Data Length: ___  (30)

  Prompt: ______________________________________________________________________
        (Name: )

  Help Text:                                        Help Screen: ________ (HNAME)
  Enter a name as LAST, FIRST MI, TITLE or "*" for a list._______________________
  (Enter a name as LAST, FIRST MI, TITLE)
  Attributes: ___________________________________________________________________
                               Data Name Attributes: LCASE
  Validation: LOOKUP/LIST________________________________________________________
          (LOOKUP/VERIFY/LIST ; NAME)
       ________________________________ Actions _______________________________
       NXTFLD CLEAN:%RES="" ; NXTFLD CQ



       ________________________________________________________________________
            Edit, edit Actions, Copy, View defaults, Delete, or Quit: _
```

Figure 3–7 CQ Field of the ADBKSEL Screen



```
     Screen: ADBKSEL        Field Name: CQ_____        Data Name: _______

  Description: Continue or Quit.______________________________________________
        Line: 8_      Column: 40_      Justify: C      Data Length: 1__

  Prompt: Continue or Quit: __________________________________________________

  Help Text:                                        Help Screen: ________
  Enter "C" to continue or "Q" to quit._______________________________________

  Attributes: DEMAND ; REQUIRED ; EXIT CLEAN__________________________________

  Validation: TABLE C,Q_______________________________________________________

       ________________________________ Actions _______________________________
       NXTSCN:%RES="Q" ; SET %RPNAME="ADBKONE" ; SET %PARM=ID
       DONP %REPDEV ; NXTSCN


       ________________________________________________________________________
                    Edit, edit Actions, Copy, Delete, or Quit: _
```

How to Call an Application Report from an Option Screen

When you define an option through the Define Option Screens Option of the Screen Driver, you can specify a report to call. The second screen at this option, the Option Information Screen, contains these two fields:

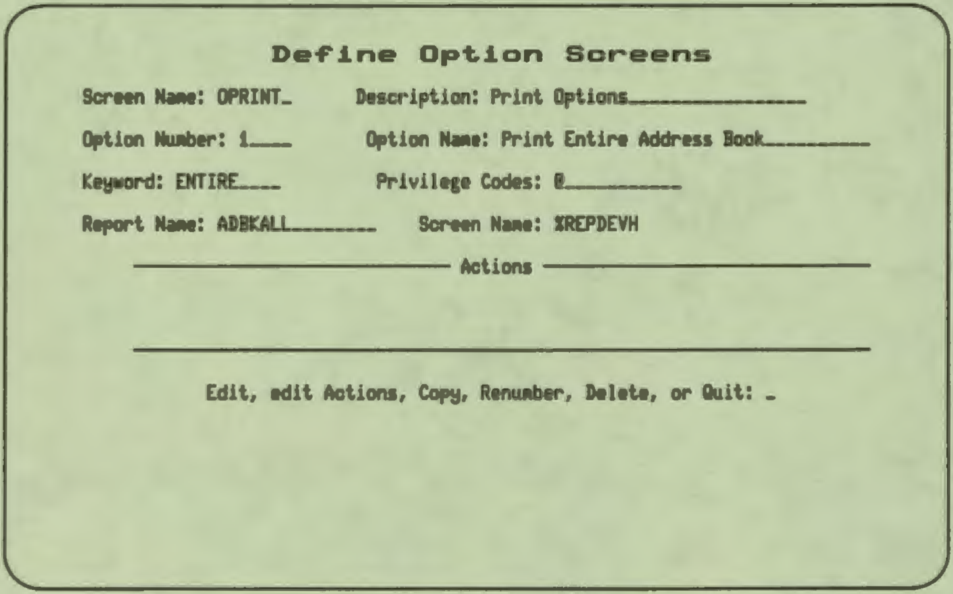
- Report Name
- Screen Name

To call an application report from an option screen, follow these steps:

1. Enter the name of the report at the "Report Name:" prompt. The DASL software sets the value of %RPNAME to that report name.
2. Enter the name of a report device selection screen (%REPDEV or %REPDEVH) at the "Screen Name:" prompt.

Figure 3–8 shows an example of a completed option screen definition for the screen OPRINT. This screen calls the application report ADBKALL and the report device selection screen %REPDEVH.

Figure 3–8 Calling Report ADBKALL from an Option Screen



```
                          Define Option Screens
   Screen Name: OPRINT_      Description: Print Options________________
   Option Number: 1___        Option Name: Print Entire Address Book_________
   Keyword: ENTIRE____         Privilege Codes: @__________
   Report Name: ADBKALL________    Screen Name: %REPDEVH
        ------------------------------- Actions -------------------------------

        ---------------------------------------------------------------------------
              Edit, edit Actions, Copy, Renumber, Delete, or Quit: _
```

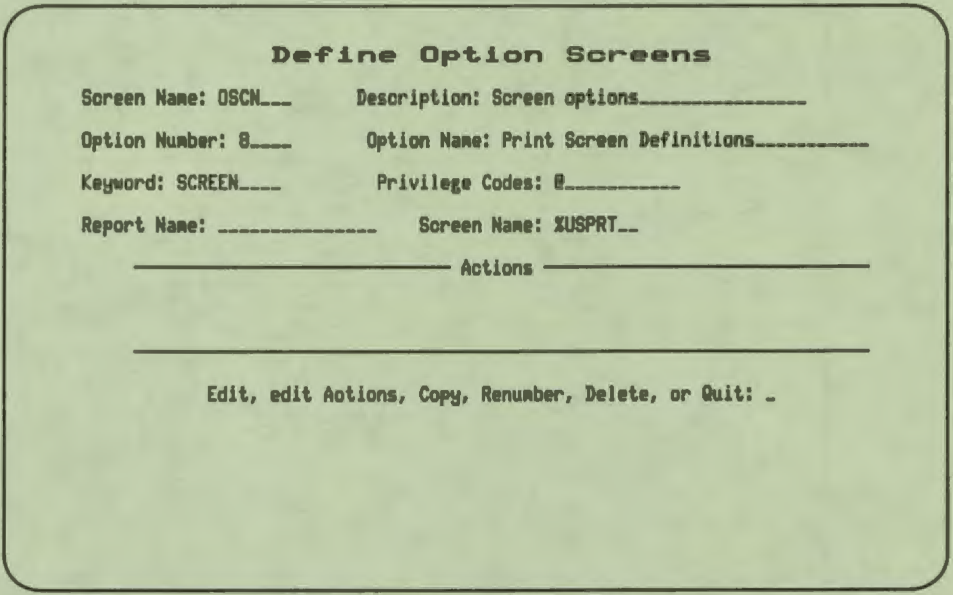
How to Call a DASL Utility Report from an Option Screen

To call a DASL utility report from an option screen, take the following steps:

1. Enter the name of a data screen that calls the report in the "Screen Name:" field. **Do not specify the name of a report in the "Report Name:" field.**
2. Specify the following actions in the data screen that you entered in the "Screen Name:" field:
    - Set the value of %RPNAME to the report name with a SET screen or field action.
    - Call the Utility Device Selection Screen %UDEV with a DO or DONP screen or field action.

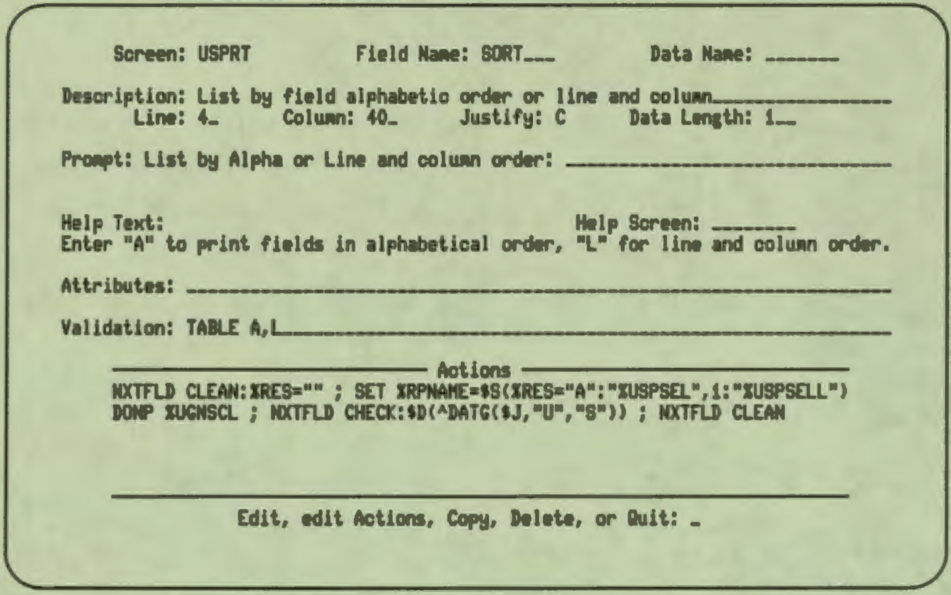
For example, Figure 3–9, Figure 3–10, and Figure 3–11 show a sequence of screen and field definitions that call the Print Screen Definitions utility report. The option screen OSCN calls the data screen USPRT. Then, the USPRT data screen definition sets the value of %RPNAME and calls the Utility Device Selection Screen %UDEV.

Figure 3–9 Option Screen OSCN Calls USPRT



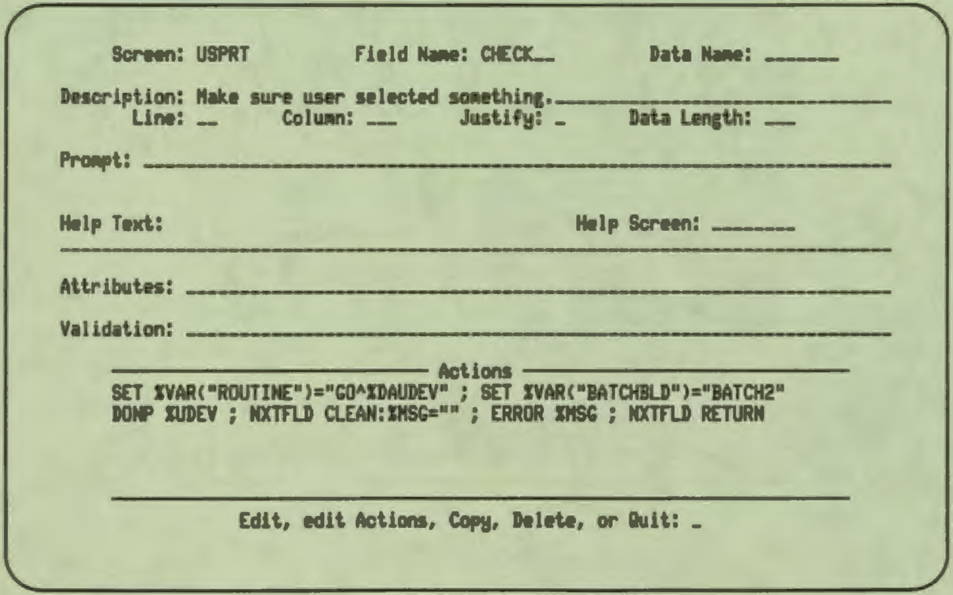
```
                        Define Option Screens

   Screen Name: OSCN___      Description: Screen options______________

   Option Number: 8____       Option Name: Print Screen Definitions__________

   Keyword: SCREEN____         Privilege Codes: @__________

   Report Name: ______________    Screen Name: %USPRT__
        ______________________________ Actions ______________________________

        _____________________________________________________________________

                  Edit, edit Actions, Copy, Renumber, Delete, or Quit: _
```

Figure 3–10 USPRT — Field SORT Defines %RPNAME



```
    Screen: USPRT          Field Name: SORT___          Data Name: _______

Description: List by field alphabetic order or line and column_______________
      Line: 4_       Column: 40_       Justify: C      Data Length: 1__

Prompt: List by Alpha or Line and column order: ____________________________


Help Text:                                        Help Screen: ________
Enter "A" to print fields in alphabetical order, "L" for line and column order.

Attributes: _________________________________________________________________

Validation: TABLE A,L________________________________________________________

    ______________________________ Actions ______________________________
    NXTFLD CLEAN:%RES="" ; SET %RPNAME=$S(%RES="A":"%USPSEL",1:"%USPSELL")
    DONP %UGNSCL ; NXTFLD CHECK:$D(^DATG($J,"U","S")) ; NXTFLD CLEAN


    _____________________________________________________________________
               Edit, edit Actions, Copy, Delete, or Quit: _
```

Figure 3–11 USPRT — Field CHECK Calls %UDEV



```
    Screen: USPRT          Field Name: CHECK__          Data Name: _______

Description: Make sure user selected something.______________________________
      Line: __       Column: ___       Justify: _      Data Length: ___

Prompt: _____________________________________________________________________

Help Text:                                        Help Screen: ________
_____________________________________________________________________________

Attributes: _________________________________________________________________

Validation: _________________________________________________________________

    ______________________________ Actions ______________________________
    SET %VAR("ROUTINE")="GO^%DAUDEV" ; SET %VAR("BATCHBLD")="BATCH2"
    DONP %UDEV ; NXTFLD CLEAN:%MSG="" ; ERROR %MSG ; NXTFLD RETURN


    _____________________________________________________________________
               Edit, edit Actions, Copy, Delete, or Quit: _
```

Using the Report Device Selection Screens

You can use the report device selection screens (%REPDEV and %REPDEVH) when you call application reports.

Both report device selection screens prompt the user for the following information:

- Report function
- Output device
- Copies
- Start time
- Start date

In addition to the preceding information, the screen %REPDEVH displays the following:

- The title of the option screen from which it is called
- The current date and time

Table 3–11 lists the report device selection information and the parameters defined by user input.

Table 3–11 Report Device Selection Information

Parameters	Item	Meaning
%RPFNC	Report function	Function to be performed. Enter D to send report to Report Directory. Enter P to print report.
%RPDEV	Output device	Print device if print requested. If Report Directory is selected and print device is given, %RPFNC is set to "DP."
%RPCOPY	Copies	Number of copies if print is requested.
%RPTIME	Start date and time	Date and time report is to be run.

The report device selection screens follow this process:

1. The report device selection screen calls the report execution routine ^%DARPDRV.
2. The execution routine ^%DARPDRV calls the ^%DARUN routine.
3. The ^%DARUN routine supplies the report name for %RPNAME and processes the report.

Using the Utility Device Selection Screen

You can use the Utility Device Selection Screen %UDEV when you call DASL utility reports.

The Utility Device Selection Screen prompts the user for the following information:

- Run as a batch job?
- Output device

> **Note:** Unlike %REPDEV and %REPDEVH, %UDEV does not direct reports to the Report Directory. %UDEV directs the output of a report to a specified device or a VMS sequential file.

When you use %UDEV, you can define the following variables in your data screen as shown in Table 3–12.

Table 3–12 Variables for %UDEV Screen

Variable	Description
%VAR("TYPE")	Types of DASL entities. Types include: S — Screen R — Report D — Data name T — Template
%VAR("ROUTINE")	Run time routine entry reference for print job process. Defaults to GO^%DAUDEV.
%VAR("IMMED")	Run time routine entry reference for non-print job process. Job specific.
%VAR("BATCHBLD")	Utility-specific section of command procedure. Defaults to BATCH2^%DAUDEV for print job. Job specific for non-print batch processes.
%VAR("VARIABLES")	Job-specific local variables.

The Utility Device Selection Screen can process both print and non-print jobs.

For print jobs, %UDEV follows this process:

1. %UDEV calls the utility driver routine ^%DAUDEV.
2. The utility driver routine ^%DAUDEV calls the ^%DARUN routine.
3. The ^%DARUN routine supplies the report name for %RPNAME and processes the report.

For non-print jobs, the following process occurs:

1. You define the variables %VAR("IMMED") and %VAR("BATCHBLD") in the data screen before calling %UDEV.
2. %UDEV calls the utility driver routine ^%DAUDEV.
3. The utility driver routine ^%DAUDEV executes the user-defined routine that is specified by %VAR("IMMED") or %VAR("BATCHBLD").

Related Sections

Define Data Screens Option, Chapter 2
Define Option Screens Option, Chapter 2
Report Directory Option, Chapter 6
Report Driver Variables

Report Driver Utilities

The Report Driver Menu includes several options that are user utilities. You can use these options to print report definitions, compare reports, search and edit reports, and produce reports of items included in your report definitions. These options include:

- Print Report Definitions
- Compare Reports
- Search/Edit Reports
- Search Multiple Reports
- Print Report Literal Strings
- Print Report MUMPS Routine Calls
- Print Report Macro Calls

Specifying Items in Utility Scroll Regions

When you use some Report Driver Utilities, the DASL software displays a scroll region where you can specify the items that you want to search or include in a report.

You can list the items you want by entering one or more of the following:

- Report names
- An asterisk (*) for all reports
- A partial name followed or preceded by an asterisk (*)
- An at sign (@) followed by the name of a defined DASL group for all reports in the group
- A range of reports in the form **name1-name2** or **-name**

If you choose a range in the form **name1-name2**, the DASL software includes all reports that fall within the range, but excludes the reports you specify for the range limits. If you choose a range in the form **-name**, the DASL software deletes the report **-name** from the selection.

If you specify ranges or use a wildcard with a partial name, you can verify the reports in the range. When the DASL software displays the "Continue, eXpand wild cards, Edit, or Quit:" prompt, enter X. The DASL software then displays all reports that fall within the range.

To exclude any of the listed reports, enter D in response to the "Continue, Delete, or Quit:" decision prompt that the DASL software displays after expanding the list. Then, you move back to the top of the scroll region. Use the scroll-region editing keys to change any of the entries.

Note: The other choices at this prompt are C to continue to the device selection screen where you specify where you want to print the report, and Q to start over again.

Choosing a Destination

The DASL software displays the device selection screen that you use to produce the report in your current process or through a batch process. After you select the process to use, you specify the report destination. You can print the report on any VMS device, or write the report to a VMS file. If you produce the report through your current process, you can also display the report on your current device.

Related Section

Define Reports Option

Print Report Definitions Option

The Print Report Definitions Option creates a report listing the definitions of reports in your application.

Explanation

Example 3–1 shows a sample Report Definition Report.

Example 3–1 Report Definition Report

```
20-Apr-88  2:54 PM              DASL                       Page 1
                  DSM Application Software Library
                      Report Definition: ADBKONE

Description: Print one address
Group(s): DEMO
                   Page Length: 62     Page Width: 80
Version:                                                Created by:
Edited on 30-Mar-88     Edited by:              Compiled on 30-Mar-88

For ID=%PARM

                                Header Level 0

[DEMOHEAD]
"Address"                                                       40  C
                                  Data Level

NAME                                                              5
AGE/PROMPT                                                        48
$LINE
STREET                                                            10
PHHOME/PROMPT                                                     46
$LINE
CITY_", ":CITY]""                                                 10
STATE_"  "
ZIP
PHWORK/PROMPT                                                     46
$LINE
PHOTH1/PROMPT:PHOTH1]""                                           45
$LINE(2)
```

Related Sections

Define Reports Option
Report Driver Utilities

Compare Reports Option

The Compare Reports Option lists any differences in the definitions of any two specified reports.

Example 3–2 shows a sample Report Comparison Report.

Example 3–2 Report Comparison Report

```
20-Apr-88  2:54 PM             DASL                     Page 1
                    DSM Application Software Library
                          Report Comparison

Compare Report: ADBKALL                With Report: ADBKONE

REPORT SPECIFICATIONS                  REPORT SPECIFICATIONS

 Description:                           Description:
  Address Book - Print All               Print one address
 Created By:                            Created By:
 Compile Date: 20-Apr-88                Compile Date: 30-Mar-88
 For 1:                                 For 1:
  ID                                     ID=%PARM
 Sort by:                               Sort by:
  NAME

HEADER LEVEL 0                         HEADER LEVEL 0

Line: 2                                Line: 2
 Item:                                  Item:
  "Entire Address Book"                  "Address"

Line: 3                                Not present

DATA LEVEL                             DATA LEVEL

Line: 1                                Line: 1
 Item:                                  Item:
  [ADBKONE]                              NAME
 Column:                                Column: 5

Not present                            Line: 2
```

As Example 3–2 shows, the Report Comparison Report lists only the differences between the two specified reports.

Related Section

Report Driver Utilities

Search/Edit Reports Option

You use the Search/Edit Reports Option to search a specified report definition for a specified string, edit any occurrences of the string, and then recompile the report.

Explanation

The Search/Edit Reports Option is a quick way to find variables and make changes in a report. When you choose the Search/Edit Reports Option, you specify the following:

- Report name
- Search string

You can specify a string from any of the following:

- Report identification
- Collection specification
- Sort list
- Format groups and print items

The DASL software displays all occurrences of the search string in the specified report and a "Save, Edit, or Quit:" decision prompt. If you enter E to edit, you can enter the scroll region and edit any of the displayed items. The DASL software evaluates any changes you make and issues error messages if you make a syntactically invalid change or specify an undefined data name.

After you finish, you can save your edits as part of the report definition and recompile the report.

Related Section

Define Reports Option

Search Multiple Reports Option

The Search Multiple Reports Option lists all occurrences of a specified string of characters in the output specifications of selection reports.

Example 3–3 shows a sample of a Multiple Search Report for the literal strings BTHDAY and PHOTH1.

Example 3–3 Search Multiple Reports Report

```
20-Apr-88  2:56 PM             DASL                           Page 1
                      DSM Application Software Library
                           Search Multiple Reports

Searching for: "BTHDAY"
               "PHOTH1"

Line    Item
_____________________________________________________________________

                            Report: ADBKONE
                             Level: Data
  12    PHOTH1/PROMPT:PHOTH1]""
_____________________________________________________________________
```

As Example 3–3 shows, the Search Multiple Reports Report lists all occurrences of the selected strings in the output specifications of the reports you list.

When you choose this option, you must specify search strings that are at least two characters long and are print items.

Related Sections

Define Reports Option
Report Driver Utilities

Print Report Literal Strings Option

The Print Report Literal Strings Option lists or specified literal strings used as print items in selected reports.

Example 3–4 shows a sample Report Literal Strings Report for the string literal "address" in the ADBKALL and ADBKONE reports.

Example 3–4 Report Literal Strings Report

```
19-May-88  4:13 PM              DASL                      Page 1
                    DSM Application Software Library
                         Report Literal Strings
Searching for: "address"

Line   Literal
__________________________________________________________________

                          Report: ADBKALL
                          Level: Header 0

2   "Entire Address Book"

=============================================================

                          Report: ADBKONE
                          Level: Header 0

2   "Address"

=============================================================
```

As Example 3–4 shows, the Report Literal Strings Report lists in alphabetical order all the reports you specify. Under each report, the Report Literal Strings Report includes all occurrences of the specified string literals. The report includes the level and line at which each string literal occurs.

In this option, you first specify the literal strings for which you want to search. You can search for all literals or selected literals. The literal strings you choose must be at least two characters long and must be print items.

Then, the DASL software displays a scroll region to let you specify which reports to search for the strings.

Related Sections

Define Reports Option
Print Items

Print Report MUMPS Routine Calls Option

The Print Report MUMPS Routine Calls Option creates a report listing all VAX DSM routine calls used in selected reports.

Example 3–5 shows a sample Report Routine Calls Report.

Example 3–5 Report Routine Calls Report

```
1-Oct-88    3:12 PM                DASL                        Page 1
                        DSM Application Software Library
                            Report Routine Calls

Report             Level        Line    Routine
______________________________________________________________________

SECSYSM            Data           7     D TMIX^%DAUDTTM
```

The Report Routine Calls Report lists in alphabetical order all the reports you specify. For each report, the Report Routine Calls Report includes all routines occurring in the report. The report lists the following for each routine:

- Level
- Line
- Entry reference

Related Sections

Define Reports Option
Print Items

Print Report Macro Calls Option

The Print Report Macro Calls Option list all macros used as print items in selected reports.

Example 3–6 shows a sample Report Macro Calls Report.

Example 3–6 Report Macro Calls Report

```
1-Oct-88   3:29 PM                DASL                           Page 1
                       DSM Application Software Library
                               Report Macro Calls

Report             Level        Line         Macro
-----------------------------------------------------------------------

SECOPT             Header 0      1           [SYSHEAD]

SECSYSM            Header 0      1           [APPLHEAD]

UDPRTB             Header 0      1           [SYSHEAD]

UTPRT              Data          2           [TEMPLATE]
                   Header 0      1           [SYSHEAD]
```

As Example 3–6 shows, the Report Macro Calls Report lists in alphabetical order all the reports you specify. For each report, the Report Macro Calls Report includes all macros occurring in the report. The report lists the following for each macro:

- Level
- Line
- Macro name

Related Sections

Define Reports Option
Print Items
Report Driver Utilities

Chapter 4
Query Driver

This chapter describes the following Query Driver Menu options, utilities, and elements:

- Define Query Option
- SELECT Statement

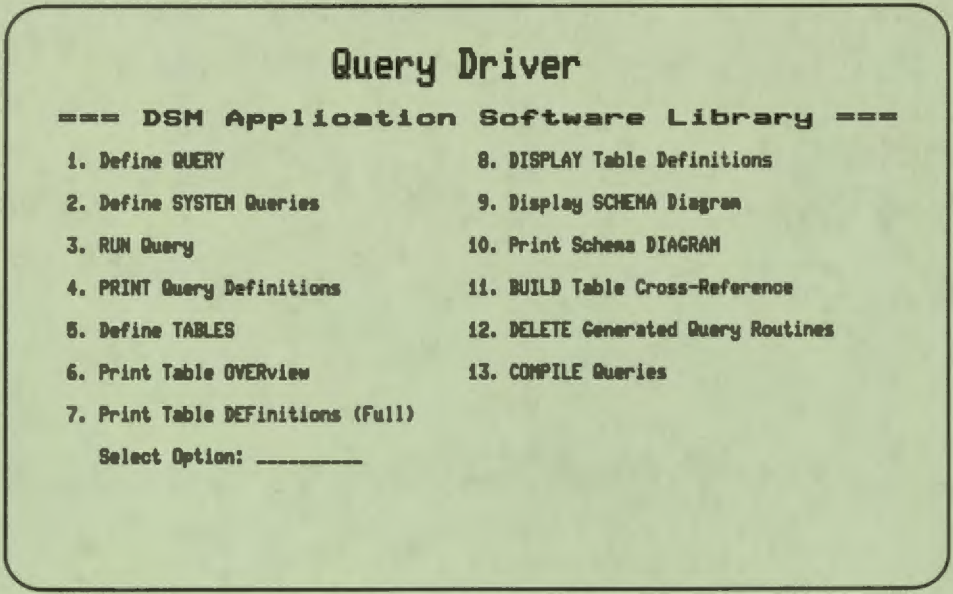
    SELECT Clause
    FROM Clause
    WHERE Clause
    GROUP BY Clause
    HAVING Clause

- Value Expressions
- Query Driver Functions
- Search Conditions
- Define System Queries Option
- Define Tables Option
- Query Driver Utilities

    Run Query Option
    Print Query Definitions Option
    Print Table Overview Option
    Print Table Definitions (Full) Option
    Display Table Definitions Option
    Display Schema Diagram Option
    Print Schema Diagram Option
    Build Table Cross-Reference Option
    Delete Generated Query Routines Option
    Compile Queries Option

Figure 4–1 shows the Query Driver Menu and its options.

Figure 4–1 Query Driver Menu Screen



```
                          Query Driver
  === DSM Application Software Library ===
   1. Define QUERY                         8. DISPLAY Table Definitions
   2. Define SYSTEM Queries                9. Display SCHEMA Diagram
   3. RUN Query                           10. Print Schema DIAGRAM
   4. PRINT Query Definitions             11. BUILD Table Cross-Reference
   5. Define TABLES                       12. DELETE Generated Query Routines
   6. Print Table OVERview                13. COMPILE Queries
   7. Print Table DEFinitions (Full)
      Select Option: _________
```

This chapter describes each option in detail. See the *DASL Programmer's Guide* for examples of how to define tables and queries for your application.

Define Query Option

You define a *query* to make inquiries to the database. To define a query, you use the Define Query Option to create a *query specification*.

Explanation

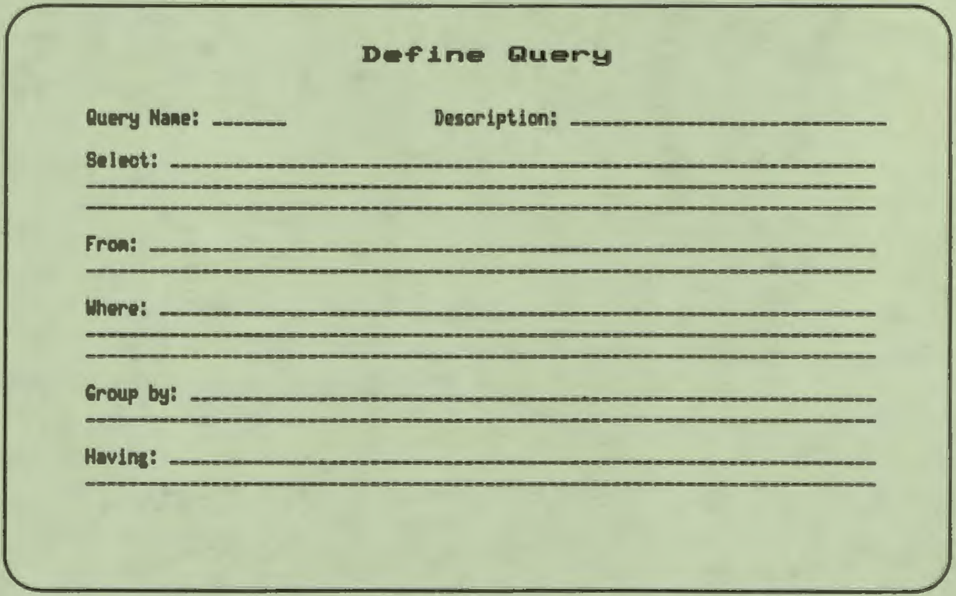
The query specification creates a *result table* derived from some combination of tables in the database. The result table is a table that consists of the intersection of rows and columns that you specify within the query specification.

The query specifications that you define conform to the syntax of the ANSI Standard Database Language SQL (Structured Query Language). For more information about the syntax of the query specification, see the "SELECT Statement" section of this chapter.

Using the Define Query Screen

Figure 4–2 shows the screen that appears when you choose the Define Query Option.

Figure 4–2 Define Query Screen



```
                          Define Query

Query Name: ______           Description: ________________________
Select: __________________________________________________________
__________________________________________________________________
__________________________________________________________________
From: ____________________________________________________________
__________________________________________________________________
Where: ___________________________________________________________
__________________________________________________________________
__________________________________________________________________
Group by: ________________________________________________________
__________________________________________________________________
Having: __________________________________________________________
__________________________________________________________________
```

Table 4–1 shows the information you supply to define a query.

> **Note:** Before you enter information in the Define Query screen, you must use the Application Parameters Option in the Development Environment to specify a prefix for all your queries.

Table 4–1 Information for Defining Queries

Item	Description	Example
Owner	The user name of the creator of an existing query. The owner appears on the screen only if the query already exists.	CLARK, JANE
Query Name	The name of the query, a standard DASL name made up of one alphabetic character followed by up to six alphanumeric characters.	NOVELS
Description	Up to 35 characters describing the query. The DASL software displays the description in lookup lists.	Listing of authors and titles
Select	A required select list of column names or value expressions separated by commas. The select list specifies what data items to search for.	NAME,TITLE
From	A required table reference of one or more tables separated by commas. The table reference specifies which tables to search.	AUTHORS,BOOKS
Where	An optional search condition consisting of predicates (conditions) joined by AND or OR statements.	AGE>60 AND SEX="F"
Group by	An optional column name that specifies how to organize columns in the result table.	NAME
Having	An optional search condition to eliminate groups, consisting of predicates (conditions) joined by AND or OR statements.	NAME LIKE "Jon%"

If the query is an existing one, the DASL software fills in the query specifications and displays the following menu prompt:

```
Run, Utilities, Edit, or Quit:
```

The following sections describe your choices at this prompt.

Running the Query

To run the query, enter R. The DASL software then displays a device selection screen whose contents depend upon the area where you are currently working. The DASL software displays one device selection screen in the development area and a different screen in the application area.

In the development area, the DASL software first prompts you to specify whether to run the query through a batch process or through your current process. If you choose to use your current process, you can display the query result on your terminal, print it on any VMS device, or write the query result to a VMS file.

In the application area, the DASL software prompts you for the following information:

- Report Directory
- Device name
- Number of copies
- Start date
- Time

Enter Y to send your query result to the Report Directory. (The default at this prompt is N.)

Enter the name of a device defined in the Device Dictionary as follows:

- If you do not send the query result to the Report Directory, you can display it on your terminal or print it on any VMS device that is defined in the Device Dictionary.
- If you send the query result to the Report Directory, you can specify only a spooled device at this field. With a spooled device, the DASL software sends reports directed to that device into a spool file for temporary storage until the device is free to print the report. The contents of the spool file are queued to the device.

Enter the number of copies you want.

Enter the date and time that you want printing to begin.

At the "Continue, Edit, or Quit:" prompt, enter C to produce the report, E to edit the device selection screen, or Q to quit the screen without producing a query result.

Using the Utilities

Enter U to delete, copy, print, or add documenting comments to the query. The DASL software then displays the following utilities decision prompt:

```
Edit Comments, Print, Copy, or Delete:
```

- To add comments to document the query, enter E. You then see a comments screen that allows you to enter text in a scroll region.
- To delete the query, enter D and verify your choice where prompted.

- To make a copy of the query under a different name, take the following steps:

  1. Enter C and specify the new query name at the "Copy to:" prompt. The DASL software then displays a "User to copy to:" prompt.
  2. To make a copy for yourself, press Return. The default at the "User to copy to:" prompt is the current user. If you copy the query to yourself, you can later edit the new query name to change any of the copied values.
  3. To make a copy for another user, enter the user name of the other user. You can also enter an asterisk (*) to see a lookup list of all users, including the system manager. The DASL software checks to determine if the user you specify has security privileges to use the tables in the query. If the user does not have appropriate privileges, the DASL software does not copy the query and returns an error message.

- To print a copy of the query definition, enter P. The DASL software displays a device selection screen that allows you to produce the report through a batch process or through your current process. If you choose to use your current process, you can display the report on your terminal, print it on any VMS device, or write the report to a VMS file.

Editing the Query Definition

Enter E to edit the query definition.

Starting Over

Enter Q to start over. The DASL software moves the cursor back to the "Query Name:" prompt. You can also use the exit keys at any point to return to the previous screen or the DASL Main Menu.

Help Screens for the Define Query Option

The Define Query Option offers comprehensive help screens. When you press Help at the "Select:" prompt, the DASL software displays a Help Menu where you can choose to view a schema diagram of the database or display table definitions.

When you select the Display Schema Diagram Option, the DASL software displays a schema diagram of all the tables in the database. Figure 4–10 shows a sample schema diagram.

When you select the Display Table Definitions Option, the DASL software displays the Display Table Definitions screen. You can then specify the tables for which you want to display definitions.

Recommended Practices for Designing Queries

Follow these recommended practices when you design your queries:

- Use the Application Parameters Option in the Development Environment to specify a prefix for all your queries before you define a query.
- Know your database structure and plan your tables accordingly.
- Before you run the entire query, run the query to your terminal screen without using the GROUP BY or HAVING clause. Confirm that the results are what you want. Then add the GROUP BY and HAVING clauses.

  A query that contains a GROUP BY or HAVING clause requires more time to display the first page to the screen than a query without these clauses. With a GROUP BY or HAVING clause, the DASL software must collect data for the entire query before determining what data to display on the first page of the query. If the database is large, the collection process takes some time.
- Run queries in batch mode when possible.
- You can press Exit or Ctrl/C to interrupt a query that is currently running.

Related Sections

FROM Clause
GROUP BY Clause
HAVING Clause
Query Driver Functions
Search Conditions
SELECT Clause
SELECT Statement
Value Expressions
WHERE Clause

SELECT Statement

The SELECT statement defines a query specification. It consists of a SELECT clause and a FROM clause and, optionally, a WHERE, GROUP BY, or HAVING clause. The SELECT statements that you define in the DASL Query Driver conform to the syntax of the ANSI Standard Database Language SQL (Structured Query Language).

Explanation

You use the clauses of the SELECT statement to create a result table from existing base tables in the database.

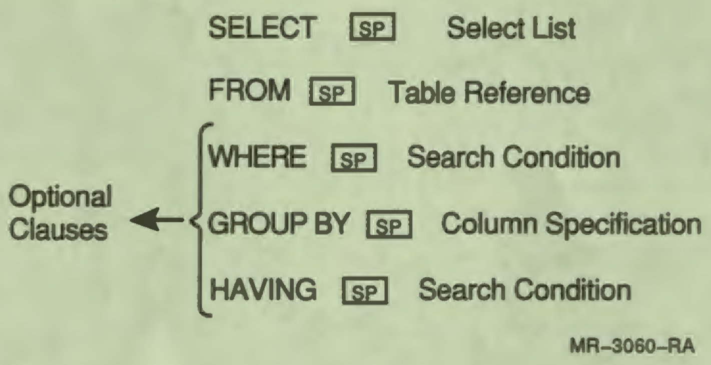
The SELECT statement clauses ensure that the result table:

- Contains only the data you want to work with
- Puts columns and rows in the order you want

Components of the SELECT Statement

Figure 4–3 shows the syntax of the SELECT statement.

Figure 4–3 Syntax of the SELECT Statement


SELECT SP Select List
FROM SP Table Reference
WHERE SP Search Condition
GROUP BY SP Column Specification
HAVING SP Search Condition
Optional Clauses
MR–3060–RA

The arguments to the clauses of a SELECT statement can contain many elements, including:

- Value expressions
- Query Driver functions
- Table references
- Column specifications
- Search conditions (predicates)

This section contains brief descriptions of the main elements you need to know to complete a SELECT statement. Each element is described in greater detail in other sections of this chapter.

Value Expressions

A *value expression* is a symbol or a string of symbols used to represent or calculate a single value. The argument to the SELECT command, the *select list*, is a series of value expressions separated by commas.

Value expressions can include any of the following:

- Literal strings enclosed in quotation marks
- Query Driver functions
- Column specifications
- Numeric literals
- Value expressions joined by arithmetic operators

The following examples illustrate various value expressions:

```
NAME,AGE            ;Column names

"Name:",NAME        ;Literal string, Column name

SUM(AGE)            ;Query Driver Function

AVG(AGE)+10         ;Query Driver Function, operator, numeric literal
```

Query Driver Functions

Query Driver *functions* are statistical expressions that calculate a single value for a collection of rows.

The Query Driver functions that you can use are COUNT, SUM, AVG, MAX, and MIN.

With most functions, you must specify a column name, for example:

```
SUM(TAX)

MAX(EMPLOYEE_SALARY)
```

Table References

The argument to the FROM clause is a *table reference.*

A table reference takes the following syntax:

table_name{ SPcorrelation_name}

where:

table_name is the name of a table defined in the Define Tables Option

correlation_name is a symbolic name for the table

For example, you can specify the following table reference for the PATIENT table:

```
From:   PATIENT P
```

In this example, PATIENT is the table name and P is the correlation name. You can use a correlation name when specifying a column within a table. You must use the correlation name when specifying a column name that is not unique. For example, NAME is a common column name shared by many tables. Use the following correlation name to specify the NAME column in the PATIENT table:

```
P.NAME
```

Column Specifications

Column specifications are the arguments to the GROUP BY clause and are also value expressions.

Column specifications take the following syntax:

{table_reference.}column_name

where:

table_reference is an optional table name or correlation name

column_name is a column name defined in the Define Tables Option

The following examples illustrate various column specifications for the same NAME column:

```
NAME
PATIENT.NAME
P.NAME
```

Predicates

The arguments to the WHERE and HAVING clauses are *search conditions*. Search conditions contain one or more *predicates* joined by any number of ANDs or ORs. Predicates are conditional expressions that evaluate to true or false.

Each search condition takes the following format:

predicate SP{AND SPpredicate}...

predicate SP{OR SPpredicate}...

where:

predicate is a conditional expression that evaluates to true or false

AND is a logical operator that requires that both predicates joined by AND have a truth value of true

OR is a logical operator that requires that either predicate joined by OR has a truth value of true

The many types of predicates include:

- Comparison predicates
- BETWEEN predicates
- IN predicates
- LIKE predicates
- NULL predicates

The following examples illustrate various search conditions:

```
Where:   AGE>10

Where:   AGE BETWEEN 12 AND 24

Where:   NAME LIKE "Sam%"

Where:   AGE NOT NULL
```

Related Sections

FROM Clause
GROUP BY Clause
HAVING Clause
Query Driver Functions
Search Conditions
SELECT Clause
Value Expressions
WHERE Clause

SELECT Clause

The SELECT clause identifies the columns that appear in the result table.

Format

SELECT SP{qualifier} SPselect list

SELECT SP*

where:

qualifier can be one of the following:

ALL to include all rows (ALL is the default.)

DISTINCT to eliminate duplicate rows

select list is a list of value expressions separated by commas

***** selects all columns of a table

Explanation

The SELECT clause identifies columns from the tables that you specify in the FROM clause. The columns that you specify in the SELECT clause must be defined for the tables that you specify in the FROM clause. If you enter an undefined column specification in the SELECT clause, the Query Driver displays an error message.

Comments

Keep the following points in mind when you use the SELECT clause:

- The ALL qualifier is the default qualifier. ALL specifies that duplicate rows should not be eliminated from the result table.
- Use the DISTINCT qualifier to eliminate duplicate rows from the result table.
- Specify a select list as the argument to a SELECT clause. The select list consists of a list of value expressions separated by commas.
- The result table displays values in the order in which you specify value expressions as the argument to the SELECT clause. For example, if you specify SELECT NAME, AGE, SEX, the result table displays the columns in the order of NAME, AGE, SEX.
- If you want to include all columns and do not care in what order they occur, you can specify an asterisk (*) rather than list column names. The columns appear in the order in which they were defined in the table definition. You can also specify an asterisk with a table reference to include all columns in that table, for example:

  ```
  AUTHORS.*
  ```

You can specify a single asterisk in the SELECT clause to select all columns from a single table, for example:

```
Select:   *

From:    NAMES
```

To select all columns from more than one table, you must use a table reference with the asterisk in the SELECT clause, for example:

```
Select:    NAMES.*,PHONES.*

From:    NAMES,PHONES
```

> **Note: In the SELECT statement, the asterisk (*) character indicates that you include all columns of a table, and does not call a lookup list. Elsewhere in the DASL software, the asterisk (*) character does call a lookup list.**

- If you specify an asterisk (*) in the SELECT clause, you cannot use the ALL or DISTINCT qualifiers.
- If you specify an asterisk (*) in the SELECT clause, the combined data length of the column names selected must be less than 255 characters. In addition, the combined data length of all characters within the columns selected must be less than 255 characters. VAX DSM limits the number of characters in a string to less than 255 characters.
- You must specify a GROUP BY clause to order the values within the columns you select. For example, to order the result table alphabetically by name, you must specify GROUP BY NAME.

Related Sections

FROM Clause
GROUP BY Clause
HAVING Clause
Query Driver Functions
SELECT Statement
Value Expressions
WHERE Clause

Examples

The following examples are all valid SELECT clauses:

```
Select:   *

Select:   DISTINCT NAME,TITLE

Select:   "Name:",P.NAME

Select:   SEX,AVG(AGE)

Select:   EMP.NAME,CURRENT_SALARY*1.05
```

FROM Clause

The FROM clause identifies the tables that the Query Driver uses to generate the result table.

Format

FROM table_reference{,table_reference} . . .

(in which **table_reference** consists of a **table_name** and, optionally, a **correlation_name**)

where:

table_name	is the name of a table defined in the Define Tables Option
correlation_name	is a symbolic name for the table

Explanation

The FROM clause is a required part of the query specification. The tables that you identify in the FROM clause must include the columns that you specify in the SELECT clause. In other words, you cannot specify any columns in the SELECT clause that do not belong to the tables in the FROM clause.

Comments

Keep the following points in mind when you use the FROM clause:

- When you specify a table in the FROM clause, you can also assign a correlation name to the table. The correlation name is a symbolic name that you can use in place of the full table name in the query specification. For example, you can assign the correlation name P to the table PATIENT with the following clause:

  ```
  From:    PATIENT P
  ```

  Then you can use the correlation name, followed by a period, in conjunction with a column name to indicate a column in a specific table, for example:

  ```
  Select:    P.NAME
  ```

- If you name more than one table in the FROM clause, the Query Driver joins them to create an intermediate result table. This intermediate result table consists of every possible combination of all the rows and columns of each table.

Related Sections

GROUP BY Clause
HAVING Clause
SELECT Clause
SELECT Statement
WHERE Clause

Examples

The following example shows how to define a correlation name in the FROM clause, and how to use the correlation name in the SELECT clause.

```
Select:   A.NAME,B.TITLE

From:   AUTHORS A,BOOKS B
```

The following example shows an incorrect query specification. The table SCORES in the FROM clause includes the columns NAME, ID, and SCORE. The column TEAM in the SELECT clause is not included in the SCORES table.

```
Select:   NAME,SCORE,TEAM

From:   SCORES
```

The DASL software displays an error message if a column name in the SELECT clause is not included in the tables specified in the FROM clause, for example:

```
TEAM is not in any of the tables.
```

WHERE Clause

The argument to the WHERE clause is a search condition that applies to the preceding FROM clause. A search condition contains one or more predicates joined by any number of ANDs or ORs. When you use the AND and OR constructs within the WHERE argument, you can create a complicated search condition.

Format

WHERE predicate SP{AND SPpredicate} . . .

WHERE predicate SP{OR SPpredicate} . . .

where:

predicate is a conditional expression that evaluates to true or false

AND is a logical operator that requires that both predicates joined by AND have a truth value of true

OR is a logical operator that requires that either predicate joined by OR has a truth value of true

Explanation

The WHERE clause follows the FROM clause in a query specification. You use the WHERE clause to retrieve data from only those rows in a table that satisfy certain criteria. In other words, use the WHERE clause to eliminate rows from the result table. The argument of the WHERE clause is a condition to search for in the tables specified in the FROM clause.

The search condition consists of one or more predicates. Multiple predicates are joined by ANDs or ORs. The types of predicates include the following:

- Comparison predicates use comparison operators to compare two value expressions. The operators you can use are <, >, =, <=, =>, and <>.
- BETWEEN predicates compare a range of values.
- IN predicates specify a quantified comparison.
- LIKE predicates specify a pattern-match comparison.
- NULL predicates test for a null value.

You can precede the BETWEEN, IN, LIKE, and NULL predicates with a NOT operator. The NOT operator requires that the predicate evaluate to false.

Comments

Keep the following points in mind when you use the WHERE clause:

- The Query Driver evaluates conditions as being either true or false.
  - — If a row in the table being searched satisfies the condition, the condition is true and that row is included in the result table.
  - — If a row in the table being searched does not satisfy the condition, then the condition is false and that row is not included in the result table.
- In the predicate, the first value expression is usually a column name and the other value expression is often a literal, for example:

  ```
  Where:   AGE>25
  ```
- If you have two tables in which a column in one table contains a pointer to a column in the other table, you can use the WHERE clause to establish the pointer relationship between the two tables. See Figure 4–4.
- You can use parentheses to group comparisons within predicates, for example:

  ```
  Where: (AGE=>60 AND SALARY>50000) OR (AGE<=30 AND SALARY>50000)
  ```
- You cannot use Query Driver functions in the predicates of a WHERE clause.

Related Sections

FROM Clause
GROUP BY Clause
HAVING Clause
Query Driver Functions
Search Conditions
SELECT Clause
SELECT Statement
Value Expressions

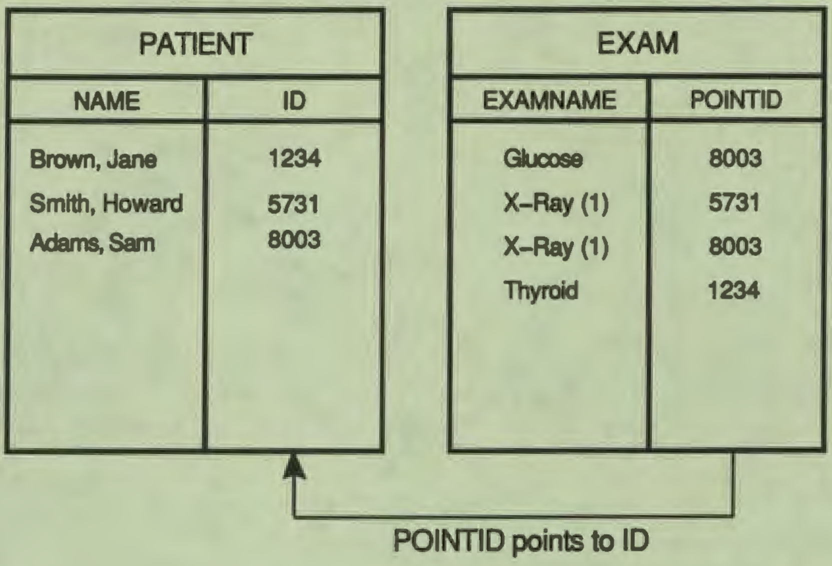
Examples

The following examples are valid uses of the WHERE clause:

```
Where:   SSN="013-78-8976"

Where:   (AGE<=21 AND SEX="F") OR (AGE<=5 AND SEX="M")

Where:   SALARY BETWEEN 20000 AND 30000

Where:   NAME LIKE "Bro%"

Where:   STATE IN ("MA","VT","CT")

Where:   PATIENT.DOCTOR = DOCTOR.DOCTOR
```

Consider a database schema that consists of two tables, the PATIENT table and the EXAM table. Figure 4–4 shows the schema for these two tables.

Figure 4–4 Schema Showing Pointers in WHERE Clause



PATIENT

NAME	ID
Brown, Jane	1234
Smith, Howard	5731
Adams, Sam	8003

EXAM

EXAMNAME	POINTID
Glucose	8003
X-Ray (1)	5731
X-Ray (1)	8003
Thyroid	1234

MR–3056–RA

The PATIENT table contains a NAME column and an ID column. The ID column contains a unique ID for each patient in the NAME column.

The EXAM table contains an EXAMNAME column and a POINTID column. The POINTID column points to the ID column in the PATIENT table.

To specify a query that displays the names of patients (NAME column, PATIENT table) and the exams that each patient has taken (EXAMNAME column, EXAM table), you can use the following query specification:

```
Select:   NAME,EXAMNAME

From:    PATIENT,EXAM

Where:   POINTID=ID
```

GROUP BY Clause

The GROUP BY clause specifies how to organize columns in the result table. Whenever you specify a column name in addition to Query Driver functions in the SELECT statement, you must use the GROUP BY clause.

Format

GROUP BY SPcolumn_specification . . .

where:

column_specification is one or more column names separated by commas

Explanation

The GROUP BY clause divides tables into groups. GROUP BY orders rows in a result table by the values in a column. For example, if you direct the Query Driver to group by name, the result table displays rows alphabetically by name.

Comments

Keep the following points in mind when you use the GROUP BY clause:

- If the argument to the SELECT clause, the select list, contains a Query Driver function and a column name, the Query Driver automatically lists all column names, with the exception of the column name in the function, as the argument to the GROUP BY clause. For example:

  ```
  Select:    SEX,AVG(AGE)

  From:    EMPLOYEES

  Group by:    SEX
  ```

- If you do not use a GROUP BY clause, the argument to the SELECT clause, the select list, must be either:

  — A list of functions only

  — A list containing no functions

- You can include columns in the GROUP BY clause that are not in the SELECT clause. In this case, the Query Driver organizes the query by the groupings indicated in the GROUP BY clause, but does not print the values associated with the GROUP BY column names.

For example, the following query groups data by DEPARTMENT but does not print the name of the department:

```
Select:   NAME,ADDRESS,PHONE

From:   EMPLOYEE_NAME

Group By:   DEPARTMENT
```

- The GROUP BY clause works as a control break; that is, it provides statistics for more than one subgroup in a larger group. For example, to calculate the average grade of all students for each course in a college curriculum, you can define the following query:

```
Select:   COURSE,AVG(GRADE)

From:   STUDENT ST,COURSES CS

Where:   ST.STUDID=CS.STUDID

Group by:   COURSE
```

- A query that contains a GROUP BY or HAVING clause requires more time to display the first page to the screen than a query without these clauses. With a GROUP BY or HAVING clause, the DASL software must collect data for the entire query before determining what data to display on the first page of the query. If the database is large, the collection process takes some time.

  To preview the format of your query, it is good practice to enter only the SELECT and FROM clauses to run the query to your terminal. Then, add other clauses before sending to a file or printer.

Related Sections

FROM Clause
HAVING Clause
Query Driver Functions
SELECT Clause
SELECT Statement
Value Expressions
WHERE Clause

HAVING Clause

Like the WHERE clause, the argument to the HAVING clause is a list of search conditions. These search conditions generally apply to the preceding GROUP BY clause, or the FROM clause if there is no GROUP BY clause. However, the HAVING clause affects groups, whereas the WHERE clause affects individual rows.

Format

HAVING SPpredicate SP{AND SPpredicate} . . .

HAVING SPpredicate SP{OR SPpredicate} . . .

where:

predicate	is a conditional expression that evaluates to true or false
AND	is a logical operator that requires that both predicates joined by AND have a truth value of true
OR	is a logical operator that requires that either predicate joined by OR has a truth value of true

Explanation

The HAVING clause follows the GROUP BY clause in a query specification and works like a WHERE clause on each group of the result table. You use the HAVING clause to eliminate groups in a result table. If there is no GROUP BY clause in the specification, the result table is considered to have only one group.

The argument of the HAVING clause is a list of conditions to search for in each group of the result table. The search conditions consist of one or more predicates. The types of predicates include the following:

- Comparison predicates use comparison operators to compare two value expressions. The operators you can use are <, >, =, <=, =>, and <>.
- BETWEEN predicates compare a range of values.
- IN predicates specify a quantified comparison.
- LIKE predicates specify a pattern-match comparison.
- NULL predicates test for a null value.

You can precede the BETWEEN, IN, LIKE, and NULL predicates with a NOT operator. The NOT operator requires that the predicate evaluate to false.

Comments

Keep the following points in mind when you use the HAVING clause:

- Typically, the HAVING clause follows a GROUP BY clause. However, you can specify a HAVING clause without a GROUP BY clause. In this case, the conditions in the HAVING clause apply to all the rows in the table created as a result of the SELECT, FROM, and WHERE clauses.
- Column names in the HAVING clause must be specified within a function if they are not included in the GROUP BY clause.
- A query that contains a GROUP BY or HAVING clause requires more time to display the first page to the screen than a query without these clauses. With a GROUP BY or HAVING clause, the DASL software must collect data for the entire query before determining what data to display on the first page of the query. If the database is large, the collection process takes some time.

  To preview the format of your query, it is good practice to enter only the SELECT and FROM clauses to run the query to your terminal. Then, add other clauses before sending to a file or printer.

Related Sections

FROM Clause
GROUP BY Clause
Query Driver Functions
Search Conditions
SELECT Clause
SELECT Statement
Value Expressions
WHERE Clause

Example

In this example, you first create an intermediate table from the FROM and WHERE clauses. You then group the rows in this table by SEX and NAME. The HAVING clause further limits the result table to groups with a SALARY greater than 75000.

```
Select:    NAME,SEX,AGE,SALARY

From:    EMPLOYEE EM

Group by:    SEX,NAME

Having:    AVG(SALARY)>75000
```

Value Expressions

A value expression is a symbol or a string of symbols used to represent or calculate a single value. Value expressions are used in the SELECT, WHERE, HAVING, and GROUP BY clauses. For example, the argument to the SELECT clause, the select list, is a series of value expressions separated by commas.

Explanation

Value expressions can include any of the following:

- Literal strings enclosed in quotation marks
- Query Driver functions
- Column specifications
- Numeric literals
- Value expressions joined by arithmetic operators

Literal Strings

You can specify a string of printable characters enclosed in quotation marks as a value expression. The printable characters consist of:

- Uppercase alphabetic characters
- Lowercase alphabetic characters
- Numerals
- Special characters such as *, #, &, and $

The Query Driver does not preserve the case distinction in character string literals. For example, NAME="Smith" and NAME="SMITH" yield the same results.

Query Driver Functions

Query Driver functions are statistical expressions that calculate a single value for a collection of rows.

The Query Driver functions that you can use are COUNT, SUM, AVG, MAX, and MIN.

See the "Query Driver Functions" section of this chapter for more information on using Query Driver functions.

Column Specifications

Columns are data elements that describe a set of data values contained in rows. Each column within a single table must have a unique column name.

You can precede a column name by a table name or a correlation name and a period to indicate which table the column belongs to. For example, the AUTHORS table has a correlation name of AU and a column NAME. You can indicate the NAME column with the following column specifications:

```
AUTHORS.NAME

AU.NAME
```

You must use a table name or correlation name when specifying columns that are not unique in a query. For example, several tables in the same query may each have a NAME column. In this case, you must specify the correlation name with the NAME column name to indicate the desired table and column.

> **Note:** You cannot use SQL keywords as column names. See Table D–1 in Appendix D for a list of keywords that are not allowed.

Numeric Literals

The numeric literals that you can use in a value expression include:

- A decimal string that consists of digits and an optional decimal point
- A decimal number in scientific notation (E notation) to indicate exponents
- A negative decimal string

The following examples show valid numbers used in value expressions:

```
123

15.678

12E4

-56.98
```

Using Arithmetic Operators

You can combine one or more value expressions with arithmetic operators. The resulting expression is called another value expression. When you use a value expression that contains arithmetic operators in a SELECT statement, the Query Driver calculates the numeric value associated with the expression and uses that value.

Table 4–2 shows the arithmetic operators you can use and their functions.

Table 4–2 Arithmetic Operators

Operator	Function
+	Add
–	Subtract
*	Multiply
/	Divide

Keep the following points in mind when you use arithmetic operators in value expressions:

- You can use values that are NUMERIC, DATE, and TIME data types in arithmetic expressions.

  Date and time are stored in the DASL software in $H format. When you add 1 to a date, you add one day. When you add 1 to a time, you add one second. However, with time, values greater than 86399 are invalid.

  When you specify a date or time in a value expression, you can specify any valid DASL date or time. You must enclose the date or time within quotation marks, for example:

  ```
  EXAMDATE="10/1/1990"

  EXAMDATE="JAN 10, 1990"

  EXAMDATE>"T-10"
  ```

- You cannot use an arithmetic operator to combine values that are STRING or NAME data types.
- You do not have to use spaces to separate arithmetic operators from value expressions.
- You can use parentheses to control the order in which the Query Driver performs arithmetic operations. The Query Driver evaluates arithmetic expressions in the following order:

  1. Value expressions in parentheses
  2. Multiplication and division, from left to right
  3. Addition and subtraction, from left to right

Related Sections

FROM Clause
GROUP BY Clause
HAVING Clause
Query Driver Functions
SELECT Clause
SELECT Statement
WHERE Clause

Examples

The following examples show valid value expressions:

```
NAME,AGE,SEX

DEPARTMENT,SUM(SALARY)

AVG(AGE+10)

PN.NAME

14E2

"Total Count ="
```

Query Driver Functions

Query Driver functions are value expressions you can specify in the SELECT statement. Query Driver functions calculate values based on a value expression for every row in a result table.

Explanation

Query Driver functions are statistical expressions that calculate a single value for a collection of rows. With most functions, you must specify a column name enclosed in parentheses, for example, SUM(SCORES).

Table 4–3 shows the Query Driver functions and their actions.

Table 4–3 Query Driver Functions

Function	Action
AVG	Calculates the average value of the rows in the specified column.
COUNT	Counts the number of rows in a result table. Can count all values or distinct values in a column.
MAX	Calculates the largest of the values in the rows of the specified column.
MIN	Determines the smallest of the values in the rows of the specified column.
SUM	Calculates the total of the values in the rows of the specified column.

Comments

Keep the following points in mind when you use Query Driver functions:

- The Query Driver does not include null values when it evaluates functions.
- You can specify DISTINCT to exclude redundant values in rows.
- The argument commonly used with a Query Driver function is a column name. However, you can also use arithmetic operators within the argument, for example:

  ```
  AVG(AGE+20)
  ```

- You cannot nest functions. A value expression used as an argument to a function cannot include a function. For example, the following is not a valid argument:

  ```
  SUM(AVG(AGE))
  ```

AVG

The Query Driver AVG function calculates the average of the values in the rows of a specified column.

Format

AVG(value_expression)

AVG(DISTINCT SP column_specification)

where:

value_expression is usually a column name whose average value you want, or a column name and number combined by an arithmetic operator

DISTINCT calculates the average of distinct values for the column name you specify

column_specification is a column name

Comments

Keep the following points in mind when you use the AVG function:

- You can use AVG only with NUMERIC, DATE, and TIME type column names.
- You can specify DISTINCT to exclude redundant values in columns.
- If there are no rows in the result table to which it is applied, AVG returns a null value.

Related Sections

COUNT
MAX
MIN
Query Driver Functions
SUM

Example

You can specify the following SELECT statement to calculate the average height of a class of students. The HGT column name contains values for height.

```
Select:   AVG(HGT)

From:   CLASS
```

COUNT

The Query Driver COUNT function counts the number of rows in a result table.

Format

COUNT(*)

COUNT(DISTINCT SPcolumn_specification)

where:

*	calculates the total number of rows in a result table
DISTINCT	calculates the number of distinct values in the column you specify
column_specification	is a column name

Comments

Keep the following points in mind when you use the COUNT function:

- You can use COUNT with all data types of column names.
- Use COUNT(*) to count all rows of a result table.
- Use COUNT(DISTINCT column_specification) to count the number of distinct rows in a specified column.

  COUNT(DISTINCT column_specification) eliminates rows with duplicate values from the number it calculates. It does not count rows with null values.
- If there are no rows in the result table, COUNT returns a zero (0).

Related Sections

AVG
MAX
MIN
Query Driver Functions
SUM

Examples

You can specify the following COUNT statement to count the number of students in a class.

```
Select:   COUNT(*)

From:   CLASS
```

You can specify the following COUNT function to count the number of different fields of study that the students represent.

```
Select:   COUNT(DISTINCT FIELD)

From:   CLASS
```

MAX

The Query Driver MAX function calculates the largest value in a specified column.

Format

MAX(value_expression)

MAX(DISTINCT [SP]column_specification)

where:

value_expression is usually a column name whose largest value you want, or a column name and number combined by an arithmetic operator

DISTINCT calculates the largest of distinct values in the column you specify

column_specification is a column name

Comments

Keep the following points in mind when you use the MAX function:

- You can use MAX with all data types of column names. If the data type is STRING or NAME, the value of MAX depends upon the collating order of values in the column.
- The results when you use MAX(DISTINCT) are identical to the results using MAX without the DISTINCT keyword.
- If there are no rows in the result table to which it is applied, MAX returns a null value.

Related Sections

AVG
COUNT
MIN
Query Driver Functions
SUM

Example

You can specify the following SELECT statement to calculate the height of the tallest member of a class of students. The HGT column name contains values for height.

```
Select:   MAX(HGT)

From:   CLASS
```

MIN

The Query Driver MIN function calculates the smallest value in a specified column.

Format

MIN(value_expression)

MIN(DISTINCT SP column_specification)

where:

value_expression is usually a column name whose smallest value you want, or a column name and number combined by an arithmetic operator

DISTINCT calculates the smallest of distinct values in the column you specify

column_specification is a column name

Comments

Keep the following points in mind when you use the MIN function:

- You can use MIN with all data types of column names. If the data type is STRING or NAME, the value of MIN depends upon the collating order of values in the column.
- The results when you use MIN(DISTINCT) are identical to the results using MIN without the DISTINCT keyword.
- If there are no rows in the result table to which it is applied, MIN returns a null value.

Related Sections

AVG
COUNT
MAX
Query Driver Functions
SUM

Example

You can specify the following SELECT statement to calculate the shortest height represented in a class of students. The HGT column name contains values for height.

```
Select:   MIN(HGT)

From:   CLASS
```

SUM

The Query Driver SUM function calculates the total of all the values in a specified column.

Format

SUM(value_expression)

SUM(DISTINCT SP column_specification)

where:

value_expression is usually a column name whose sum of values you want, or a column name and number combined by an arithmetic operator

DISTINCT calculates the sum of distinct values for the column name that you specify

column_specification is one or more column names separated by commas

Comments

Keep the following points in mind when you use the SUM function:

- You can use SUM only with NUMERIC, DATE, and TIME type column names.
- You can specify DISTINCT to exclude redundant values in rows.
- If there are no rows in the result table to which it is applied, SUM returns a zero (0).

Related Sections

AVG
COUNT
MAX
MIN
Query Driver Functions

Example

You can specify the following SELECT statement to calculate the total annual payroll of the faculty at UDASL.

```
Select:   SUM(SALARY)

From:   UDASL
```

Search Conditions

The argument to the WHERE and HAVING clauses are search conditions. A search condition contains one or more predicates joined by any number of ANDs or ORs. Predicates are conditional expressions that evaluate to true or false.

Format

predicate SP{AND SP predicate} . . .

predicate SP{OR SP predicate} . . .

where:

predicate is a conditional expression that evaluates to true or false

AND is a logical operator that requires that both predicates joined by AND have a truth value of true

OR is a logical operator that requires that either predicate joined by OR has a truth value of true

Explanation

Within predicates, you can use two types of operators to describe a search condition. The types of operators you use within a predicate are:

- Conditional operators (BETWEEN, IN, LIKE, and NULL)
- Comparison operators such as =, >, and <

Table 4–4 describes these operators and the different kinds of predicates they create.

Table 4–4 Operators Within Predicates

Operator	Description
Comparison	Compares two values using comparison operators such as = and >
BETWEEN	Compares one value with a range of values
IN	Compares one value with another value or range of values
LIKE	Searches character string literals for pattern matches
NULL	Tests for null values in value expressions

See the following sections for descriptions of each comparison and conditional operator.

Using Logical Operators to Create Complex Search Conditions

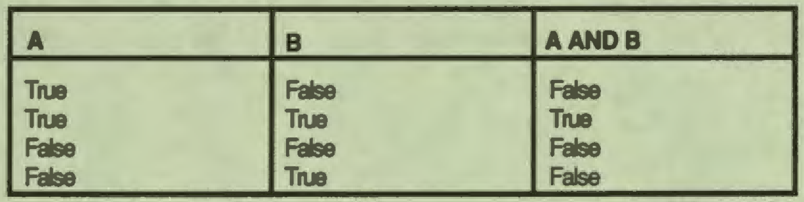
You can create a complex search condition when you combine any number of predicates with the logical operators AND and OR. You can set off individual predicates within a complex predicate by using parentheses.

The Query Driver evaluates parts of a complex predicate in the following order:

1. Predicates enclosed in parentheses
2. Predicates combined with OR
3. Predicates combined with AND

Figures 4–5 and 4–6 show *truth tables* that summarize how the Query Driver evaluates predicates combined with logical operators.

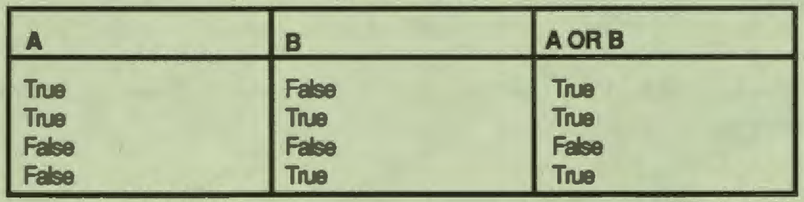
Figure 4–5 Logical Operators: AND



A	B	A AND B
True	False	False
True	True	True
False	False	False
False	True	False

MR–2898–RA

Figure 4–6 Logical Operators: OR



A	B	A OR B
True	False	True
True	True	True
False	False	False
False	True	True

MR–2899–RA

Comments

Keep the following points in mind when you specify predicates:

- If either value being compared in a predicate is null, the value of the predicate is false.
- You can not use Query Driver functions in the predicates of a WHERE clause.

- You can precede the BETWEEN, IN, LIKE, and NULL predicates with a NOT operator. The NOT operator requires that the predicate evaluates to false, for example:

  ```
  DEATH_DATE NOT NULL

  STATE NOT IN (ME,MA,VT)
  ```

- All queries and searches on character strings are case insensitive. For example, the following two searches produce the same results:

  ```
  NAME BETWEEN "A" AND "H"

  NAME BETWEEN "a" AND "h"
  ```

Related Sections

BETWEEN Predicate
Comparison Predicates
IN Predicate
LIKE Predicate
NULL Predicate

Examples

The following examples illustrate valid search conditions:

```
(AGE>25 AND SEX="F") OR (AGE>60 AND SEX="M")

PATIENT.AGE BETWEEN 20 AND 25

PATIENT.NAME LIKE "B%,J%"

DOCTOR.PHONE NOT NULL

SEX IN (U,F)
```

Comparison Predicates

Comparison predicates use comparison operators to compare two value expressions. You can use a comparison predicate as the argument to a WHERE or HAVING clause.

Format

value_expression comparison_operator value_expression

where:

value_expression	is a symbol or a string of symbols used to represent or calculate a single value
comparison_operator	is an operator that compares the two value expressions

Explanation

Table 4–5 shows the comparison operators you can use and the truth values of the predicates.

Table 4–5 Comparison Operators

Operator	Predicate is:
=	True if the two value expressions are equal
<>	True if the two value expressions are not equal
<	True if the first value expression is less than the second value expression
<=	True if the first value expression is less than or equal to the second value expression
>	True if the first value expression is greater than the second value expression
=>	True if the first value expression is greater than or equal to the second value expression

Comments

Keep the following points in mind when you use the comparison predicates:

- You can only use predicates with the WHERE and HAVING clauses.
- It is not necessary to place a space between comparison operators and value expressions.

Related Sections

BETWEEN Predicate
IN Predicate
LIKE Predicate
NULL Predicate

Examples

The following examples show some valid comparison predicates:

```
Where:    AGE>10

Having:    SALARY<=50000
```

BETWEEN Predicate

The BETWEEN predicate compares a value with a range of values. You can use the BETWEEN predicate with the WHERE and HAVING clauses.

Format

value_expression SP BETWEEN SP value_expression SP AND SP value_expression

where:

value_expression is a symbol or a string of symbols used to represent or calculate a single value (The value expressions following the BETWEEN operator encompass a range of values.)

AND is a logical operator that defines the range of values

Explanation

BETWEEN predicates are a simpler way of representing conditions that can be represented using comparison operators. For example, the following two predicates produce the same results, but the BETWEEN predicate is less complex:

```
AGE BETWEEN 55 AND 65
```

```
AGE>=55 AND AGE<=65
```

Comments

Keep the following points in mind when you use the BETWEEN predicate:

- You can only use the BETWEEN predicate as an argument to the WHERE and HAVING clauses.
- The order of the arguments following BETWEEN does not affect how the Query Driver evaluates a BETWEEN predicate. For example, both of the following predicates produce the same result table:

  ```
  SALARY BETWEEN 50000 AND 75000
  ```

  ```
  SALARY BETWEEN 75000 AND 50000
  ```

- You can use the BETWEEN predicate with character strings. Enclose each character string in quotation marks. For example, to retrieve the authors whose names begin with the letters S and T:

  ```
  AUTHORS BETWEEN "S" AND "T"
  ```

  To find only those authors whose names begin with S, you can specify the following:

  ```
  AUTHORS BETWEEN "S" AND "Szzzzz"
  ```

All queries and searches on character strings are case insensitive. For example, the following two searches produce the same results:

```
NAME BETWEEN "A" AND "H"

NAME BETWEEN "a" AND "h"
```

- You can precede the BETWEEN predicate with a NOT operator. The NOT operator requires that the predicate evaluates to false, for example:

  ```
  NAME NOT BETWEEN "A" AND "Fzzz"
  ```

Related Sections

Comparison Predicates
IN Predicate
LIKE Predicate
NULL Predicate

IN Predicate

The IN predicate compares a value with another value or a collection of values. You can use the IN predicate with the WHERE and HAVING clauses.

Format

value_expression SPIN SP(value_list)

where:

value_expression is a symbol or a string of symbols used to represent or calculate a single value

value_list is a list of values separated by commas and enclosed in parentheses. Character strings in the list must be enclosed in quotation marks.

Comments

Keep the following points in mind when you use the IN predicate:

- You can use the IN predicate only with the WHERE and HAVING clauses.
- IN predicates represent conditions that can also be represented using comparison operators. For example, the following two predicates produce the same results:

  ```
  NAME IN "Smith, James"
  ```

  ```
  NAME="Smith, James"
  ```
- IN predicates are useful when you want to compare a value to a list of values. For example, you can use the following IN predicate to select a state in New England:

  ```
  STATE IN ("CT","RI","MA","ME","VT","NH")
  ```
- You can precede the IN predicate with a NOT operator. The NOT operator requires that the predicate evaluates to false, for example:

  ```
  AGE NOT IN (10,20,35)
  ```

Related Sections

BETWEEN Predicate
Comparison Predicates
LIKE Predicate
NULL Predicate

LIKE Predicate

The LIKE predicate searches character string literals for pattern matches. You can use the LIKE predicate with the WHERE and HAVING clauses.

Format

column_specification SPLIKE SP"pattern"{ SPESCAPE SP"escape_character"}

where:

column_specification	is a column name
pattern	is a string of characters enclosed in double quotation marks
ESCAPE	is an optional keyword used to specify an escape character
escape_character	is a character enclosed in double quotation marks that you specify to allow the use of wildcard characters as character string literals within the pattern

Comments

Keep the following points in mind when you use the LIKE predicate:

- You can use the LIKE predicate only with the WHERE and HAVING clauses.
- The Query Driver interprets the values of the column as character string literals and compares them to the pattern that you use as an argument to LIKE.
- The LIKE predicate does not distinguish between uppercase and lowercase characters.
- There are two wildcard characters that you can use within the LIKE pattern argument:
  - — The percent sign (%) stands for any string of characters, including no characters at all.
  - — The underscore (_) stands for any single character.

  For example, you can use the following predicate to find all AUTHORS whose names begin with "BRO":

  ```
  AUTHORS LIKE "BRO%"
  ```

  All other characters represent themselves.

- If you want to include the percent sign (%) and the underscore (_) as part of the character string literal in the pattern, you must use the ESCAPE keyword to define an escape character. You then precede the percent sign (%) or the underscore (_) with the escape character when you specify the pattern. For example, suppose you want to specify the following character string literals as the pattern:

  ```
  100%

  EMPLOYEE_CODE
  ```

  You can specify the following predicates for these literal strings:

  ```
  SCALE LIKE "100.%" ESCAPE "."

  REPORT LIKE "EMPLOYEE^_CODE" ESCAPE "^"
  ```

- You can precede the LIKE predicate with a NOT operator. The NOT operator requires that the predicate evaluates to false, for example:

  ```
  AUTO NOT LIKE "TOYOTA"
  ```

Related Sections

BETWEEN Predicate
Comparison Predicates
IN Predicate
NULL Predicate

NULL Predicate

The NULL predicate tests for null values in value expressions. You can use the NULL predicate with the WHERE and HAVING clauses.

Format

column_specification SP IS SP NULL

where:

column_specification	is a column name
IS	is a required keyword

Comments

Keep the following points in mind when you use the NULL predicate:

- You can use the NULL predicate only with WHERE and HAVING clauses.
- If the specified column is null, the predicate is true. If the specified column is not null, the predicate is false.
- You must use a NULL predicate to retrieve rows with null values in a particular column. Other predicates such as NOT LIKE or <> (not equal) do not include rows with null values in the result table.
- You can precede the NULL predicate with a NOT operator. The NOT operator requires that the predicate evaluates to false, for example:

  ```
  INSURANCE_CODE IS NOT NULL
  ```

Related Sections

BETWEEN Predicate
Comparison Predicates
IN Predicate
LIKE Predicate

Example

In the following example, you can use a NULL predicate to count the rows that have no values stored in the DEATHDATE column:

```
Select:   COUNT(*)

From:   AUTHORS

Where:   DEATHDATE IS NULL
```

Define System Queries Option

Application managers can use the Define System Queries Option to define new system queries or edit system queries that they or application users have previously defined.

System Queries and User Queries

Application users and system managers can create two types of queries, distinguishable by ownership:

- *User queries* are queries that individual application users define for their own use. User queries are owned by the user who defines them and are available only to their owner.
- *System queries* are available to all application users who have the Security System privileges to use the query. System queries are owned by the system manager. Application users cannot edit or delete system queries.

Application users can use the Define Query Option to create user queries and system queries. To create a system query, complete the query specification, then copy the query to the system manager. When you copy the query, specify a new name for the query. Once you have copied a query to the system manager, the query becomes a system query and is available to other users. You no longer own the renamed query and cannot edit or delete it.

System managers can use the Define System Queries Option to create, edit, or delete system queries.

> **Note:** All tables that you use in a system query must be available to all users. Therefore, the tables must have the @ privilege.

Queries and System Security

The DASL Security System is privilege-based; that is, it requires user and device identification and validation.

Briefly, each user has a classification code that determines the privilege codes assigned to the user. Each device is associated with a privilege code or codes. System statuses are also associated with a string of privilege codes. The intersection of the system status privileges and the device privileges with the user privileges determines the current process privileges of the user.

All login screens and menu screens are integrated into the Security System through the Screen Driver. A user can only access those screens allowed by the current process privileges. The "Security System Option" section of Chapter 6 describes the DASL Security System.

Likewise, the tables and queries you define through the Query Driver are integrated with the DASL Security System. You assign privileges to tables, and queries are owned by individual users or the system manager.

Keep the following comments about system security in mind when you define tables and queries:

- Tables are associated with privilege codes. The privileges of tables that you specify when you define a query must match your current process privileges. If you specify a table for which you do not have privileges, the DASL software displays an error message.
- Individual users can view and use their own user queries and also view and use system queries.
- Users can copy queries to another user. However, for the other user to view or use the copied query, the user must have the appropriate privileges to use the tables in the query.
- Each query has a unique name. Whenever you copy a query to another user or to the system manager, you do not own the copied query (which you have renamed). You can, however, edit or delete the original query which you own. It is possible to have several edited copies of a popular query owned by individual users under different names. In this case, it is good practice to define commonly used queries as system queries so that editorial control is maintained by the system manager.
- When you display a lookup list of query names at the "Query Name:" prompt of the Define Query Option, the DASL software displays both your own user queries and system queries, but does not indicate ownership in the lookup screen. When you select a query from the list, the DASL software displays the owner in the Define Query Option screen.

Using the Define System Queries Screen

The Define System Queries screen is identical to the Define Query Option screen. You enter the same specifications and respond to the same prompts. See the "Define Query Option" section of this chapter for more information about defining queries and the information you enter in the Define System Queries screen.

Related Sections

Define Query Option
Define Tables Option
Security System Option, Chapter 6

Define Tables Option

To use the Query Driver, you must represent the database as a relational database, a collection of base tables containing columns and rows. The Query Driver searches specified base tables for requested information and produces a result table in response to a query.

Explanation

Each table contains columns and rows. Columns are like fields, and rows are like records in a database.

- Columns are data elements that describe a set of data values.

  Each column in a table must have a unique name and a data type. All the values in that column are of the same data type and exist in the same table. Columns are like the headings for vertical collections of data in a table.

  Columns are ordered from left to right.

  When you define a query for a table, you specify column names as the data elements in the query.

- Rows contain data values for each column of the table.

  While columns are like headings for vertical collections of data, rows are horizontal collections of data that fill in data values for each column.

  A table can contain multiple rows of data values. Each row contains a value for every column of the table, even if that value is null.

Defining Tables and Columns

Table 4–6 shows the information you specify to define a table. Note that the information you use to define tables must be data that is described in the Data Dictionary. You cannot perform queries on data that cannot be described in the Data Dictionary.

Table 4–6 Information for Defining Tables

Item	Description	Examples
Table Name	The name of the table, made up of one alphabetic character followed by up to 17 alphanumeric characters or underscores (_). **You cannot use SQL keywords as table names.**	AUTHORS

Table 4–6 (Cont.) Information for Defining Tables

Item	Description	Examples
Desc	A free text description for documentation and lookup purposes. Helps users make accurate selections from lookup lists.	Author info
Primary Key(s)	A primary key data name that points to the dependent data names contained in this table. Each table **must** have at least one primary key.	AID
Group(s)	The group or groups from the Group Dictionary to which the table belongs.	DEMO,CLASS
Privilege Code	A list of privilege codes, defined in the Privilege Code Dictionary of the Development Environment Menu. At least one of the table's privilege codes must match one of the user's privilege codes for the table to be available and displayed to a user. If you leave this field blank, the table is not available to users.	@
Comments that display on Schema Diagram	Lines of comments describing the table that you want to display on the Schema Diagram.	Table of author info

For each table, you define columns. Table 4–7 shows the information you specify to define columns.

Table 4–7 Information for Defining Columns

Item	Description	Examples
Column Number	An option index number from 0 to 99.99.	2 3.5
Data Name	A dependent data name or primary key data name from the Data Dictionary.	AID ANAME
Column Name	The name of the column, made up of one alphabetic character followed by up to 17 alphanumeric characters or underscores (_). Each name within a table must be unique. **You cannot use SQL keywords as column names.**	NAME TITLE
Description	Up to 35 characters describing the column name. The DASL software displays the description in lookup lists.	Title of Book

Table 4–7 (Cont.) Information for Defining Columns

Item	Description	Examples
Data Type	One of the five DASL data types assigned to the column name.	DATE, TIME, NAME, STRING, NUMERIC
Data Length	The maximum number of characters that can be entered in fields associated with the column name. You can override the data name default for length. If the data type is NUMERIC, you can specify a length, decimal separator (period (.) or comma (,)), and number of characters to appear at the right of the decimal separator. The length you specify must include the decimal separator.	30 6.2
Attributes	Commands that specify how the DASL software handles input. Usually, attributes default from the data-name definition in the Data Dictionary. The Query Driver uses the LCASE attribute in the WHERE and HAVING clauses, converting lowercase characters to uppercase characters to compare strings. You should specify the LCASE attribute only if your database contains lowercase data. All other data-name attributes have no effect in the Query Driver.	FULL, LCASE, NOECHO, NOFILTER, REQUIRED, SPACES

Using the Define Tables Option

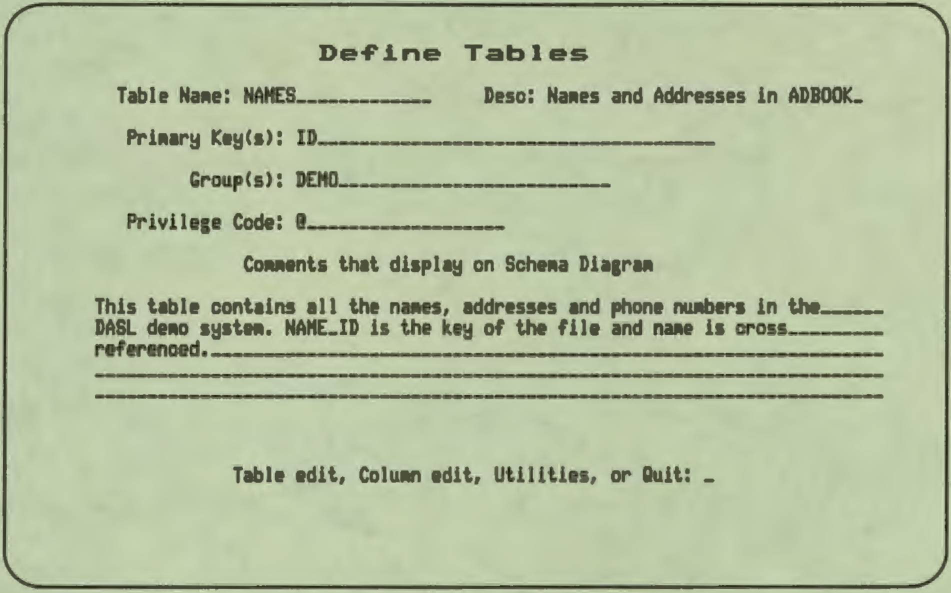
You use the Define Tables Option to define a table and the columns of that table. You can also use this option to edit, delete, copy, print, and add comments to existing tables.

The Define Tables Option has two screens. In the first screen, you provide table definitions. In the second screen, you define each column in the table.

Filling In Table Definitions

Figure 4–7 shows the first Define Tables screen.

Figure 4–7 Define Tables Screen #1



```
                          Define Tables

       Table Name: NAMES____________     Desc: Names and Addresses in ADBOOK_

        Primary Key(s): ID________________________________

             Group(s): DEMO________________________

        Privilege Code: @_________________

                  Comments that display on Schema Diagram

     This table contains all the names, addresses and phone numbers in the______
     DASL demo system. NAME_ID is the key of the file and name is cross________
     referenced._________________________________________________________________
     ____________________________________________________________________________
     ____________________________________________________________________________


                  Table edit, Column edit, Utilities, or Quit: _
```

To define a new table, enter a new name (one letter followed by up to 17 alphanumeric characters or underscores) after the "Table Name:" prompt. You can then fill in the remaining fields.

Keep in mind the following comments when you define your table:

- You cannot use SQL keywords as table names or column names. See Table D–1 in Appendix D for a list of the keywords that are not allowed.
- You **must** specify a key or keys for each table. If you specify a second key, the second key must be related to the first key. The Query Driver then combines the two keys to determine the primary key for the table.
- Define the keys as columns if you want the application user to form relationships among tables based upon the keys. For example, if you define the key ID as a column in the NAMES table, you can then point to ID from another table in the WHERE clause of the SELECT statement.
- You must specify a privilege code to make the table available to a user with the appropriate privilege.

When you complete your table definition, the "Save, Edit, or Quit:" prompt appears.

When you enter S, the DASL software displays the following prompt:

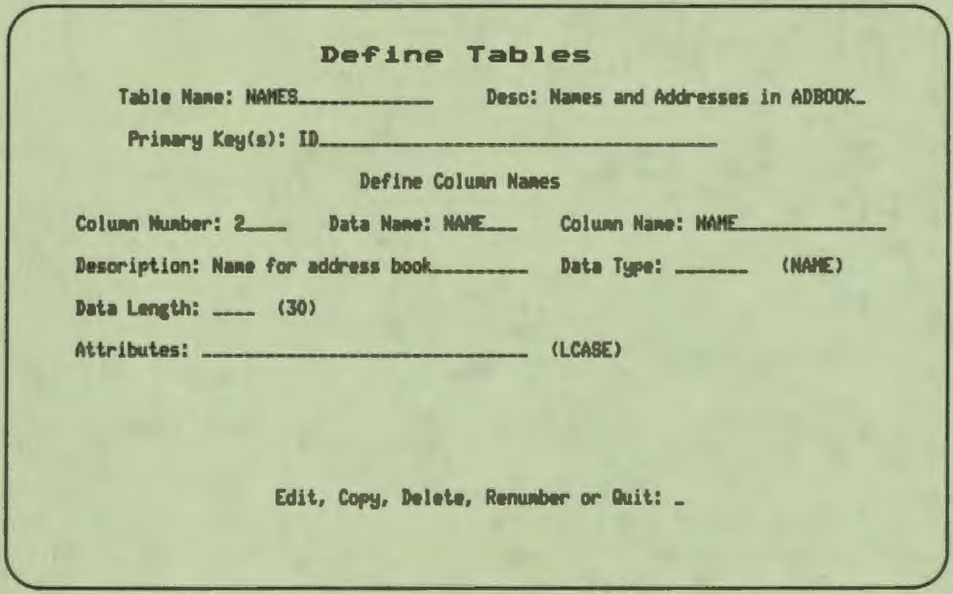
```
Table edit, Column edit, Utilities, or Quit:
```

At this prompt, enter T to edit the table definition, C to edit column names, U to use the utilities, or Q to quit from the screen.

Filling in Column Name Definitions

When you select C at the decision prompt, you see the second Define Tables screen as shown in Figure 4–8.

Figure 4–8 Define Tables Screen #2



```
                           Define Tables
       Table Name: NAMES____________     Desc: Names and Addresses in ADBOOK_
        Primary Key(s): ID_________________________________
                          Define Column Names
    Column Number: 2____   Data Name: NAME___   Column Name: NAME_____________
    Description: Name for address book_________  Data Type: ______   (NAME)
    Data Length: ____  (30)
    Attributes: ______________________________  (LCASE)



                   Edit, Copy, Delete, Renumber or Quit: _
```

You must first enter a column number at the "Column Number:" prompt. You can then fill in the remaining fields if the column is a new one. After you complete the column name definition, the DASL software displays the "Save, Edit, or Quit:" prompt.

If the column is an existing one, the DASL software fills in the screen and displays the following prompt:

```
Edit, Copy, Delete, Renumber or Quit:
```

Enter C at the decision prompt to make a copy of the column under a different name. You can later edit the new column name to change any of the copied values.

Enter R at the decision prompt to renumber an existing column. The DASL software prompts you for a new column number as follows:

```
Renumber:
```

The DASL software then repositions the columns by column number in ascending numeric order.

Keep the following points in mind when you define column names:

- If you specify a data name at the "Data Name:" prompt, the DASL software fills in many of the remaining fields with information that defaults from the Data Dictionary.

- You must specify a data type for each column, or use the default data type from the Data Dictionary. The data type of the column controls how the DASL software interprets values for that column.
- You can add column names to an existing table by giving each new column name a decimal number that indicates the position in which you want that column to appear in the table. For example, if the table consists of columns numbered 1 to 4, you can add a new column name numbered 1.1. Your new column name 1.1 appears in the table between column 1 and column 2. The DASL software automatically renumbers the columns.

Editing an Existing Table

To edit an existing table, enter the table name at the "Table Name:" prompt on the first Define Tables screen and then enter E at the "Save, Edit, or Quit:" decision prompt. Or, you can enter T at the "Table edit, Column edit, Utilities, or Quit:" prompt.

You can change any value in the first Define Tables screen, or you can add new information.

Using the Utilities

To delete, copy, print, or add documenting comments to any existing table, enter the table name and then choose the utilities when the DASL software displays the "Table edit, Column edit, Utilities, or Quit:" prompt. The DASL software then displays a utilities decision prompt, as follows:

```
Edit comments, Delete, Copy, or Print:
```

To add comments to document the table, enter E. You then see a Comments Screen. Use the Comments Screen to specify the creator, editor, and version number of the table. You can also enter text in a scroll region to document the table name.

To delete the table, enter D and verify your choice where prompted.

To copy the table values to a new table, enter C and specify the new table name. You can later edit the new table to change any of the copied values.

To print the table definition, enter P and, where prompted, take the following steps:

1. Enter Y or N to indicate whether to use a batch (detached) process to print the data name.
2. Enter the VMS device or VMS file name you want to use.

Related Sections

Define Query Option
Define System Queries Option
Display Schema Diagram Option
Display Table Definitions Option
Print Schema Diagram Option
Print Table Definitions (Full) Option

Query Driver Utilities

The Query Driver Menu includes several options that are user utilities. You can use these options to compile and run queries, to run various reports about tables and queries you have defined, to rebuild table cross-references, and to delete generated query routines. These options include:

- Run Query Option
- Print Query Definitions Option
- Print Table Definitions (Full) Option
- Display Table Definitions Option
- Display Schema Diagram Option
- Print Schema Diagram Option
- Build Table Cross-Reference Option
- Delete Generated Query Routines Option
- Compile Queries Option

Specifying Queries or Tables in Utility Scroll Regions

When you use some Query Driver utilities, the DASL software displays a scroll region so you can specify the queries or tables to compile or include in a report.

List the queries or tables by entering one or more of the following:

- Query names or table names
- An asterisk (*) for all queries or tables
- A partial name preceded or followed by an asterisk (*)
- An at sign (@) followed by the name of a defined DASL group for all queries or tables in that group
- A range of queries or tables in the form name1-name2

If you choose a range in the form name1-name2, the DASL software includes all queries or tables that fall within the specified range alphabetically by name.

If you specify a range of queries or tables, you can verify the queries or tables in the range. When the DASL software displays the "Continue, eXpand wildcards, Edit, or Quit:" prompt, enter X. The DASL software then displays all queries or tables that fall within the range.

To exclude any of the listed queries or tables, enter D in response to the "Continue, Delete, or Quit:" prompt that the DASL software displays after expanding the list. You then move back to the top of the scroll region. You can use the scroll region editing keys to change any of the entries.

> **Note:** The other choices at this prompt are C to continue to the device selection screen, E to edit the list of queries or tables, or Q to start over again.

Choosing the Destination

When you finish choosing queries or tables, the DASL software displays a device selection screen. Use the device selection screen to run queries or produce reports through a batch process or in your current process.

The DASL software displays one device selection screen in the development area and a different screen in the application area.

In the development area, the DASL software first prompts you to specify whether to run the query or produce a report through a batch process or through your current process. If you choose to use your current process, you can display the query result or report on your terminal, print it on any VMS device, or write it to a VMS file.

In the application area, the DASL software prompts you for the following information:

- Report Directory
- Device name
- Number of copies
- Start date
- Time

Enter Y to send your query result or report to the Report Directory. (The default at this prompt is N.)

Enter the name of a device defined in the Device Dictionary as follows:

- If you do not send the query result or report to the Report Directory, you can display it on your terminal or print it on any VMS device that is defined in the Device Dictionary.
- If you send the query result or report to the Report Directory, you can specify only a spooled device at this field. With a spooled device, the DASL software sends reports directed to that device into a spool file for temporary storage until the device is free to print the report. The contents of the spool file are queued to the device.

Enter the number of copies you want.

Enter the date and time that you want printing to begin.

At the "Continue, Edit, or Quit:" prompt, enter C to produce the report, E to edit the device selection screen, or Q to quit the screen without producing a report or query result.

Related Sections

Define Query Option
Define System Queries Option
Define Tables Option

Run Query Option

The Run Query Option runs the VAX DSM routine associated with any compiled DASL query.

Explanation

When you use the Run Query Option, the DASL software prompts you for the query name. At the "Query Name:" prompt, you can enter any of the following:

- An exact name
- An asterisk to view a lookup list
- A partial query name

You can specify any previously compiled query. The DASL software then fills in the completed query definition in the Run Query Screen and displays a "Run or Quit:" prompt.

After you verify your choice at the "Run or Quit:" decision prompt, the DASL software runs the query you specify.

Running Queries from the VAX DSM Environment

You can also run a query from the VAX DSM environment for testing purposes. To run a query from DSM, follow these steps:

1. Type D ^%DAQ at the DSM prompt (>).
2. Press Return. The DASL software then prompts you for a query name.
3. Enter the query name and press Return.

Related Sections

Compile Queries Option
Define Query Option
Define System Queries Option

Print Query Definitions Option

The Print Query Definitions Option runs a report that lists all specifications for any queries you specify.

Example 4–1 shows a sample Query Definitions Report.

Example 4–1 Query Definitions Report

```
18-Apr-89 12:35 PM              DASL                       Page 1
                   DSM Application Software Library
                              Print Queries


                           Query Name: ADBK1
__________________________________________________________________

Select NAME,CITY,STATE
From NAMES


                           Query Name: ADBK10
__________________________________________________________________

Select MAX(AGE),MIN(AGE),COUNT(*)
From NAMES


                          Query Name: ADBK12
__________________________________________________________________

Select NAME,STREET,CITY,STATE,ZIP
From NAMES
Where AGE BETWEEN 40 AND 60 AND STATE IN (MA,VT)
```

Related Sections

Define Query Option
Define System Queries Option

Print Table Overview Option

The Print Table Overview Option prints a brief report that lists the contents of any tables you specify. Application users can access this option on the Query Database Menu.

Example 4–2 shows a sample Table Contents Listing. This example shows the overview for a sample table, AUTHORS.

Example 4–2 Table Contents Listing

```
1-Mar-90    4:41          AUTHORS AND BOOKS                     Page 1
                             January 1989
                         Table Contents Listing


    Table:  AUTHORS
    Description: Author Information

    Key
  # Seq Column Name          Data Type   Length   Description
   ---------------------------------------------------------------

   1  1 ANO                  NUMERIC          6   Key data name
   2    NAME                 NAME            30   Author Name
   3    CITY                 STRING          30   Place of residence
   4    BIRTHDATE            NUMERIC         10   Date of Birth
```

When you choose the Print Table Definitions (Full) Option, the DASL software displays a scroll region where you can specify the tables you want.

Related Sections

Define Tables Option
Display Table Definitions Option
Print Table Definitions (Full) Option

Print Table Definitions (Full) Option

The Print Table Definitions (Full) Option lists the contents of any tables you specify.

Example 4–3 shows a sample Print Table Definitions report. The Print Table Definitions report lists all values entered for the tables you choose to list. This example shows a portion of a definition for a sample table, AUTHORS.

Example 4–3 Print Table Definitions Report

```
24-Mar-89 10:21 AM              DASL                           Page 1
                   DSM Application Software Library
                        Print Table Definitions

                       Definition of Table: AUTHORS
_____________________________________________________________________

Desc: Author Information
Group(S): CLASS
                        Edited on:23-Mar-89 by
Primary Key(S): AID
Privilege Code: @
_____________________________________________________________________

Column Number: 1    Data Name: AID          Column Name: ID
Desc: Key data name                         Data Type: NUMERIC
Data Length: (6)


Column Number: 2    Data Name: ANAME        Column Name: NAME
Desc: Author Name                           Data Type: (NAME)
Data Length: (30)


Column Number: 3    Data Name: APLACE       Column Name: PLACE
Desc: Place of birth                        Data Type: STRING
Data Length: (30)
```

Explanation

When you choose the Print Table Definitions (Full) Option, the DASL software displays the following prompt:

```
Sort by data name or column name?
```

Enter D to sort the report items by data name. Press Return or enter C to sort the report items by column name.

The DASL software then displays a scroll region where you can specify the tables you want.

Related Sections

Define Tables Option
Display Table Definitions Option

Display Table Definitions Option

The Display Table Definitions Option displays a brief table definition for a specified table on your terminal screen.

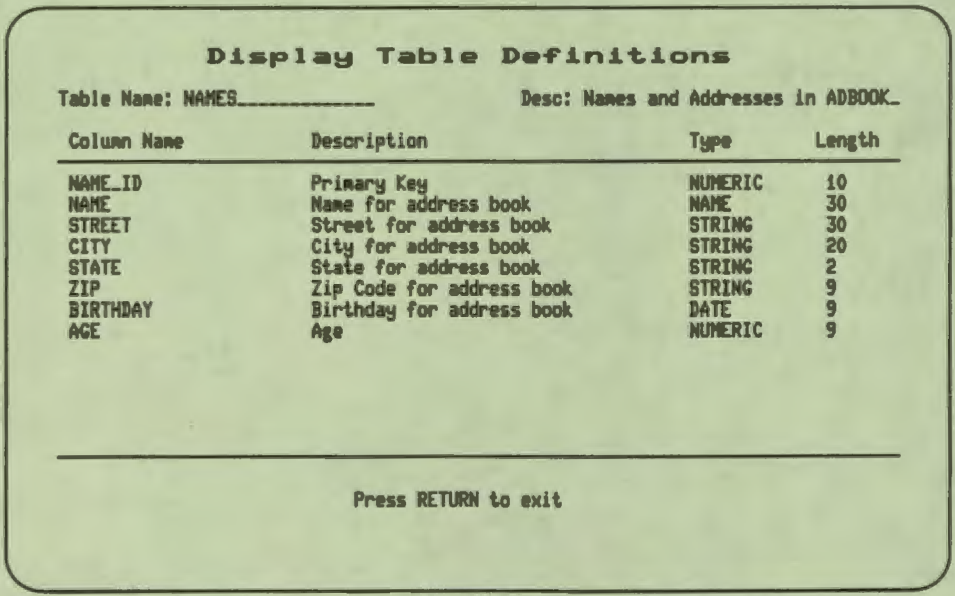
Explanation

When you use the Display Table Definitions Option, the DASL software displays a "Table Name:" prompt. You can enter any of the following:

- An exact name
- An asterisk to view a lookup list
- A partial table name

The DASL software then displays a brief table definition for the specified table. Figure 4–9 shows a sample Display Table Definitions screen for the AUTHORS table.

Figure 4–9 Display Table Definitions Screen



```
                    Display Table Definitions

Table Name: NAMES_____________          Desc: Names and Addresses in ADBOOK_

 Column Name          Description                        Type       Length
 _________________________________________________________________________
 NAME_ID              Primary Key                        NUMERIC     10
 NAME                 Name for address book              NAME        30
 STREET               Street for address book            STRING      30
 CITY                 City for address book              STRING      20
 STATE                State for address book             STRING      2
 ZIP                  Zip Code for address book          STRING      9
 BIRTHDAY             Birthday for address book          DATE        9
 AGE                  Age                                NUMERIC     9



 _________________________________________________________________________

                          Press RETURN to exit
```

To view the Display Table Definitions screen, select the Display Table Definitions Option, or press Help at the "Select:" prompt in the Define Query Option screen.

Related Sections

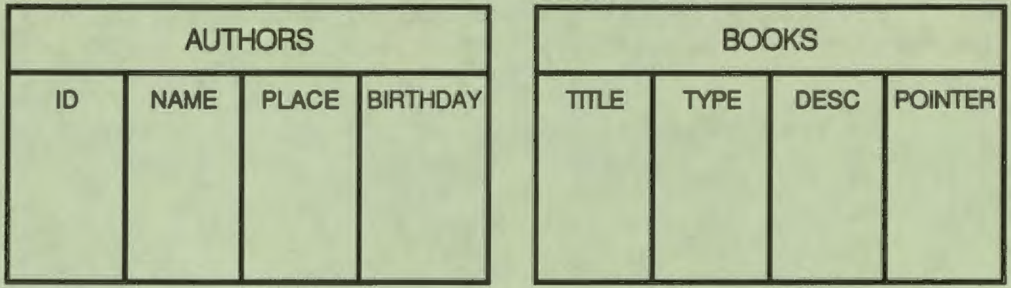
Define Query Option
Display Schema Diagram Option
Print Table Definitions (Full) Option

Display Schema Diagram Option

The Display Schema Diagram Option displays a schema diagram of the tables in the database and comments about the tables on the terminal screen. The DASL software only displays those tables that you have privileges to use.

Figure 4–10 represents a sample Schema Diagram screen that displays the AUTHORS and BOOKS tables.

Figure 4–10 Schema Diagram Screen



AUTHORS			
ID	NAME	PLACE	BIRTHDAY

BOOKS			
TITLE	TYPE	DESC	POINTER

MR–3057–RA

To view the Schema Diagram screen, select the Display Schema Diagram Option or press Help at the "Select:" prompt in the Define Query Option Screen.

Related Sections

Define Query Option
Display Table Definitions Option

Print Schema Diagram Option

You can use the Print Schema Diagram Option to print a schema diagram of the tables in the database in a 132-column report. You can print a schema diagram only for those tables that you have privileges to use.

Explanation

When you choose the Print Schema Diagram Option, the DASL software displays a device selection screen that allows you to produce the report through a batch process or through your current process. If you choose to use your current process, you can display the report on your terminal, print it on any VMS device, or write the report to a VMS file.

Related Sections

Define Query Option
Display Schema Diagram Option
Display Table Definitions Option

Build Table Cross-Reference Option

The Build Table Cross-Reference Option recreates cross-reference indexes for the tables you specify.

Explanation

The Build Table Cross-Reference Option is a utility for application developers.

Because table definitions require some non-standard cross references, the normal DASL cross-reference builder utility does not completely rebuild table cross references.

Use this utility to rebuild table cross references. You need to run this utility only when the cross-reference global ^DATDX has been damaged.

When you choose the Build Table Cross-Reference Option, the DASL software displays the following prompt:

```
Continue or Quit:
```

Enter C at this prompt to rebuild table cross-reference indexes, or enter Q to quit.

Related Section

Define Tables Option

Delete Generated Query Routines Option

Application developers use the Delete Generated Query Routines Option to delete query routines.

Delete Generated Query Routines Screen

When you select the Delete Generated Query Routines Option, the DASL software displays the following prompt:

```
Do you want to delete generated query routines?
```

Enter Y to delete query routines, or N to quit the screen without deleting query routines.

Explanation

Whenever application users create a query, the DASL software generates a MUMPS query routine. The DASL software gathers data from three sources to generate the query routine:

- Query definitions
- Table definitions
- Data Dictionary definitions (data names and templates)

In general, application users can modify a query definition, but not table definitions or Data Dictionary definitions. Whenever a query definition is modified, the DASL software automatically deletes the old MUMPS query routine and generates a current routine. Application developers, however, frequently do modify table definitions and Data Dictionary definitions. Unlike query definitions, the DASL software does not automatically regenerate MUMPS query routines after tables, data names, or templates have been modified.

To run queries in the development area, application developers must use the Delete Generated Query Routines Utility after modifying table definitions or Data Dictionary definitions. The DASL software then detects that a query does not have an associated MUMPS routine and automatically generates a MUMPS routine when the query is next run.

The DASL software always calls the Delete Generated Query Routine Utility when you run the ^%DACVRT routine for version updates.

Related Sections

Compile Queries Option
Define Query Option
Define System Queries Option

Compile Queries Option

The Compile Queries Option converts query definitions into executable VAX DSM routines.

You can use the Compile Queries Option to compile all queries or compile only the queries you specify. To compile all queries, enter Y in response to the "Compile all queries:" prompt. To compile specified queries, enter N.

Compiling Specified Queries

If you choose to compile specified queries, you then see a scroll region to list the queries you want to compile.

Choosing a Destination

When you finish specifying queries, the DASL software asks whether to compile the queries through a batch process or in your current process. If you are compiling more than one query, compile through a batch process.

Related Sections

Define Query Option
Define System Queries Option

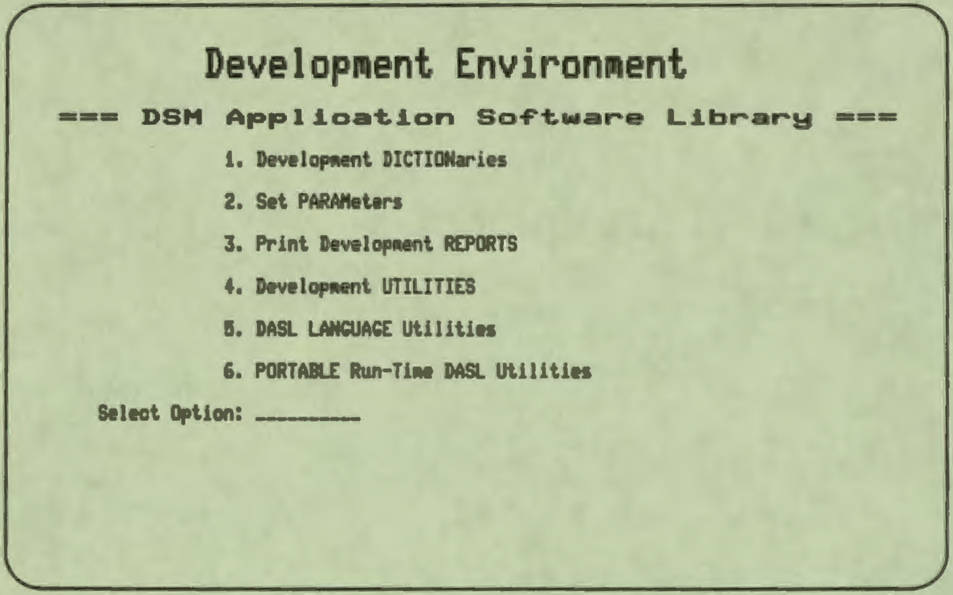
Chapter 5
Development Environment

This chapter describes the following Development Environment Menu options and their submenus:

- Development Dictionaries Option
- Set Parameters Option
- Print Development Reports Option
- Development Utilities Option
- DASL Language Utilities
- Portable Run-Time DASL Utilities

Figure 5–1 shows the options of the Development Environment Menu.

Figure 5–1 Development Environment Menu



```
              Development Environment
=== DSM Application Software Library ===
          1. Development DICTIONaries
          2. Set PARAMeters
          3. Print Development REPORTS
          4. Development UTILITIES
          5. DASL LANGUAGE Utilities
          6. PORTABLE Run-Time DASL Utilities
  Select Option: _________
```

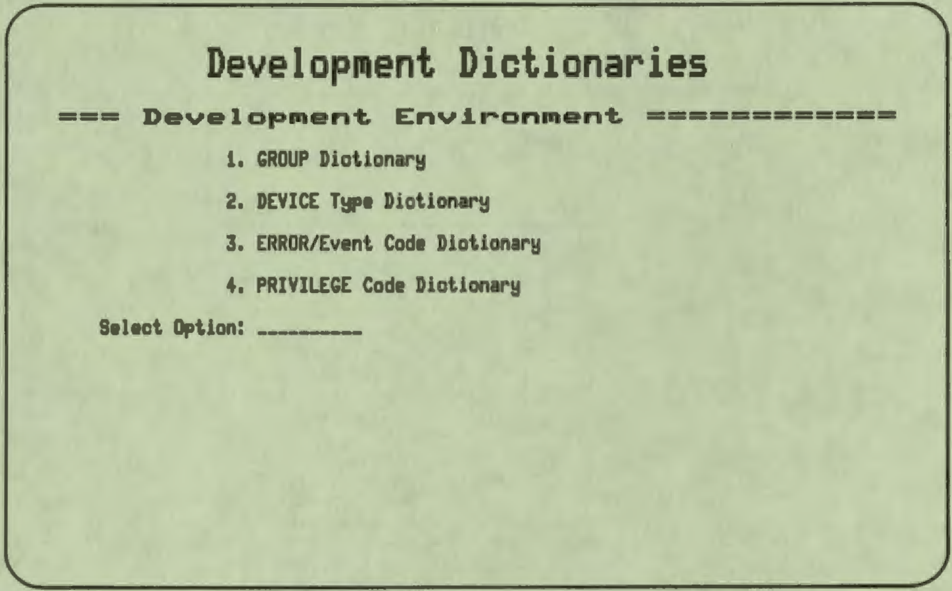
This chapter describes each option in detail. See the *DASL Programmer's Guide* for a discussion of using programmer utilities in the DASL software.

Development Dictionaries Option

The Development Dictionaries Menu contains several dictionaries where you can define groups, device types, privilege codes, or error/event messages for your application.

Figure 5–2 shows the options of the Development Dictionaries Menu.

Figure 5–2 Development Dictionaries Menu



```
                Development Dictionaries
=== Development Environment ============
                1. GROUP Dictionary

                2. DEVICE Type Dictionary

                3. ERROR/Event Code Dictionary

                4. PRIVILEGE Code Dictionary

   Select Option: _________
```

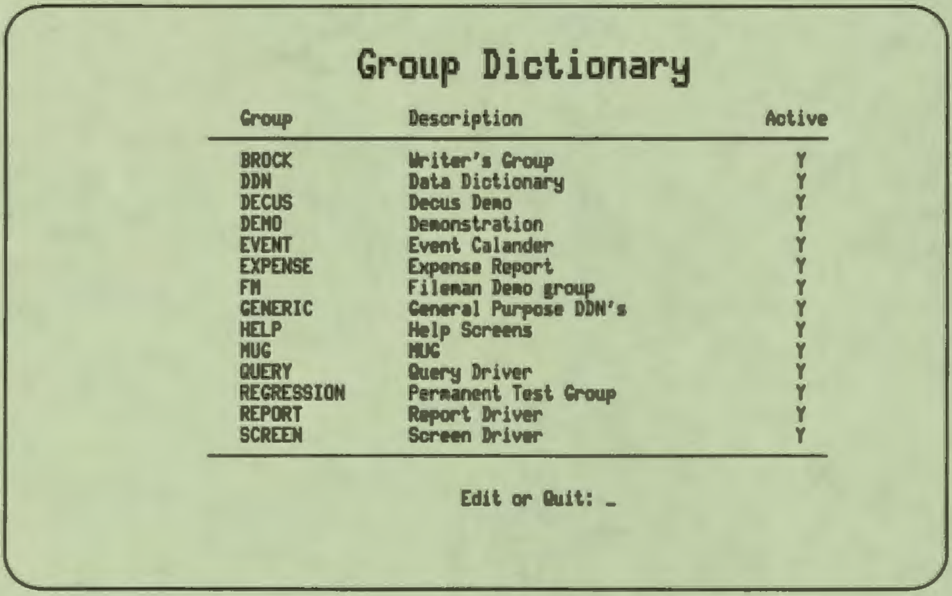
The following sections describe each option of the Development Dictionaries Menu.

Group Dictionary Option

Groups are keywords you assign to data names, tables, screens, and reports. Each group name identifies the data names, tables, screens, and reports to which it is assigned as part of a functional unit (for example, an application or application module). You use the Group Dictionary Option to add to, change, and activate or deactivate application groups.

Figure 5–3 shows a sample of the screen you see when you choose the Group Dictionary Option.

Figure 5–3 Group Dictionary Screen



```
                    Group Dictionary

   Group           Description                       Active
   ---------------------------------------------------------
   BROCK           Writer's Group                       Y
   DDN             Data Dictionary                      Y
   DECUS           Decus Demo                           Y
   DEMO            Demonstration                        Y
   EVENT           Event Calander                       Y
   EXPENSE         Expense Report                       Y
   FM              Fileman Demo group                   Y
   GENERIC         General Purpose DDN's                Y
   HELP            Help Screens                         Y
   MUG             MUG                                  Y
   QUERY           Query Driver                         Y
   REGRESSION      Permanent Test Group                 Y
   REPORT          Report Driver                        Y
   SCREEN          Screen Driver                        Y
   ---------------------------------------------------------

                       Edit or Quit: _
```

Explanation

When you choose the Group Dictionary Option, you see a scroll region and an "Edit or Quit:" decision prompt. To edit, enter E. The DASL software moves you to the top of the scroll region. You can then perform either of the following tasks:

- Create new groups.
- Change existing groups.

Creating New Groups

To create a new group, create a new line at any point you desire in the scroll region and enter a group name and description. The group name must be unique and can be up to 10 characters long. The description can be up to 30 characters long.

The DASL software automatically assigns an active status to any new group you create. That is, the DASL software sets the active flag to Y.

Enter S at the "Save, Edit, or Quit:" prompt to save the new definition.

Changing Existing Groups

To change an existing group, use the scroll-region editing keys to move the cursor to the group. Then, move to the scroll-region field you want to change. You cannot change the group name, but you can change the description or the activity flag by entering the new information over the existing information.

To delete a group, enter D in the first scroll field, next to the group you want to delete. The DASL software then displays the following prompt:

```
Deleting a Group can be dangerous. Enter D again to delete.
```

Enter D again and press Return to delete the group.

To deactivate the group, enter N over the Y in the Active Flag field.

> **Note:** When you delete a group, you can no longer assign DASL data names, screens, or reports to that group. You can no longer use the group name as a wild card in print or compile operations.

How to Use DASL Groups

Any DASL data name, table, screen, or report can belong to one or several groups. You can specify a DASL group to perform the following operations for all entities within that group:

- Compile by group
- Save by group
- Print by group

Related Sections

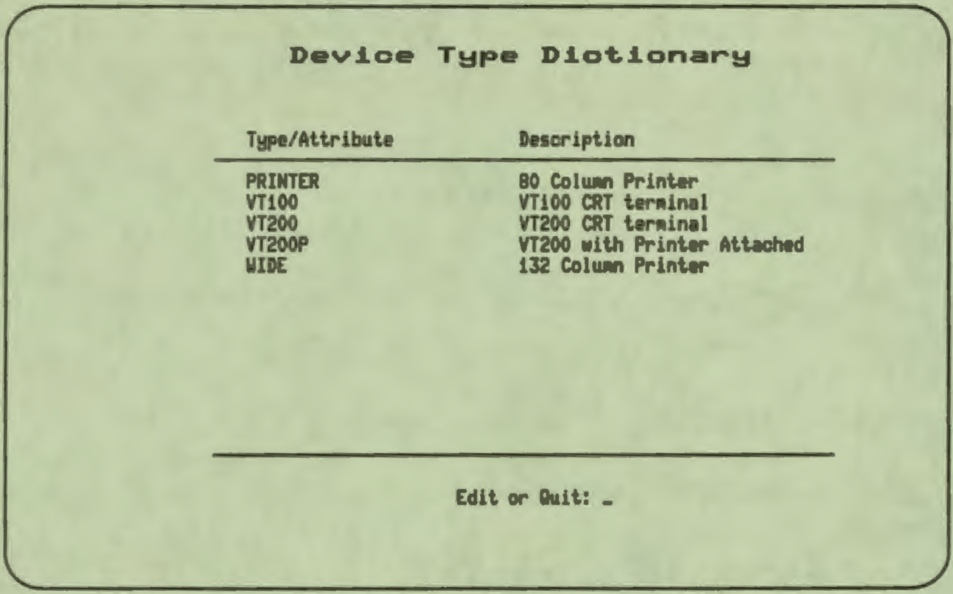
Define Data Names Option, Chapter 1
Define Data Screens Option, Chapter 2
Define Reports Option, Chapter 3
Define Tables Option, Chapter 4
Groups, Chapter 1

Device Type Dictionary Option

You use the Device Type Dictionary Option to define the attributes you assign to devices in the Device Dictionary of the Security System.

Figure 5–4 shows a sample Device Type Dictionary screen.

Figure 5–4 Device Type Dictionary Screen



```
                    Device Type Diotionary

        Type/Attribute          Description
        ------------------------------------------------
        PRINTER                 80 Column Printer
        VT100                   VT100 CRT terminal
        VT200                   VT200 CRT terminal
        VT200P                  VT200 with Printer Attached
        WIDE                    132 Column Printer

        ------------------------------------------------

                         Edit or Quit: _
```

Explanation

When you first choose the Device Type Dictionary Option, you see a scroll region with several listed device attributes and an "Edit or Quit:" decision prompt. The device attributes listed in the scroll region are defined with the DASL software. They are the basic device types that all applications use. Table 5–1 lists these basic devices.

Table 5–1 Basic DASL Device Attributes

Attribute	Meaning
PRINTER	The device is an 80-column hardcopy printer.
VT100	The device is a VT100 terminal.
VT200	The device is a VT200-series terminal.
VT200P	The device is a VT200-series terminal with an attached printer (such as an LA50).
WIDE	The device is a 132-column hardcopy printer.

As these attributes specify basic devices that any application can use, do not delete these attributes. However, you need to define other attributes to describe special-purpose devices attached to your system.

To define new attributes, enter E after the "Edit or Quit:" decision prompt. The cursor then moves to the top of the scroll region. You can add new attributes.

For each attribute, you supply an attribute name and description. The name can be up to 10 characters. The description can be up to 30 characters. When you finish, you save your changes in response to the "Save, Edit, or Quit:" decision prompt.

> **Note:** The DASL software does not accept device attributes in the Device Dictionary that you have not defined through the Device Type Dictionary Option. If you enter an undefined device attribute in response to the "Device Attribute:" prompt in the Device Dictionary of the Security System, the DASL software returns an error.

Related Sections

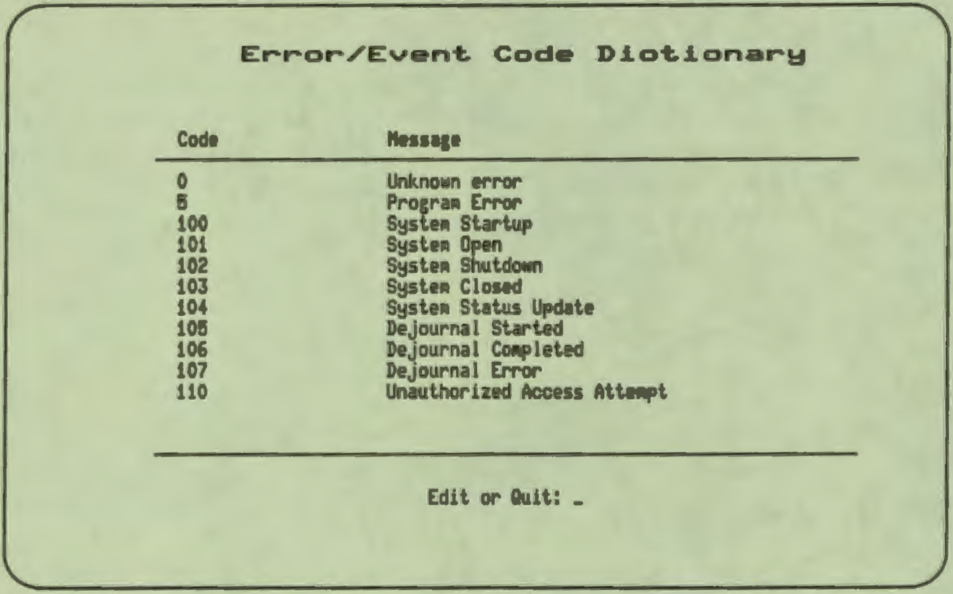
Device Attributes, Chapter 6
Device Dictionary Option, Chapter 6
Security System Option, Chapter 6

Error/Event Code Dictionary Option

You use the Error/Event Code Dictionary Option to define the error or event messages that you want the Security System's error and event logging facility to register in the System Event Log.

Figure 5–5 shows the screen you see when you choose the Error/Event Code Dictionary Option.

Figure 5–5 Error/Event Code Dictionary Screen



```
                Error/Event Code Diotionary

     Code              Message
     ------------------------------------------------
     0                 Unknown error
     5                 Program Error
     100               System Startup
     101               System Open
     102               System Shutdown
     103               System Closed
     104               System Status Update
     105               Dejournal Started
     106               Dejournal Completed
     107               Dejournal Error
     110               Unauthorized Access Attempt

     ------------------------------------------------

                      Edit or Quit: _
```

Explanation

When you first choose the Error/Event Code Dictionary Option, you see a scroll region with several listed error or event codes and an "Edit or Quit:" decision prompt. The error and event codes listed in the scroll region are defined with the DASL software. They are the basic codes that all applications use. Table 5–2 lists these basic codes.

Table 5–2 Basic DASL Error and Event Codes

Code	Meaning
Code:	0
Message:	Unknown Error
Meaning:	The DASL software cannot explicitly identify this error; usually an error in a compiled screen or report routine.
Code:	5
Message:	Program Error
Meaning:	The error is a VAX DSM software error in a compiled screen or report routine. The System Event Log entry for the error contains the VAX DSM error message explaining the error. See the *VAX DSM Programmer's Guide* for details about the full messages that accompany program errors.
Code:	100
Message:	System Startup
Meaning:	The application was started at the time specified in the message.
Code:	101
Message:	System Open
Meaning:	The application is available for general use. This message appears in conjunction with the system startup message (100) and is entered into the log when you run the command procedure that starts your application.
Code:	102
Message:	System Shutdown
Meaning:	The application was shut down at the specified time. This message is entered into the System Event Log when you run the command procedure that shuts down your application.
Code:	103
Message:	System Closed
Meaning:	The application is not available for general use. This message usually appears in conjunction with the system shutdown message (102), and is entered into the log when you close your application through the System Control Option of the Application Environment Menu or when you run a site-specific command procedure to close the application.

Table 5–2 (Cont.) Basic DASL Error and Event Codes

Code	Meaning
Code:	104
Message:	System Status Update
Meaning:	You or a privileged application user has used the System Control Option to change the access status of the system.
Code:	105
Message:	Dejournal Started
Meaning:	A dejournaling operation was started at the specified time. This message is entered into the log when a site-specific dejournaling procedure is run. See the *VAX DSM Installation and Management Guide* for an explanation of how to write dejournaling procedures.
Code:	106
Message:	Dejournal Completed
Meaning:	A dejournaling operation was completed at the specified time. This message is entered into the System Event Log when the your site-specific dejournaling procedure completes running. The full message confirms the number of records restored.
Code:	107
Message:	Dejournal Error
Meaning:	An error occurred while restoring records from the journal file. This message is entered into the System Event Log during the execution of your site-specific dejournaling procedure.
Code:	110
Message:	Unauthorized Access Attempt
Meaning:	A user attempted to log in to your application with an unrecognized User ID or password. The full event message in the System Event Log identifies the User ID that was entered and the device on which the login was attempted.

Codes 0–99 indicate individual errors, while codes 100–199 indicate system events. As these codes specify basic errors or events that any application encounters, do not delete them. You can, however, define other codes to describe errors or events specific to your system. Applications can use codes 200–999 for application-specific errors and events.

To define new codes, enter E after the "Edit or Quit:" decision prompt. The cursor then moves to the top of the scroll region. You can now add new codes.

For each code, you supply a name and an error or event message. The name can be up to 10 characters. Although the codes supplied with the DASL software are numeric, you can specify codes that are alphanumeric.

The error or event message is the text the DASL software writes with the error to the System Event Log. The error or event message can be up to 45 characters. When you finish, save your changes in response to the "Save, Edit, or Quit:" decision prompt.

> **Note:** The DASL software does not register any error or event in the System Event Log that you have not defined in the Error/Event Code Dictionary.

Using User-Defined Error and Event Codes

To use the error and event codes that you have defined for your application, you must do the following:

1. Set an error trap to trap for the application-specific error or event.
2. In the error trap, set the variable %ERR to the error number, and then run the LOG^%DAERRT routine.

For example, when a user makes three unsuccessful attempts to enter a correct User ID or password in the login screen, the DASL software enters an event message (110) in the Error/Event Log. The error trap for this error contains the following MUMPS code:

```
S %ERR="110\User ID: "_USRID,%FLGN=0 S %USR="" D LOG^%DAERRT
```

The Error/Event Code Report for this error contains the error code and description, as well as the User ID of the user who made the error, as follows:

```
                               Introduction to
                                      DASL

                         Demonstration System Event Log
                      Log File Created on 7-May-90    3:30

Demonstration System  7-MAY-1990 15:30:44  Event 110: Unauthorized Access Attempt
        User ID: KSDKF
        Device: RTA14: Option: Login
```

Related Sections

Dejournaling (*VAX DSM Installation and Management Guide*)
Event Logging Option, Chapter 6
Journaling (*VAX DSM Installation and Management Guide*)
Print System Event Log Option, Chapter 6
Security System Option, Chapter 6
System Control Option, Chapter 6
VAX DSM Errors (*VAX DSM Programmer's Guide*)

Privilege Code Dictionary Option

You use the Privilege Code Dictionary Option to define the privilege codes you assign to classifications in the Classification Dictionary, devices in the Device Dictionary, and system statuses in the System Status Dictionary of the Security System. You also assign privilege codes to option screens and database tables.

Explanation

Privilege codes are the basic element in the DASL security system. Developers assign privilege codes to option screens and database tables. Application users then build the security dictionaries (Device Dictionary, Classification Dictionary, and User Dictionary) based on the privilege codes.

To use privilege codes, follow these steps:

1. Associate the privileges with options through the Define Option Screens Option of the Screen Driver, and tables through the Define Tables Option of the Query Driver.
2. Associate the privileges with devices through the Device Dictionary Option of the Security System.
3. Group the privileges into classifications through the Classification Dictionary Option of the Security System.
4. Assign the classifications to users through the User Dictionary Option of the Security System.

You also assign privilege codes to system statuses through the System Status Dictionary Option of the Security System.

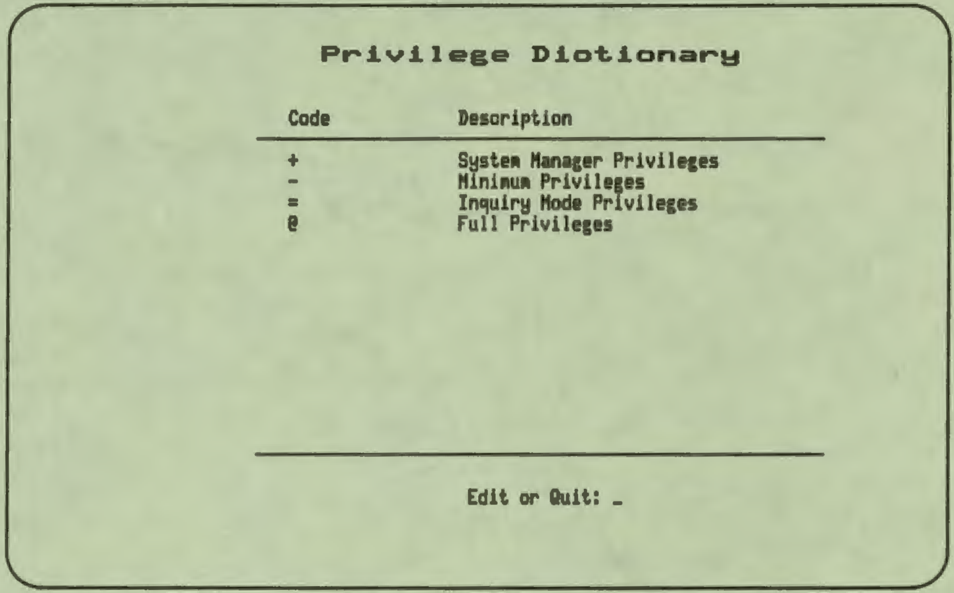
See the "Security System Option" section of Chapter 6 for a discussion of the DASL security system and how privileges interact to allow access to DASL options.

Editing the Privilege Code Dictionary

When you first choose the Privilege Code Dictionary Option, you see a scroll region with several listed privilege codes and an "Edit or Quit:" decision prompt.

Figure 5–6 shows the screen you see when you choose the Privilege Code Dictionary Option. The privilege codes in this screen are basic privilege codes provided by the DASL software.

Figure 5–6 Privilege Code Dictionary Screen



```
                    Privilege Dictionary

         Code                   Description
         ------------------------------------------------
          +                     System Manager Privileges
          -                     Minimum Privileges
          =                     Inquiry Mode Privileges
          @                     Full Privileges




         ------------------------------------------------

                             Edit or Quit: _
```

As these privileges specify basic functions that any application may need, do not delete these privileges. You can, however, define other privileges for your application system.

To define new privileges, enter E after the "Edit or Quit:" decision prompt. The cursor then moves to the top of the scroll region. You can now add new privileges.

For each privilege, you supply a code and a description. The code must be a single alphanumeric character. The code cannot be a punctuation character. The description can be up to 35 characters. When you finish, save your changes in response to the "Save, Edit, or Quit:" decision prompt.

> **Note:** The DASL software does not accept privilege codes in the Classification Dictionary that you have not defined through the Privilege Code Dictionary Option. If you enter an undefined privilege in the Classification Dictionary of the Security System, the DASL software returns an error.

Related Sections

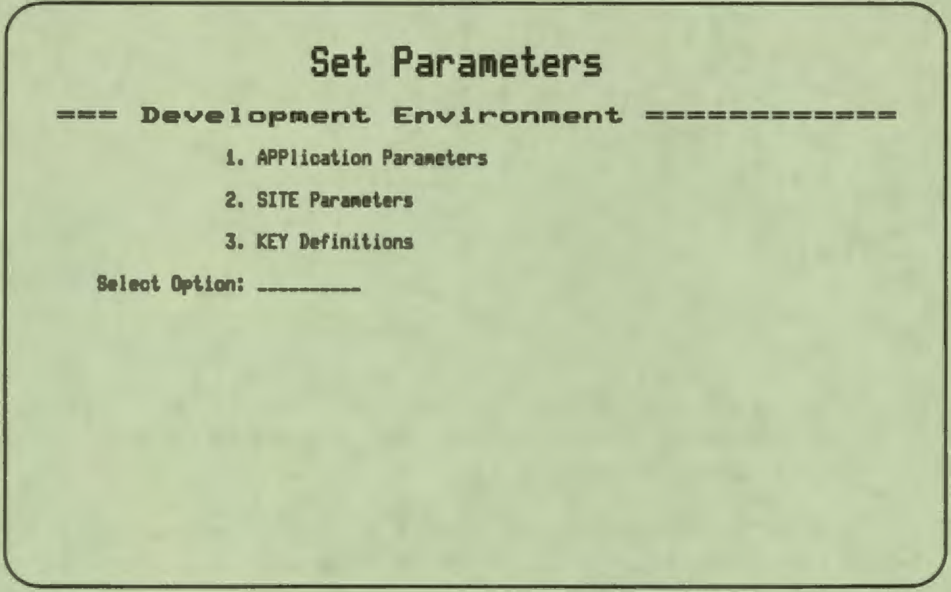
Classification Dictionary Option, Chapter 6
Device Dictionary Option, Chapter 6
Security System Option, Chapter 6
User Dictionary Option, Chapter 6

Set Parameters Option

The Set Parameters Option is a menu of options that you can use to set system parameters for your application, set site-specific parameters, and redefine function keys for your application.

Figure 5–7 shows the options of the Set Parameters Menu.

Figure 5–7 Set Parameters Menu



```
                    Set Parameters
=== Development Environment ============
              1. APPlication Parameters
              2. SITE Parameters
              3. KEY Definitions
     Select Option: _________
```

As Figure 5–7 shows, the Set Parameters Menu has the following options:

- Application Parameters

  You use the Application Parameters Option to set system parameters that describe basic operating conditions under which your application operates.

- Site Parameters

  You use the Site Parameters Option to set parameters, such as date and time output format, which are relevant to a specific application site.

- Key Definitions

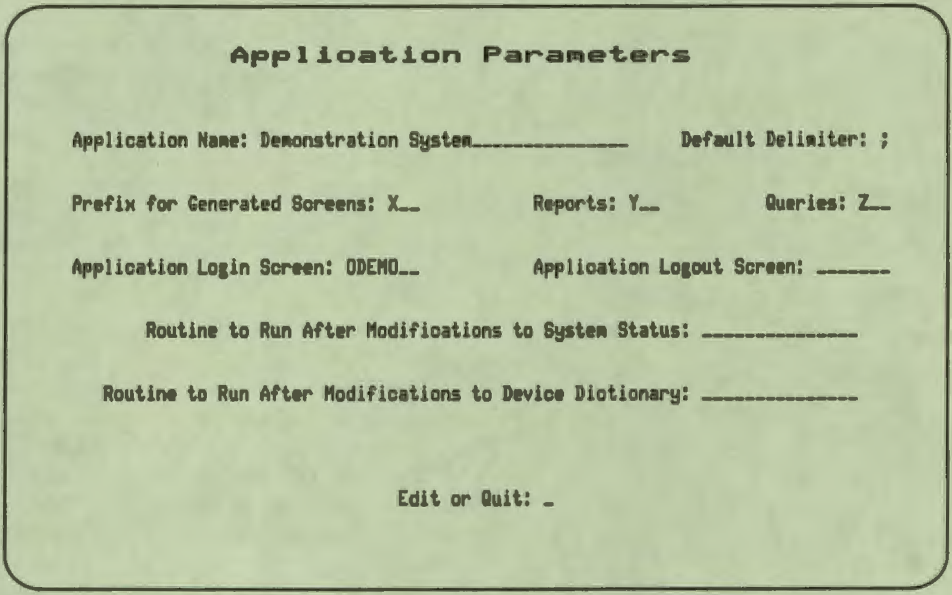
  You use the Key Definitions Option to define function keys for your application or for the DASL software that replace the function keys supported by the DASL software.

Application Parameters Option

You use the Application Parameters Option to set certain basic operating characteristics of your applications. The operating characteristics set by application parameters apply to all sites that use your applications.

Figure 5–8 shows a sample of the screen you see when you choose the Application Parameters Option.

Figure 5–8 Application Parameters Screen



```
                      Application Parameters

  Application Name: Demonstration System______________    Default Delimiter: ;

  Prefix for Generated Screens: X__          Reports: Y__          Queries: Z__

  Application Login Screen: ODEMO__          Application Logout Screen: ______

          Routine to Run After Modifications to System Status: ______________

       Routine to Run After Modifications to Device Dictionary: ______________


                               Edit or Quit: _
```

Explanation

Table 5–3 lists the parameters you can set through the Application Parameters Option and outlines their meanings.

Table 5–3 Application Parameters

Parameter	Explanation
Application name	The name of your application. The name you specify can be up to 20 characters.
Default delimiter	The single punctuation character DASL uses to separate values for dependent data names stored in the database. You can use any punctuation character.

Table 5–3 (Cont.) Application Parameters

Parameter	Explanation
Prefix for generated screens	The prefix the DASL software assigns to the randomly-generated names for the VAX DSM routines compiled from screen specifications. DASL prefixes screen-routine names with X. You can enter up to three alphabetic characters that you want to use as the prefix for screen routines.
Prefix for generated reports	The prefix the DASL software assigns to the randomly-generated names for the VAX DSM routines compiled from report specifications. DASL prefixes report-routine names with Y. You can enter up to three alphabetic characters you want to use as a report routine prefix.
Prefix for generated queries	The prefix the DASL software assigns to the randomly-generated names for the VAX DSM routines compiled from query specifications. DASL prefixes query-routine names with Z. You can enter up to three alphabetic characters to use as a query routine prefix.
Application login screen	The name of the screen you want to use as the application login screen. When you create a DASL application, you normally need to specify a login screen on which application users enter a user name and password. A login screen lets you use the DASL Security System to control application access. The name you specify must follow the conventions for DASL screen names and must refer to a defined data screen.
Application logout screen	The name of the screen to use as the application logout screen (the screen that users see after exiting from the application). The logout screen does not have to be a data screen.
System status Modification routine	The name of a VAX DSM routine to trigger whenever users modify the system status to open or close the DASL application. The DASL software does not provide the trigger routine.
Device Dictionary Modification routine	The name of a VAX DSM routine to trigger whenever users modify the Device Dictionary to add new application-recognized devices or change the characteristics of existing devices. The DASL software does not provide the trigger routine.

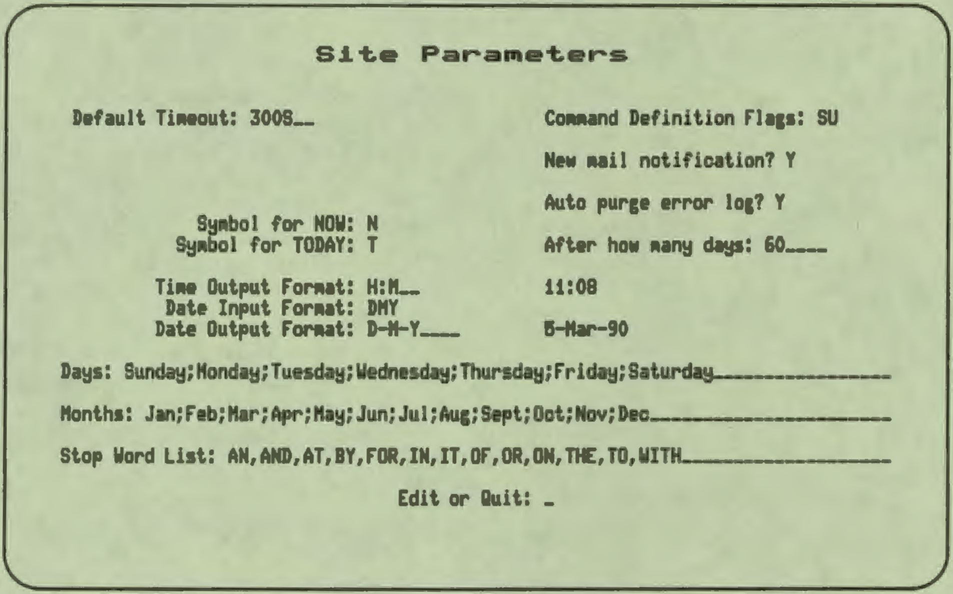
Note: You must set application parameter values before you load the DASL demonstration system or before you build your application.

Site Parameters Option

You use the Site Parameters Option to set basic operating characteristics of your applications. Site parameters can be different for specific sites that use your applications.

Figure 5–9 shows a sample of the screen you see when you choose the Site Parameters Option.

Figure 5–9 Site Parameters Screen



```
                              Site Parameters

   Default Timeout: 300S__                       Command Definition Flags: SU

                                                 New mail notification? Y

                                                 Auto purge error log? Y
              Symbol for NOW: N
            Symbol for TODAY: T                  After how many days: 60____

          Time Output Format: H:M__              11:08
           Date Input Format: DMY
          Date Output Format: D-M-Y____          5-Mar-90

  Days: Sunday;Monday;Tuesday;Wednesday;Thursday;Friday;Saturday_______________

  Months: Jan;Feb;Mar;Apr;May;Jun;Jul;Aug;Sept;Oct;Nov;Dec_____________________

  Stop Word List: AN,AND,AT,BY,FOR,IN,IT,OF,OR,ON,THE,TO,WITH__________________

                              Edit or Quit: _
```

Explanation

As Figure 5–9 shows, the Site Parameters Option provides a number of defaults for basic operating parameters. These default values are displayed in the fields when you first choose the option.

Table 5–4 lists the parameters you can set through the Site Parameters Option, outlines their meanings, and notes any default values supplied.

Table 5–4 Site Parameters

Parameter	Default	Explanation
Default timeout	300	The number of seconds the DASL software is to wait for input before transferring control to the currently defined exit field.
Command definition	SU	Two characters that specify whether users can create systemwide commands and user-specific commands. By default, the DASL software allows the creation of systemwide (S) and user-specific (U) commands. You can change this default to prohibit the creation of systemwide commands or the creation of user-specific commands. Enter S (system) to allow only systemwide commands. Enter U to allow only user-specific commands.
New mail notification	N	A Y or an N that specifies whether the DASL software notifies application users that they have received a mail message during an active session. Normally, the DASL software notifies users of incoming mail at login. If you specify Y for new mail notification, the DASL software also notifies users of incoming mail when they access any option screen during an active session.
Auto purge error log	N	A Y or an N that specifies whether the DASL software automatically purges the error log. Normally, the DASL software purges the error log whenever you use the Purge Error Log Option of the Error/Event Log Menu in the Application Environment. If you specify Y to automatically purge the error log, the DASL software prompts you for the number of days for which you want to keep error log records.
After how many days	None	The number of days for which you want to keep records when the DASL software purges the error log. For example, if you enter 20 at this field, the DASL software purges errors that occurred more than 20 days before the current date.
NOW symbol	N	The single letter users can enter at TIME-validated fields to cause the DASL software to display the current time. Normally, application users can enter N to specify the current time.
TODAY symbol	T (Today)	The single character application users can enter to display the current date.

Table 5-4 (Cont.) Site Parameters

Parameter	Default	Explanation
Time Output	H:M A	The format for displaying times of day. Normally, the DASL software displays times in North American format, set as H(ours):M(inutes) A. You can, however, change the time format.
Input Date	MDY	The format the DASL software uses to redisplay dates that application users enter in DATE-validated fields. Normally, the DASL software uses the North American method of redisplaying dates. You can change the default input display to other standards. Enter the letters MDY with no intervening spaces in the order you want months, days, and years displayed.
Output Date	D-M-Y	The format in which the DASL software displays data values retrieved from the application database. The DASL software normally uses the North American date format. You can change that default by entering M, D, and Y with intervening dashes in the order you want months, days, and years displayed. You can also enter W, N, CY, S, or Z.
Day names	English names	The names of the days of the week. The DASL software normally uses the English names for days. You can change from the default English by entering the day names you want separated by semicolons and no spaces.
Month names	English names	The names of the month. By default, the DASL software displays months by their English names. You can change this default by entering the month names you want separated by semicolons.
Stop Word List	None	A list of words, usually common articles and prepositions, that the DASL software checks when writing data values to a KWIC-type cross-reference global.

You do not have to set site parameter values before you create your application. You can change these values at any time. You do, however, have to set these values before users enter real data in the application.

Related Sections

Application Parameters Option
Cross References, Chapter 1
DATE, Chapter 2
TIME, Chapter 2

Key Definitions Option

You use the Key Definitions Option to define function keys for your application or for the DASL software that replace the function keys supported by the DASL software.

Explanation

The DASL software supports a number of function keys that you use on VT100™, VT200-series, and VT300™-series terminals to enter and edit data. Appendix C shows the keys defined by the DASL software and the functions they perform.

To redefine the DASL function keys, use the Key Definitions Option. You can also disable function keys using this option.

You can use the Key Definitions Option in the development area or in the customization area. Define function keys for your application in the development area. To define function keys for the DASL software, use the Key Definitions Option in the customization area.

After you have defined keys and saved the changes, you must recompile all screens that are affected by the changes. You must recompile application screens in the development area, and DASL screens in the customization area.

> **Note:** If you want to include the DASL Application Environment screens in your application, and you have redefined function keys, you must recompile the DASL Application Environment screens in the customization area.

Function Keys Supported by the DASL Software

On VT200-series and VT300-series terminals, you can use the following keys as functions keys:

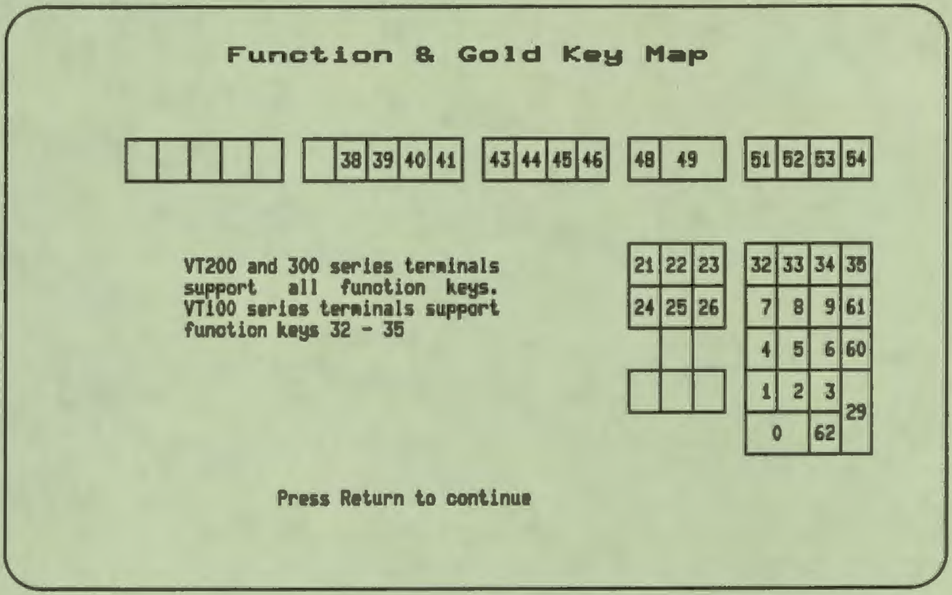
- Top row function keys
- Editing keypad
- Auxiliary keypad

On VT100 terminals, use only the four top keys on the auxiliary keypad as function keys.

In addition, you can use the GOLD key or PF1 key on the auxiliary keypad in conjunction with another key on the main keypad to perform some functions on all VT-type terminals.

Each function key supported by the DASL software has an assigned number for key definition purposes. Figure 5–10 shows the function keys supported by the DASL software and their assigned numbers.

Figure 5–10 Function and Gold Key Map



Function & Gold Key Map

VT200 and 300 series terminals support all function keys. VT100 series terminals support function keys 32 - 35

Press Return to continue

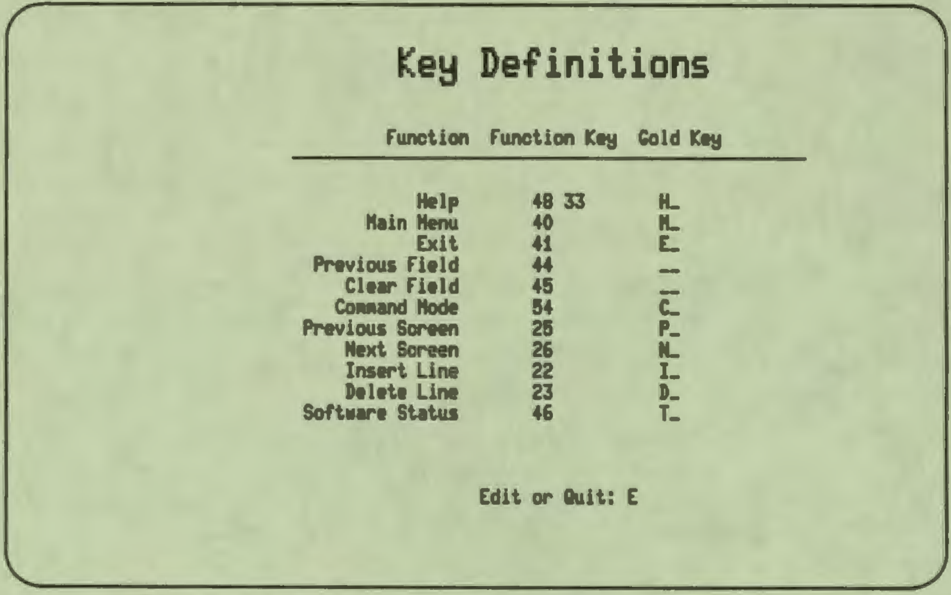
The DASL software displays the Function and Gold Key Map on your terminal screen when you press Help twice at the "Edit or Quit:" prompt in the Key Definitions Screen.

You can also use the Print Key Definitions Option of the Print Development Reports Menu to print a report listing the current key definitions.

Key Definitions Screen

When you select the Key Definitions Option, the DASL software displays the screen shown in Figure 5–11.

Figure 5–11 Key Definitions Screen



```
                        Key Definitions

            Function  Function Key  Gold Key
         ____________________________________________

                  Help     48 33         H_
             Main Menu     40            M_
                  Exit     41            E_
        Previous Field     44            __
           Clear Field     45            __
          Command Mode     54            C_
       Previous Screen     25            P_
           Next Screen     26            N_
           Insert Line     22            I_
           Delete Line     23            D_
       Software Status     46            T_


                     Edit or Quit: E
```

In this screen, the DASL software displays the information described in Table 5–5.

Table 5–5 Key Definitions Information

Item	Description
Function	Brief description of the function that the key performs.
Function Key	Number, from the Function and Gold Key Map, of the key that you press to perform the described function. Notice that the Help function can be associated with two function keys, and a GOLD key sequence.
Gold Key	Name of the key on the main keyboard that you press in conjunction with the GOLD or PF1 key to perform the described function.

When you first select the Key Definitions Option, the DASL software displays the current function keys and their corresponding numbers and characters in the Key Definitions screen. You can edit the numbers and characters in the Function Key and Gold Key fields to redefine the function keys.

At the "Edit or Quit:" prompt, enter E to edit the current definitions, or enter Q to quit the screen without making changes.

Disabling Function Keys

To disable a function, enter a null character for that function in the Function Key and Gold Key fields. For example, you can use the Software Status key while you develop your application. Then, to disable the Software Status key for application users, enter null characters for that function in the Key Definitions screen. After you recompile your application screens, the Software Status function is disabled.

Related Sections

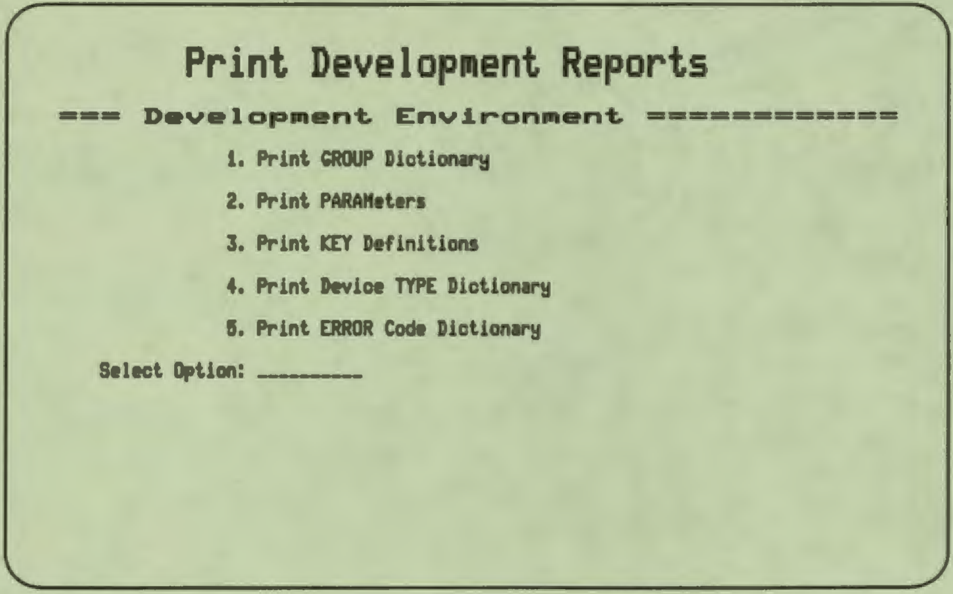
Table C–1, Appendix C
DASL Management Guide
Print Development Reports Option

Print Development Reports Option

You use the Print Development Reports Option to access a menu of options you use to print listings of the Development Environment dictionaries and screens.

Figure 5–12 shows the Print Development Reports Menu.

Figure 5–12 Print Development Reports Menu



```
              Print Development Reports
=== Development Environment ============
                 1. Print GROUP Dictionary
                 2. Print PARAMeters
                 3. Print KEY Definitions
                 4. Print Device TYPE Dictionary
                 5. Print ERROR Code Dictionary
   Select Option: _________
```

Explanation

Each option on the Print Development Reports Menu creates a complete list of the current contents of the dictionary or screen listed in its title. Whenever you choose one of the report options, the DASL software displays the device selection screen that you use to produce the report in your current process or through a batch process.

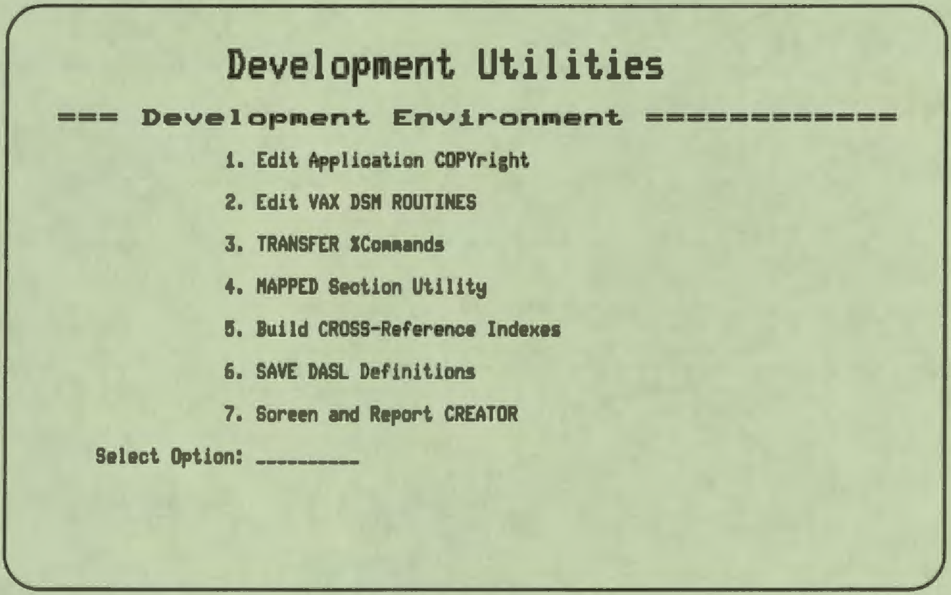
After you select the process to use, you specify the report destination. You can print the report on any VMS device or write the report to a VMS file. If you produce the report through your current process, you can also display the report on your current device.

Development Utilities Option

The Development Utilities Option is a menu of options that you can use to perform programmer tasks when you are developing your DASL application.

Figure 5–13 shows the options of the Development Utilities Menu.

Figure 5–13 Development Utilities Menu



```
                    Development Utilities
  === Development Environment ===========

                    1. Edit Application COPYright

                    2. Edit VAX DSM ROUTINES

                    3. TRANSFER %Commands

                    4. MAPPED Section Utility

                    5. Build CROSS-Reference Indexes

                    6. SAVE DASL Definitions

                    7. Screen and Report CREATOR

        Select Option: _________
```

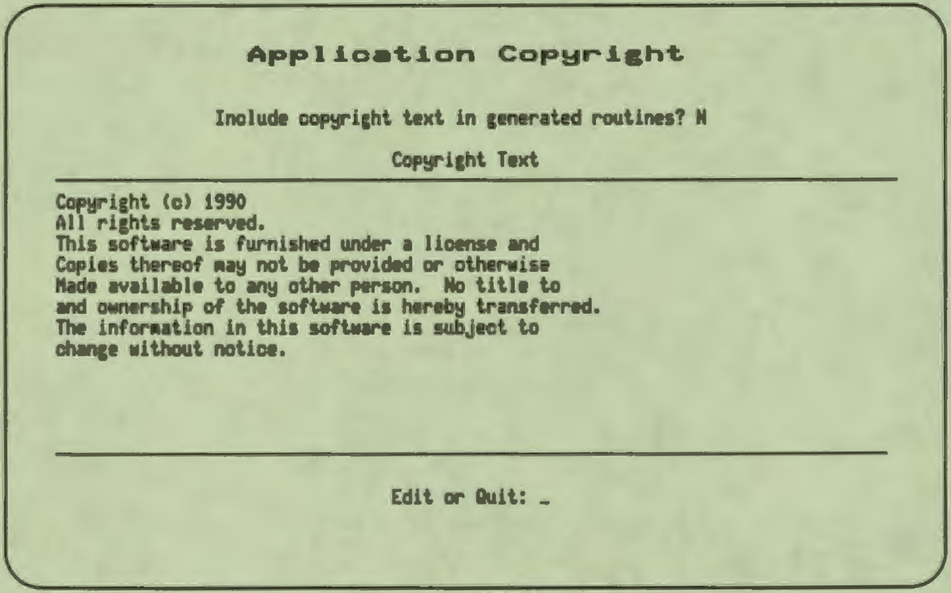
The following sections describe each option of the Development Utilities Menu.

Edit Application Copyright Option

You use the Edit Application Copyright Option to modify the default application copyright supplied with the DASL software. Use this option to include the application copyright as header comments in all the VAX DSM routines generated from your screen and report specifications.

Figure 5–14 shows a sample of the screen you see when you select the Edit Application Copyright Option.

Figure 5–14 Edit Application Copyright Screen



```
                    Application Copyright

         Include copyright text in generated routines? N

                         Copyright Text
____________________________________________________________
Copyright (c) 1990
All rights reserved.
This software is furnished under a license and
Copies thereof may not be provided or otherwise
Made available to any other person.  No title to
and ownership of the software is hereby transferred.
The information in this software is subject to
change without notice.



____________________________________________________________

                         Edit or Quit: _
```

When you choose the Edit Application Copyright Option, you see a scroll region containing the default application copyright and an "Edit or Quit:" decision prompt. To edit, enter E. The DASL software displays the following prompt:

```
Include copyright text in generated routines? N
```

Enter Y to include the copyright as comment lines in all routines compiled from your screen and report specifications. Press Return if you do not want to include the copyright.

The cursor then moves to the scroll region containing the default copyright statement. You can use any of the scroll-region editing keys and techniques to change any part of the default copyright. When you exit from the scroll region, save your edits at the "Save, Edit, or Quit:" decision prompt.

Edit VAX DSM Routines Option

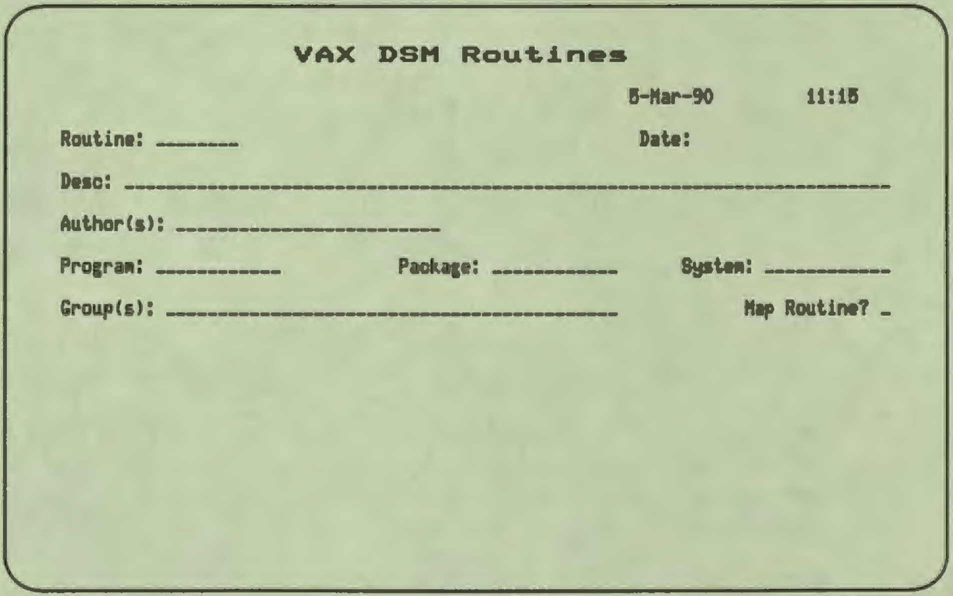
You use the Edit VAX DSM Routines Option to call the EDT editor to create or edit VAX DSM routines from within the DASL software. This utility performs the following tasks:

- Formats the first line of the routine
- Creates a new routine without using the ZSAVE command
- Locks the routine name, preventing two developers from modifying the same routine

Explanation

Figure 5–15 shows the screen you see when you choose the Edit VAX DSM Routines Option.

Figure 5–15 Edit VAX DSM Routines Screen



```
                          VAX DSM Routines

                                             5-Mar-90        11:15

  Routine: ________                          Date:

  Desc: ________________________________________________________________

  Author(s): _______________________

  Program: ___________      Package: ___________     System: ___________

  Group(s): ________________________________________     Map Routine? _
```

As shown in Figure 5–15, you first supply the name of the routine to edit. The name must conform to the *ANSI MUMPS Standard*. That is, the name must consist of one alphabetic character followed by up to seven alphanumeric characters.

> **Note:** Because VAX DSM and the DASL software reserve the percent sign (%) for their own use, you cannot begin routine names with the percent sign.

If the name you specify is new, the DASL software prints the following message at the bottom of the screen:

```
This is a new routine.
```

Table 5–6 describes the routine identification information you specify for each routine.

Table 5–6 Routine Identification Information

Item	Description
Description	A short (no more than 62 characters) description of the routine and its purpose.
Author(s)	The programmers who wrote the routine. If you need to enter more than one name, separate the names with a slash character (/).
Program	The name of the set of routines to which this routine belongs. Program is a term defined in the *ANSI MUMPS Standard*. This field is not required and is used only for identification purposes.
Package	The name of the *subsystem* to which the program belongs. Package is a term defined in the *ANSI MUMPS Standard* and is not a required item.
System	The name of the application to which the package belongs. System is also a term defined in the *ANSI MUMPS Standard* and is not a required item.
Group(s)	Specifies the DASL groups that use the routine. This field is not required.
Map routine	Specifies whether you want the routine mapped. Enter Y to map the routine or N to leave it unmapped.

After you enter the identifying information and save it, the DASL software calls the EDT editor. You can then use EDT to create or edit the routine.

When you finish, press Ctrl/Z, then type EXIT to save the routine or QUIT to delete the routine. Press Return. If you save the routine, the DASL software stores it in a sequential file in your current directory. The name of the file is the name of the routine. The file type is DAT.

When you use the Edit VAX DSM Routines Option to edit an existing routine, the DASL software locks the name of the routine to prevent other developers from editing the routine at the same time.

Related Sections

DO, Chapter 2
Transfer %Commands Option

Transfer %Commands Option

The DASL Screen Driver and Report Driver support direct incorporation of application written subroutines into compiled routines. You can specify a DASL %command for use as a screen action, screen validation, input transform, or output transform.

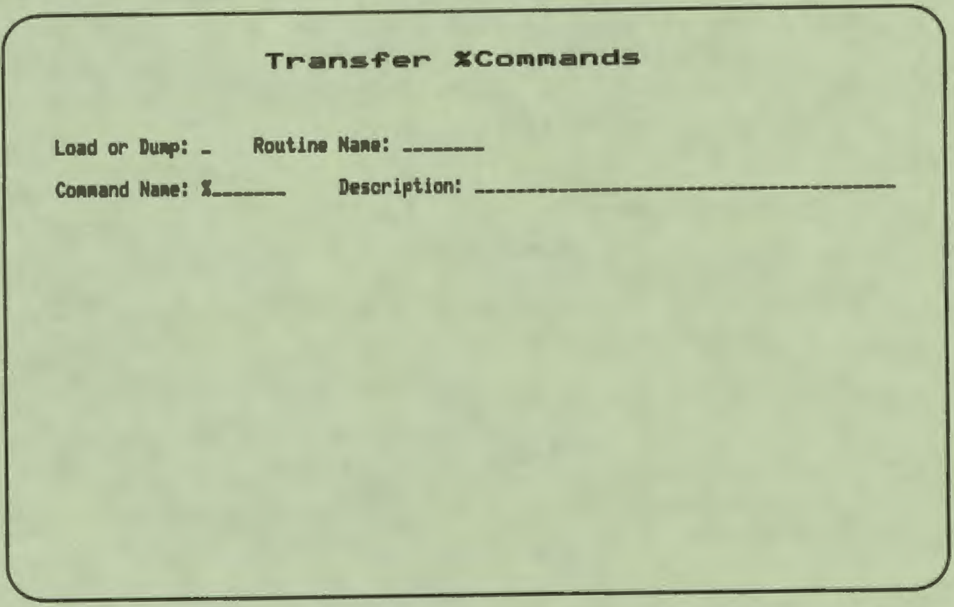
Explanation

The VAX DSM code for %commands is stored in a global variable, and loaded into a screen or report at compile time. You use the Transfer %Commands Option to perform the following tasks:

- Copy VAX DSM routines from your current routine buffer to a DASL storage global and associate a %command name with that routine.
- Copy VAX DSM routines from the DASL storage global to your current routine buffer.

Figure 5–16 shows a sample of the screen you see when you choose the Transfer %Commands Option.

Figure 5–16 Transfer %Commands Screen



```
                    Transfer %Commands

   Load or Dump: _    Routine Name: ________
   Command Name: %_______     Description: ________________________________________
```

When you first choose the Transfer %Commands Option, the DASL software displays the following prompt:

```
Load or Dump:
```

Loading a Routine

Enter L in response to the "Load or Dump:" prompt to transfer a copy of a routine from your current routine directory to the DASL storage global. You then supply the following:

- Routine name
- %Command name
- Description

If you have not previously loaded the routine, the DASL software associates the %command name you supply with the routine. You then use that %command name to call that routine as a DASL action or validation.

Dumping a Routine

Enter D in response to the "Load or Dump:" prompt to copy the routine from the DASL storage global to your current routine directory. You then supply the name under which you want the routine to be stored and the current %command name.

If you already have a routine with the name you specify in your current routine directory, the DASL software asks you whether you want to overwrite that routine. (The overwriting process is identical to the process of saving an already saved routine.) Enter Y if you want to overwrite the routine in the routine directory. Enter N if you do not want to overwrite the routine in the routine directory.

Related Sections

DO, Chapter 2
Routine Directory (*VAX DSM Language Reference Manual*)
Validations, Chapter 2

Mapped Section Utility Option

You use the Mapped Section Utility Option to define a mapped section file for the routines in your current routines directory or volume. You can then use the file with the VAX DSM ^RMBLD Utility to map the routines.

Explanation

When you choose the Mapped Section Utility Option, the DASL software displays a screen that prompts you for an output file name.

Type a valid VMS file name at the "Output File:" prompt and press Return to create a mapped section file.

Mapping Routines

To create a mapped section, follow these steps:

1. Use the Mapped Section Utility to create a file that lists the application routines that you want to map. When you define a screen or report, you indicate whether you want to map the compiled screen or report. The DASL software then flags the routines that you want to map.

   The Mapped Section Utility Option checks the routines for a flag that indicates mapping and creates a file that lists each routine.

2. At the VMS prompt, enter the DSM/MANAGER command to log in to the DSM manager account.
3. At the DSM prompt, enter D ^RMBLD to use the VAX DSM ^RMBLD Utility. The ^%RMBLD Utility builds a mapped section. At the "routine(s)" prompt, use the following syntax to enter the file name that you defined in the Mapped Section Utility screen:

   ```
   routine(s)  <filename
   ```

The ^RMBLD Utility prompts you for other information as well. See the *VAX DSM Installation and Management Guide* for more information about ^RMBLD and about building mapped sections.

Including Your Own Routines in the Mapped Section

You can also flag your own routines for inclusion in the mapped section by using the following DASL programming convention in the second line of the routine:

```
;DASL;Y
```

The second piece following the semicolon delimiter must be DASL, and the third piece must be Y. The third piece corresponds to the value of the mapped section flag. This is the format built by the screen compiler, the report compiler, and the Edit VAX DSM Routines Utility.

A modification to the mapped section flag in screens and reports does not take effect until the routine is generated.

Related Sections

Edit VAX DSM Routines Option
Mapped Sections (*VAX DSM Installation and Management Guide*)

Build Cross-Reference Indexes Option

You use the Build Cross-Reference Indexes Option to generate a routine to rebuild cross-reference indexes.

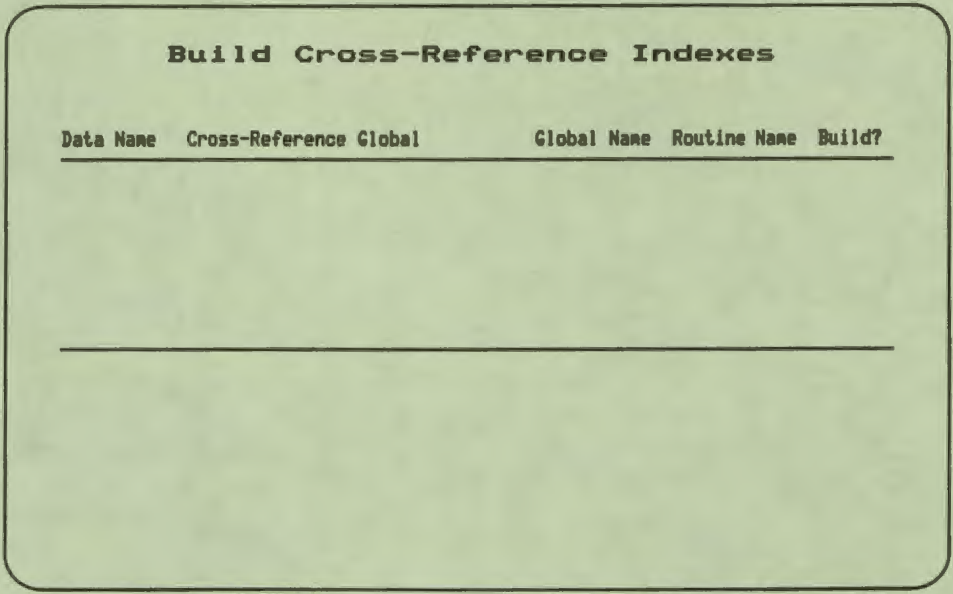
You can rebuild a cross-reference index in the following circumstances:

- To replace a corrupt cross-reference index
- To restructure a cross-reference index
- To build a new cross-reference index for an existing database

Build Cross-Reference Indexes Screen

Figure 5–17 shows a sample of the screen you see when you choose the Build Cross-Reference Indexes Option.

Figure 5–17 Build Cross-Reference Indexes Screen



```
                Build Cross-Reference Indexes

Data Name  Cross-Reference Global        Global Name  Routine Name  Build?
```

Explanation

When you choose the Build Cross-Reference Indexes Option, the DASL software displays a scroll region. You then use the scroll region to list all the data names for which you want to rebuild cross-reference indexes.

When you enter a data name, the DASL software fills in the name of the current cross-reference index under the "Cross-Reference Global" and "Global Name" fields. The DASL software then moves the cursor to the "Global Name" field.

The "Cross-Reference Global" field specifies the name of the current cross-reference index. The "Global Name" field specifies the name of the new cross-reference index to create.

You now have three courses of action you can follow:

- Rebuild the cross-reference index with the same name.
- Recreate the cross-reference index under another name.
- Build the cross-reference routine, but do not run it.

Rebuilding the Cross-Reference Index

To rebuild the cross-reference index, press Return under the "Global Name" field and enter a name to give to the rebuilding routine under the "Routine Name" field. Enter Y under the "Build?" field.

When you specify C at the "Continue, Edit or Quit:" decision prompt, the DASL software generates and runs rebuilding routines with the names you specify. The DASL software also stores the rebuilding routines in your current routine directory.

Because the old and new cross-reference indexes have the same global names, the rebuilding routines delete the old cross-reference index, then build the new cross-reference index with the same name. Your original cross-reference index is lost.

Recreating the Index

You can follow a safer procedure by rebuilding the index under another name so that your original cross-reference index is not immediately lost. To rebuild the cross-reference index under a different name, enter a global name under the "Global Name" field. This is the name the DASL software gives to the recreated cross-reference file. You can enter lowercase characters for the global name and use extended global syntax in this field.

You then follow the procedure outlined in the previous section. Enter a name for the rebuilding routine under the "Routine Name" field, and enter Y under the "Build?" field.

When you specify C at the "Continue, Edit or Quit:" decision prompt, the rebuilding routines create new cross-reference indexes with the names you specify. The rebuilding routines leave the old cross-reference indexes untouched.

After you save the old indexes through the ^%GTO Utility, you can copy the old indexes to the new indexes by using the ^%GC Utility. See the *VAX DSM Language Reference Manual* for more information on the ^%GTO and the ^%GC Utilities.

Building but not Running the Rebuilding Routine

You can check the generated rebuilding routine before you run it. You enter a cross-reference index name (using either of the methods outlined previously) and a name for the routine. However, enter N under the "Build?" field.

The DASL software generates the routine and stores it in the current routine directory, but does not run the routine or rebuild the cross-reference index. You can now inspect the routine in the routine directory. If the routine meets your needs, you can then use the Build Cross-Reference Indexes Utility again, specifying the name of that routine under the "Routine Name" field.

Related Sections

Cross References, Chapter 1
^%GC (*VAX DSM Language Reference Manual*)
^%GTO (*VAX DSM Language Reference Manual*)

Save DASL Definitions Option

You use the Save DASL Definitions Option to store all or selected data name, template, screen, or report definitions in a specified VMS sequential file.

Explanation

When you choose the Save DASL Definitions Option, the DASL software displays the Save DASL Definitions screen. In this screen, enter a descriptive header and destination for the definitions. The destination must be a VMS sequential file. Then, specify the definitions to store.

The DASL software prompts you to specify (by entering Y or N) whether to save each of the following:

- Data names
- Templates
- Tables
- Screens
- Reports
- Queries
- VAX DSM routines
- Global data

For each type of definition you specify, the DASL software displays a scroll region to list the items within that type. Specify the definitions by entering one or more of the following:

- Names
- An asterisk (*) for all definitions
- A partial name followed by an asterisk (*)
- An at sign (@) followed by a defined DASL group name for all definitions in the group
- A range of definitions in the form name1-name2

If you choose a range in the form name1-name2, the DASL software includes all definitions that fall within the range, including the definitions you specified as limits. You can verify the definitions within the range. When DASL displays the "Continue, eXpand wild cards, Edit, or Quit:" decision prompt, enter X. The DASL software then displays all definitions that fall within the range.

The following are exceptions to the list of items that you can enter:

- You cannot specify a group name for a template, query, or global data.
- You can save only system queries; user queries cannot be saved.
- You must specify a global name or global subtree to save global data. You cannot use a wildcard to select global data.

When you finish, the DASL software displays the device selection screen that you use to produce the file in your current process or through a batch process. If you included a large number of definitions, specify a batch process.

Restoring DASL Definitions

After you have saved DASL definitions in a VMS sequential file, you can restore them to your directory by using the DASL ^%DALOAD routine. See Appendix B for a description of the ^%DALOAD routine.

Related Section

Appendix B

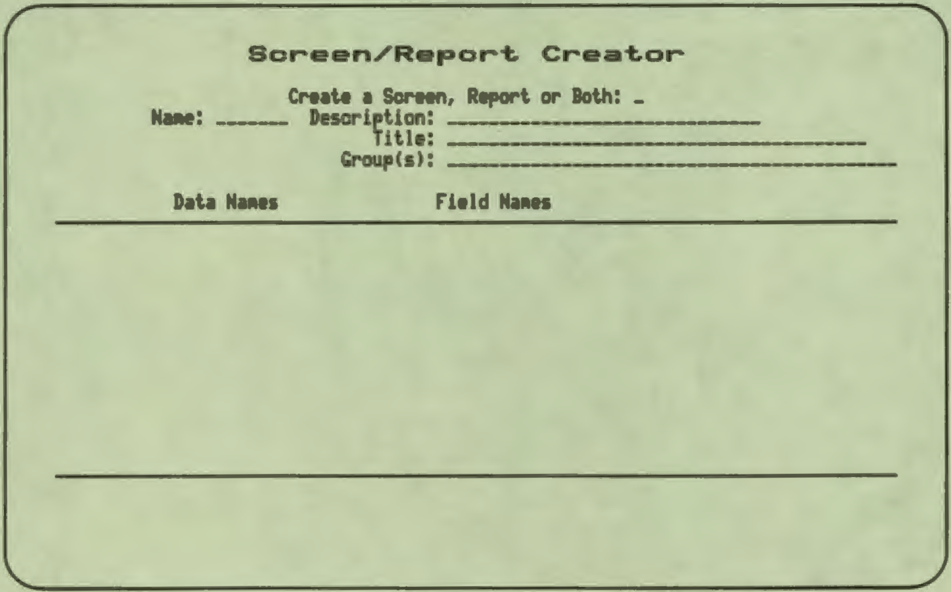
Screen and Report Creator Option

The DASL Screen and Report Creator creates simple data screens and reports that use data names previously defined in the Data Dictionary. You can use the Screen and Report Creator as a starting point to build more complex screens and reports. You may need to modify the screens and reports you create with this option before they compile without errors or run correctly.

Explanation

When you select the Screen and Report Creator Option, the DASL software displays the screen shown in Figure 5–18.

Figure 5–18 Screen and Report Creator Screen



```
                    Screen/Report Creator

                Create a Screen, Report or Both: _
      Name: ______ Description: ___________________________
                         Title: ____________________________________
                      Group(s): _______________________________________

         Data Names                 Field Names
   ______________________________________________________________________
```

As shown in Figure 5–18, the DASL software first displays the “Create a Screen, Report or Both:” prompt.

- Enter S to create a screen only.
- Enter R to create a report only.
- Enter B to create both a screen and a report.

Table 5–7 shows the descriptive information you provide when you create a screen or a report.

Table 5–7 Screen and Report Creator Definition

Item	Description
Name	The name of the screen or report, a standard DASL name made up of one alphabetic character followed by up to six alphanumeric characters.
Description	A text description of the screen or report that the DASL software displays in lookup lists.
Title	The title of the screen or report as it appears on the screen or report.
Group(s)	The names of the DASL groups, from the Group Dictionary Option of the Development Environment, to which the screen or report belongs. The group name is a required item that the DASL software uses to classify screens and reports by application or package.

If the name you enter at the "Name:" prompt is an existing one, the DASL software displays either or both of the following prompts:

```
This screen already exists, do you want to overwrite?
```

```
This report already exists, do you want to overwrite?
```

You can enter Y to overwrite any previously defined screens or reports with the same name.

After you have entered the information described in Table 5–7, the DASL software displays a scroll region in which you enter the following information:

- Data Names

  In this scroll region, you must enter data names that you have previously defined in the Data Dictionary. Enter data names in the order in which you want them to appear on the screen or report. The DASL software displays the data names in the order in which you list them.

  The DASL software uses information about the data name from the Data Dictionary to determine the data-name prompt, length, and validations.

- Field Names

  When you first enter a data name, that data name appears as the field name in the scroll region. You can override that default if you want field names that are different from the data names. You can edit the field name by entering E at the "Save, Edit or Quit:" prompt that appears after you have entered all data names in the scroll region.

  The DASL software uses the field name as the field name in a screen. You must assign a unique name to each field.

- Counter Data Name

  If the first data name you enter in the scroll region is not a primary key data name, the DASL software prompts you for a counter data name. The DASL software increments the counter data name with the COUNT screen action in the FILE field of the data screen. You must enter a counter data name that you have previously defined in the Data Dictionary.

After you finish entering data names in the scroll region, the DASL software displays the "Save, Edit, or Quit:" prompt.

When you enter S to save the screen, the DASL software displays either or both of the following prompts:

```
Do you want to compile screen?

Do you want to compile report?
```

You can enter Y at these prompts to compile the screen or report that you have created.

Editing Created Screens and Reports

The screens and reports that you create with the Screen and Report Creator can be rough drafts for more detailed screens and reports.

Use the Define Data Screens Option and the Display Designer of the Screen Driver to edit and make additions to the data screens you create. Using the Screen Driver options, you can:

- Use the Display Designer to arrange fields on the screen.
- Use the Define Data Screens Option to check the defaults in each field definition to ensure that the defaults are correct.

  Default items such as Help Text, Prompt, and Validations in the Data Dictionary are specified in the Data Dictionary when the DASL software defines the data name. Check these defaults for appropriateness in each field.
- Use the Define Data Screens Option to edit the validations for the first field of your screen and the EDIT field.

  The Screen and Report Creator creates a field named EDIT if the first field listed is not a primary key. The EDIT field is identical to the first field, using the same data name. For example, if the first field you list is the NAME field for the NAME data name (not a primary key), the DASL software creates an EDIT field identical to the NAME field. You use the EDIT field to edit entries in the NAME field.

  Use the Define Data Screens Option to change the validations in the first field or the EDIT field. The default validations in the first field, if it is a name-type field, usually include LOOKUP and NAME. Do not include a LOOKUP validation in an EDIT field.

- Use the Define Data Screens Option to create scroll regions, if desired. The Screen and Report Creator does not create scroll regions.

You can also use the Define Reports Option of the Report Driver to edit and make additions to the reports you create.

Related Sections

Define Data Names Option, Chapter 1
Define Reports Option, Chapter 3
Display Designer, Chapter 2
Group Dictionary Option

DASL Language Utilities

You can use the DASL Language Utilities to create a DASL application that accepts and displays non-English text.

Explanation

Although the DASL default language is US English, you can use the DASL software to create applications that display text in many Western European languages. Using the DASL software, you can generate applications that display text in languages whose alphabetic characters are included in the DEC Multinational Character Set or the ISO Latin-1 Character Set.

General Issues in Translation

Whenever you create a non-English application, keep the following general issues in mind:

- Expanding the character set

  For many non-English languages, you must expand the character set to include characters whose decimal codes are greater than the ASCII decimal code 127. The DEC Multinational Character Set and the ISO Latin-1 Character Set contain characters with decimal codes of 128 to 255, such as ä or Ç.

- Performing pattern matches

  At present, the MUMPS pattern match validation does not account for characters whose decimal codes are greater than 127. Proponents of the MUMPS language are currently addressing this problem by developing enhancements to the MUMPS language. In the meantime, you can use the DASL software to specify a pattern match for a multinational character. As the DASL software continues to evolve with the MUMPS language, the DASL pattern match solution aids in migration to future MUMPS enhancements.

- Establishing collating order

  The MUMPS language collates characters according to their ASCII decimal code. When you use multinational characters, you must collate data in an order other than the ASCII decimal code to take the multinational characters into account. Furthermore, different languages can require a different collating order for the same character. You can set up the correct collating order for a language that uses multinational characters by defining cross-reference transforms in the DASL Data Dictionary.

See the "Choose Multinational Characters Option" section in this chapter for more information on selecting multinational characters, pattern matching, and collating.

DASL Issues in Translation

Whenever you create a non-English application using the DASL software, keep the following issues in mind:

- Translating DASL messages and prompts

  DASL applications display many standard DASL prompts and messages to end users. You must translate these messages and prompts into the non-English language, and you must recompile all the DASL screens and reports that display the prompts and messages. See the "Translate DASL Messages Option" section of this chapter for more information.

- Translating Application Environment screens and reports

  The DASL Application Environment module contains screens and reports that you can include in your application to provide system security, error and event logging, and report management. To include these screens and reports in your application, you must translate and then compile each screen and report. See "Translating the Application Environment Screens and Reports" in this section for more information.

Preparing for Translation

When you perform some translation tasks, you must modify or recompile DASL system screens and reports. You must modify or recompile DASL system screens and reports when you:

- Select multinational characters for the DASL software and your application.
- Translate and include DASL Application Environment screens and reports in your application.
- Translate standard DASL messages and prompts.

To modify the DASL software, you must work in the customization area (in the DSM manager's UCI). To work in the customization area, enter the following command at the VMS prompt to log in to DSM:

```
$ DSM/DASL_CUSTOMIZE Return
```

Then, log in to the DASL software to perform translation tasks.

If you attempt to access most DASL Language Utilities options from the development area or the application area, the DASL software displays a message directing you to work in the customization area.

Use the customization area to copy and recompile DASL Application Environment screens and reports, and to recompile the DASL software when modifications to the DASL software are necessary.

In addition, if you use the DASL software to create applications in more than one non-English language, you may want to set up a DASL development area for each language.

See the *DASL Management Guide* for information on how to establish and log in to DASL work areas.

Creating a Non-English Application

To create an application that displays non-English text, complete the following application development tasks:

1. Select the multinational characters to include in non-English text; for example, characters such as ø and æ. When you use the Choose Multinational Character Option to select a multinational character, you can also specify a DASL pattern match for the character.
2. Recompile selected DASL screens and reports that display or accept multinational characters. You can use a DASL group name to specify all the required screens and reports. You can then enter the new characters when defining prompts in fields.
3. Specify a language for your application and translate standard DASL prompts and messages into the specified language. Use the Translate DASL Messages Option to specify a language and translate DASL messages and prompts.
4. Choose the current language for your application, and regenerate three language-relevant DASL utility routines. Use the Choose Current Language Option to perform these tasks. The language you choose must be previously defined in the Translate DASL Messages Screen.
5. Recompile DASL screens and reports that display standard messages and prompts. You can use a DASL group name to specify all the required DASL screens and reports.
6. Develop your application. When you compile application screens and reports, the changes you made using the DASL Language Utilities take effect.

In addition to the previous tasks, you can optionally do the following:

- Define data-name templates in the Data Dictionary for language-specific items such as currency and postal codes.
- Define cross-reference transforms for data names in the Data Dictionary.

  You can use cross-reference transforms to specify a language-specific collating order for characters in reports and lookup lists. You can use the MUMPS $TRANSLATE function in cross-reference transforms to solve sorting problems.

See the "Choose Multinational Characters Option" section of this chapter for more information on how to use cross-reference transforms to collate multinational characters.

- Translate each DASL Application Environment screen and report to display non-English text, if you want to include the Application Environment screens and reports in your application.

Translating the Application Environment Menu Screens and Reports

If you want the DASL Application Environment Menu screens and reports that are visible to application users to display non-English text, you must translate each individual screen and report.

For a listing of these screens and reports, see the *DASL Programmer's Guide*.

To redesign each screen and report for a non-English application, take the following steps:

1. In the Group Dictionary of the Development Environment Menu, assign a new group name to the translated screens and reports.
2. Make a copy of each screen and report, giving each screen and report a new name and a new group name.

   Note: You can edit Application Environment screens, reports, and data names directly. However, when you update to a new version of the DASL software, these routines and globals are overwritten. If you edit the original Application Environment screens and reports, save copies of your modified screens and reports before updating the DASL software.

3. Using the new copy, edit the English text that appears in each screen or report:
   - In screens, translate prompts, headers, and help text to the desired non-English text.
   - In reports, translate headers, footers, and other relevant data to the desired non-English text.
4. Wherever a screen or report calls another Application Environment screen or report, correct the reference to indicate the translated screen or report name.
5. In the Site Parameters Option of the Development Environment Menu, enter the correct date and time formats for the non-English text.
6. Recompile the Application Environment screens and reports using the Compile Screens Option of the Screen Driver Menu and the Compile Reports Option of the Report Driver Menu. If you have assigned a group name to the translated screens and reports, you can specify the group when recompiling.

Recompiling Screens and Reports

After you have entered all translation information through the DASL Language Utilities Menu options, you must recompile all application and DASL screens and reports visible to application users.

You can use the Compile Screens Option of the Screen Driver Menu to compile all screens, and the Compile Reports Option of the Report Driver Menu to compile all reports. Use these options in the customization area.

Related Sections

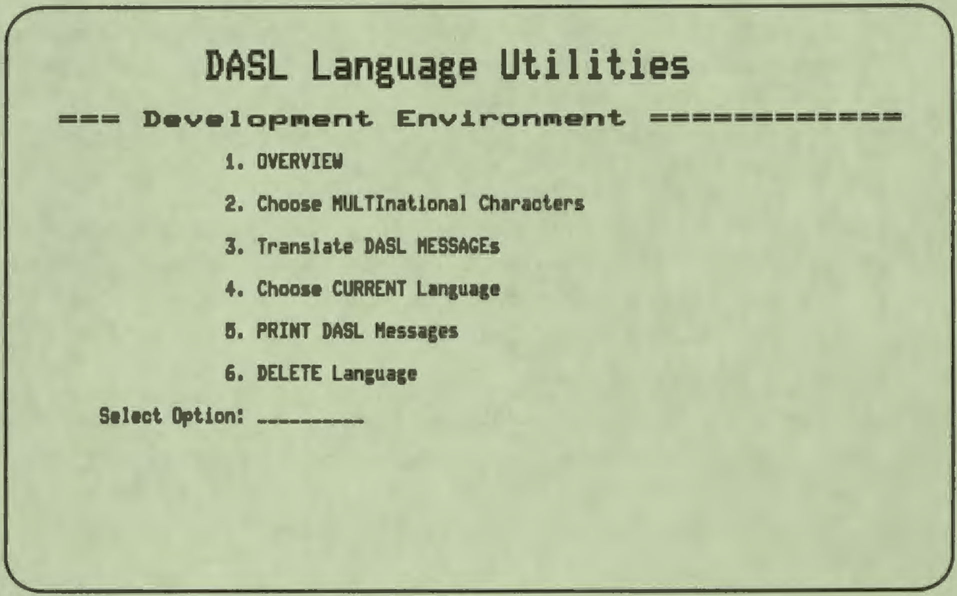
Choose Current Language Option
Choose Multinational Characters Option
Cross References, Chapter 1
DASL Management Guide
DASL Programmer's Guide
Delete Language Option
Group Dictionary Option
Print DASL Messages Option
Site Parameters Option
Translate DASL Messages Option
$TRANSLATE (*VAX DSM Language Reference Manual*)

DASL Language Utilities Menu

You can complete most translation tasks using the DASL Language Utilities Menu options. You must work in the customization area to access most of the language options.

When you select the DASL Language Utilities Menu, you see the screen shown in Figure 5–19.

Figure 5–19 DASL Language Utilities Menu



```
              DASL Language Utilities
=== Development Environment ============
                 1. OVERVIEW
                 2. Choose MULTInational Characters
                 3. Translate DASL MESSAGEs
                 4. Choose CURRENT Language
                 5. PRINT DASL Messages
                 6. DELETE Language
      Select Option: _________
```

The following sections describe the options of the DASL Language Utilities Menu.

Choose Multinational Characters Option

VAX DSM and the DASL software store data as strings of ASCII (American Standard Code for Information Exchange) characters. However, many non-English text applications require characters that are contained in the DEC Multinational Character Set (MCS) or the ISO Latin-1 Character Set (ISO). You use the Choose Multinational Characters Option to allow the use of MCS or ISO characters as data in your application.

Explanation

The ASCII character set consists of 128 characters. Each character is represented by a corresponding decimal code; for example, the character a (lowercase a) is represented by a decimal code of 97. MUMPS uses the ASCII character set as its default and collates data according to its ASCII decimal code.

Many video terminals can recognize and display MCS or ISO characters in addition to ASCII characters. Digital VT200-series terminals can recognize and display MCS characters, and Digital VT300-series terminals can recognize and display either MCS characters or ISO characters.

The MCS and ISO character sets contain characters 1 through 128 of the ASCII character set, and add characters 129 through 255. While many multinational characters are represented by the same decimal code in both MCS and ISO character sets, there are some differences. See Appendix A for tables of ASCII, ISO, and MCS characters with their representative decimal codes.

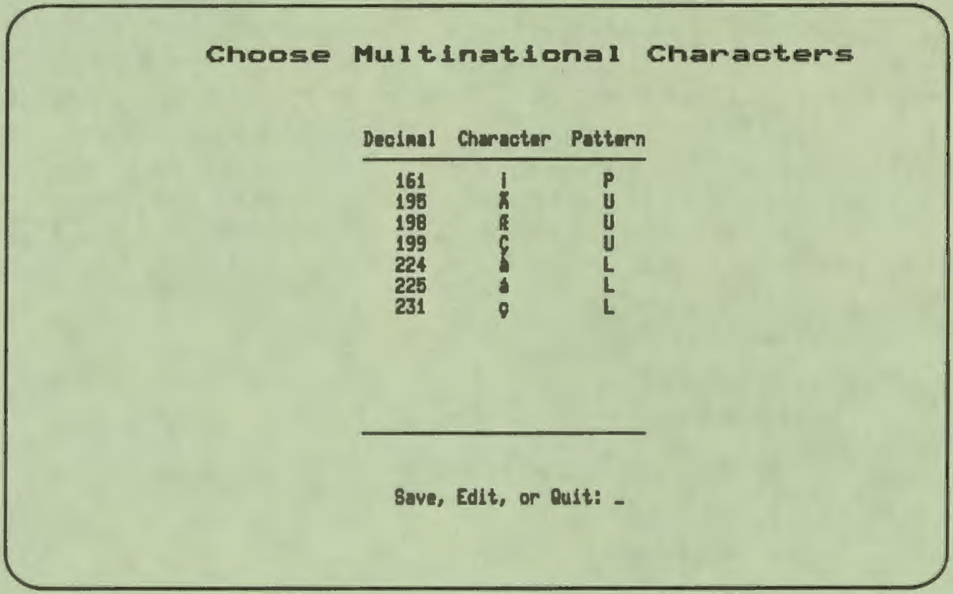
The DASL software allows the use of MCS or ISO characters with decimal codes of 161 through 255 in your application. The multinational character set you see depends on which set your terminal recognizes.

> **Note:** To use the Choose Multinational Characters Option, your terminal must be set up to recognize and display multinational characters. Check the operating manual of your video terminal to learn how to set up your terminal to recognize and display multinational characters.

Choose Multinational Characters Screen

You can use the Choose Multinational Character Option in the development area or in the customization area. When you select the Choose Multinational Characters Option, you see a screen similar to the screen shown in Figure 5–20.

Figure 5–20 Choose Multinational Characters Screen



```
                 Choose Multinational Characters


                  Decimal  Character  Pattern
                  ___________________________
                     161       ¡         P
                     195       Ã         U
                     198       Æ         U
                     199       Ç         U
                     224       à         L
                     225       á         L
                     231       ç         L

                  ___________________________

                      Save, Edit, or Quit: _
```

The DASL software displays a scroll region where you enter the information described in Table 5–8.

Table 5–8 Multinational Character Information

Column Prompt	Meaning
Decimal	A decimal code (161 through 255) of a character in the MCS or ISO character set.
Character	The multinational character, displayed by the DASL software, that is represented by the decimal code entered.
Pattern	A letter representing the type of MUMPS pattern match validation that the DASL software applies to the multinational character. The three types of pattern matches are: P Punctuation U Uppercase L Lowercase

As shown in Figure 5–20, you can specify multiple characters in the scroll region.

When you have finished entering characters, the DASL software displays the "Save, Edit, or Quit:" prompt.

Pattern Matches in Translation

When users enter data in a data-entry field, the DASL software normally stores the input data in the variable %RES. For translation purposes, however, the DASL software translates any characters defined in the Choose Multinational Characters Option and stores them in an internal form in the variable %REST. The DASL software then uses the contents of %REST to perform pattern match validations. The pattern match code you specify in the Choose Multinational Characters screen for each character determines how that character is stored in %REST.

The following example shows the use of %RES and %REST.

```
Pattern match codes specified:

Character            Pattern
----------           -------
   Ñ                   U
   ¡                   P
   ñ                   L

Input at field:  Ñ ¡ ñ

Value of %RES:  %RES="Ñ ¡ ñ"

Value of %REST: %REST="A . a"
```

Using Cross-Reference Transforms

VAX DSM collates characters for lookups and report collections according to their ASCII decimal code. To collate data containing multinational characters in an order other than the ASCII decimal code, you can define cross-reference transforms in the Data Dictionary. The DASL software uses cross-reference transforms to create cross-reference globals that store user input in internal format for lookups and report collections. Using cross-reference transforms, you can specify a collating order that considers language-specific collating practices. The DASL software uses the transformed values in the cross-reference only for collating, not for display.

You can use the MUMPS $TRANSLATE function within a cross-reference transform to solve problems of sorting. See the *VAX DSM Language Reference Manual* for a discussion of the $TRANSLATE function. The DASL software uses the transformed values in the cross-reference transform only for collating, not for display.

You define cross-reference transforms for data names in the second screen of the Define Data Names Option in the Data Dictionary. See the "Cross References" section of Chapter 1 for information on how to create cross-reference transforms.

Recompiling Screens and Reports

Whenever you add or delete a multinational character, you must recompile all application or reports in which that character appears. You must also recompile DASL screens to allow the entry of multinational characters in the Define Data Names screens, in prompts, and in help text.

Select US as the current language in the Choose Current Language screen before you recompile DASL screens to include new characters. Then, after the compilation is complete, set the current language to the language of your application.

Use the Compile Screens Option of the Screen Driver to recompile screens, and the Compile Reports Option of the Report Driver to recompile reports.

Related Sections

Choose Current Language Option
Cross References, Chapter 1
Delete Language Option
Print DASL Messages Option
Translate DASL Messages Option
$TRANSLATE (*VAX DSM Language Reference Manual*)

Translate DASL Messages Option

DASL-created applications display many standard DASL prompts and messages to end users. You use the Translate DASL Messages Option to specify a non-English language and to translate these prompts and messages into that language. You can also use this option to specify an alternate English text for DASL prompts and messages.

Explanation

The DASL messages and prompts that appear to end users of DASL applications include:

- Error messages from DASL validations, for example:

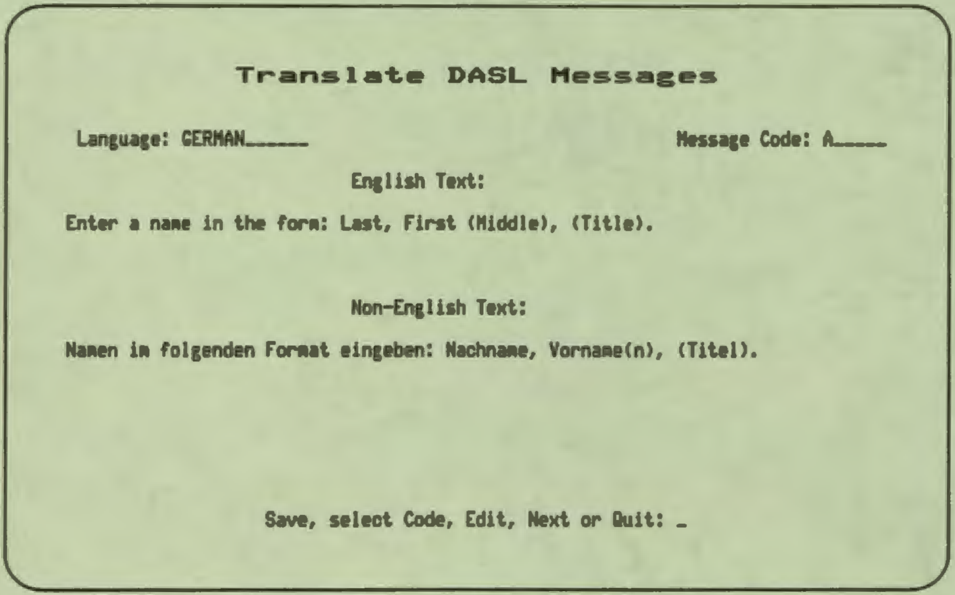
  ```
  Illegal character in input.
  ```

- Prompts for lookup lists or option screens, for example:

  ```
  Enter number of selected item or RETURN if not found.
  ```

You can specify a language and translate these messages and prompts using the Translate DASL Messages screen as shown in Figure 5–21.

Figure 5–21 Translate DASL Messages Screen



```
                     Translate DASL Messages

  Language: GERMAN______                           Message Code: A_____
                           English Text:

 Enter a name in the form: Last, First (Middle), (Title).


                           Non-English Text:

 Namen im folgenden Format eingeben: Nachname, Vorname(n), (Titel).




                   Save, select Code, Edit, Next or Quit: _
```

Table 5–9 shows the information you supply in the Translate DASL Messages screen.

Table 5–9 DASL Translation Information

Item	Description
Language	The name of the language you use for the non-English text.
Message Code	A message code provided by the DASL software that represents a standard DASL prompt or message.
English Text	The English text of a standard DASL message.
Non-English Text	The non-English text that replaces the English text of a standard DASL message.

When you first select the Translate DASL Messages Option, the DASL software prompts you to enter a language name in the Language field. You can enter an existing language name, a new language name, or an asterisk (*) for a lookup list of existing languages.

After you enter a language name, the DASL software displays the following information:

- Code A in the Message Code field
- The English text for Code A in the English Text field

  DASL messages appear in the English Text field in one of the following three forms:

  — Text, for example:

  ```
  Enter a name in the form: Last, First (Middle), (Title).
  ```

  — Text strings with DASL variables that are defined at run time, for example:

  ```
  "Enter an Option Number,"_$S(%K]"":" a Keyword,",1:"")_" a
  Command (* for a list) or RETURN to exit."
  ```

  — Executable MUMPS code that contains messages, for example:

  ```
  S %MSG="Enter a date between "_%Z_" and "_%DTX S:%DTX=""
  %MSG="Enter a date not before "_%Z
  ```

- The following message in the Non-English Text field, if you enter a new language name:

  ```
  This message needs language translation
  ```

- The “Save, select Code, Edit, Next, or Quit:” prompt

At the “Save, select Code, Edit, Next, or Quit:” prompt, you can do the following:

- Enter S to save the screen and any changes you have made.
- Enter C to move to the “Message Code” field and enter a message code.

At this field, you can also enter an asterisk (*) to see a lookup list of the available message codes. You can then select a message code from the lookup list.

After you have selected a message code, the DASL software displays the English text of that code in the "English Text" field. You can then enter your non-English text for the same code in the "Non-English Text" field.

- Enter E to edit the screen. In the "Non-English Text" field, enter the non-English translation of the English text.

  The DASL software provides three lines for the non-English translation. The text you enter must evaluate to a string whose length is not more than 80 characters at run time.

  If the DASL message contains DASL variables or executable MUMPS code as well as English text, use the same format and the same variables in your translated message. Executable MUMPS code must remain executable MUMPS code after translation.

- Enter N to select the next consecutive message code. When you enter N, the DASL software displays the next message code in the Message Code field and the English text in the English Text field. You can then enter or edit non-English text in the Non-English Text field.
- Enter Q to quit the screen without saving changes.

Developing and Compiling Screens and Reports

After you have entered all translation information in the Translate DASL Messages Screen, you can develop your application screens and reports.

The new messages you entered in the Translate DASL Messages Screen become active after you recompile all applicable screens and reports (application or the DASL software) in the customization area.

You can use the Compile Screens Option of the Screen Driver Menu to compile all screens, and the Compile Reports Option of the Report Driver Menu to compile all reports.

Related Sections

Choose Current Language Option
Choose Multinational Characters Option
Delete Language Option
Group Dictionary Option
Print DASL Messages Option

Choose Current Language Option

You use the Choose Current Language Option to specify the current language of your application.

Explanation

In the Choose Current Language screen, the DASL software displays the current application language at the "Current Language" prompt. At this prompt, you can enter a language name or an asterisk (*) to see a lookup list of available languages.

After you enter a current language, the DASL software displays the "Save, Edit, or Quit" prompt.

When you enter S to save a screen, the DASL software displays the following prompt:

```
Do you want to regenerate selected routines?
```

Enter Y at this prompt to regenerate selected DASL routines to reflect language translation changes. The routines that the DASL software regenerates are:

- DASL date and time utility routines
- Lookup utility routines
- Report listing routines

Related Sections

Choose Multinational Characters Option
Delete Language Option
Print DASL Messages Option
Translate DASL Messages Option

Print DASL Messages Option

You use the Print DASL Messages Option to print a report listing the DASL message codes and corresponding message text in a specified language. Using this option, you can print the English or non-English text of all DASL messages as an aid to translation.

Print DASL Messages Report

Example 5–1 shows an excerpt from a Print DASL Messages Report for the language US2. In this report, the DASL software prints each message code and corresponding message text.

Example 5–1 Print DASL Messages Report

```
22/Mar/88 10:51 AM            DASL                             Page 1
                    DSM Application Software Library
                            Messages for: US2
Code        Message
------------------------------------------------------------------------

A           Enter name as Last, First (Middle), (Title).

B           First Name is required.

C           Not the correct format.

CUR         Enter left, right cursor command, DELETE, Text or RETURN to
            exit.

D           Last name is required.

DATE        Enter date in the format

DATE1       S %MSG="Enter date between "_%Z_" and "_%DTX S:%DTX=""
            %MSG="Enter date not before "_%Z

DETAIL      Make your entry more detailed.

EOF         END of the file

F           +%W_" character"_$S(+%W=1:"",1:"s")_" required"
```

Explanation

In the Print DASL Messages screen, enter the name of the language for the report text. You can specify the DASL default language code (US), or a language for which you have translated text in the Translate DASL Messages screen. If you enter a language for which there is no translated text, the DASL software prints the following message for each code:

```
This message needs language translation
```

You can also enter an asterisk (*) at the "Language:" prompt to see a lookup list of available languages.

Choosing a Report Destination

Whenever you choose a report option, the DASL software displays the device selection screen that you use to produce the report in your current process or in a batch process.

After you select the process to use, specify the report destination. You can print the report on any VMS device, or write the report to a VMS file. If you produce the report through your current process, you can also display the report on your current device.

Related Sections

Choose Current Language Option
Delete Language Option
Translate DASL Messages Option

Delete Language Option

You use the Delete Language Option to delete a language that you have entered in the Translate DASL Messages screen.

Explanation

At the "Language to be deleted:" prompt, enter the name of a language that you have previously entered in the Translate DASL Messages screen, or enter an asterisk for a lookup list.

You cannot delete the DASL default language (US) or the current language of your application.

After you enter the name of the language to be deleted, the DASL software displays the "OK to delete this language?" prompt.

- Enter Y at this prompt to delete the language. After you delete the language, the cursor returns to the "Language to be deleted:" prompt, and you can delete another language.
- Enter N to return to the "Language to be deleted:" prompt without deleting the specified language. You can then delete another language.
- Press Exit to quit the screen.

Related Sections

Choose Current Language Option
Print DASL Messages Option
Translate DASL Messages Option

Portable Run-Time DASL Utilities

When you use VAX DSM and the DASL software to develop an application, your DASL application runs on the VAX computer. The Portable Run-Time DASL Utilities are a set of utilities that you can use to modify your DASL application to run on a *target system*. A target system can be any of the following:

- A VAX DSM system on another VAX computer
- A DSM-11 system on a PDP-11 computer
- Other MUMPS implementations on other computers (for example, a MUMPS implementation running on a PC)

You use the options of the Portable Run-Time DASL Utilities Menu to create translation tables for VAX DSM code and routines that handle input and output on target machines.

You must use the DASL software and VAX DSM on the VAX computer to design your application database and create application screens and reports. After you have developed and debugged your DASL application, you can port the completed application to a target machine other than the VAX computer.

The DASL software provides the translation tables and routines you need to run DASL applications on PDP-11 computer systems. The DASL software also includes examples of translation tables and routines for MGlobal's CCSM™ and DataTree's DTM-PC™ implementations of MUMPS on PCs.

DASL Applications at Run Time

DASL applications at run time consist of the following interacting globals and routines:

- DASL library routines

  DASL library routines reside in the DSM manager's UCI and begin with the characters %DA. These routines perform tasks such as lookups and date and time conversions. For example, the DASL Date and Time Utility %DAUDTTM is a DASL library routine.

- DASL globals

  DASL globals, identified by the initial characters ^%DA, provide some messages that application users see at run time, including error messages and help text. For example, your application accesses these DASL %DA globals if you use date and time validations in your application.

- DASL application routines

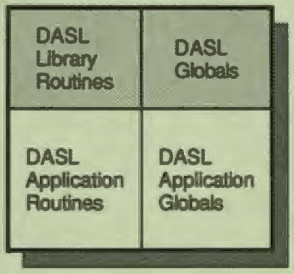
  Application routines are the routines that the DASL software compiles from the screen and report definitions you have provided and any MUMPS routines that you have specified.

- DASL application globals

  Application globals are globals that you set up that contain code used by the DASL application. Some application globals, identified by the initial characters ^DA, are set up by the DASL software. For example, when you specify the TABLE validation, you create a table of values that is stored in an application global. You also can create your own application globals.

Figure 5–22 shows the components of a DASL application.

Figure 5–22 DASL Applications at Run Time



MR–4576–RA

When you port your DASL application, you must port some of the DASL library routines and the DASL globals as well as your own application routines and globals to the target system.

> **Note:** To use the DASL library routines and DASL globals on a target system other then VAX DSM, you must have a DASL Run-Time License.

Creating a Portable DASL Application

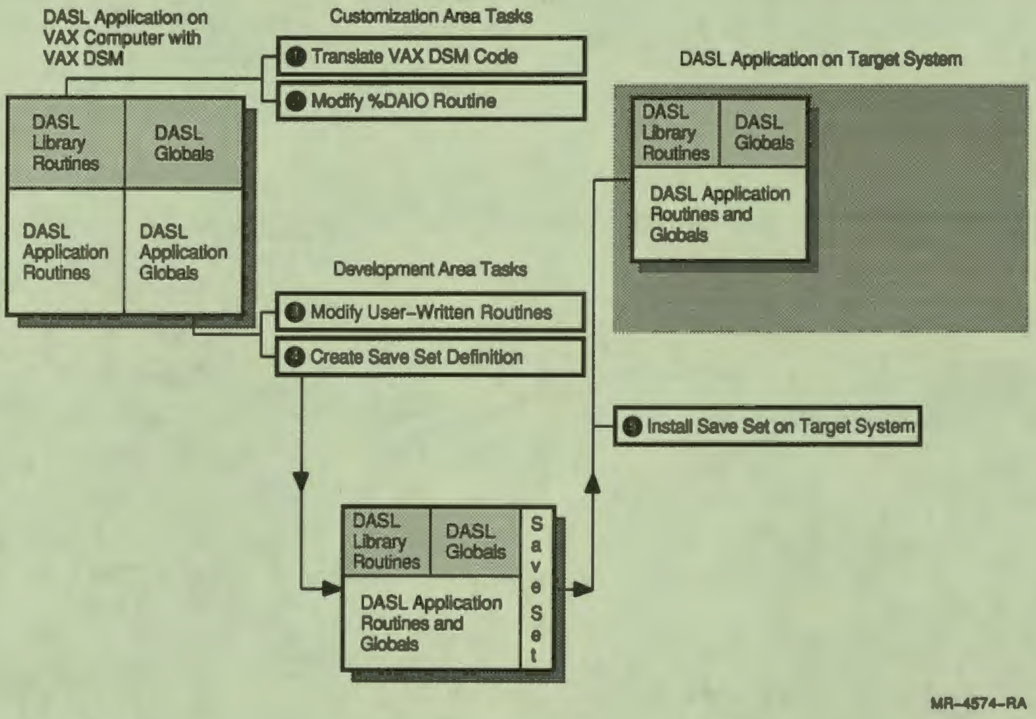
To port your DASL application to a target system other than the VAX computer that uses another vendor's MUMPS implementation, follow these steps:

1. Translate application MUMPS code that is specific to VAX DSM to the target vendor's MUMPS code.
2. Modify the DASL routine (%DAIO) that handles input and output for the target system devices.
3. Modify for the target system any VAX DSM routines that you have written.

4. Define a target system save set and run the save set to recompile the application globals, routines, screens, and reports for the target system.
5. Install the save set on the target system device.

Figure 5–23 shows the process that you follow to create a portable DASL application.

Figure 5–23 Porting DASL Applications to a Target Device


DASL Application on VAX Computer with VAX DSM
DASL Library Routines
DASL Globals
DASL Application Routines
DASL Application Globals
Customization Area Tasks
1 Translate VAX DSM Code
2 Modify %DAIO Routine
Development Area Tasks
3 Modify User-Written Routines
4 Create Save Set Definition
DASL Library Routines
DASL Globals
DASL Application Routines and Globals
Save Set
5 Install Save Set on Target System
DASL Application on Target System
DASL Library Routines
DASL Globals
DASL Application Routines and Globals
MR-4574-RA

Selecting a DASL Work Area for Portability Tasks

You must work in the DASL customization area to complete the following tasks:

- Translate VAX DSM code to target system code.
- Modify the %DAIO routine for the target device.

To work in the customization area, use the following DSM command line to log in to VAX DSM:

```
$ DSM/DASL_CUSTOMIZE Return
```

The DSM/DASL_CUSTOMIZE command places you in the DASL manager's UCI and translates global references of ^DA* to ^%DA*. You can work in the customization area to modify the DASL library routines and DASL library globals.

You must work in the DASL development area to complete the following tasks:

- Modify any VAX DSM routines that you have written for your application.
- Define and run a target system save set.

The DASL development area is the work area in which you create your DASL application. You must work in the development area to modify your application routines and globals. To work in the DASL development area, use your developer's UCI to log in to VAX DSM, for example:

```
$ DSM/UCI=developers_UCI Return
```

Table 5–10 summarizes the portability tasks you perform in each DASL work area.

Table 5–10 DASL Work Areas and Portability Tasks

Area	Command Line	Tasks
Customization	DSM/DASL_CUSTOMIZE	Translate VAX DSM code Modify %DAIO routine
Development	DSM/UCI=developers_UCI	Modify your own VAX DSM code Define and run a target system save set

See the *DASL Management Guide* for more information about DASL work areas.

Translating VAX DSM Code to Target MUMPS Code

The DASL software contains MUMPS code that is specific to VAX DSM. The DASL global ^%DASY("MUMPS","VAX DSM") isolates this implementation-specific code so that you can easily translate it to another vendor's MUMPS code.

You can use the Translate VAX DSM Code Option on the Portable DASL Run-Time Utilities Menu to set up a *translation table* that translates VAX DSM code to the target vendor's MUMPS code. The translation table contains a listing of VAX DSM code and the equivalent target system code. You create a translation table for each target system that you specify in this screen.

The screen displays the VAX DSM code to be translated; you must provide the equivalent target MUMPS code. (Consult the VAX DSM documentation set and the documentation for the target vendor's MUMPS implementation for helpful information at this step in the translation process.)

The screen also displays the translated DSM-11 code. For example, the screen displays the following code for VAX DSM and DSM-11:

```
VAX DSM:  $ZC(%BASEDIT,P1,%X)

DSM-11:   $$BASEDIT^%DAIO(P1,%X)
```

In the previous example, the VAX DSM code calls the VMS BASEDIT utility. The BASEDIT utility converts string format according to a mask value; for example, it can strip leading and trailing spaces. The DSM-11 code performs the same functions, but uses standard MUMPS code.

In some cases, you may have to write target MUMPS code that performs the functions of existing VAX DSM code. You can add subroutines containing this code to the %DAIO routine (described in the following section).

> **Note:** You must work in the DASL customization area to use the Translate VAX DSM Code Option.

For more information about translating VAX DSM code in the DASL software, see the "Translate VAX DSM Code Option" section of this chapter.

Modifying the DASL Input and Output Routine

The DASL software uses VAX DSM USE commands to handle input and output to devices. The %DAIO routine translates these VAX DSM USE commands to DSM-11 code. You must write a subroutine for %DAIO to translate VAX DSM USE commands to your target system code.

> **Note:** You must work in the customization area to modify the %DAIO routine.

The %DAIO routine contains a subroutine that takes the following format:

USE("DEV","ARG{:ARG}")

where:

DEV is the name of the target device

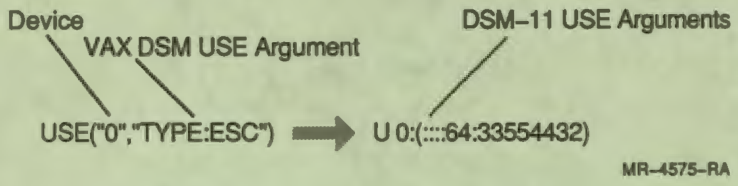
ARG is the VAX DSM argument to the USE command (Multiple arguments are separated by a colon (:).)

The arguments to the VAX DSM USE command can be any of the following:

CON
CTRAP
ECHO
ESC
NOCON
NOESC
NOECHO
NOTYPE
TERM=value
TYPE
WIDTH=value

The USE subroutine converts the VAX DSM USE command code to DSM-11 USE arguments. Figure 5–24 shows the structure of the USE subroutine when the target system is DSM-11.

Figure 5–24 Structure of USE Subroutine



If your target is another vendor's MUMPS implementation, you must modify the %DAIO routine as follows:

1. Write a new subroutine that contains the MUMPS code for the target system's USE command.

   Make sure that you use the NEW command for all variables in the subroutine.

2. Make an entry in the subroutine USE("DEV","ARG") to branch to the new subroutine if the variable %MUMPS is set to the target system name.

3. Quit the subroutine with the QUIT argument variable set to the target system's USE command syntax.

You can use the Edit VAX DSM Routines Option of the Development Environment to edit the %DAIO routine and create new routines.

The DASL software contains examples of modified %DAIO routines for MGlobal and DataTree MUMPS implementations.

Modifying VAX DSM Routines That You Have Written

If you have written any VAX DSM routines for your application that contain VAX DSM-specific code, you must modify these routines for the target system.

> **Note:** You must work in the DASL development area (the developer's UCI) to modify VAX DSM routines that you have written.

To modify a VAX DSM-specific routine for a target system, make a copy of the VAX DSM routine, giving the routine a new name. Then, use the Edit VAX DSM Routines Option of the Development Environment to edit the new routine name and replace VAX DSM code with target system MUMPS code.

You may replace the VAX DSM code in one of three ways:

- Replace the VAX DSM code with the equivalent target system code.
- Translate the USE command, using the %DAIO routine as an extrinsic function.
- Use a DASL-provided extrinsic function when you want to replace variables for an existing entry in the translation table.

Replacing VAX DSM Code in User-Written Routines

If you have written routines for your application that contain VAX DSM code, you must translate the VAX DSM code for your target system. The VAX DSM code that you must translate can include:

- VAX DSM code that is listed in the translation table of VAX DSM code
- VAX DSM code that you have used in your routines that is not listed in the translation table of VAX DSM code

You can use the Add Entries to the Translation Table Option to add VAX DSM code that needs to be translated to the translation table. With this option, you also specify the equivalent MUMPS code for each target system.

In all cases, you must modify your own routines so that they can use the translation table to translate VAX DSM code to target system code.

To modify the routines, isolate VAX DSM code. Then, modify the code so that it is stored in the ^%DASY global in the subtree MUMPS. The ^%DASY global takes the following syntax:

^%DASY(S1,S2,S3)=code

where:

S1 is the string literal "MUMPS"

S2 is the name of the target system (contained in the variable %MUMPS)

S3 is an entry reference to the VAX DSM code

code is the target system MUMPS code

In the following example, the $ZT function is a VAX DSM function that must be translated. This example shows the form that your edits must take:

```
Old Code:

S $ZT="TRAP^%DASDFED"

Edited Code:

S ??^%DASY("MUMPS",%MUMPS,"$ZT")??="TRAP^%DASDFED"
```

Note that you must enclose the global reference to the ^%DASY global in double question marks. The double question marks indicate to the DASL software that this code must be translated.

Translating the USE Command in User-Written Routines

You must edit your routines as follows so that the USE command can be translated to the target code:

1. Isolate the USE command in the routines.
2. Edit the routine to use the ^%DAIO routine as an extrinsic function.
3. Enclose the extrinsic function in double question marks.

The following example shows the form that your edits must take:

```
Old Code:

U 0:NOECHO

Edited Code:

??$$USE^%DAIO(0,"NOECHO")??
```

Replacing Variables for User-Written Routines

The DASL BASEDIT routine is a routine that normalizes data, such as stripping leading and trailing spaces, converting lowercase input to uppercase, and so on. The input for the BASEDIT routine can come from any number of different variables, such as %RES for user input. The translation table of VAX DSM code uses the P1 variable to contain the names of the variables used in the BASEDIT routine.

You must use an extrinsic function in your routines to replace P1 with the variables that are specific to your routines.

Suppose that you want to translate the following VAX DSM code:

```
S %RES=$ZC(%BASEDIT,%RES,168)
```

Note that the translation table for VAX DSM code contains code that is identical to that in the previous example with the exception that the variable P1 is used in place of %RES:

```
$ZC(%BASEDIT,P1,168)
```

The following example shows the form that your edits must take:

```
Old Code:

S %RES=$ZC(%BASEDIT,%RES,168)

Edited Code:

S %RES=??$$REPLACE^%DAIO(^%DASY("MUMPS",%MUMPS,"$ZC(%BASEDIT,P1,168)"),"P1","%RES")??
```

In the previous example, the $$REPLACE extrinsic function replaces the P1 variable with the %RES variable when the code is translated to the target system code.

Saving the Portable DASL Application

After you have set up a translation table for the target vendor's MUMPS implementation, modified the %DAIO routine, and modified application VAX DSM routines that you have written, you must use the Define Target Save Set Option of the Portable Run-Time DASL Utilities Menu to create and run a save set for your portable DASL application.

The save set is a listing of all the DASL library routines, DASL library globals, application routines, and application globals that you want to move to the target system. When you run the save set, you recompile all the DASL application screens and reports that use the target system MUMPS code. You need only define a save set once for a target system: if you change any existing application screens and reports, you can run the previously defined save set to modify your portable application.

After you run the save set, you can then install the save set on your target system.

You must have a DASL Run-Time License for each target machine to use your portable DASL application. The DASL save set contains DASL utility routines and globals as well as your application routines and globals.

For more information about creating a save set, see the "Define Target Save Set Option" section of this chapter.

Limitations to Portable DASL Applications

With the current version of the DASL software, there are limitations to portable DASL applications. You cannot port the following:

- Database queries and tables
- Application Environment Menu options, such as the Security System and the Message Center

Therefore, do not include DASL Application Environment options in your portable application.

There are also limitations for applications that run on PDP-11 computers as follows:

- DSM-11 has a fixed partition size that limits the size of routines in the symbol table. This limitation can affect some longer DASL routines.
- You can not create large screens for PDP-11 systems.

Portable Run-Time DASL Examples

The DASL software includes examples of the ^%DAIO routine and the translation tables for MGlobal's CCSM and DataTree's DTM-PC implementations of MUMPS on PCs.

These examples are stored in the SYS$EXAMPLES directory in the following files:

File	Description
DSM$DASL_PRTD_ROU.SAV	^%DAIO routine example
DSM$DASL_PRTD_GLO.SAV	Translation table for DTM-PC and CCSM implementations

To load these files, you must use the VAX DSM manager's account and the SYS$EXAMPLES directory. At the DSM prompt:

- Use the %GTI utility to load the translation tables in the DSM$DASL_PRTD_GLO.SAV file.
- Use the %RS utility to load the ^%DAIO routine in the DSM$DASL_PRTD_ROU.SAV file.

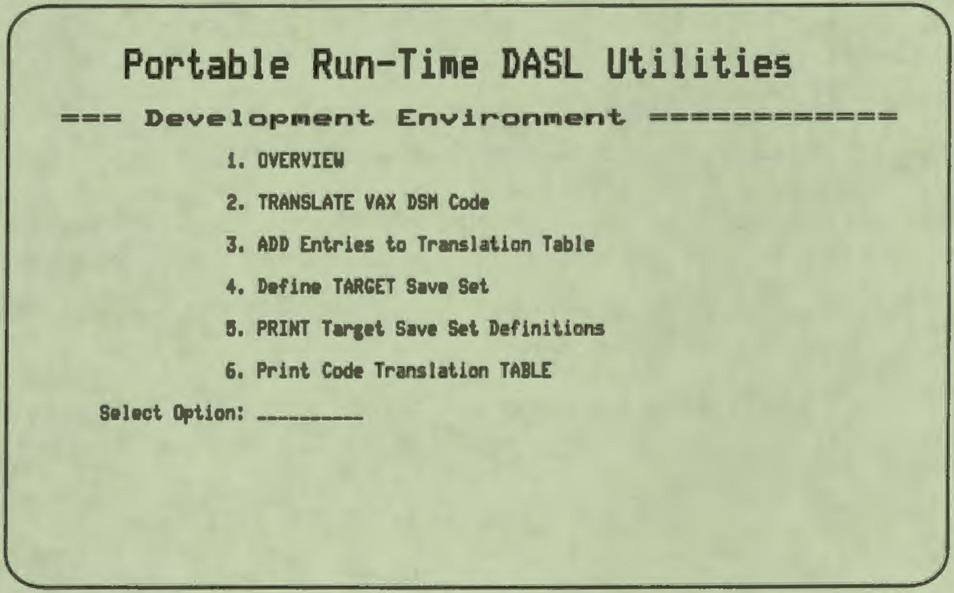
Note: The DSM$DASL_PRTD_ROU.SAV and DSM$DASL_PRTD_GLO.SAV files are examples only and are not supported software.

Portable Run-Time DASL Utilities Menu

You can use the options of the Portable Run-Time DASL Utilities Menu to modify your DASL application so that you can port it to a computer other than the VAX computer and a MUMPS implementation other than VAX DSM.

When you select the Portable DASL Run-Time Utilities Menu, you see the screen shown in Figure 5–25.

Figure 5–25 Portable Run-Time DASL Utilities Menu



```
         Portable Run-Time DASL Utilities
=== Development Environment ============

              1. OVERVIEW

              2. TRANSLATE VAX DSM Code

              3. ADD Entries to Translation Table

              4. Define TARGET Save Set

              5. PRINT Target Save Set Definitions

              6. Print Code Translation TABLE

      Select Option: _________
```

The following sections describe the options of the Portable Run-Time DASL Utilities Menu.

Translate VAX DSM Code Option

A DASL application contains some DASL library routines as well as application routines generated by the DASL Screen Driver and Report Driver. These routines contain MUMPS code that is specific to VAX DSM. You use the Translate VAX DSM Code Option to set up a translation table that translates VAX DSM code to a target vendor's MUMPS code (other than VAX DSM).

Explanation

Each translation table contains a listing of VAX DSM code and the equivalent target system code. You create a translation table for each target system for your DASL application.

The DASL global ^%DASY("MUMPS","VAX DSM") isolates VAX DSM code so that you can easily translate it to another vendor's MUMPS code. The Translate VAX DSM Code Option accesses the ^%DASY("MUMPS","VAX DSM") global and sets up a translation table that the DASL software uses to convert VAX DSM code to the target system's MUMPS code.

The DASL software stores the translated MUMPS code in a global that takes the following form:

^%DASY(S1,S2,S3)=code

where:

S1	is the string literal "MUMPS"
S2	is the name of the target system (also contained in the %MUMPS variable)
S3	is an entry reference to the VAX DSM code
code	is the target system MUMPS code

The following example shows the global node that stores one line of VAX DSM code and the corresponding global node that stores the equivalent line of DSM-11 code:

```
^%DASY("MUMPS","VAX DSM","$ZC(%SQRT,P1)")=$ZC(%SQRT,P1)

^%DASY("MUMPS","DSM-11","$ZC(%SQRT,P1)")=$$SQROOT^%DAIO(P1)
```

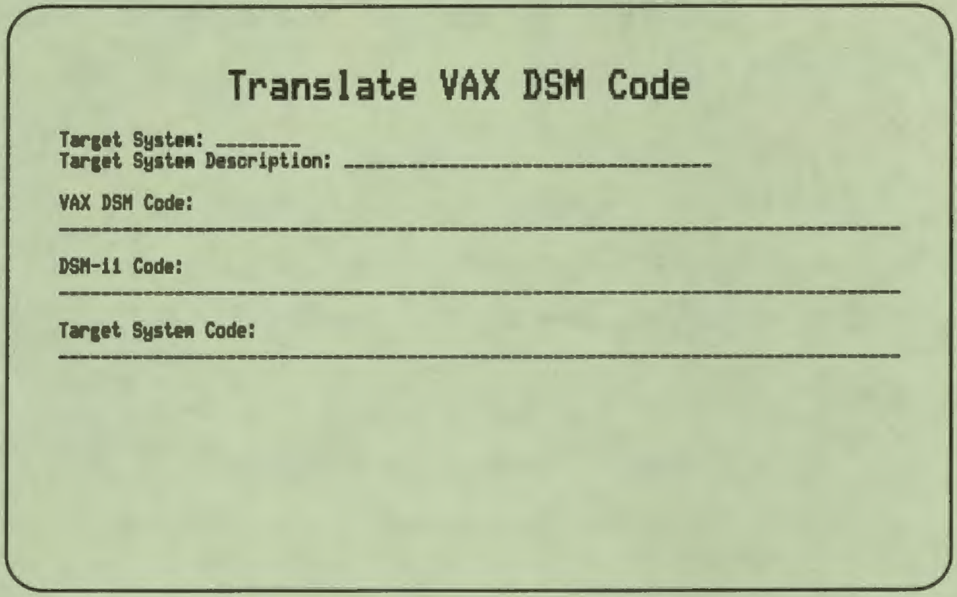
The DASL software provides the translation table for DSM-11 code, and also provides examples of translation tables for MGlobal and DataTree MUMPS code.

You must specify the corresponding MUMPS code for any other target system. In addition, you may need to produce new MUMPS code for the target system that performs the same functions as existing VAX DSM code. You can use the Add Entries to Translation Table Option to add new MUMPS code to the translation table.

Translate VAX DSM Code Screen

You must work in the DASL customization area to use the Translate VAX DSM Code Option. Figure 5–26 shows the screen you see when you choose the Translate VAX DSM Code Option.

Figure 5–26 Translate VAX DSM Code Screen



```
                 Translate VAX DSM Code

Target System: ________
Target System Description: ___________________________________

VAX DSM Code:
______________________________________________________________

DSM-11 Code:
______________________________________________________________

Target System Code:
______________________________________________________________
```

Table 5–11 describes the information that you enter in the Translate VAX DSM Code screen.

Table 5–11 Translation Table of VAX DSM Code Information

Item	Description
Target System	The name of the target system (other than VAX DSM). The name must contain a leading alphabetic character followed by up to 6 alphanumeric characters.
Target System Description	A text description of the target system up to 35 alphanumeric characters in length. The text description is displayed in lookup lists.
VAX DSM Code	VAX DSM code, displayed in this field, to translate to the target system MUMPS code.
DSM-11 Code	DSM-11 code, displayed in this field, that the DASL software supplies for the translation table.
Target System Code	Equivalent MUMPS code that you must substitute for the VAX DSM code in the target system.

When you first select the Translate VAX DSM Code Option, the DASL software prompts you to enter the name of a target system. You can enter an existing target system name, a new name, or an asterisk (*) to view a lookup list of existing target systems.

If you enter a new name, the system displays the following prompt:

```
New entry, OK to add?
```

Enter Y to add the entry, or N to start again.

If you enter an existing target system name, the system displays the following prompt:

```
Edit, Print, Delete, or Quit:
```

At this prompt, you can enter one of the following:

- Enter E to edit existing entries for the target system.
- Enter P to print a copy of the translation table.
- Enter D to delete the target system translation table. (The system then asks if you are sure that you want to delete. Enter Y to delete.)
- Enter Q to quit without making changes.

Then, the DASL software prompts you to enter a description of the target system if it is a new name, or displays the description of the target system if it is an existing name.

After you enter a target system name and description, the DASL software displays the following information:

- VAX DSM code in the VAX DSM Code field
- DSM-11 code in the DSM-11 field

Then the DASL software displays a "Target System Code:" prompt where you can enter the equivalent target system MUMPS code.

> **Note:** The target system code that you enter cannot exceed one line in length. If the target system MUMPS code that you enter is longer than one line, you must enter a call to an extrinsic function, and also write the code for the extrinsic function.

After you finish entering target system MUMPS code, the system displays the following prompt:

```
Edit, Print, Delete, or Quit:
```

Enter E to edit the translation table, P to print a copy of the translation table, D to delete the target system from the translation table, or Q to quit the screen without saving any changes.

Modifying VAX DSM or DSM-11 Code

In rare circumstances, you may want to modify the existing VAX DSM and DSM-11 code as well as add MUMPS code for target systems. If you modify VAX DSM and DSM-11 code, you must save a version of the translation table that includes the modified code so that you can use the modified code in future versions of the DASL software. (When you install a new version of the DASL software, the new version overwrites any existing VAX DSM or DSM-11 code that you have edited.)

> **Note:** Please notify the VAX DSM or DASL developers if you need to make changes to existing VAX DSM or DSM-11 code.

Related Sections

Add Entries to Translation Table Option
Define Target Save Set Option
Print Code Translation Table Option
Print Target Save Set Definitions Option

Add Entries to Translation Table Option

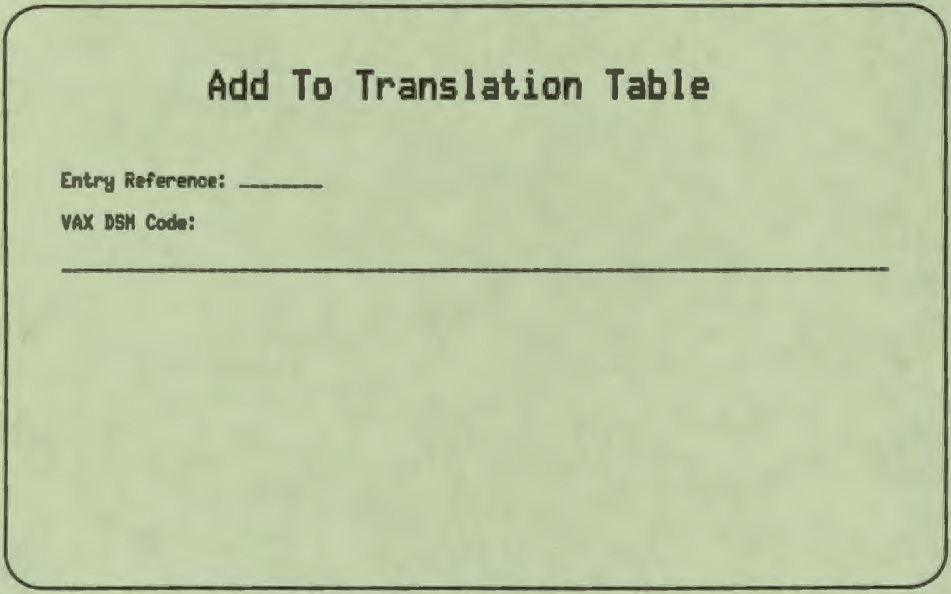
You use the Add Entries to Translation Table Option to add entries of VAX DSM code to the translation table. You can then translate the entries to target system code.

Explanation

If you write your own routines that contain VAX DSM code that is not currently in the translation table, you can add the VAX DSM code to the translation table.

Figure 5–27 shows the screen you see when you choose the Add Entries to Translation Table Option.

Figure 5–27 Add Entries to Translation Table Screen



```
                Add To Translation Table

Entry Reference: _______
VAX DSM Code:
____________________________________________________________
```

Table 5–12 shows the information that you enter in the Add to Translation Table screen.

Table 5–12 Information for the Add to Translation Table Screen

Item	Description
Entry Reference	An entry reference consisting of one alphabetic character followed by up to seven alphanumeric characters that refers to the VAX DSM code to be translated.
VAX DSM Code	The VAX DSM code that you want to translate.
Target System Code	The target system code that is equivalent to the VAX DSM code.

When you first select the Add Entries to Translation Table Option, the DASL software prompts you to enter the name of an entry reference. You can enter an existing entry reference, a new entry reference, or an asterisk (*) to view a lookup list of existing entry references.

If you enter a new entry reference, the system displays the following prompt:

```
New entry, OK to add?
```

Enter Y to add the entry, or N to start again.

If you enter an existing entry reference, the system displays the following prompt:

```
Edit or Quit:
```

At this prompt, you can enter one of the following:

- Enter E to edit existing entries for the target system.
- Enter Q to quit without making changes.

The system then displays the VAX DSM code field where you can enter the VAX DSM code that you want to translate.

After you enter the VAX DSM code, the system displays a field in which you enter the equivalent code for each target system that you have defined through the Translation Table of VAX DSM Code Option.

After you finish entering equivalent target system MUMPS code, the system displays the "Edit or Quit:" prompt again.

Related Sections

Define Target Save Set Option
Print Code Translation Table Option
Print Target Save Set Definitions Option
Translate VAX DSM Code Option

Define Target Save Set Option

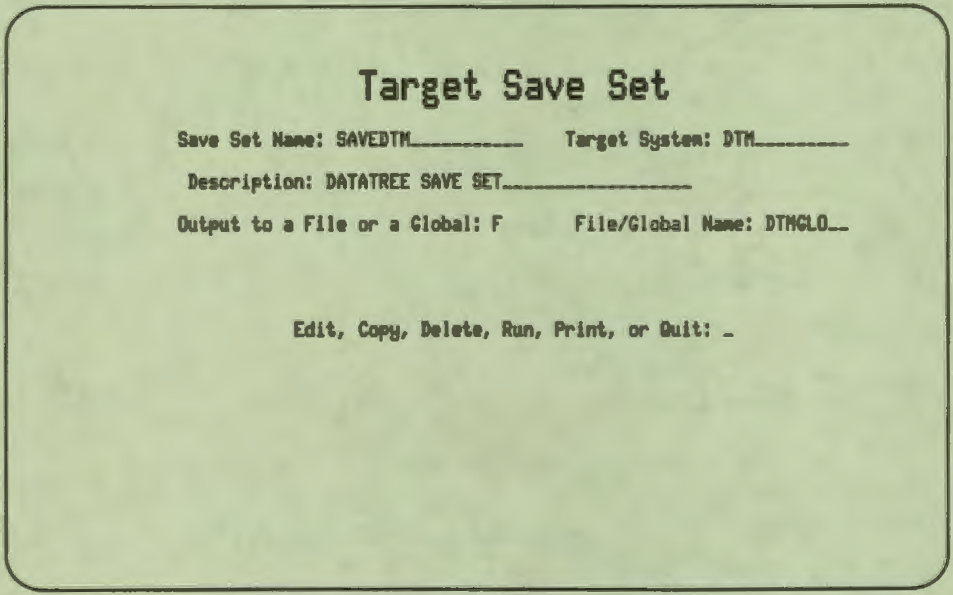
You use the Define Target Save Set Option to define a save set of routines, globals, screens, and reports for your portable DASL application. You can run the save set to create the portable DASL application and then install the save set on your target system.

You must have a DASL Run-Time License for each target machine to use your portable DASL application. The DASL save set contains DASL utility routines and globals as well as your application routines.

Define Target Save Set Screen

Figure 5–28 shows the screen you see when you choose the Define Target Save Set Option.

Figure 5–28 Define Target Save Set Screen



```
                    Target Save Set

Save Set Name: SAVEDTM__________   Target System: DTM________

 Description: DATATREE SAVE SET________________

Output to a File or a Global: F     File/Global Name: DTMGLO__


          Edit, Copy, Delete, Run, Print, or Quit: _
```

Table 5–13 shows the information that you first enter in the Define Target Save Set screen.

Table 5–13 Information for the Define Target Save Set Screen

Item	Description
Save Set Name	A name consisting of one alphabetic character followed by up to 16 alphanumeric characters.
Target System	The name of a target system previously defined in the Translation Table for VAX DSM Code screen.
Description	A text description of the target save set that DASL displays in lookup lists.
Output to a File or a Global	An F to store the save set in a VMS sequential file, or a G to store the save set output in a MUMPS global.
File/Global Name	The name of the file or global in which the save set is stored, consisting of one alphabetic character followed by up to seven alphanumeric characters.

Saving to Files or Globals

As shown in Table 5–13, you can direct the save set output to VMS files or MUMPS globals.

- Save the save set to VMS files if you want to transport the save set by tape or electronic mail.
- Save the save set to MUMPS globals if you want to transport the save set using Distributed Data Processing (DDP).

When you save the save set to VMS files, the DASL software creates four VMS files with the following file extensions:

```
filename.GL            filename.RL
filename.G             filename.R
```

When you save the save set to MUMPS globals, the DASL software creates four global subtrees with the following names:

```
^filename("GL")        ^filename("RL")
^filename("G")         ^filename("R")
```

Listing Items for Translation

After you enter the information shown in Table 5–13, the DASL software displays scroll regions where you can list the screens, reports, routines, and globals to include in the save set.

You specify the screens, reports, routines, and globals by entering one or more of the following in the scroll region:

- Names
- An asterisk (*) for all items in the category
- A partial name followed by an asterisk (*)

- An at sign (@) followed by a defined DASL group name for all items in the group
- A range of items in the form name1-name2

If you choose a range in the form name1-name2, DASL includes all items that fall within the range, including the items you specified as limits.

The following are exceptions to the list of items that you can enter:

- You cannot specify a group name for global data.
- You must specify a global name to save global data. You cannot use a wildcard (*) to select global data.

> **Note:** You should use wildcards as much as possible to specify screens, reports, and routines. If you make modifications to your application at a later time, the use of wildcards in the save set definition ensures that all your modifications will be saved when you run the save set again.

Listing Routines for Translation

When you list routines for translation in the scroll region, you must specify both a master routine name and a run-time routine name.

The master routine name is the current name of the routine that you want to translate. For example, a routine can be a user-written routine that contains VAX DSM code enclosed in double quotation marks.

The run-time routine name is the name that the DASL software gives to the routine when it is translated to the target system code.

> **Note:** You must specify a run-time routine name that is different from the master routine name. If you specify an identical name for master and run-time routines, no translation occurs.

In this scroll region, you should use wildcards as much as possible, for example:

```
            Routine Selection

      Master                  Run-Time
      ---------------------------------
      KB*                     KL*
```

Responding to the Decision Prompt

When you select a save set name from a lookup list or when you finish entering screens, reports, routines, and globals, the system displays the following decision prompt:

```
Edit, Copy, Delete, Run, Print, or Quit:
```

Enter E to edit the save set definition.

Enter C to copy the save set definition to another save set name. Enter a name at the "Copy to:" prompt.

Enter D to delete the save set. The system then asks if you are sure you want to delete. Enter Y to delete, or N to quit.

Enter R to run the save set. When you run the save set, you create a portable DASL application from the save set definitions you have specified.

Enter P to print a copy of the target save set definition. The DASL software then displays the device selection screen that you use to produce the report in your current process or through a batch process.

Enter Q to quit the screen without saving changes.

Related Sections

Add Entries to Translation Table Option
Print Code Translation Table Option
Print Target Save Set Definitions Option
Translate VAX DSM Code Option

Print Target Save Set Definitions Option

You use the Print Target Save Set Definitions Option to print a copy of the save set definitions for a target system.

Explanation

Example 5–2 shows a sample Print Target Save Set Definitions Report.

Example 5–2 Print Target Save Set Definitions Report

```
7-Jun-90  11:27 AM            DASL                          Page 1
                       DSM Application Software Library
                       Print Target Save Set Definitions
__________________________________________________________________
    Save Set Name: LAB
    Description: Health Care Lab System
    Target System: DSM-11
    Description:  Digital Standard MUMPS for the PDP
    Output to a File or a Global: F

                                  Screen
                    ------------------------------------------

                             LAB*
                             GN*
                                  Report
                    ------------------------------------------

                             LAB*
                                  Routine
                    ------------------------------------------

                        LMSORT                     LASORT
                        LMFILE                     LAFILE
                        LMCAL                      LACAL
                                  Global
                    ------------------------------------------

                             ^LAB
                             ^LABCODE
                             ^LABSTA
                             ^LABEX
                             ^LABTEST
```

When you choose the Print Target Save Set Definitions Option, the DASL software displays the following prompt:

```
Print All Save set Definitions?
```

Enter Y at this prompt to print all save set definitions.

If you enter N at the prompt, the system prompts you for the name of the save set for which you want a Target Save Set Definitions Report. You can enter a previously defined save set name or an asterisk (*) to choose from a lookup list of save set names.

Choosing a Destination

After you specify the save set, the system displays the device selection screen that you use to select the process to produce the report and to specify the report destination.

Related Section

Define Target Save Set Option

Print Code Translation Table Option

You use the Print Code Translation Table Option to print a copy of the code translation table for a target system.

Explanation

Example 5–3 shows a portion of a Print Code Translation Table Report for DataTree's DTM-PC implementation of MUMPS.

Example 5–3 Print Code Translation Table Report

```
27-Feb-90  9:37 AM              DASL                           Page 1
                       DSM Application Software Library
                         Print Code Translation Table
________________________________________________________________________

    VAX DSM:    $ZB#256
    DTM:        $S($L($zb)=1:$A($zb),1:$C(27))

    VAX DSM:    $ZB\256
    DTM:        $ZZB

    VAX DSM:    $ZC(%BASEDIT,P1,%X)
    DTM:        $$BASEDIT^%DAIO(P1,%X)

    VAX DSM:    $ZC(%BASEDIT,P1,136)
    DTM:        $$BASEDIT^DAIO(P1,136)

    VAX DSM:    $ZC(%BASEDIT,P1,168)
    DTM:        $$BASEDIT^DAIO(P1,168)

    VAX DSM:    $ZC(%BASEDIT,P1,288)
    DTM:        $$BASEDIT^DAIO(P1,288)

    VAX DSM:    $ZC(%SQRT,P1)
    DTM:        $$SQROOT^DAIO(P1)
```

When you choose the Print Code Translation Table Option, the system prompts you for the name of the target system for which you want a Print Code Translation Table Report.

Choosing a Destination

After you specify the target system, the system displays the device selection screen that you use to select the process to produce the report and to specify the report destination.

Related Section

Translate VAX DSM Code Option

Chapter 6
Application Environment

This chapter describes the following Application Environment Menu options and concepts:

- Security System Option

  Classification Dictionary Option
  User Dictionary Option
  Device Dictionary Option
  System Status Dictionary Option
  System Control Option
  Security Reports Option
  Edit User Password Option
  Edit Installation Name Option

- Report Directory Option
- Message Center Option

  Send Messages Option
  Read Messages Option
  Send System Messages Option
  Read System Messages Option

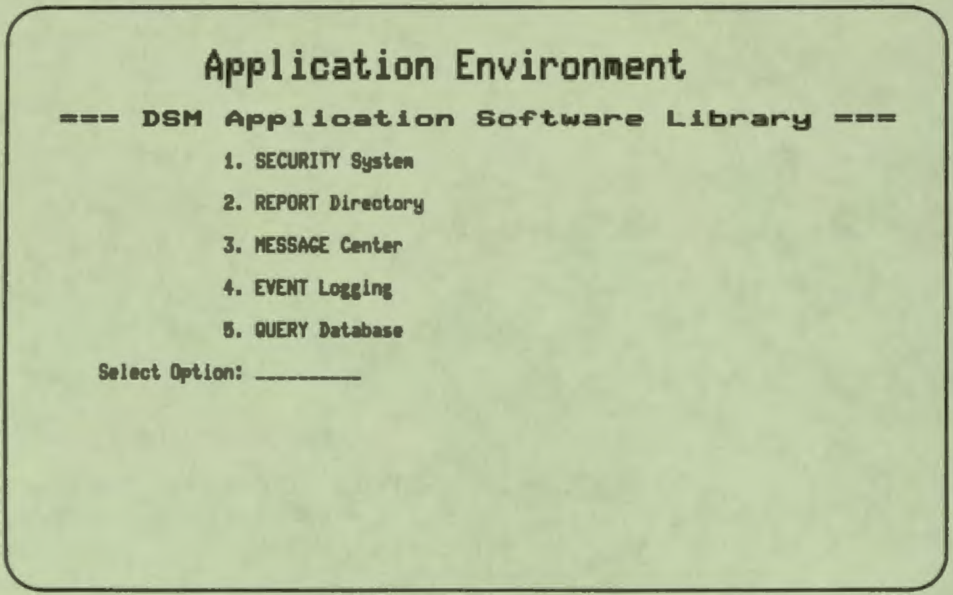
- Event Logging Option

  Assign Event Logging Device Option
  Print System Event Log Option
  Purge System Event Log Option

- Query Database Option

Figure 6–1 shows the options of the Application Environment Menu.

Figure 6–1 Application Environment Menu



```
                Application Environment

  === DSM Application Software Library ===

           1. SECURITY System

           2. REPORT Directory

           3. MESSAGE Center

           4. EVENT Logging

           5. QUERY Database

     Select Option: _________
```

See the *DASL Programmer's Guide* for instructions on how to include Application Environment options in your application.

Security System Option

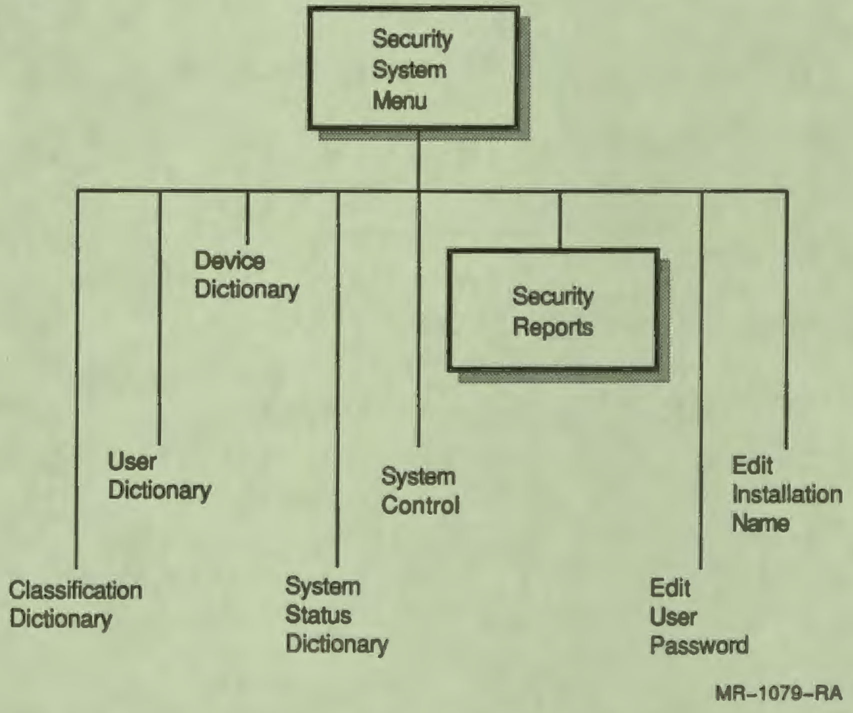
The Security System controls access to the various application options and devices. System security is a privilege-based operation, requiring user and device identification and validation. All login screens and menu screens are integrated into the Security System through the Screen Driver. Database tables are also associated with security system privileges. The Security System options also provide system manager functions for security maintenance, and a full set of security and utilization reports.

> **Note:** Only a user with system manager (+) privileges can access the Security System options.

Security System Menu

Figure 6–2 diagrams the options you can access using the Security System Menu.

Figure 6–2 Security System Menu Options



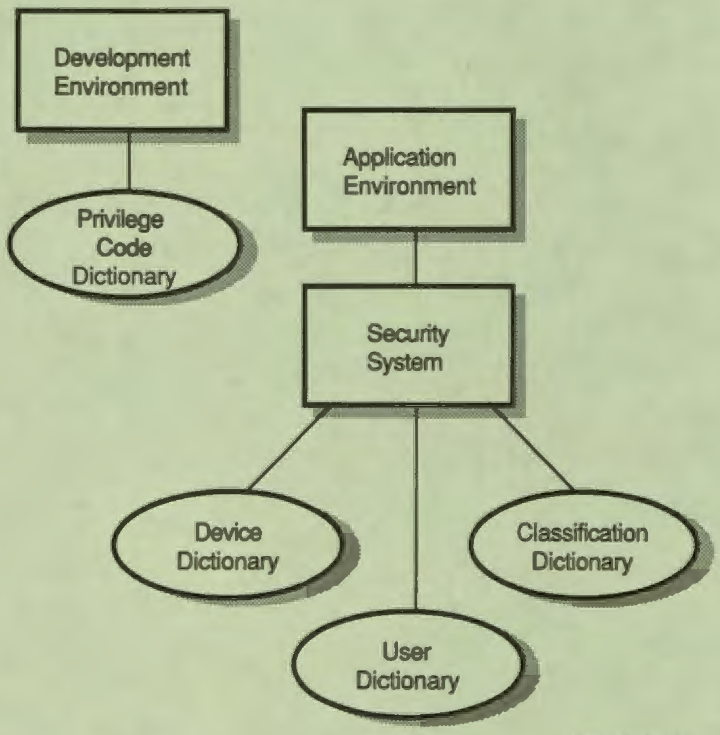
The Security System is designed for use within a DASL-created application (in the application area rather than in the development area). To design the Security System, however, you must complete some design tasks within the DASL development area. These tasks include:

- Define privilege codes in the Privilege Code Dictionary of the Development Environment Menu.
- Attach privilege codes to each option screen you define through the Define Option Screens Option of the Screen Driver Menu.
- Attach privilege codes to database tables in the Define Tables Option of the Query Driver Menu.
- Define any specific device attributes in the Device Type Dictionary of the Development Environment Menu.

Privilege Dictionaries

A group of four privilege dictionaries form the framework of the Security System. Figure 6–3 displays the location of these dictionaries within the DASL software.

Figure 6–3 Location of Security System Dictionaries



MR–1077–RA

You enter privilege codes in the Privilege Code Dictionary on the Development Environment Menu. Each privilege code is a single letter or character that corresponds to one or more DASL options. Any user who has a particular privilege can access the options associated with that privilege. For example, a user with the – privilege can access any option that also has the – privilege.

The Classification Dictionary Option is a choice on the Security System Menu. The Classification Dictionary contains user classifications. A user classification describes a type of user. Each user classification includes the following:

- Classification code
- Classification name
- Privilege list of privileges associated with the classification

DASL users can only run those options specified by the list of privilege codes associated with their classification. A user can have more than one classification.

The User Dictionary is a choice on the Security System Menu. It contains a series of user records, each describing an individual DASL user. Each user record contains the following information:

- User's name
- User's identification code
- User's title
- Reset password flag
- Active flag
- User's classification list

The Device Dictionary contains information about system input and output devices. It contains the following information about each device:

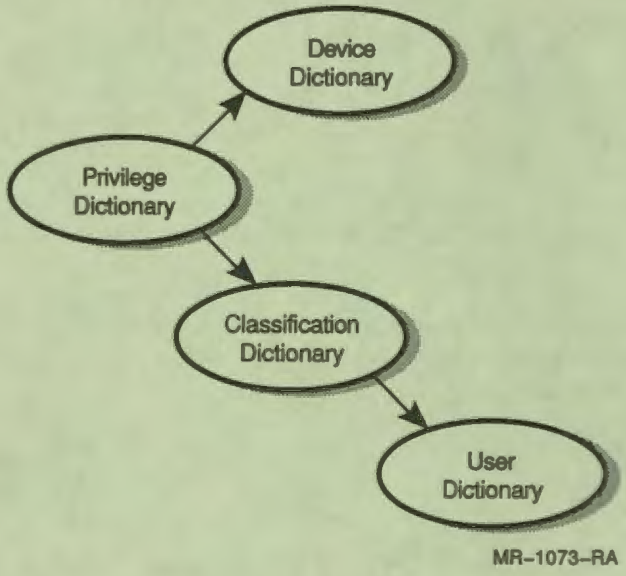
- Device name
- Device port
- Location/description
- Active flag
- Queue name
- Device attributes
- Device privileges

To use a specific device, the user must have privileges corresponding to the device privileges, and both the user and the device must have privileges corresponding to the privileges of the option being run.

Relationship of the Privilege Dictionaries

Figure 6–4 illustrates the relationship of the Privilege Dictionaries of the Security System.

Figure 6–4 Relationship of Security System Privilege Dictionaries



- Each privilege in the Privilege Code Dictionary describes one or more accessible facilities or options.
- Each classification in the Classification Dictionary contains one or more privileges.
- Each user record in the User Dictionary contains one or more classifications.
- Each device record on the Device Dictionary contains one or more privileges.

The DASL software determines the options available to a user by examining the privilege codes of:

- The user's User Classifications
- The user's current input and output device

The DASL software then compares these with the privileges attached to the option the user attempts to run.

If both the user and the device have the privilege to access an option, the user can run the option. If either the user or the device does not have the privilege, the user cannot run the option.

System Status Privileges

In addition to user privileges and device privileges, you use the System Status Dictionary on the Security System Menu to define system status codes for use with the System Control Option. Each system status record includes the following information:

- Name of the status
- Description
- Privilege codes
- Description of privilege codes

Each system status is associated with a system status code. Each system status code, in turn, is associated with a string of privilege codes. When a particular system status is in effect, only the privilege codes associated with its system status code are recognized by the DASL software.

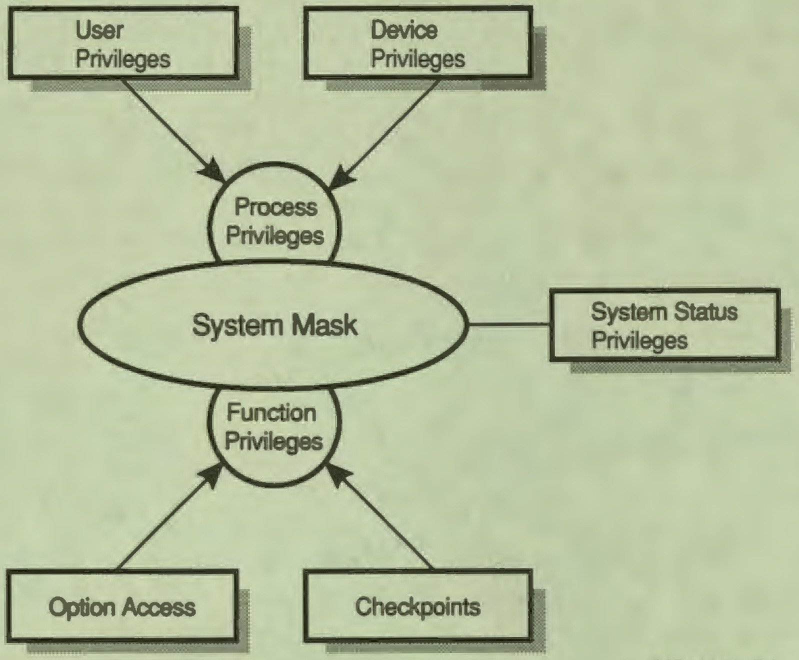
The system manager can change the system status at any time. When a system status is in effect, only those users whose classification codes include privileges that correspond to those of the current system status code can access DASL options. When users with an invalid code execute a checkpoint or attempt to access an option screen, the DASL software exits those users from the system.

For example, the CLOSED system status code has + privileges (system manager privileges). When the CLOSED status is in effect, only those users with + privileges can access options. Other current users are returned to the DASL Main Menu and logged out as they complete their current tasks.

Creating a System Mask

Setting a system status through the System Control Option creates a *system mask* that effectively modifies the privileges available to all users. Figure 6–5 illustrates how a system mask controls system security.

Figure 6–5 Function of the System Mask



MR–1078–RA

A user's ability to access options within the application system depends upon the system mask, which is the intersection of three sets of privileges:

- The user's process privileges
- The system status privileges
- The application's function privileges

The user's process privileges are determined at login: they are the intersection of the user's privileges described by the classification code, and the device privileges of the user's current input and output device.

The system status privileges further define a user's access to options at login. The system status privileges determine the current system mask that logically integrates overall system control with individual process privileges. A system manager can change the system status at any time during a session, creating a new system mask.

The application's function privileges are determined at the entry to each option screen and at each checkpoint in the application. The programmer assigns privileges to each option and defines checkpoints within options while developing the application in the development area. The function privilege determines option availability in option screens and function continuation at checkpoints.

Designing the Security System

The Security System Menu is intended for use in a completed application. However, you design the Security System in the DASL development area. Any user with system manager privileges can modify the Security System in the application, with the exceptions of the Privilege Code Dictionary and the Device Type Dictionary.

You must follow these steps to plan the Security System:

1. Define privilege codes in the Privilege Code Dictionary of the Development Environment Menu.
2. Assign privileges to menus and options through the Define Options Screen Option of the Screen Driver. To help with this task, you can view logical movement through menus and options by obtaining an Option Structure Report through the Print Option Structure Option of the Screen Driver.
3. Use the CHECKPT screen attribute to designate checkpoints within screens as you design each screen, if desired.
4. Assign privileges to tables through the Define Tables Option of the Query Driver Menu.
5. Define a set of user classifications and assign privileges to each classification in the Classification Code Dictionary of the Application Environment Menu.
6. Establish system users and assign classifications to each user in the User Dictionary of the Application Environment Menu. Define a generic system manager user record, or assign yourself system manager privileges, so that you or a designated system manager have access to all Security System Options in the completed application.
7. Define device attributes in the Device Type Dictionary of the Development Environment Menu.
8. Assign names, attributes, and privileges to devices in the Device Dictionary of the Application Environment Menu.

Incorporating Security System Options in an Application

After you have designed the Security System, you can incorporate all of the Security System Options in your application by specifying the screen name %OSEC in the screen definition for your application's Main Menu.

Note that all of the Security System Options have the + privilege code, requiring that a user have system manager privileges to use the options. If you do not want to restrict the entire set of Security System Options to users with system manager privileges, you must enter each option individually in your application menu, and assign the appropriate privilege code to each option.

See the *DASL Programmer's Guide* for more information on how to incorporate individual Application Environment screens and reports in the application.

Circumventing the Security System

You may not want to use the DASL Security System for some applications. You can take the following steps to circumvent the Security System:

1. Place one entry in each of the Security System dictionaries, particularly the User Dictionary and the Device Dictionary.
2. In the User Dictionary, assign a User ID that you can use with the DSM$DASL_USRID logical name. For example, you can assign AAA as the User ID.
3. Assign the @ privilege code to all entries in the Security System dictionaries.
4. Define the logical name DSM$DASL_USRID in the LOGIN.COM file. DSM$DASL_USRID equates to the User ID from the User Dictionary. For example, you can define DSM$DASL_USRID as follows:

   ```
   $ DEFINE/JOB DSM$DASL_USRID AAA
   ```

5. You can then assign each user the User ID of AAA. Because that User ID is assigned the @ privilege code, each user can access all options.

Related Sections

Classification Dictionary Option
CHECKPT, Chapter 2
Device Dictionary Option
Privilege Code Dictionary Option, Chapter 5
Security Reports Option
System Control Option
System Status Dictionary Option
User Dictionary Option

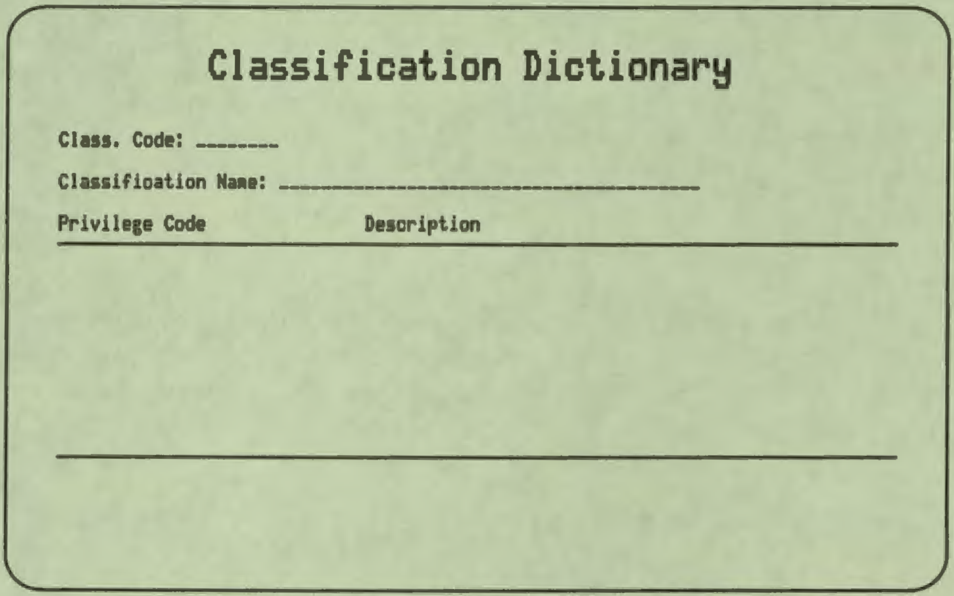
Classification Dictionary Option

The Classification Dictionary contains user classifications. A user classification is a definition of a type of user. You use the Classification Dictionary Option to create and edit user classifications.

Explanation

Figure 6–6 shows the Classification Dictionary screen.

Figure 6–6 Classification Dictionary Screen



```
                Classification Dictionary

Class. Code: ________

Classification Name: ________________________________________

Privilege Code               Description
```

Creating User Classifications

When you first select this option, the cursor moves to the "Class. Code:" prompt. You can enter a new classification code or enter an asterisk (*) to view a lookup list of existing codes.

Each classification includes the information described in Table 6–1.

Table 6–1 Classification Dictionary Information

Item	Description
Classification Code	A site-defined code that represents a group of users. This code can be an abbreviation such as NRS for nurse.
Classification Name	A description that reflects what the classification represents. For example, the classification name associated with a SYSMGR code can be "System Manager".
Privilege Code	A list of one or more privilege codes from the Privilege Code Dictionary that determine what options are available to users in this user classification. Users can only run those options specified by the privilege codes associated with their classification.
Description	A brief description of the privileges associated with the privilege code (from the Privilege Code Dictionary).

After you enter new information, the DASL software displays a "Save, Edit or Quit:" prompt. Enter S to save the new classification. You can then create more classifications, or press Exit to return to the Security System Menu.

Enter E to edit existing information, or Q to quit the screen without making changes.

Editing User Classifications

If you choose a code from the lookup list, the DASL software displays the Classification Dictionary entry for that code and an "Edit or Quit:" prompt.

When you edit an existing user classification, you can change the classification name, but you cannot change the classification code. You can also add privileges to, or delete privileges from, the classification.

Related Sections

Privilege Code Dictionary Option, Chapter 5
Security System Option

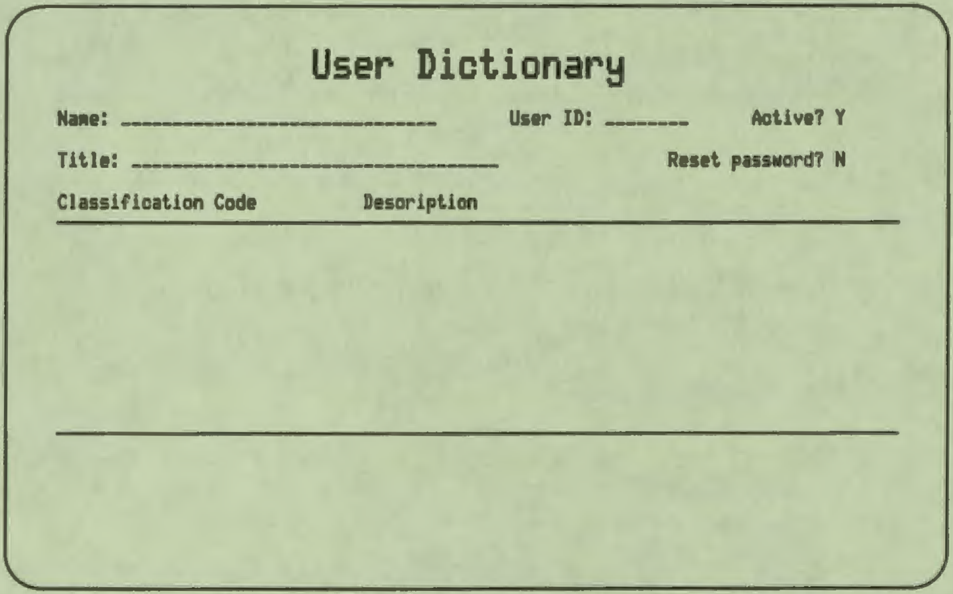
User Dictionary Option

The User Dictionary contains a series of user records. Each user record describes an individual DASL application user. You use the User Dictionary Option to create new user records or edit existing records.

Explanation

When you select the User Dictionary Option, the DASL software displays a screen like the one shown in Figure 6–7.

Figure 6–7 User Dictionary Screen



```
                     User Dictionary

Name: ____________________________     User ID: _______      Active? Y

Title: _________________________________          Reset password? N

Classification Code        Description
______________________________________________________________________
```

Table 6–2 shows the information you supply for each user record.

Table 6–2 User Dictionary Information

Item	Description
Name	The user's name in standard DASL name format of *lastname, firstname initial.*
User ID	The user name this user enters when logging in to the DASL application.
Active	A Y or an N that indicates if the user is active. (The default is Y.)
Title	The job title of the user or functional description of the user's job.
Reset password	A Y or an N that indicates whether DASL prompts the user for a new password at login.
Classification Code	A list of one or more classification codes from the Classification Code Dictionary.
Description	A brief description of the classification codes.

The DASL software recognizes two types of users:

- Active users
- Inactive users

Active users have an active flag value of Y. Active users can log in to the DASL application and perform whatever operations are permitted for their classification.

Inactive users have an active flag value of N. Inactive users cannot log in to the DASL application. However, the DASL software requires that inactive users be kept on file for reference purposes. You cannot delete a user from the User Dictionary.

Creating New User Records

When you create a new user record, enter all the information required, and save the record at the decision prompt. You can then create more user records, or press Exit to return to the Security System Menu.

Editing User Records

You normally edit existing user records to reset:

- The active flag value of a user
- The user's password

Deactivating or Reactivating a User

To deactivate a user, change the active flag value in the user's record from Y to N. That user then becomes inactive and can no longer log in to the application.

To reactivate an inactive user, change the active flag value from N to Y. That user then becomes active, and enters a new password when first logging in.

Setting and Changing Passwords

When you first create a user record, the DASL software places a Y at the "Reset password?" prompt. When the user logs in to the DASL application for the first time, the login screen prompts the user for a password for all subsequent logins.

After the user specifies a password, the DASL software changes the reset password flag value to N. If, after using the DASL application for a period of time, a user forgets the password, you can edit the user's record in the User Dictionary and enter Y at the "Reset password?" prompt. The next time the user logs in, the DASL software prompts for a new password.

> **Programmer's Note:** If you set the logical name DSM$DASL_USRID in your LOGIN.COM file, you can bypass the login screen for your application and test the application as though you had entered through the login screen.

Related Sections

Privilege Code Dictionary Option, Chapter 5
Security System Option

Device Dictionary Option

The Device Dictionary stores information about the DASL application's input and output devices. You create entries in the Device Dictionary for all input and output devices you use with your application. You cannot use any device in the application area that does not have a Device Dictionary record.

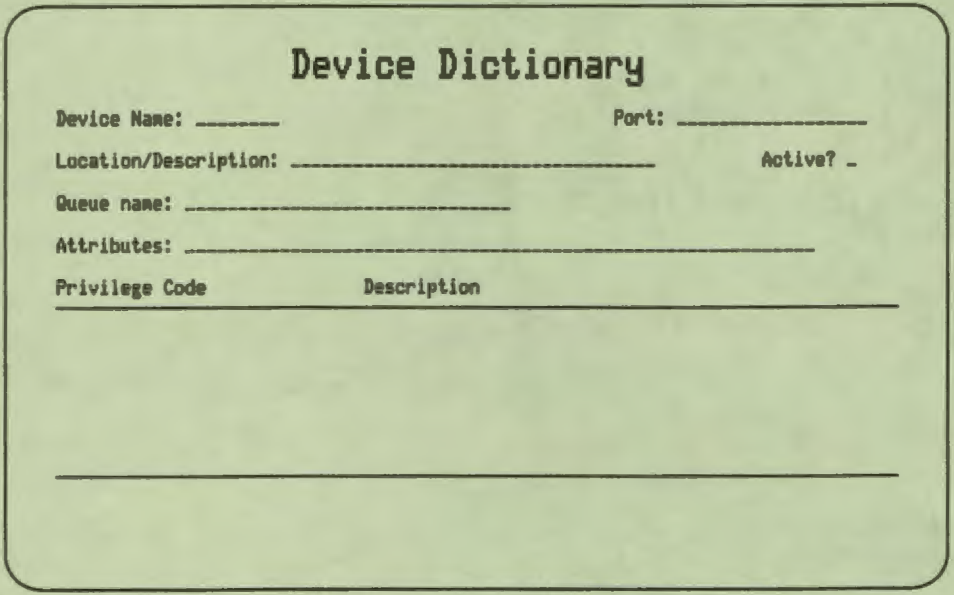
Explanation

The types of devices you enter in the Device Dictionary are physical devices or device queues for spooled devices. The Device Dictionary performs the following services:

- Provides the VMS operating system with information about the types of devices performing DASL application operations
- Establishes certain operating characteristics of the devices
- Restricts access to the DASL application according to the device being used

Figure 6–8 is a sample Device Dictionary screen.

Figure 6–8 Device Dictionary Screen



```
                    Device Dictionary
Device Name: _______                          Port: _______________
Location/Description: ______________________________       Active? _
Queue name: ____________________________
Attributes: ______________________________________________________
Privilege Code              Description
___________________________________________________________________

___________________________________________________________________
```

The Device Dictionary maintains the information shown in Table 6–3 for each input and output device.

Table 6–3 Device Dictionary Information

Item	Description
Device Name	A unique name that identifies the device to the application user. Users enter this code when a DASL screen prompts for output devices. DASL device names can be from one to eight characters in length.
Port	The port on the VMS device controller that connects the input and output device to the computer.
Location/Description	A description of the device or the area where the device is located.
Active	A Y or an N that indicates if the device is active (the default is Y).
Queue name	The name of the VMS print queue to be used when printing on this device.
Attributes	One or more codes identifying the device's type and operating characteristics.
Privilege Code	One or more access privilege codes from the Privilege Code Dictionary that specify the operations that can be performed using this device.
Description	A brief description of the privilege code.

Creating Device Entries

When you create a new device entry, enter all the information required, and save the information at the decision prompt. You can then create more entries, or press Exit to return to the Security System Menu.

Editing Device Entries

When you edit an existing device entry, you can change the device port, but you cannot change the device name. You can also add privileges to, or delete privileges from, the device entry.

Creating Device Dictionary Entries for a Terminal Switch System

On a DASL system that uses a terminal switching device, user terminals do not always have the same device port. When a user logs in to a system that uses a terminal switching device, the switching device assigns whatever line is currently available. A user may be assigned TTE6 one day, and TTB2 the next day.

If your system uses a terminal switching device, you must create entries in the Device Dictionary for all terminal ports. This covers all possible lines the switch can assign to users. As the device name, you can assign either a specific port designator such as TTD5, or a partial port designator name with an asterisk such as TT* or RT*. For each entry, use the port designator as the device name, and VT100 as the device attribute.

Give all terminal entries full privileges (the @ privilege).

Comments

See the "Device Ports" and "Device Attributes" sections of this chapter for more discussion of device ports and device attributes.

Sending Reports to a VMS Print Queue

You can specify the name of a VMS print queue for a device in the Device Dictionary. When you print reports, the reports are directed to the print queue.

> **Note:** Before you can print reports to a device to which you have assigned a print queue, the print queue must already exist at the VMS level. You must set up the print queue before you use the device to print reports.

Related Sections

Device Attributes
Device Ports
Device Specifications
Privilege Code Dictionary Option, Chapter 5

Device Specifications

Whenever you print a DASL report or one of the reports you create, you enter the device specification, the name of the device to which you want to send the reports.

Explanation

The device specifications you use depend on whether you are in:

- Development area
- Application area

Development Area Device Specifications

The DASL development area is an open area similar to DSM programmer mode. That is, the development area is the one in which application developers run DSM and then call the DASL software to create data names, screens, and reports.

When you are in the development area, you can specify any valid VMS device as the destination of the report. That is, you can send any DASL report to any of the following:

- A terminal
- A printer
- A file
- A queue
- Another network node

You can also enter the following shorthand device specifications:

. (period)	To display a DASL report on your current device (The DASL software displays the VMS specification of your current device as confirmation.)
.P	To print the report on a printer attached to your terminal (such as an LA50)

Application Area Device Specifications

The DASL application area is a closed area similar to DSM application mode. That is, the application area is the area in which users of your application operate.

In the application area, the DASL Security System controls the use of devices. You must create entries in the Security System Device Dictionary for all input and output devices you use with your application.

> **Note:** You cannot use any device in the application area that does not have a Device Dictionary Record.

The types of devices you can specify to the Device Dictionary are physical devices or device queues for spooled devices. You can designate a generic queue name for a physical device, and then specify the physical device to send reports to the generic queue. You cannot direct application reports to other network nodes or files. After you build your Device Dictionary, however, your users can specify the period (.) and .P conventions to display reports to their terminals or print them on their terminal printers.

> **Programmer's Note:** You can enter a routine to run after modifications to the Device Dictionary in the Application Parameters Option of the Development Environment Menu.

Related Sections

Application Parameters Option, Chapter 5
Device Attributes
Device Dictionary Option
Device Ports
Security System Option

Device Attributes

Device attributes are codes you specify in a device's Device Dictionary entry to describe that device's:

- Type
- Operating characteristics
- Relationship to the application you are creating

Explanation

Each device you enable for your applications must have one or more device attributes associated with it. You specify these attributes in the Device Dictionary of the Security System Option in the Application Environment.

However, you define device attributes in the Device Type Dictionary Option of the Development Environment Menu. **The DASL software does not accept device attributes in the Device Dictionary that you have not already defined through the Device Type Dictionary Option.** Table 6–4 lists the basic device attributes any application can use.

Table 6–4 Device Attributes

Attribute	Meaning
PRINTER	The device is an 80-column hardcopy printer.
VT100	The device is a VT100-type video display terminal.
VT200	The device is a VT200-type video display terminal.
VT200P	The device is a VT200-series terminal with an attached printer (such as an LA50).

You can also define other attributes to describe special-purpose devices attached to your system. See the "Device Type Dictionary Option" section of Chapter 5 for more information on defining device attributes.

Related Sections

Device Dictionary Option
Device Ports
Device Type Dictionary Option, Chapter 5
Privilege Code Dictionary Option, Chapter 5

Device Ports

Device ports are the ports on the VMS device controller that connect the device to the computer. The device port identifies the actual location of the device within the hardware configuration of your system.

Explanation

You specify a device port for each device in the Device Dictionary. The DASL software has two types of port designations, specific and generic. The specific port designation consists of a device mnemonic, a controller mnemonic, and a unit number. A colon (:) terminates all specific port designations. The generic port designation consists of a device mnemonic and a wildcard character, the asterisk (*).

For video display terminals, laser printers, and dot-matrix printers, the port designation takes the following format:

TT*CU*:

TT*

where:

TT	is a mnemonic for device type
C	is the controller designation (an uppercase letter)
U	is the unit number of the device
(:)	is the colon that terminates all specific port designations
*	is the wildcard character that terminates all generic port designations

Typical port designations of terminals are:

```
TTA0:

TTB4:

TT*
```

For line printers, port designations usually take the following form:

LP*CU*:

LP*

where:

LP	is a mnemonic for device type
C	is the controller designation (an uppercase letter)
U	is the unit number of the device
(:)	is the colon that terminates all port designations
*	is the wildcard character that terminates all generic port designations

Typical port designations for line printers are:

```
LPA0:
```

```
LP*
```

Comments

For more information on device designations and controller names, consult the *VMS Introduction to System Services* and the *VMS DCL Dictionary*.

Related Sections

Device Attributes
Device Dictionary Option
Device Specifications
VMS DCL Dictionary
VMS Introduction to System Services

System Status Dictionary Option

The System Status Dictionary defines system status codes. Each system status that you declare through the System Control Option has a system status code associated with it. You use the System Status Dictionary Option to create new system status codes or edit existing ones.

Explanation

Each system status is associated with one or more privilege codes that define access to the application system. When a particular status is in effect, the DASL software recognizes only the privilege codes associated with that status. Only those options associated with the recognized privilege codes are open to users. All other application options and facilities are closed.

The DASL software provides three predefined system status codes:

- OPEN
- CLOSED
- INQUIRY

The OPEN system status code has the @ privilege (full privileges). When the system status code is OPEN, all options and facilities are available to users with the privileges to access those options and facilities.

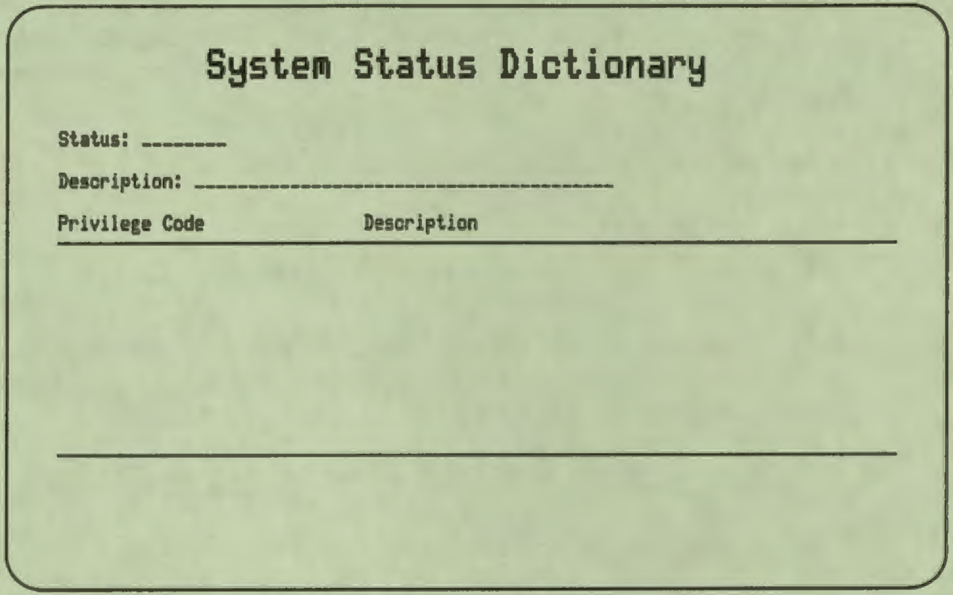
The CLOSED system status code has the + privilege (system manager privileges). When the system status code is CLOSED, the application is closed to users, except those users with system manager privileges. A user with system manager privileges can reopen the system.

The INQUIRY system status code has the = privilege (inquiry mode privileges). When the system status code is INQUIRY, any user with the = privilege can read data from the database, but cannot write anything to the database. See the "Inquiry Mode" section for information on the INQUIRY system status.

System Status Dictionary Screen

When you select the System Status Dictionary Option, the DASL software displays the screen shown in Figure 6–9.

Figure 6–9 System Status Dictionary Screen



```
                 System Status Dictionary

Status: --------
Description: ---------------------------------------
Privilege Code              Description
```

At the "Status:" prompt, you can enter the name of a new status, or enter an asterisk (*) and press Return for a lookup list of existing statuses.

Creating a New System Status

Table 6–5 describes the information you provide to create a new system status.

Table 6–5 System Status Information

Item	Description
Status	The name of a system status
Description	Brief description of the status
Privilege Code	A list of one or more privilege codes associated with the status
Description	A description of each privilege code

After you complete this information, a "Save, Edit, or Quit" prompt is displayed.

Editing an Existing Screen

If you enter an * at the "Status:" field, the DASL software displays a lookup list of previously defined system status codes. When you select an item from the lookup list, you can then edit information about that status code.

At the "Edit, Delete, or Quit:" prompt, you can:

- Enter E to edit existing information.
- Enter D to delete the system status code.

  Note: You cannot delete or modify the system status codes provided by the DASL software. You can, however, delete or modify system status codes you have created.

- Enter Q to quit the screen without saving changes.

Related Sections

Inquiry Mode
Privilege Code Dictionary Option, Chapter 5
System Control Option

Inquiry Mode

Inquiry mode is a system status that the system manager can set through the System Control Option. Inquiry mode provides users with read-only access to the application database, and is used in conjunction with a DSM write-locked database. Inquiry mode is especially useful when the system manager is performing an integrity check or a backup operation.

Implementing Inquiry Mode in the Development Area

The DASL software uses the scratch global, ^DATG, as a temporary sort global for reports, and for some scroll regions. When the INQUIRY system status is not in effect and the database is write-locked, users cannot run reports or use some scroll regions. If the INQUIRY system status is in effect, users can still perform inquiries that read information from the database, such as reports.

To use inquiry mode, you must have at least two volume sets in your database; for example, VOL 0 and VOL N. Store the main database in VOL 0, and store scratch globals, including the global ^DATG, in VOL N.

> **Note:** You can have any number of volume sets from 0 to N, where N is any number other than 0. However, you must have at least two volume sets.

At login, the DASL software checks the variable %WRITE, a flag that indicates system status. If VOL 0 is write-locked, the DASL software sets the value of the variable %WRITE to 0. The DASL software assumes that the status of VOL 0 is representative of all database volume sets, with the exception of VOL N, where your scratch globals are stored.

If the database is write-locked, the DASL software does the following:

- Does not set or kill the global ^DAJH, used for user utilization statistics, and the global ^DAJC, a list of current users
- Does not show the user system messages when logging in
- Logs system errors in a limited way

To include an inquiry mode capability in your application, you must give the = privilege code to all options you want users to access while in inquiry mode. You can also use the %WRITE flag to branch to different logic within data screens or routines. For example, you can specify that a user go to a Press Return to Exit field after an EVAL action has been performed, rather than proceeding to an Edit or Quit field.

You cannot perform any SET or KILL actions in the write-locked database.

Invoking Inquiry Mode in the Application area

You can then follow this procedure to extend inquiry mode to your application:

1. Set system status to INQUIRY.
2. Log out of the DASL software and DSM.
3. Log in to the DSM manager account using the DSM/MGR command.
4. In DSM, use the ^VOLINH Volume Set Access utility to change the main database volume set to no-write access (read-only).
5. Log out of the DSM manager account.
6. Log in to the DASL application using the application UCI. You can now only read data from the database; you cannot write to it.

When you change from inquiry mode to another system status, you must reverse this process as follows:

1. Log out of the DASL application.
2. Log in to the DSM manager account using the DSM/MGR command.
3. In DSM, use the ^VOLINH utility to change the main database volume set to allow users to write to the database.
4. Log in to the DASL application using the application UCI.
5. Change the system status from INQUIRY to the desired status.

Performing SET and KILL Actions

When in inquiry mode, the DASL software does not perform KILL or SET actions on DASL globals, other than the ^DATG scratch global.

The application developer must ensure that the application does not perform any global SET or KILL actions.

Related Sections

System Control Option
System Status Dictionary Listing
^VOLINH Utility, *VAX DSM Database Operations Guide*

System Control Option

You use the System Control Option to change the system status of the DASL application system. Using this option, you can close parts of the DASL application system while leaving other parts open to users. Because the DASL Security System is menu-based, changing the system status affects which options are available to users.

Closing an application means that you prevent further use of all or part of an application. Users cannot log in to parts of the application that are closed. Current users of those parts are returned to the application Main Menu and logged out as they finish their current tasks, because the DASL software performs a checkpoint at each option screen.

Using the System Status Dictionary

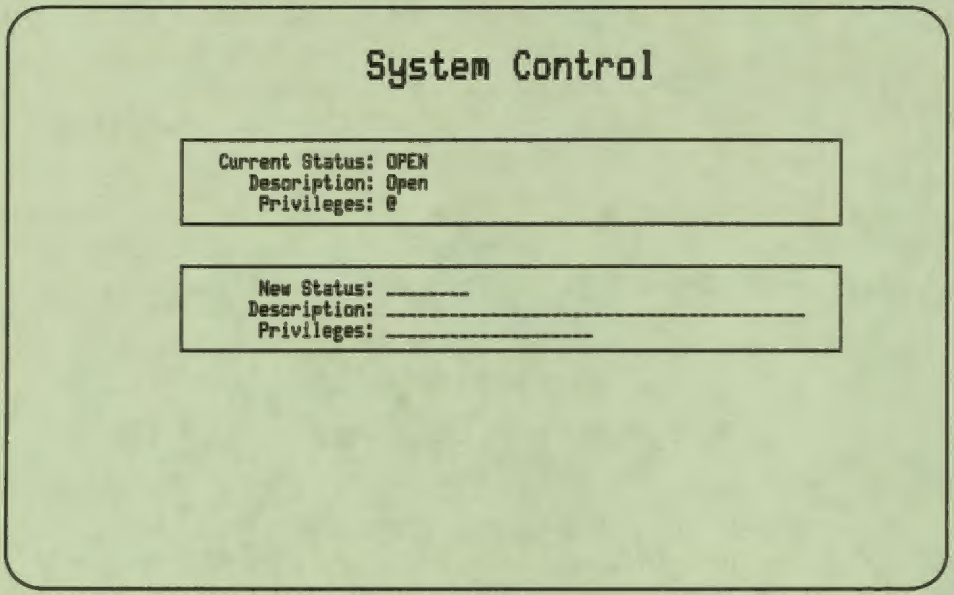
The System Status Dictionary defines system status codes for use with the System Control Option. Each system status code is associated with a string of privilege codes. When a particular system status code is in effect, only the privilege codes associated with that system status code are recognized by the DASL software. Furthermore, only those options associated with the recognized privilege codes are open to users.

For example, the CLOSED system status code is associated with the + privilege. When the CLOSED system status code is in effect, only users who have the + privilege (system manager) are allowed access to the application system and its options.

Changing the System Status

When you select the System Control Option, the DASL software displays a screen similar to the screen shown in Figure 6–10.

Figure 6–10 System Control Screen



```
                    System Control

     Current Status: OPEN
        Description: Open
         Privileges: @

         New Status: _______
        Description: ________________________________________
         Privileges: ___________________
```

The DASL software displays the current system status with its description and associated privileges. To change the system status, enter a new system status code at the "New Status:" prompt. Enter an asterisk (*) to view a lookup list of predefined status codes.

Setting Up a Temporary System Status Code

To set up a temporary status code (one that is not defined in the System Status Dictionary), press Return at the "New Status:" prompt. The cursor moves to the "Description:" prompt. Enter a brief description of your temporary system status code. At the "Privileges:" prompt, enter the privilege codes you want to associate with the temporary system status code. The temporary system status code stays in effect until you enter a new system status code through the System Control Option.

Programmer's Note: The DASL software performs a checkpoint at each option screen when you use CHECKPT as a screen attribute. At the checkpoint, the DASL software takes these steps:

1. Evaluates the system status code from the database, and concatenates a + with the system status code.
2. Performs a logical AND operation on system privileges and the user's process privileges.
3. Checks the system privileges and the user's process privileges for a match with option privileges, after performing the AND operation.
4. Exits from the option if the match fails.

Related Sections

CHECKPT, Chapter 2
Inquiry Mode
Privilege Code Dictionary Option, Chapter 5
System Status Dictionary Option

Security Reports Option

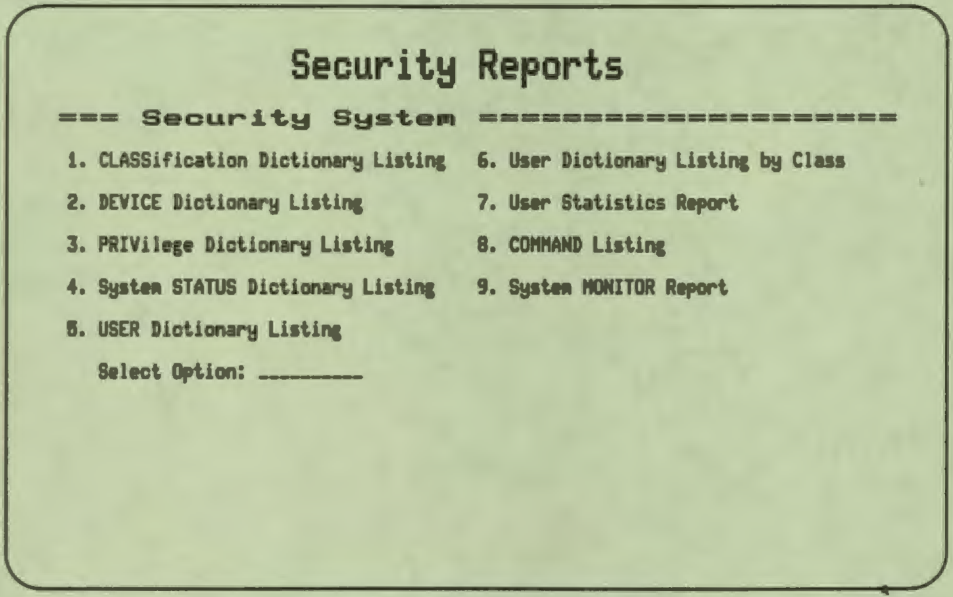
You use the Security Reports Option to produce reports on DASL privileges, classifications, users, and devices. You can also produce reports that list system status definitions and system commands.

> **Note:** You can only run security reports while you are in the application area.

Explanation

When you select the Security Reports Option, the DASL software displays a menu of report options you can choose. Figure 6–11 shows the Security Reports Menu.

Figure 6–11 Security Reports Menu



```
                     Security Reports
=== Security System ===================
 1. CLASSification Dictionary Listing   6. User Dictionary Listing by Class
 2. DEVICE Dictionary Listing           7. User Statistics Report
 3. PRIVilege Dictionary Listing        8. COMMAND Listing
 4. System STATUS Dictionary Listing    9. System MONITOR Report
 5. USER Dictionary Listing
    Select Option: _________
```

The following sections describe each security report.

Choosing a Destination

When you choose one of the Security Reports options, the DASL software displays a device selection screen that prompts you for the following information:

- Report Directory
- Device name
- Number of copies

- Start date
- Time

Enter Y to send your report to the Report Directory. (The default at this prompt is N.)

Enter the name of a device defined in the Device Dictionary as follows:

- If you do not send the report to the Report Directory, you can display the report on your terminal or print it on any VMS device that is defined in the Device Dictionary.
- If you send the report to the Report Directory, you can specify only a spooled device at this field. With a spooled device, the DASL software sends reports directed to that device into a spool file for temporary storage until the device is free to print the report. The contents of the spool file are queued to the device.

Enter the number of copies you want.

Enter the date and time that you want printing to begin.

At the “Continue, Edit, or Quit:” prompt, enter C to produce the report, E to edit the device selection screen, or Q to quit the screen without producing a report.

Some reports require further specifications. See the reference sections that follow on each security report for more information.

Related Sections

Classification Dictionary Listing
Command Listing
Device Dictionary Listing
System Monitor Report
System Status Dictionary Listing
User Dictionary Listing
User Dictionary Listing by Class
User Statistics Report

Classification Dictionary Listing

The Classification Dictionary Listing Option runs a report that lists all DASL user classifications. The report includes the classification codes, the privileges associated with them, and a brief description of the classification.

Example 6–1 is a sample Classification Dictionary Listing created through the Classification Dictionary Listing Option.

Example 6–1 Classification Dictionary Listing

```
8-Apr-88  10:21 AM     Introduction to                   Page 1
                            DASL

               Classification Dictionary Listing
------------------------------------------------------------------------
Classification          Privilege            Description
Code                      Code
------------------------------------------------------------------------
MINUS                       -                MINUS
                                             Minimum Privileges

PLUS                        +                PLUS
                                             System Manager Privileges

PROG                                         Programmer
                            +                System Manager Privileges
                            @                Full Privileges

------------------------------------------------------------------------
```

Related Sections

Classification Dictionary Option
Report Directory Option
Security Reports Option

Device Dictionary Listing

The Device Dictionary Listing Option runs a report that lists all DASL input and output devices. The report includes each device's name, attributes, port, and privileges.

Example 6–2 is a sample Device Dictionary Listing created through the Device Dictionary Listing Option.

Example 6–2 Device Dictionary Listing

```
8-Apr-88   10:23 AM   Introduction to                       Page 1
                          DASL

             Device Dictionary Listing by Device Name
--------------------------------------------------------------------
Device             Port                        Description
                   Attributes
                   Active Flag
                   Privileges
--------------------------------------------------------------------
ALLVT              TT*                         All VT terminals
                   VT100,VT200,VT200P
                   Y
                   @                           (Full privileges)

LASER              TTD1:                       LASER PRINTER
                   PRINTER
                   Y
                   @                           (Full privileges)
--------------------------------------------------------------------
```

When you choose the Device Dictionary Listing Option, the DASL software displays the following decision prompt:

```
Sort by Device or Port name:
```

Enter D to print out a listing sorted by device name. Enter P to print out a listing sorted by port name.

Related Sections

Device Dictionary Listing
Report Directory Option
Security Reports Option

Privilege Dictionary Listing

The Privilege Dictionary Listing Option runs a report that lists all DASL privilege codes used at your installation. The report includes the codes and descriptions of all privileges.

Example 6–3 is a sample Privilege Dictionary Listing created through the Privilege Dictionary Listing Option.

Example 6–3 Privilege Dictionary Listing

```
8-Apr-88     10:25 AM     Introduction to                       Page 1
                               DASL

                             Privilege File Listing
--------------------------------------------------------------------
Privilege Code               Description
--------------------------------------------------------------------

  @                          Full Privileges
  =                          Inquiry Mode Privileges
  -                          Minimum Privileges
  +                          System Manager Privileges

--------------------------------------------------------------------
```

Related Sections

Privilege Code Dictionary Option, Chapter 5
Report Directory Option
Security Reports Option

System Status Dictionary Listing

The System Status Dictionary Listing Option runs a report that lists the system status codes defined in the System Status Dictionary. The report includes the system status codes and a brief description of each status.

Example 6–4 is a sample System Status Dictionary Listing created through the System Status Dictionary Listing Option.

Example 6–4 System Status Dictionary Listing

```
8-Apr-88    10:26 AM    Introduction to               Page 1
                            DASL

                      System Status Report
--------------------------------------------------------------------
Status Name     Privilege Name      Description
--------------------------------------------------------------------
CLOSED                +             Closed
                                    System Manager Privileges

INQUIRY               =             Read Access Only
                                    Inquiry Mode Privileges

OPEN                  @             Open
                                    Full Privileges
--------------------------------------------------------------------
```

Related Sections

Report Directory Option
Security Reports Option
System Status Dictionary Option

User Dictionary Listing

The User Dictionary Listing Option runs a report that lists all DASL application users. The report includes each user's classification code and description, user ID, and title. The default User Dictionary Listing is sorted alphabetically by user name.

Example 6–5 is a sample User Dictionary Listing created through the User Dictionary Listing Option.

Example 6–5 User Dictionary Listing

```
6-Apr-88    10:28 AM    Introduction to                 Page 1
                           DASL

                    User Dictionary Listing
----------------------------------------------------------------------
                         Active Users
----------------------------------------------------------------------
   Name: BROWN, JANE                          Title: Writer
User ID: JB
  Class: PROG                                  Desc: Programmer

   Name: HILL, HARRY                          Title: Programmer
User ID: HOH
  Class: PROG                                  Desc: Programmer

   Name: SMITH, DALE                          Title: System Manager
User ID: DS
  Class: PROG                                  Desc: Programmer
         SYSMGR                                      System Manager

----------------------------------------------------------------------
```

Related Sections

Classification Dictionary Option
Report Directory Option
Security Reports Option
User Dictionary Listing by Class
User Dictionary Option

User Dictionary Listing by Class

The User Dictionary Listing by Class Option runs a report that lists all DASL application users by class. This report is similar to the User Dictionary Listing: it includes each user's classification code, user ID, and title. However, the list is sorted alphabetically by user classification.

Example 6–6 is a sample User Dictionary Listing by Class created through the User Dictionary Listing by Class Option.

Example 6–6 User Dictionary Listing by Class

```
8-Apr-88    10:30 AM    Introduction to                      Page 1
                            DASL

             User Dictionary Listing by Classification
-----------------------------------------------------------------------
Classification: PROG - Programmer

   Name: BROWN, JANE                          User ID: JB
  Title: Writer

   Name: HILL, HARRY                          User ID: HOH
  Title: Programmer

Classification: SYSMGR - System Manager

   Name: SMITH, DALE                          User ID: DS
  Title: Programmer
-----------------------------------------------------------------------
```

Related Sections

Classification Dictionary Option
Report Directory Option
Security Reports Option
User Dictionary Listing
User Dictionary Option

User Statistics Report

The User Statistics Report Option runs a report that lists the total number of people logged in to the system and the amount of time the users have been logged in. The report is sorted by date first, then alphabetically by user.

Example 6–7 is a portion of a User Statistics Report created through the User Statistics Report Option.

Example 6–7 User Statistics Report

```
8-Apr-88   10:32 AM        Introduction to                  Page 1
                                DASL

                          User Statistics Report

                     From: 15-Mar-88     To: 8-Apr-88

Date           User                 Log-ons          Total Time
----------------------------------------------------------------------
31-Mar-88      BROWN, JANE             1                0:05

1-Apr-88       BROWN, JANE             2                0:03
               HILL, HARRY             1                0:10

3-Apr-88       SMITH, DALE             3                0:40

7-Apr-88       BROWN, JANE             1                0:02

8-Apr-88       BROWN, JANE             1                0:03
                                      ---              ------
Totals:        15 Users               58                5:34

----------------------------------------------------------------------
```

For this report, you must supply the following information:

- From Date
- To Date

Enter the date from which you want the report to begin at the “From Date:” prompt. Enter the date at which you want the report to end at the “To Date:” prompt.

Related Sections

Report Directory Option
Security Reports Option

Command Listing

The Command Listing Option runs a report that lists all defined system and user commands that enable DASL application users to move to application options without accessing those options from menu screens. The report lists the commands in alphabetical order, and includes the user name of the command's owner and a description of the options accessed by the commands.

Example 6–8 is a sample Command Listing created through the Command Listing Option.

Example 6–8 Command Listing

```
8-Apr-88  10:34 AM        Introduction to              Page 1
                              DASL
                     Security Command Listing
-----------------------------------------------------------------
User                      Command          Description
-----------------------------------------------------------------
System                    DDN              Data Dictionary
                          BOOK             Address Book
                          DD               Device Dictionary
                          SC               System Control
                          UTILS            User Utilities
-----------------------------------------------------------------
```

Related Sections

Command Mode, Chapter 2
Define Option Screens Option, Chapter 2
Report Directory Option
Security Reports Option

System Monitor Report

The System Monitor Report Option runs a report listing the users currently logged in to the DASL application. For each user, the report includes the option being used, the user's device, and the time the user logged in.

Example 6–9 is a sample System Monitor Report created through the System Monitor Report Option.

Example 6–9 System Monitor Report

```
8-Apr-88    10:35 AM      Introduction to                Page 1
                               DASL

                         System Monitor Report

                  Start        Elapsed Time
User              Time         (hh:mm:ss)     Device    Option
-------------------------------------------------------------------------

SMITH, DALE        3:20 PM     9:20:36        TTD4:     Print Address Book
                   1:49 PM     1:49:09        TTA4:     %LOGIN
BROWN, JANE       10:20 AM     0:15:12        TTE6:     System Monitor Report
                  11:33 AM     1:33:22        LASER     SEC
HILL, HARRY       10:44 AM     2:44:29        TTE5:     %LOGIN

-------------------------------------------------------------------------
```

Related Sections

Report Directory Option
Security Reports Option

Edit User Password Option

You use the Edit User Password Option to change your existing password.

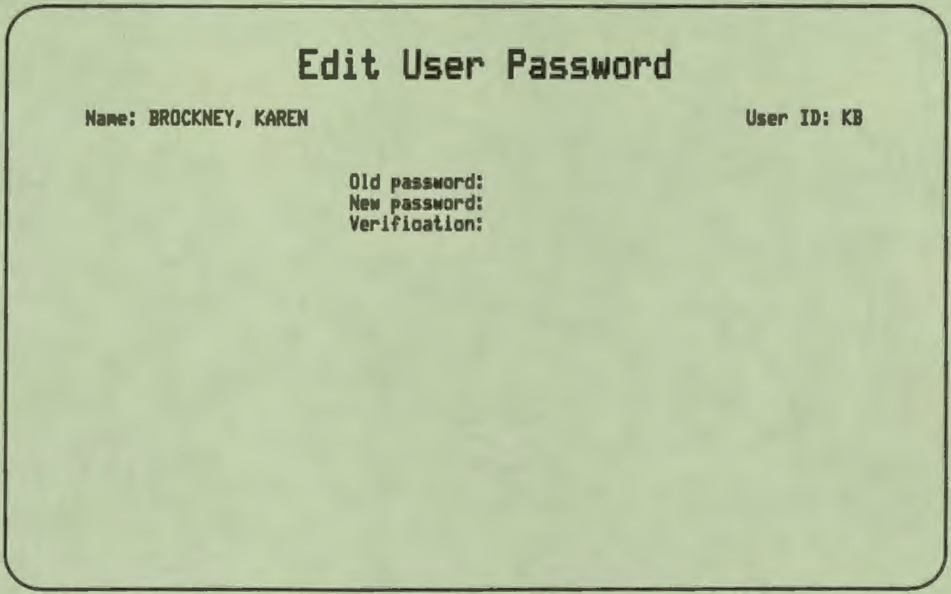
Explanation

For each application, the DASL software displays a login screen that prompts you for a User ID and Password. You enter a password into the database the first time you log in to the application system. These passwords are not stored in any DASL dictionary and cannot be reviewed by any DASL user, including the system manager.

If a user forgets the password, the system manager can reset that user's password through the User Dictionary Option. When the system manager resets a user's password, the DASL software prompts the user for a new password the next time that user logs in to the application.

Alternatively, you can change your own password at any time by using the Edit User Password Option. When you select this option, the DASL software displays a screen similar to the screen shown in Figure 6–12.

Figure 6–12 Edit User Password Screen



```
                    Edit User Password

Name: BROCKNEY, KAREN                              User ID: KB

                      Old password:
                      New password:
                      Verification:
```

In the Edit User Password screen, the DASL software displays your user name and user ID.

Take the following steps to change your password:

1. Enter your current password at the "Old password:" prompt.
2. Enter your new password at the "New password:" prompt.
3. Enter your new password again for verification at the "Verification:" prompt.

   If the verification fails, the DASL software displays the message "Verification failed, password not changed". Enter your new password again at the "New password:" prompt, and verify it at the "Verification:" prompt.

When you have entered and verified your new password, the DASL software displays the "Save, Edit, or Quit:" prompt. Enter S and press Return to save your new password.

Related Sections

Classification Dictionary Option
User Dictionary Option

Edit Installation Name Option

You use the Edit Installation Name Option to specify two lines of text to appear on the application login screen. This text is the name of your installation or your application.

Explanation

When you select the Edit Installation Name Option, the DASL software displays the Edit Installation Name screen. This screen prompts you for the following information:

```
Installation Name Line 1:
Installation Name Line 2:
```

Creating a New Installation Name

To create a new installation name, enter the first line of text that you want to display on the application login screen at the "Installation Name Line 1:" prompt.

Enter the second line of text that you want to display on the application login screen at the "Installation Name Line 2:" prompt.

The DASL software centers the text you specify on the application login screen.

After you enter the installation name, the DASL software displays the "Save, Edit, or Quit:" prompt. Enter S and press Return to save the new installation name.

Editing an Existing Installation Name

If there is an existing installation name, the DASL software displays that name on the screen and then displays an "Edit or Quit:" prompt.

Report Directory Option

Application users can access the DASL Report Directory through the Report Directory Option. The Report Directory is a combined VAX DSM and VMS facility that allows application users to store reports and queries in VMS sequential files, and control those reports and queries through an application option.

Explanation

Reports and queries in the Report Directory are stored in a VMS directory in files. The Report Directory is a MUMPS global. It is an archive of the reports and queries that users want to keep for future reference or later printing. Once submitted to the Report Directory, a report or query remains there until the user who submitted it (the owner) deletes it.

Sending a Report or Query to the Report Directory

Each DASL application has the ability to produce reports and queries. The Main Menu for each application can list the options used to produce reports in a submenu called Print Options. The Query Database Menu contains options to define or run queries.

The DASL software displays a device selection screen when a user selects a report or runs a query. In this screen, the user can enter Y at the "Report Directory?" prompt to send the report or query result to the Report Directory.

Users can send a report to the Report Directory and to an output device in the same device selection screen.

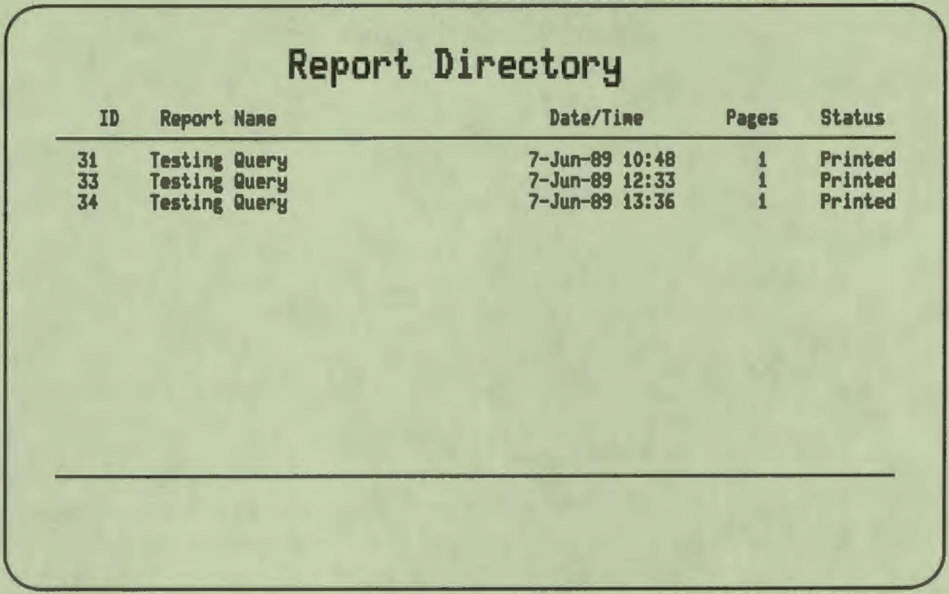
Report Directory Screen

When users select the Report Directory Option, the Report Directory screen is displayed. The Report Directory screen is a scroll region that lists reports and queries sent to the Report Directory from any part of the DASL application system.

The Report Directory screen only shows reports or queries that the user has privileges to submit to the Report Directory. For example, if a user does not have privileges to access security reports, the user does not see any security reports listed in the Report Directory.

Figure 6–13 is a sample Report Directory screen.

Figure 6–13 Report Directory Screen



```
                         Report Directory
  ID   Report Name                         Date/Time       Pages  Status
 ---------------------------------------------------------------------------
 31    Testing Query                   7-Jun-89 10:48      1    Printed
 33    Testing Query                   7-Jun-89 12:33      1    Printed
 34    Testing Query                   7-Jun-89 13:36      1    Printed
```

Table 6–6 describes the information that the Report Directory maintains on each report.

Table 6–6 Report Directory Information

Item	Meaning
ID	The internally generated number that the DASL software uses as part of the file name of the VMS file in which the completed report is stored. For example, the report listed as 33 in the Report Directory is the VMS file filename_33.DAT.
Report Name	A brief description of the report or query, such as “Address Book — Print All.”
Date/Time	The date and time when the report was collected, or will be collected.
Pages	The number of pages in the report.
Status	The report status, one of the following: *Queued* — The report is not yet compiled. *Prepared* — The report is compiled and written to a file, but not printed. *Printed* — The report was printed at least once. *Error* — An error occurred during compilation.

The *DASL Handbook* contains instructions for DASL application users on how to:

- Move the cursor within the scroll region
- Access online help
- Print reports and queries from the Report Directory
- Delete reports and queries
- List the contents of the Report Directory
- List the owner of the report or query
- Maintain the Report Directory

Routines Used by the Report Driver

The DASL Report Directory consists of a set of routines to coordinate the collection and cataloging of reports and queries. The routine ^%DARPDRV schedules and runs reports using the parameters %RPNAME, %RPFNC, %RPDEV, %RPCOPY, %RPTIME, and %RPPAGE. The variable %RPFNC indicates whether a report or query is sent to the Report Directory. You must set %RPFNC equal to D to send reports and queries to the Report Directory.

See the "Report Driver Variables" and the "Calling Reports from Data Screens and Option Screens" sections in Chapter 3 for more information about Report Driver Variables.

Related Sections

Calling Reports from Data Screens and Option Screens, Chapter 3
DASL Handbook
Report Driver Variables, Chapter 3

Message Center Option

Application users use the Message Center to communicate with other DASL application users. The Message Center operates only within the application area (within a DASL-created application).

Explanation

The Message Center handles two types of messages:

- System messages
- User messages

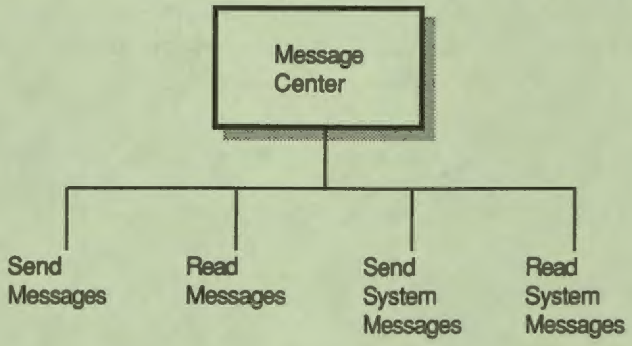
System messages usually contain information about the DASL application that every user needs to know. Users view system messages at login. **Only users with + privileges (system manager) can send system messages.**

User messages contain information from other application users. Any application user with – privileges can send and read user messages using the DASL Message Center Options. The DASL software notifies users of new mail at login and at option screen entry.

Message Center Options

Figure 6–14 illustrates the options of the Message Center Menu.

Figure 6–14 Message Center Options



MR–1074–RA

Application users can choose the following options:

- Send Messages

  With the Send Messages Option, application users can send user messages to other users.

- Read Messages

  With the Read Messages Option, application users can read user messages, and then select from a list of utilities to delete, print, list messages on the screen, continue, or quit.

- Send System Messages

  With the Send System Messages Option, anyone with + privileges (system manager) can send system messages to all application users.

- Read System Messages

  With the Read System Messages Option, anyone with + privileges (system manager) can perform a variety of tasks that include deleting selected messages for redistribution to the system, printing system messages, or listing system messages.

All DASL users can read system messages at login, and use the Message Center to read and send user messages. However, only users with system manager privileges can use the Message Center Options for sending and deleting system messages.

The Message Center application user options are described in the *DASL Handbook*. The Message Center system manager options are described in the following sections of the *DASL Reference Manual*. The following table shows where you can find information about each option:

Option	Location
Message Center Menu	Chapter 6, *DASL Reference Manual*
Send Messages	*DASL Handbook*
Read Messages	*DASL Handbook*
Send System Messages	Chapter 6, *DASL Reference Manual*
Read System Messages	Chapter 6, *DASL Reference Manual*

Including the Message Center Options in the Application

Enter %OMESS in the “Screen Name” field of the Define Option Screens Screen when you create the menu for the Message Center to include all four Message Center Options in your application. If you do not specify %OMESS, and incorporate each Message Center Option separately, specify screen names and set the variable %OPP as follows:

- For the Send Messages Option, specify the screen name, %MESSWR, and set the variable %OPP="USR."
- For the Read Messages Option, specify the screen name, %MESSRD, and set the variable %OPP="USR."
- For the Send System Messages Option, specify the screen name, %MESSWR, and set the variable %OPP="SYS."
- For the Read System Messages Option, specify the screen name, %MESSRD, and set the variable %OPP="SYS."

The variable %OPP is a. flag that indicates whether an option is a user option or a system manager option.

Related Sections

DASL Handbook
Privilege Code Dictionary Option, Chapter 5
Read System Messages Option
Send System Messages Option

Send System Messages Option

While all application users can use the Message Center to read and send user messages, only the application's system manager (or a user with + privileges) can use the Message Center options for reading and sending system messages. The system manager can use the Send System Messages Option of the Message Center to send messages to all application users.

Explanation

System messages contain information that all DASL application users need to know.

The DASL software displays system messages to users after they log in, but before it displays the application Main Menu. The messages are displayed in reverse chronological order with the most recent message appearing first. All users (including the system manager) must read system messages before beginning an application session.

Reading System Messages at Login

At application login, the DASL software notifies you of messages received in the following order:

- First, the DASL software displays the number of unread user messages received. You must choose the Read User Messages Option to read your user messages.
- Then, any system messages are displayed on the screen. System messages are displayed in reverse chronological order, beginning with the most recent system message. Press Return after you read a system message to erase the message and display the next system message. Press Exit to skip reading the system messages.

  Once you erase a system message, you cannot return to it without logging out and logging in again. Therefore, be sure that you understand a message before pressing Return.

Send System Messages Screen

The Send System Messages Screen has three areas:

- A heading that indicates that the message will be sent to all users:

  ```
  To: All
  ```

  In this area, the DASL software also displays a "cc:" prompt and a "Subject:" prompt.
- A scroll region in which to enter the message

- A decision prompt (not displayed when you first enter the option)

After the DASL software displays the screen, it moves the cursor to the "Subject:" prompt. The DASL software sends all system messages to all users; you cannot edit the "To:" or "cc:" fields.

At the "Subject:" prompt, enter the subject of the message. Press Return to move to the scroll message region. Enter the text of the message you want to send.

After you type each line of the message, press Return. Do not type more than 70 characters on one line because the Message Region does not display more than 70 characters on a line.

When you finish entering the message, press Exit.

After you press Exit, a "Save, Edit, or Quit?" decision prompt appears. Enter one of the following:

- Enter S to send the message to all system users. The DASL software sends the message, clears the screen, and displays the Message Center Menu again.

  When you enter S, the DASL software asks if you want to print the last message. Enter Y at this prompt to save a copy of the message. The DASL software then displays the device selection screen. You can send the mail message to the Report Directory, specify a VMS file, or print it on any VMS device.

- Enter E to edit the subject or the message text.

  You can use the arrow keys for editing. The up and down arrow keys move from line to line. You can also press ⇐ or ⇒ to use the DASL field editor.

  Each line of the message is a complete unit. Therefore, if you need to change a word, you must retype the entire line containing the word. If you do not retype the entire line, only the part that you retype is entered when you press Return.

  If you want to add a blank line to the message, press Insert Here at the position where you want the blank line to appear.

  If you need to change an entire line of text, press Remove to delete that line. Then, retype the entire corrected line and press Return.

  When you finish editing the message, press Exit to display the decision prompt again. Then, enter S to send the message. The DASL software sends the message, clears the screen, and displays the Message Center Menu again.

- Enter Q to quit the message screen. By entering Q, you prevent the message from being sent. The DASL software clears the screen and displays the Message Center Menu again.

Related Sections

DASL Handbook
Message Center Option
Privilege Code Dictionary Option, Chapter 5
Read System Messages Option

Read System Messages Option

You use the Read System Messages Option to perform a variety of system manager tasks. Only the application's system manager (or a user with + privileges) can use this option. The system manager can use the Read System Messages Option of the Message Center to delete system messages, print system messages on a printing device, and list system messages on the terminal screen.

Explanation

Users continue to see a system message until the system manager explicitly deletes that message through the Read System Messages Option.

When you select the Read System Messages Option, the DASL software displays the Read System Messages Screen. The DASL software displays the most recent system message you sent, and the following decision prompt at the bottom of the screen asking you what you want to do with the message.

```
Delete, Print, List, Continue, or Quit:
```

Enter one of the following at the decision prompt:

- Enter D to delete the message and remove it from distribution.
- Enter P to print the message on a hardcopy device. The DASL software then prompts you for the name of an output device.

  All subsequent messages printed from the Message Center in this session use the same printer that you specified for the first message.

  After you enter the output device, the DASL software displays the decision prompt. You can choose any of the other decision prompt options.
- Enter L to list the system messages in a lookup list on the terminal screen. The system messages appear on the list in reverse chronological order. You can then select a message from the list.
- Enter C to continue. The previous system message (if any) appears.
- Enter Q to quit the screen without performing any actions. The DASL software erases the screen and displays the Message Center Menu again.

Related Sections

DASL Handbook
Message Center Option
Privilege Code Dictionary Option, Chapter 5
Read System Messages Option

Event Logging Option

The Event Logging Option is a menu of options you and users of your applications can use to control and maintain the System Event Log.

System Event Log

During the routine operation of a DASL application, events and errors occur. Events can be normal system events such as changing system status, or events and errors can be caused by hardware failure, software errors, or incorrect use of the system. The DASL software logs such events and errors in a System Event Log, which is a file in the DASL application system.

The system manager uses the Error/Event Code Dictionary Option of the Development Environment Menu to define error or event messages that the error and event logging facility registers in the System Event Log. **The DASL software does not register any error or event in the System Event Log that is not defined in the Error/Event Code Dictionary.**

Event messages fall into two categories:

- Program error messages
- Status messages

A *program error message* reports a software-related problem. For example, an error in a DASL application routine produces a program error message in the System Event Log. If a user encounters a program error during a work session, The DASL software does the following:

1. Logs the error in the System Event Log (and, optionally, displays the event message on the event logging device).
2. Sends a message to the user's terminal instructing the user to contact the system manager.
3. Logs the user out of the system.

A *status message* reports the occurrence of a specific DASL operation. Generally, status messages require no corrective action.

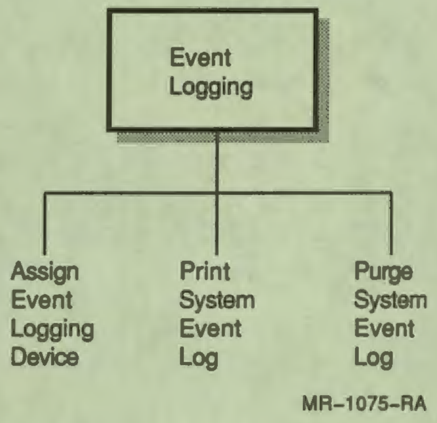
The DASL software does not report status events to most system users. It does, however, report some events that affect the entire system (such as system close) to all users.

The application system manager regularly reviews the information in the System Event Log and decides what action to take to remedy errors.

Event Logging Menu Options

Figure 6–15 illustrates the options on the Event Logging Menu.

Figure 6–15 Event Logging Menu Options



Explanation

As Figure 6–15 shows, the Event Logging Menu has three options:

- Assign Event Logging Device

  You use the Assign Event Logging Device Option to specify a terminal or printer to receive all error and event messages.

- Print System Event Log

  You use the Print System Event Log Option to produce a report listing the error and status events that have occurred since the last time you purged the Event Log.

- Purge System Event Log

  You use the Purge System Event Log Option to delete all current entries from the System Event Log and open the System Event Log for new events.

Related Sections

Assign Event Logging Device Option
Device Dictionary Option
Error/Event Code Dictionary Option
Print System Event Log Option
Purge System Event Log Option

Assign Event Logging Device Option

You use the Assign Event Logging Device Option to specify a device on which to print all event or error messages sent to the System Event Log.

Explanation

When you select the Assign Event Logging Device Option, the DASL software displays the Assign Event Logging Device screen. This screen prompts you for the name of the event logging device.

To send error and status messages to a hardcopy device (as well as to the System Event Log), enter the name of the device you want to assign after the "Event Log Device:" prompt. The device you specify must be listed in the Device Dictionary of the Security System.

After you specify an event logging device, all error and status messages go to that device until you redirect event logging to another device. Therefore, whenever you choose the Assign Event Logging Device Option after your initial device assignment, the name of the device you first specified appears as a default after the "Event Log Device:" prompt.

To change event logging devices, type the new event logging device name over the old event logging device name at the "Event Log Device:" prompt.

> **Note:** You do not have to assign an event logging device. If you do not specify an event logging device, the DASL software sends all event and error messages only to the System Event Log.

Related Sections

Device Dictionary Option
Error/Event Code Dictionary Option
Print System Event Log Option
Purge System Event Log Option

Print System Event Log Option

The Print System Event Log Option produces a report listing the error and status events that occurred since the last time you purged the System Event Log. **This report is the only permanent record of system events.** You must be in the application area to run this option.

Example 6–10 shows an excerpt from a System Event Log.

Example 6–10 System Event Log

```
                      DEMO Event Log
                   27-FEB-1988  11:25:25

DEMO  27-FEB-1988  11:25:25  Event 5: Program Error
     DO+5%DARUN, %DSM-E-INTVER, Interpreter / Code Mismatch
     Device: TTA5:  User: BROWN, JANE  Option: DEMO
DEMO  27-FEB-1988  14:31:02  Event 5: Program Error
     EXIT+1^%DANAME, %DSM-E_ZTRAP, software trap received, Undefined Screen
     Device: TTE6:  User: HILL, HARRY  Option: Print Entire Address Book
```

Explanation

When you select the Print System Event Log Option, the DASL software prompts you for the name of the device to which you want to send the log. You can only specify the terminal you are currently using or a device listed in the Device Dictionary as a printer.

The System Event Log lists events chronologically, with the earliest event at the top of the log. The System Event Log includes the following information for each event:

- Date of the event
- Time the event occurred
- Message identification number
- Category of the event (error or status)
- Description of the event
- Device
- User name
- Option where event occurred

Related Sections

Device Dictionary Option
Error/Event Code Dictionary Option
Purge System Event Log Option

Purge System Event Log Option

You use the Purge System Event Log Option to delete all data from the System Event Log and open the System Event Log for new events.

Explanation

When you select this option, the DASL software displays the following prompt:

```
Are you ready to purge the Event Log?
```

Enter Y and press Return to purge the System Event Log. Enter N if you decide not to purge the log at this session.

Run this option only after you print the System Event Log. Otherwise, you lose a record of all logged events.

Unless you are experiencing frequent problems, purge the system event log weekly. When you run the Purge System Event Log Option, VMS automatically recovers the disk space the file occupied.

Using the Site Parameters Option to Purge the Event Log

You can use the Site Parameters Option of the Development Environment to direct the DASL software to automatically purge the system event log.

Enter Y at the "Auto purge error log:" prompt in the Site Parameters screen. The DASL software then displays the "After how many days?" prompt.

Enter the number of days for which you want to keep system event log records. For example, if you enter 10 days at this prompt, the DASL software purges errors and events that occurred more than 10 days before the current date.

Related Sections

Error/Event Code Dictionary Option, Chapter 5
Print System Event Log Option
Site Parameters Option, Chapter 5

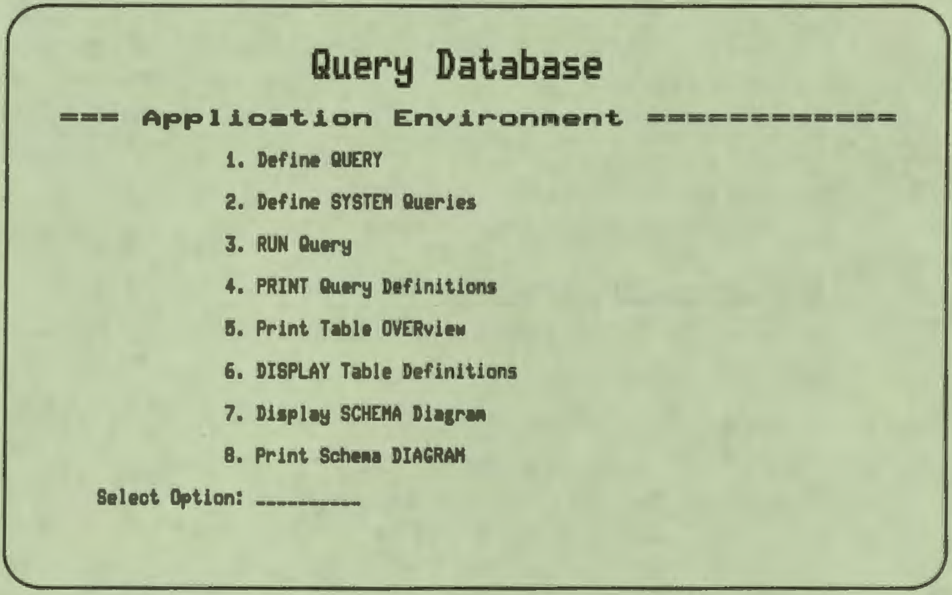
Query Database Option

You use the Query Database Option to define and run database queries for your application. You can also use the options on this menu to view or print a schema diagram that displays the tables in the database and to print reports of query definitions or table definitions.

Query Database Options

Figure 6–16 shows the screen you see when you select the Query Database Option.

Figure 6–16 Query Database Screen



```
                     Query Database
=== Applioation Environment ============
              1. Define QUERY
              2. Define SYSTEM Queries
              3. RUN Query
              4. PRINT Query Definitions
              5. Print Table OVERview
              6. DISPLAY Table Definitions
              7. Display SCHEMA Diagram
              8. Print Schema DIAGRAM
       Select Option: __________
```

You can choose any of the following options:

- Define Query

  Application users can define user queries or system queries for the database. Users can also use this option to run, edit, delete, copy, print, or add comments to existing queries.

- Define System Queries

  Application managers can define or edit system queries for the database. Application managers can also use this option to run, edit, delete, copy, print, or add comments to existing system queries.

- Run Query

  You can use this option to run an existing query.

- Print Query Definitions

  You can use this option to print a report that displays the definitions of requested queries.

- Print Table Overview

  You can use this option to print a brief overview of the contents of tables that you specify. You can direct the report to the Report Directory, your terminal, or a printer.

- Display Table Definitions

  You can display a brief report of table definitions on the terminal screen.

- Display Schema Diagram

  You can display a schema diagram of all the tables in the database on the terminal screen.

- Print Schema Diagram

  You can print the schema diagram of all the tables in the database in a 132-column report.

See Chapter 4 for descriptions of each of the Query Database options.

Related Sections

Define Query Option, Chapter 4
Define System Queries Option, Chapter 4
Display Schema Diagram Option, Chapter 4
Display Table Definitions Option, Chapter 4
Print Query Definitions Option, Chapter 4
Print Table Overview Option, Chapter 4
Run Query Option, Chapter 4

Appendix A
ASCII and Multinational Character Sets

This appendix contains tables describing the ASCII Character Set, the DEC Multinational Character Set, and the ISO Latin-1 Character Set.

Figure A–1 lists characters of the ASCII character set with decimal codes of 1 through 128.

Figure A-1 ASCII Character Set

Decimal Code	Character	Decimal Code	Character	Decimal Code	Character
0	NUL	49	1	98	b
1	SOH	50	2	99	c
2	STX	51	3	100	d
3	ETX	52	4	101	e
4	EOT	53	5	102	f
5	ENQ	54	6	103	g
6	ACK	55	7	104	h
7	BEL	56	8	105	i
8	BS	57	9	106	j
9	HT	58	: (colon)	107	k
10	LF	59	; (semicolon)	108	l
11	VT	60	<	109	m
12	FF	61	=	110	n
13	CR	62	>	111	o
14	SO	63	?	112	p
15	SI	64	@	113	q
16	DLE	65	A	114	r
17	DC1	66	B	115	s
18	DC2	67	C	116	t
19	DC3	68	D	117	u
20	DC4	69	E	118	v
21	NAK	70	F	119	w
22	SYN	71	G	120	x
23	ETB	72	H	121	y
24	CAN	73	I	122	z
25	EM	74	J	123	{
26	SUB	75	K	124	|
27	ESC	76	L	125	}
28	FS	77	M	126	~
29	GS	78	N	127	DEL (RUBOUT)
30	RS	79	O		
31	US	80	P		
32	SP (blank)	81	Q		
33	!	82	R		
34	"	83	S		
35	#	84	T		
36	$	85	U		
37	%	86	V		
38	&	87	W		
39	' (apostrophe)	88	X		
40	(	89	Y		
41	)	90	Z		
42	*	91	[		
43	+	92	\		
44	, (comma)	93	]		
45	- (hyphen)	94	^		
46	. (period)	95	_ (underscore)		
47	/	96	`		
48	0	97	a		

MR-1366-RA

Figure A–2 lists characters of the DEC Multinational Character Set with decimal codes of 161 through 255.

Figure A–2 DEC Multinational Character Set

B8 B7 B6 B5 BITS		8		9		10		11		12		13		14		15	
B4 B3 B2 B1	Row	1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1	
0 0 0 0	0	▨	200 128 80	▨	220 144 90	▨	240 160 A0	°	260 176 B0	À	300 192 C0		320 208 D0	à	340 224 E0		360 240 F0
0 0 0 1	1	▨	201 129 81	▨	221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1
0 0 1 0	2	▨	202 130 82	▨	222 146 92	¢	242 162 A2	²	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2
0 0 1 1	3	▨	203 131 83	▨	223 147 93	£	243 163 A3	³	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3
0 1 0 0	4	▨	204 132 84	▨	224 148 94		244 164 A4		264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4
0 1 0 1	5	▨	205 133 85	▨	225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5
0 1 1 0	6	▨	206 134 86	▨	226 150 96		246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6
0 1 1 1	7	▨	207 135 87	▨	227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	Œ	327 215 D7	ç	347 231 E7	œ	367 247 F7
1 0 0 0	8	▨	210 136 88	▨	230 152 98	¤	250 168 A8		270 184 B8	È	310 200 C8	Ø	330 216 D8	è	350 232 E8	ø	370 248 F8
1 0 0 1	9	▨	211 137 89	▨	231 153 99	©	251 169 A9	¹	271 185 B9	É	311 201 C9	Ù	331 217 D9	é	351 233 E9	ù	371 249 F9
1 0 1 0	10	▨	212 138 8A	▨	232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA
1 0 1 1	11	▨	213 139 8B	▨	233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB
1 1 0 0	12	▨	214 140 8C	▨	234 156 9C		254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC
1 1 0 1	13	▨	215 141 8D	▨	235 157 9D		255 173 AD	½	275 189 BD	Í	315 205 CD	Ÿ	335 221 DD	í	355 237 ED	ÿ	375 253 FD
1 1 1 0	14	▨	216 142 8E	▨	236 158 9E		256 174 AE		276 190 BE	Î	316 206 CE		336 222 DE	î	356 238 EE		376 254 FE
1 1 1 1	15	▨	217 143 8F	▨	237 159 9F		257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF	▨	377 255 FF

MR–2908–RA

Figure A–3 lists characters of the ISO Latin-1 Character Set with decimal codes of 161 through 255.

Figure A–3 ISO Latin-1 Character Set

B8 B7 B6 B5 BITS		8		9		10		11		12		13		14		15	
		1 0 0 0		1 0 0 1		1 0 1 0		1 0 1 1		1 1 0 0		1 1 0 1		1 1 1 0		1 1 1 1	
B4 B3 B2 B1	Row																
0 0 0 0	0		200 128 80		220 144 90	NBSP	240 160 A0	°	260 176 B0	À	300 192 C0	Ð	320 208 D0	à	340 224 E0	ð	360 240 F0
0 0 0 1	1		201 129 81		221 145 91	¡	241 161 A1	±	261 177 B1	Á	301 193 C1	Ñ	321 209 D1	á	341 225 E1	ñ	361 241 F1
0 0 1 0	2		202 130 82		222 146 92	¢	242 162 A2	²	262 178 B2	Â	302 194 C2	Ò	322 210 D2	â	342 226 E2	ò	362 242 F2
0 0 1 1	3		203 131 83		223 147 93	£	243 163 A3	³	263 179 B3	Ã	303 195 C3	Ó	323 211 D3	ã	343 227 E3	ó	363 243 F3
0 1 0 0	4		204 132 84		224 148 94	¤	244 164 A4	´	264 180 B4	Ä	304 196 C4	Ô	324 212 D4	ä	344 228 E4	ô	364 244 F4
0 1 0 1	5		205 133 85		225 149 95	¥	245 165 A5	µ	265 181 B5	Å	305 197 C5	Õ	325 213 D5	å	345 229 E5	õ	365 245 F5
0 1 1 0	6		206 134 86		226 150 96	¦	246 166 A6	¶	266 182 B6	Æ	306 198 C6	Ö	326 214 D6	æ	346 230 E6	ö	366 246 F6
0 1 1 1	7		207 135 87		227 151 97	§	247 167 A7	·	267 183 B7	Ç	307 199 C7	×	327 215 D7	ç	347 231 E7	÷	367 247 F7
1 0 0 0	8		210 136 88		230 152 98	¨	250 168 A8	¸	270 184 B8	È	310 200 C8	Ø	330 216 D8	è	350 232 E8	ø	370 248 F8
1 0 0 1	9		211 137 89		231 153 99	©	251 169 A9	¹	271 185 B9	É	311 201 C9	Ù	331 217 D9	é	351 233 E9	ù	371 249 F9
1 0 1 0	10		212 138 8A		232 154 9A	ª	252 170 AA	º	272 186 BA	Ê	312 202 CA	Ú	332 218 DA	ê	352 234 EA	ú	372 250 FA
1 0 1 1	11		213 139 8B		233 155 9B	«	253 171 AB	»	273 187 BB	Ë	313 203 CB	Û	333 219 DB	ë	353 235 EB	û	373 251 FB
1 1 0 0	12		214 140 8C		234 156 9C	¬	254 172 AC	¼	274 188 BC	Ì	314 204 CC	Ü	334 220 DC	ì	354 236 EC	ü	374 252 FC
1 1 0 1	13		215 141 8D		235 157 9D	SHY	255 173 AD	½	275 189 BD	Í	315 205 CD	Ý	335 221 DD	í	355 237 ED	ý	375 253 FD
1 1 1 0	14		216 142 8E		236 158 9E	®	256 174 AE	¾	276 190 BE	Î	316 206 CE	Þ	336 222 DE	î	356 238 EE	þ	376 254 FE
1 1 1 1	15		217 143 8F		237 159 9F	¯	257 175 AF	¿	277 191 BF	Ï	317 207 CF	ß	337 223 DF	ï	357 239 EF	ÿ	377 255 FF

▨ Characters inserted in the positions that were left for future standardization in DEC MCS

□ Characters in the same position as characters in DEC MCS

MR–2909–RA

Appendix B
Entry Points

DASL provides entry points to the DASL internal software. You can use the DASL entry points as VAX DSM entry references when you construct your applications.

Use entry points in the following ways:

- In VAX DSM as arguments to the DO command
- In DASL data screens as arguments to DO or DONP actions and validations

You can execute all entry points from VAX DSM. To enter the DASL environment, for example, you use the entry point %DASL, as follows:

```
>D ^%DASL Return
```

You can use entry points as actions and validations on data screens. To use the entry points, you simply specify DO or DONP actions or validations with the entry point as an entry reference.

Table B–1 lists the DASL entry points and their purposes.

Table B–1 DASL Entry Points

Entry Point	Use
^%DADEV	Validates devices specified by DASL application users to print reports. The function performed depends on the line label you specify in the entry reference you use. The line labels you can use are as follows: %DADEV — validates that device is in Device Dictionary. OPEN — validates that device is in Device Dictionary and opens it.
^%DAERROR	Lists, reloads, and purges an error logged by the routine ^%DAERRT. You can use this entry point as a support tool to investigate errors.
^%DAERRT	Serves as DASL application error handler. To use, you must include the line label ERR. At ERR, ^%DAERRT logs the error, saves the local symbol table, and returns to the calling routine after an error occurs. You can use the line label LOG to send a message to the application's event logging device and the event log file.
^%DAINIT	Initializes the DASL software after VAX DSM is installed.
^%DALOAD	Interactively loads DASL entities from dump files you create through the Save DASL Definitions Option of the Development Environment Menu or through the ^%DAUDMP routine.
^%DALOGIN	Calls the Login Screen when an application user logs into DSM. Also initializes the user's environment, including the Security System. You can use the line label SCREEN^%DALOGIN to call a screen from a MUMPS routine. Pass the name of the screen to be called in the variable %NAME.
^%DANAME	Returns the name of a routine generated from a DASL screen or report. You can call ^%DANAME from a routine or an interactive session.
^%DAQ	Prompts for a query name and then runs the specified query interactively for testing purposes.
^%DAR	Prompts for a report name and then runs the specified report interactively for testing purposes.
^%DARUN	Runs the generated routine of a DASL report, query, or screen from inside the DASL development area. ^%DARUN does not initialize or clean up any DASL variables. You must provide the report, query, or screen name in the variable %NAME.

Table B–1 (Cont.) DASL Entry Points

Entry Point	Use
^%DAS	Prompts for a screen name and then runs the specified screen. You can use the line label READ^%DAS to bypass the DASL error trap. If you use READ, reset the terminal by executing the statement: X ^%DASY("RESET"). See Programmer's Note in Appendix C. To employ this routine in the application area, the DASL Security System must be operating.
^%DASL	Calls the DASL Main Menu.
^%DAUDTTM	Converts dates and times from external format to internal format or internal format to external format. The function performed depends on the line label you specify in the entry reference you use. The line labels you can use are as follows: DTIX — convert internal date to external date. DTXI — convert external date to internal date. NOW — convert present internal time and date to external time and date. TMIX — convert internal time to external time. TMXI — convert external time to internal time. When you use this utility, the following variables change: %DTX — External date %DTI — Internal date %TMX — External time %TMI — Internal time %MSG — Error message

Appendix C
Function Keys

DASL defines function keys on the VT100, VT200-series, and VT300-series terminals. These function keys are default settings that you can redefine using the Key Definitions Option of the Development Environment Menu. Table C–1 shows the function keys and describes the meaning of the keys.

Table C–1 DASL Function Keys

VT100 Key	VT200 Key VT300 Key	Function
PF2	Help	Displays a line of help text when pressed once; displays a help screen when pressed twice at any field that has an associated help screen.
GOLD/H	GOLD/H	Displays a line of help text when pressed once; displays a help screen when pressed twice at any field that has an associated help screen.
?	?	Displays a line of help text when pressed once; displays a help screen when pressed twice at any field that has an associated help screen.
Ctrl/C	Ctrl/C	In development area, interrupts the current activity and returns to the DSM prompt (>). In some cases, returns to the current screen when you press Ctrl/C during report or query collections from the database. For application users, Ctrl/C is disabled or returns to the current screen when pressed during report or query collections. See the Programmer's Note that follows this table.
Ctrl/W	Ctrl/W	Repaints the screen.

Table C–1 (Cont.) DASL Function Keys

	VT200 Key	
VT100 Key	**VT300 Key**	**Function**
GOLD/M	Main Menu F9	Returns the user directly to the Main Menu (a Super Exit).
GOLD/E	Exit F10	Transfers control to the currently defined exit field.
BACKSPACE	Previous Field F12	Transfers control to the previous field on the display unless the current field is the exit field (one whose actions handle exit processing for the screen) or it carries the NOBACK attribute (which specifies that a user cannot back up from the field).
LINE FEED	Clear Field F13	Deletes the data entered in the current field.
DELETE	⌫	Deletes the character to the left of the cursor.
RETURN	Return	Accepts data entered at the field and transfers control to the next field on the screen.
GOLD/C	Command Mode F20	Enters Command Mode at an option screen where command definition is enabled. Command definition must be allowed in the option screen definition and the Site Parameters Screen.
GOLD/N	Next Screen	Displays the next scroll window.
GOLD/P	Prev Screen	Displays the previous scroll window.
GOLD/I	Insert Here	Inserts a blank line in the scroll region above the cursor position.
GOLD/D	Remove	Deletes the line displayed in the scroll region at the cursor position.
GOLD/T	Software Status F14	Returns the first line of the current program. Includes the program name, version number, date and time of compilation, screen name, screen description, and field name.
⇑	⇑	Moves the cursor to the previous line in a scroll region.
⇓	⇓	Moves the cursor to the next line in a scroll region.
⇐ and ⇒	⇐ and ⇒	Field editor keys that enter line editing mode. Move the cursor within the line. You can insert or delete characters at the cursor.

Programmer's Note: After you press Ctrl/C, you may need to reset your terminal. To reset the terminal, execute the ^%DASY("RESET") global as follows:

```
X ^%DASY("RESET")
```

The ^%DASY("RESET") global resets the terminal characteristics, clears the screen, resets the scroll region index, and moves the cursor to the top of the screen.

Use the ^%DASY("RESET") global also after a screen error in the development area, or when a break point is set.

Appendix D SQL Keywords

SQL keywords are basic elements in an SQL statement. The DASL Query Driver uses SQL statements to create tables and queries. The SQL Standard states that you cannot use SQL keywords as table names or column names.

Table D–1 shows the SQL keywords that you cannot use.

Table D–1 SQL Keywords Not Allowed as Column or Table Names

AND	ANY	AS	ASC
AVG	BETWEEN	BY	COUNT
DELETE	DESC	DISTINCT	ESCAPE
EXEC	EXISTS	FOR	FROM
GOTO	GRANT	HAVING	IN
IS	JOIN	LIKE	MAX
MIN	NOT	NULL	OF
OR	ORDER	PRIVILEGES	SELECT
SET	SOME	SUM	TABLE
TO	UNION	UNIQUE	UPDATE
USER	VALUES	VIEW	WHENEVER
WHERE	WITH		

Note: Table D–1 does not represent a comprehensive listing of SQL keywords. More SQL keywords will be added to this list as the SQL Standard incorporates more keywords.

Index

Reader's Comments

VAX DSM
DASL Reference Manual
AA–KM67D–TE

Your comments and suggestions help us improve the quality of our publications.

Please rate the manual in the following categories:	Excellent	Good	Fair	Poor
Accuracy (product works as described)	□	□	□	□
Completeness (enough information)	□	□	□	□
Clarity (easy to understand)	□	□	□	□
Organization (structure of subject matter)	□	□	□	□
Figures (useful)	□	□	□	□
Examples (useful)	□	□	□	□
Table of contents (ability to find topic)	□	□	□	□
Index (ability to find topic)	□	□	□	□
Page design (overall appearance)	□	□	□	□
Print quality	□	□	□	□

What I like best about this manual: ______________________________

What I like least about this manual: ______________________________

Additional comments or suggestions:

I found the following errors in this manual:

Page Description

For which tasks did you use this manual?

□ Installation
□ Maintenance
□ Marketing
□ Operation/Use
□ Programming
□ System Management
□ Training
□ Other (please specify) ______________

Name/Title ______________________________ Date ____________

Company ______________________________ Dept ____________

Mailing Address ______________________________

______________________________ Phone ____________

Do Not Tear - Fold Here and Tape

digital™

AFFIX
STAMP
HERE

DIGITAL EQUIPMENT CORPORATION
CORPORATE USER PUBLICATIONS
MRO1–3/L12
P.O. BOX 1001
MARLBOROUGH, MA 01752–9840

Do Not Tear - Fold Here and Tape

DECLIT AA VAX KM67D

VAX DSM DASL reference manual

digital

Printed in U.S.A.