

PeopleTools 8.4: SQR for PeopleSoft Language Reference

PeopleSoft®

PeopleTools 8.4: SQR for PeopleSoft Language Reference

SKU Tr84SQL-B 0302

PeopleBooks Contributors: Teams from PeopleSoft Product Documentation and Development.

Copyright © 2002 PeopleSoft, Inc. All rights reserved.

Printed in the United States.

All material contained in this documentation is proprietary and confidential to PeopleSoft, Inc. ("PeopleSoft"), protected by copyright laws and subject to the nondisclosure provisions of the applicable PeopleSoft agreement. No part of this documentation may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, including, but not limited to, electronic, graphic, mechanical, photocopying, recording, or otherwise without the prior written permission of PeopleSoft.

This documentation is subject to change without notice, and PeopleSoft does not warrant that the material contained in this documentation is free of errors. Any errors found in this document should be reported to PeopleSoft in writing.

The copyrighted software that accompanies this document is licensed for use only in strict accordance with the applicable license agreement which should be read carefully as it governs the terms of use of the software and this document, including the disclosure thereof.

PeopleSoft, the PeopleSoft logo, PeopleTools, PS/nVision, PeopleCode, PeopleBooks, *PeopleTalk*, and Vantive are registered trademarks, and "People power the internet." and Pure Internet Architecture are trademarks of PeopleSoft, Inc. All other company and product names may be trademarks of their respective owners. The information contained herein is subject to change without notice.

Contents

SQR for PeopleSoft Language Reference Preface

About This PeopleBook.....	ix
Before You Begin.....	ix
<i>PeopleSoft Application Fundamentals</i>	x
Related Documentation	x
Hard-copy Documentation.....	xi
PeopleBooks Standard Field Definitions.....	xi
Typographical Conventions and Visual Cues.....	xiii
Page and Panel Introductory Table.....	xiv
Comments and Suggestions.....	xv

Chapter 1

Understanding SQR for PeopleSoft

Understanding SQR for PeopleSoft Tools.....	1-1
Understanding SQR.....	1-1
SQR Program Structure	1-2
SQR Syntax Conventions.....	1-3
Rules for Entering SQR Commands	1-3
Understanding the SQR Command Line	1-4
SQR Command-Line Arguments.....	1-4
SQR Command-Line Flags.....	1-5
Understanding SQR Data Elements	1-13
Columns	1-13
Variables	1-14
Literals	1-18
Viewing Sample Reports.....	1-18

Chapter 2

SQR Command Reference

ADD.....	2-2
ALTER-COLOR-MAP	2-2
ALTER-CONNECTION	2-4
ALTER-LOCALE.....	2-6
ALTER-PRINTER.....	2-13

ALTER-REPORT	2-15
ARRAY-ADD, ARRAY-DIVIDE, ARRAY-MULTIPLY, ARRAY-SUBTRACT.....	2-16
ASK.....	2-19
BEGIN-DOCUMENT.....	2-20
BEGIN-EXECUTE	2-21
BEGIN-FOOTING.....	2-24
BEGIN-HEADING	2-25
BEGIN-PROCEDURE.....	2-27
BEGIN-PROGRAM.....	2-30
BEGIN-SELECT.....	2-30
BEGIN-SETUP	2-34
BEGIN-SQL.....	2-35
BREAK	2-40
CALL, CALL SYSTEM	2-40
CLEAR-ARRAY	2-46
CLOSE	2-46
COLUMNS	2-47
COMMIT	2-48
CONCAT	2-49
CONNECT	2-50
CREATE-ARRAY	2-51
CREATE-COLOR-PALETTE.....	2-54
#DEBUG	2-55
DECLARE-CHART.....	2-56
DECLARE-COLOR-MAP.....	2-65
DECLARE-CONNECTION	2-66
DECLARE-IMAGE	2-68
DECLARE-LAYOUT.....	2-69
DECLARE-PRINTER	2-76
DECLARE-PROCEDURE	2-83
DECLARE-REPORT.....	2-85
DECLARE-TOC	2-87
DECLARE-VARIABLE.....	2-89
#DEFINE.....	2-92
DISPLAY	2-93
DIVIDE	2-97
DO	2-98
#ELSE	2-99
ELSE	2-100

ENCODE	2-100
END-DECLARE, END-DOCUMENT, END-EVALUATE, END-FOOTING, END-HEADING	2-101
#END-IF, #ENDIF	2-102
END-IF	2-102
END-PROCEDURE, END-PROGRAM, END-SELECT, END-SETUP, END-SQL, END-WHILE, END-EXECUTE	2-102
EVALUATE	2-103
EXECUTE (SYBASE and MS-SQL Server).....	2-105
EXIT-SELECT.....	2-108
EXTRACT	2-109
FIND	2-110
GET	2-111
GET-COLOR.....	2-113
GOTO	2-115
GRAPHIC BOX, GRAPHIC HORZ-LINE, GRAPHIC VERT-LINE.....	2-115
#IF	2-118
IF	2-120
#IFDEF	2-121
#IFNDEF.....	2-122
#INCLUDE	2-122
INPUT	2-123
LAST-PAGE.....	2-126
LET	2-127
LOAD-LOOKUP.....	2-158
LOOKUP	2-162
LOWERCASE	2-163
MBTOSBS	2-164
MOVE.....	2-164
MULTIPLY	2-168
NEW-PAGE.....	2-169
NEW-REPORT	2-170
NEXT-COLUMN	2-171
NEXT-LISTING	2-172
OPEN	2-174
PAGE-NUMBER.....	2-176
POSITION	2-177
PRINT	2-178
PRINT-BAR-CODE	2-197
PRINT-CHART	2-200
PRINT-DIRECT	2-208

PRINT-IMAGE	2-209
PUT	2-210
READ	2-212
ROLLBACK	2-215
SBTOMBS	2-215
SECURITY	2-216
SET-COLOR	2-217
SET-GENERATIONS	2-219
SET-LEVELS	2-219
SET-MEMBERS	2-220
SHOW	2-220
STOP	2-225
STRING	2-226
SUBTRACT	2-227
TOC-ENTRY	2-228
UNSTRING	2-229
UPPERCASE	2-230
USE	2-230
USE-COLUMN	2-231
USE-PRINTER-TYPE	2-232
USE-PROCEDURE	2-232
USE-REPORT	2-234
WHILE	2-235
WRITE	2-237

Chapter 3

Generating HTML Output

HTML General Purpose Procedures	3-1
HTML Heading Procedures	3-5
HTML Highlighting Procedures	3-7
HTML Hypertext Link Procedures	3-10
HTML List Procedures	3-11
HTML Table Procedures	3-16

Chapter 4

Invoking SQR Execute

Running SQR Execute	4-1
SQR Execute Flags	4-1

Chapter 5

Using SQR Print

Understanding SQR Print	5-1
Generating Output from the Command Line.....	5-1
SQR Print Command-Line Flags.....	5-2
Generating Output in Microsoft Windows	5-5

Chapter 6

Avoiding Older SQR Commands

Understanding Older SQR Commands.....	6-1
Older SQR Command Details	6-2
BEGIN-REPORT	6-2
DATE-TIME	6-2
DECLARE PRINTER.....	6-4
DECLARE PROCEDURE	6-10
DOLLAR-SYMBOL	6-11
GRAPHIC FONT.....	6-13
MONEY-SYMBOL	6-14
NO-FORMFEED	6-16
PAGE-SIZE	6-16
PRINT ...CODE	6-17
PRINTER-DEINIT	6-18
PRINTER-INIT.....	6-18

Chapter 7

Using SQR Sample Programs

Understanding SQR Samples	7-1
SQR Sample Programs.....	7-1

Chapter 8

Using the PSSQR.INI File

Installation of PSSQR.INI	8-1
[Default-Settings] Section	8-2
Processing-Limits Section	8-6
[Environment: <i>environment</i>] Sections	8-8
[Locale: <i>locale-name</i>] Section.....	8-9
[Fonts] Section.....	8-11
Adding [Fonts] entries	8-12
Specifying Character Sets in Windows.....	8-12
[HTML-Images] Section	8-12

[PDF Fonts] Section 8-13

[Enhanced-HTML] Section 8-14

[Colors] Section 8-14

Chapter 9

Understanding SQR Messages

Unnumbered Messages 9-1

Numbered Messages 9-3

Glossary

Index

SQR for PeopleSoft Language Reference Preface

This reference describes and demonstrates the structure, command set, and syntax of the Structured Query Report (SQR) language.

It also provides a directory to the library of sample SQR programs, an overview of the SQR initialization file, and a listing of SQR messages.

This reference is intended for SQR and SQL developers who must report on data from a wide range of enterprise Data Sources. Before using this reference, familiarize yourself with the Data Sources from which you are reporting and the connectivity between those Data Sources and your operating system.

The “About This PeopleBook” section contains general product line information, such as related documentation, common page elements, and typographical conventions. This book also contains a glossary with useful terms that are used in PeopleBooks.

See **PeopleSoft Glossary**.

About This PeopleBook

This book provides you with the information that you need for implementing and using *PeopleTools 8.4* applications. Complete documentation for this release is provided on the CD-ROM PT84PBR0.

Note. Your access to PeopleSoft PeopleBooks depends on which PeopleSoft applications you've licensed. You may not have access to all of the PeopleBooks.

This section contains information that you should know before you begin working with PeopleSoft products and documentation, including PeopleSoft-specific documentation conventions, information specific to each PeopleSoft product line, and information on ordering additional copies of our documentation.

Before You Begin

To benefit fully from the information covered in this book, you should have a basic understanding of how to use PeopleSoft applications. We recommend that you complete at least one PeopleSoft introductory training course.

You should be familiar with navigating the system and adding, updating, and deleting information by using PeopleSoft windows, menus, and pages. You should also be

comfortable using the World Wide Web and the Microsoft® Windows or Windows NT graphical user interface.

Because we assume that you already know how to navigate the PeopleSoft system, much of the information in these books is not procedural. That is, these books do not typically provide step-by-step instructions on using tables, pages, and menus. Instead, we provide you with the information that you need to use the system most effectively and to implement your PeopleSoft application according to your organizational or departmental needs. PeopleBooks expand on the material covered in PeopleSoft training classes.

PeopleSoft Application Fundamentals

Each PeopleSoft application PeopleBook provides implementation and processing information for your PeopleSoft database. However, there is additional, essential information describing the setup and design of your database contained in a companion volume of documentation called *PeopleSoft Application Fundamentals*.

PeopleSoft Application Fundamentals contains important topics that apply to many or all PeopleSoft applications across each product line. Whether you are implementing only one PeopleSoft application, some combination of products within a product line, or an entire PeopleSoft system, you should be familiar with the contents of this central PeopleBook. It contains fundamental information such as setting up control tables and administering security.

The PeopleSoft Applications Fundamentals PeopleBook contains common information pertinent to all applications in each product line, such as defining general options. If you're upgrading from a previous PeopleSoft release, you may notice that we've removed some topics or topic headings from the individual application PeopleBooks and consolidated them in this single reference book. You'll now find only application-specific information in your individual application PeopleBooks. This makes the documentation as a whole less redundant. Throughout each PeopleBook, we provide cross-references to *PeopleSoft Application Fundamentals* and other PeopleBooks.

Related Documentation

You can order printed, bound versions of the complete PeopleSoft documentation delivered on your PeopleBooks CD-ROM and additional copies of the PeopleBooks CDs through the Documentation section of the PeopleSoft Customer Connection website:

<http://www.peoplesoft.com/corp/en/login.asp>

You can find updates and additional documentation for this release, as well as previous releases, on PeopleSoft Customer Connection (<http://www.peoplesoft.com/corp/en/login.asp>). Through the Documentation section of Customer Connection, you can download files to add to your PeopleBook library. You'll find a variety of useful and timely materials, including updates to the full PeopleSoft documentation delivered on your PeopleBooks CD.

Important! Before you upgrade, it is *imperative* that you check PeopleSoft Customer Connection for updates to the upgrade instructions. We continually post updates as we refine the upgrade process.

Hard-copy Documentation

To order printed, bound volumes of the complete PeopleSoft documentation delivered on your PeopleBooks CD-ROM, visit the PeopleSoft Press website from the Documentation section of PeopleSoft Customer Connection. The PeopleSoft Press website is a joint venture between PeopleSoft and Consolidated Publications Incorporated (CPI), our book print vendor.

We make printed documentation available for each major release shortly after the software is shipped. Customers and partners can order printed PeopleSoft documentation by using any of the following methods:

Internet

From the main PeopleSoft Internet site, go to the Documentation section of Customer Connection. You can find order information under the Ordering PeopleBooks topic. Use a Customer Connection ID, credit card, or purchase order to place your order.

PeopleSoft Internet site: <http://www.peoplesoft.com/>.

Telephone

Contact Consolidated Publishing Incorporated (CPI) at **800 888 3559**.

Email

Send email to CPI at callcenter@conpub.com.

PeopleBooks Standard Field Definitions

Throughout our product documentation, you will encounter fields and buttons that are used on many application pages or panels. This section lists the most common fields and buttons and provides standard definitions.

Field	Definition
As of Date	The last date for which a report or process includes data.
Business Unit	An identification code that represents a high-level organization of business information. You can use a business unit to define regional or departmental units within a larger organization.
Description	Freeflow text up to 30 characters.

Field	Definition
Effective Date	Date on which a table row becomes effective; the date that an action begins. For example, if you want to close out a ledger on June 30, the effective date for the ledger closing would be July 1. This date also determines when you can view and change the information. Pages or panels and batch processes that use the information use the current row. For more information about effective dates, see <u>Understanding Effective Dates in Using PeopleSoft Applications.</u>
EmplID (employee ID)	Unique identification code for an individual associated with your organization.
Language or Language Code	The language in which you want the field labels and report headings of your reports to print. The field values appear as you enter them. Language also refers to the language spoken by an employee, applicant, or non-employee.
Process Frequency group box	Designates the appropriate frequency in the Process Frequency group box: Once executes the request the next time the batch process runs. After the batch process runs, the process frequency is automatically set to Don't Run. Always executes the request every time the batch process runs. Don't Run ignores the request when the batch process runs.
Report ID	The report identifier.
Report Manager	This button takes you to the Report List page, where you can view report content, check the status of a report, and see content detail messages (which show you a description of the report and the distribution list).
Process Monitor	This button takes you to the Process List page, where you can view the status of submitted process requests.
Run	This button takes you to the Process Scheduler request page, where you can specify the location where a process or job runs and the process output format.
	For more information about the Report List page, the Process List page, and the Process Scheduler, see <u>Process Scheduler Basics in the PeopleTools documentation.</u>
Request ID	A request identification that represents a set of selection criteria for a report or process.

Field	Definition
User ID	The system identifier for the individual who generates a transaction.
SetID	An identification code that represents a set of control table information or TableSets. A TableSet is a group of tables (records) necessary to define your company's structure and processing options.
Short Description	Freeflow text up to 15 characters.

Typographical Conventions and Visual Cues

We use a number of standard conventions and visual cues in our online documentation.

The following list contains our typographical conventions and visual cues:

`(monospace font)`

Indicates a PeopleCode program or other program example.

Bold

Indicates field names and other page elements, such as buttons and group box labels, when these elements are documented below the page on which they appear. When we refer to these elements elsewhere in the documentation, we set them in Normal style (not in bold).

We also use boldface when we refer to navigational paths, menu names, or process actions (such as **Save** and **Run**).

Italics

Indicates a PeopleSoft or other book-length publication. We also use italics for *emphasis* and to indicate specific field values. When we cite a field value under the page on which it appears, we use this style: *field value*.

We also use italics when we refer to words as words or letters as letters, as in the following: Enter the number *0*, not the letter *O*.

KEY+KEY

Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For ALT+W, hold down the ALT key while you press W.

Cross-references

The phrase **For more information** indicates where you can find additional documentation on the topic at hand. We include the navigational path to the referenced topic, separated by colons (:). Capitalized titles in *italics* indicate the title of a PeopleBook; capitalized titles in normal font refer to sections and specific topics within the PeopleBook. Here's an example:

For more information, see **Documentation on CD-ROM** in *About These PeopleBooks*: Additional Resources.

Note. Text in this bar indicates information that you should pay particular attention to as you work with your PeopleSoft system. If the note is preceded by **Important!**, the note is crucial and includes information that concerns what you need to do for the system to function properly.

Text in this bar indicates cross-references to related or additional information.

Warning! Text within this bar indicates a crucial configuration consideration. Pay very close attention to these warning messages.

Page and Panel Introductory Table

In the documentation, each page or panel description in the application includes an introductory table with pertinent information about the page. Not all of the information will be available for all pages or panels.

Usage	Describes how you would use the page or process.
Object Name	Gives the system name of the panel or process as specified in the PeopleTools Application Designer. For example, the Object Name of the Detail Calendar panel is <code>DETAIL_CALENDAR1</code> .
Navigation	Provides the path for accessing the page or process.
Prerequisites	Specifies which objects must have been defined before you use the page or process.
Access Requirements	Specifies the keys and other information necessary to access the page. For example, SetID and Calendar ID are required to open the Detail Calendar page.

Comments and Suggestions

Your comments are important to us. We encourage you to tell us what you like, or what you would like to see changed about our documentation, PeopleBooks, and other PeopleSoft reference and training materials. Please send your suggestions to:

PeopleSoft Product Documentation Manager
PeopleSoft, Inc.
4460 Hacienda Drive
Pleasanton, CA 94588

Or send comments by email to the authors of the PeopleSoft documentation at:

DOC@PEOPLESOFT.COM

While we cannot guarantee to answer every email message, we will pay careful attention to your comments and suggestions. We are always improving our product communications for you.

CHAPTER 1

Understanding SQR for PeopleSoft

This chapter provides overviews of:

- SQR for PeopleSoft tools.
- SQR.
- SQR command line.
- SQR data elements.
- Sample reports.

Understanding SQR for PeopleSoft Tools

SQR for PeopleSoft is a powerful enterprise reporting system that provides direct access to multiple Data Sources. The SQR for PeopleSoft tools make it possible to create clear, elegant reports from complex arrays of information systems.

This reference describes the following SQR for PeopleSoft tools:

- SQR is a flexible, 4GL reporting language with a lexicon of over 110 commands.

The procedural design of SQR makes development, implementation, and distribution of complex reports more systematic and efficient than most reporting tools.

- SQR Execute enables you to run previously compiled SQR programs.
- SQR Print enables you to configure reports for most printers.
- SQR Samples is a library of SQR programs and output that provides a framework for creating configured reports.

Understanding SQR

SQR is a specialized programming language for accessing, manipulating, and reporting enterprise data. With SQR, you build complex procedures that perform multiple calls to multiple Data Sources and implement nested, hierarchical, or object-oriented program logic.

SQR has several important benefits:

- Flexibility and scalability.
- Comprehensive facilities for combined report and data processing.
- Multiple platform availability.
- Multiple Data Source compatibility.

With SQR, you design reports by defining the page size, headers, footers, and layout. SQR enables you to generate a variety of output types, such as complex tabular reports, multiple page reports, and form letters. You can display data in columns; produce special formats, such as mailing labels; and create HTML, PDF, or configured output for laser printers and phototypesetters.

The high-level programming capabilities that SQR provides enable you to add procedural logic and control to Data Source calls. You can use SQR to write other types of applications, such as database manipulation and maintenance, table load and unload, and interactive query and display.

This section describes:

- SQR program structure.
- SQR syntax conventions.
- Rules for entering SQR commands.

SQR Program Structure

SQR for PeopleSoft processes source code from a standard text file and generates your report. The text file containing source code comprises a simple set of sections that you delimit with BEGIN-section and END-section commands. The following examples show the general structure of SQR.

- The SETUP section describes overall characteristics of the report.

```
BEGIN-SETUP
  {setup commands}...
END-SETUP
```

- The HEADING and FOOTING sections specify what is printed in the header and footer on each page of the report.

```
BEGIN-HEADING {heading_lines}
  {heading commands}...
END-HEADING
BEGIN-FOOTING {footing_lines}
  {footing commands}...
END-FOOTING
```

- The PROGRAM section runs the procedures contained in the report.

```
BEGIN-PROGRAM
  {commands}...
END-PROGRAM
```

- The PROCEDURE section accomplishes the tasks associated with producing the report.

```
BEGIN-PROCEDURE {procedure_name}
  {procedure commands}...
END-PROCEDURE
```

SQR Syntax Conventions

The following table describes the SQR syntax conventions.

<i>Symbol</i>	<i>Description</i>
{ }	Braces enclose required items.
[]	Square brackets enclose optional items.
...	Ellipses indicate that the preceding parameter can be repeated.
	A vertical bar separates alternatives inside brackets, braces, or parentheses.
!	An exclamation point begins a single-line comment that extends to the end of the line. Each comment line must begin with an exclamation point.
'	A single quote starts and ends a literal text constant or any argument with more than one word. Important! If you are copying codes directly from the examples in the PDF file, change the slanted quotes to regular quotes; otherwise, you receive an error message.
,	A comma separates multiple arguments.
()	Parentheses must enclose an argument or element.
BOLD UPPERCASE	SQR commands and arguments are specified in bold uppercase.
<i>Italic</i>	Information and values that you must supply are specified in italics.

Rules for Entering SQR Commands

Use these command rules as you develop SQR programs:

- SQR commands are *not* case-sensitive.

Many SQR programmers use uppercase for SQR commands, but SQR ignores case as it compiles your source code.

- You must separate command names and arguments by at least one space or tab character.
- You must begin each command on a new line; however, you can develop commands that extend beyond one line.
- You can break a line in any position between words *except* inside a quoted string.
- You can use a hyphen (-) at the end of a line to indicate that it continues on the next line; however, SQR ignores hyphens and carriage returns in commands.
- You must begin each comment line with an exclamation point (!).

To display the exclamation point (!) or single quote (') symbols in your report, type these symbols twice to indicate that they are text. For example, DON'T is typed DON" T.

Note. This rule about typing quotation and exclamation marks twice in the report does not apply in the DOCUMENT section of form-letter reports.

Understanding the SQR Command Line

SQR for PeopleSoft comprises SQR, SQR Execute, and SQR Print. Each has a command-line interface.

To begin running SQR, enter the following command:

```
SQR [program] [connectivity] [flags...] [args...] [@file...]
```

Note. If you are running this under Microsoft Windows, the executable names are SQRW, SQRWT, SQRWP, and SQRWV.

See Also

“Invoking SQR Execute”

“Using SQR Print”

SQR Command-Line Arguments

The following table describes the SQR command-line arguments.

Argument	Description
<i>program</i>	Name of your text file containing source code. The default file type or extension is .sqr. If entered as “?” or omitted, SQR for PeopleSoft prompts you for the report program name. On UNIX-based systems, if your shell uses the question character as a WILD CARD character, you must precede it with a backslash (\).

Argument	Description
<i>connectivity</i>	Information needed by SQR for PeopleSoft to connect to the database. If entered as “?” or omitted, SQR for PeopleSoft prompts you for the information. - DB2: Subsystem name and SQL authorization ID. <i>Ssname/SQLid</i> - Microsoft Windows or DB2: Name of the database, your username, and the password for the database. <i>[Database] / [Username] / [Password]</i> - Informix: Name of the database. <i>Database[/username/password]</i> - ODBC: Name that you give to the ODBC driver when you set up the driver, and your username and password for the database. This port has been certified against DB2 and Microsoft SQL Server. <i>Data_Source_Name/[Username]/[Password]</i> - Oracle: Your username and password for the database, and an optional connection string for the database (for example, @sales.2cme.com). <i>[Username]/[Password[@Database]]</i> - SYBASE: Your username and password for the database. <i>Username/[Password]</i>
<i>flags</i>	Any of the flags listed under SQR Command-Line Flags .
<i>args...</i>	Arguments used by SQR for PeopleSoft while the program is running. Arguments listed here are used by the ASK and INPUT commands rather than prompting the user. You must enter arguments on the command line in the sequence that the program expects—first all ASK arguments, in order, followed by INPUT arguments.
<i>@file...</i>	File containing program arguments—one argument per line. Arguments listed in the file are processed one at a time—first all ASK arguments, in order, followed by INPUT arguments. The command-line arguments (<i>program</i> , <i>connectivity</i> , and <i>args</i>) can be specified in this file for non-Microsoft Windows platforms.

SQR Command-Line Flags

SQR supports a number of command-line flags. Each flag begins with a hyphen (-). When a flag takes an argument, the argument must follow the flag with no intervening space.

The following table describes the SQR command-line flags.

Flag	Description
-A	Appends the output to an existing output file carrying the same name as the source of the output. If the file does not exist, a new one is created. This is useful when you want to run the same report more than once but want only a single output file.
-Bnn	(ODBC, Oracle, SYBASE CT-Lib) Indicates the number of rows to buffer each time SQR for PeopleSoft retrieves data from the database. The default is 10 rows. Regardless of the setting, all rows are retrieved. When used on the command line, -B controls the setting for all BEGIN-SELECT commands. Inside a program, each BEGIN-SELECT command can also have its own -B flag for further optimization.
-BURST:{xx}	Specifies the type of bursting to perform. See <i>SQR for PeopleSoft Developer's Guide</i>, "Working With HTML." -BURST:T generates the Table of Contents file only. -BURST:S generates the report output according to the symbolic Table of Contents entries set in the program with the <i>level</i> argument of the TOC-ENTRY command. In -BURST:S[{l}], {l} is the level on which to burst. The setting -BURST:S is equivalent to -BURST:S1. -BURST:P generates the report output, grouped by report page numbers. In -BURST:P[{l} , {s} [, {s}] ...] , {l} is the number of logical report pages that each .HTM file contains and {s} is the page selection: {n}, {n}-{m}, -{m}, or {n}-. The setting -BURST:P is equivalent to -BURST:P0,l- when using -PRINTER:HT or -BURST:P1 when using -PRINTER:EH. See <i>SQR for PeopleSoft Developer's Guide</i>, "Working With HTML", "Bursting" and Demand Paging. Note. -BURST:P and -BURST:S require -PRINTER:EH or -PRINTER:HT. The Page range selection feature of -BURST:P requires -PRINTER:HT. -BURST:T requires -PRINTER:HT
-C	(Microsoft Windows) Specifies that the Cancel dialog box appears while the program runs so that you can easily stop the program.
-CB	(Microsoft Windows, Callable SQR) Forces the communication box to use.
-Dnn	(Non-Microsoft Windows) Causes SQR for PeopleSoft to display the report output on the terminal at the same time that it is being written to the output file. The value for <i>nn</i> is the maximum number of lines to display before pausing. If no number is entered after -D, the display scrolls continuously. Note. The printer type must be <i>LP</i>; otherwise, the display is ignored. If the program is producing more than one report, the display is for the first report only.
-DBdatabase	(SYBASE) Causes the SQR program to use the specified database, which overrides any USE command in the SQR program.

Flag	Description
-DEBUG [xxx]	Compiles lines preceded by #DEBUG. Without this flag, SQR for PeopleSoft ignores these lines. See “SQR Command Reference,” #DEBUG.
-DNT: {xx}	Specifies the default behavior for numeric variables. The value for xx can be INTEGER, FLOAT, DECIMAL, or V30. To specify a precision for DECIMAL, append it with a colon delimiter (:). For example, -DNT:DECIMAL:20. See the DEFAULT argument for the DECLARE-VARIABLE command in “SQR Command Reference,” DECLARE-VARIABLE. The DEFAULT argument in the DECLARE-VARIABLE command takes precedence, if used.
-E[file]	Directs error messages to the named file or to the default file program.err. If no errors occur, no file is created.
-EH_APPLETS:dir	Specifies the directory location of the enhanced HTML applets. If you include an applet, SQR for PeopleSoft must know where it resides. SQR for PeopleSoft usually checks for the applet in a default directory; the default directory for these applets is IMAGES. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.
-EH_BQD	Generates a {report}.bqd file from the report data. Also associates a BQD (query format file) icon with {report}.bqd in the navigation bar. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.
-EH_BQD:file	Associates the BQD icon with the specified file. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.
-EH_BROWSER:xx	Generates HTML, determines the browser, and displays HTML. When set to ALL, SQR for PeopleSoft generates Frame.html, which contains JavaScript to determine the browser on the user’s machine (that is, the person reading the report, not the person writing it). When set to BASIC, SQR for PeopleSoft generates HTML suitable for all browsers. When set to IE, SQR for PeopleSoft generates HTML designed for Microsoft Internet Explorer. When set to NETSCAPE, SQR for PeopleSoft generates HTML designed for Netscape. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.
-EH_CSV	Generates a {report}.csv file from the report data. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.
-EH_CSV:file	Associates the CSV icon with the specified file. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.

Flag	Description
-EH_CSVONLY	Creates a CSV file, but does not create an HTML file. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.
-EH_FULLHTML:xx	Specifies the level of the generated enhanced HTML code. This can be 30, 32, or 40. Note. For upward compatibility, a value of TRUE is equivalent to 40 and FALSE is 30.
-EH_Icons:dir	Specifies the directory in which the HTML should find the referenced icons. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.
-EH_IMAGES:dir	Specifies the directory path for the .GIF files used by the navigation bar. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.
-EH_KEEP	Copies (does not move) the files when used in conjunction with -EH_ZIP. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.
-EH_LANGUAGE:xx	Sets the language used for the HTML navigation bar. You can specify English, French, German, Portuguese, or Spanish. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.
-EH_PDF	Associates a PDF icon with {report}.pdf in the navigation bar. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.
-EH_Scale:{nn}	Sets the scaling factor from 50 to 200. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.
-EH_XML:file	Associates the Extensible Markup Language (XML) icon with the specified file. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.
-EH_ZIP[:file]	Moves the generated files to the specified file or {report}.zip if {file} is not specified. Note. This flag is applicable only when either the -PRINTER:EH or the -PRINTER:EP flag is specified.

Flag	Description
-F[library file(member)]	(AS400) Overrides the default output file name library/lis(program). The default action places the LIS in a source file called LIS in the same library as the member {program}. To use the current library, specify -F without an argument. To change the name of the output file, specify -F with the new name. If the new name does not specify a library, the file is created in the current library. The output file is not created until data is actually printed on the page. If no data is printed, no output file is created. To specify an alternate file name, use -FSQRDIR/LIS(file).
-F[file directory]	Overrides the default output file name, program.lis. The default action places the program.lis in the same directory as the program.sqr file. To use the current directory, specify -F without an argument. To change the name of the output file, specify -F with the new name. If the new name does not specify a directory, the file is created in the current directory. The output file is not created until data is actually printed on the page. If no data is printed, no output file is created. Specify the file name and directory for different operating systems as follows: • VMS Directory character is] or : -FSYS\$USER:[REPORTS] -FSYS\$SCRATCH: • UNIX Directory character is / -F\$HOME/reports/ • MVS Directory character is (-FDSN:SQR.REPORTS(
-Gfile_mode	(VM) Specifies the file mode to use when the report output file is created.
-Gfile_attributes	(VMS, OpenVMS) Specifies the file attributes to use for the report output file. You can specify up to 10 sets of attributes, separated by commas (,).
-GPRINT=YES NO	(MVS) -GPRINT=YES causes the SQR report output file to have ANSI control characters written to the first column of each record of the file. -GPRINT=NO omits the control characters.

Flag	Description
-Idir_list	Specifies the list of directories that SQR for PeopleSoft searches when processing the #INCLUDE directive if the include file does not exist in the current directory and no path was specified for the file. The directory names must be separated by either commas (,) or semicolons (;). For UNIX-based systems, if your shell uses semicolons as command delimiters, you must precede each semicolon with a backslash (\). Always append the directory character to the end of each directory name. See the -F flag for a list of directory characters, sorted by operating system. For example, under UNIX: <code>sqr myreport sammy/baker - I/home/sqr/inc/,/usr/sqr/ incl/</code>
-ID	(Non-Microsoft Windows) Displays the copyright banner on the console.
-KEEP	Creates SPF file output and LIS files for each report that your program generates. See <i>SQR for PeopleSoft Developer's Guide</i> , "Printing Issues" for information about LIS and SPF files.
-LL{s d}{c i}	LOAD-LOOKUP: • s = SQR. • d = DB. • c = Case-sensitive. • i = Case-insensitive. See the "SQR Command Reference," LOAD-LOOKUP.
-Mfile	Defines a startup file containing sizes to assign to internal parameters—extremely small, large, or complex reports. <i>Mfiles</i> are text files that have individual switches in the INI files that are unique to a run.
-NOLIS	Prevents the creation of LIS files. Instead, SPF files are created.
-O[file]	Directs log messages to the specified file or to program.log if no file is specified. By default, the file sqr.log is used in the current working directory.
-P	(MVS, AS400) Suppresses printer control characters from column 1.
-PB	(Informix) Causes column data to retain trailing blanks.

Flag	Description
-PRINTER:xx	Uses printer type xx when creating output files. The xx represents: • EH = Enhanced HTML. -PRINTER:EH • EP = Enhanced HTML or PDF. -PRINTER:EP • HP = HP LaserJet. -PRINTER:HP • HT = HTML 2.0. -PRINTER:HT • LP = Line printer. -PRINTER:LP • PD = PDF. -PRINTER:PD • PS = PostScript. -PRINTER:PS • WP = Microsoft Windows. -PRINTER:WP Types LP, HP, and PS produce files with the .lis extension. Types EH and HT produce .htm file output. Type HT produces version 2.0 HTML files with the report content inside of <PRE></PRE> tags. Type EH produces reports in which content is fully formatted with version 3.0 or 3.2 HTML tags. On Microsoft Windows systems, the WP extension sends the output to the default Microsoft Windows printer. To specify a Microsoft Windows printer that is not the default, enter -PRINTER:WP:{Printer Name}. {Printer Name} is the name assigned to your printer. For example, to send output to a Microsoft Windows printer named <i>NewPrinter</i>, use -PRINTER:WP:NewPrinter. If your printer name has spaces, enclose the entire argument in quotes. To also create an SPF file, use -KEEP.
-RS	Saves the program in a runtime file. The program is scanned, compiled, and checked for correct syntax. Queries are validated and compiled. Then, the executable version is saved in a file with the name program.sqt. Note. SQR for PeopleSoft does not prompt ASK variables after compilation.

Flag	Description
-RT	Uses the runtime file saved with the -RS flag. This skips all syntax and query checking, and processing begins immediately. Note. SQR for PeopleSoft does not prompt ASK variables after compilation.
-S	Requests that the status of all cursors be displayed at the end of the report run. Status includes the text of each SQL statement, the number of times each was compiled and run and the total number of rows selected. The output appears directly on the screen. This information can be used for debugging SQL statements and enhancing performance and tuning.
-Tnn	Specifies that you want to test your report for <i>nn</i> pages. To save time during testing, SQR for PeopleSoft ignores all ORDER BY clauses in SELECT statements. If the program is producing more than one report, SQR for PeopleSoft stops after producing the specified number of pages defined for the first report.
-T{B}	(OS400/DB2, SYBASE CT-Lib, ODBC) Trims trailing blanks from database character columns. (MVS/DB2) Prevents SQR for PeopleSoft from removing trailing blanks from database character columns.
-T{Z}	(MVS/DB2) -TZ prevents SQR for PeopleSoft from removing trailing zeros from the decimal portion of numeric columns. (OS400/DB2) -TZ trims trailing zeros from the decimal portion of numeric columns.
-Vserver	(SYBASE) Uses the named server.
-XB	(Non-Microsoft Windows) Suppresses the SQR banner and the <i>SQR... End of Run</i> message.
-XC	(Callable SQR) Suppresses the database commit when the report has finished running.
-XCB	(Microsoft Windows) Do not use the communication box. Requests for input are made in Microsoft Windows dialog boxes.
-XI	Prevents user interaction during a program run. If an ASK or INPUT command requires user input, an error is produced and the program ends.
-XL	Prevents SQR for PeopleSoft from logging on to the database. Programs that you run in this mode cannot contain SQL statements. -XL enables you to run SQR for PeopleSoft without accessing the database. You still must supply at least an empty slash (/) on the command line as a placeholder for the connectivity information. For example: <pre>sqr myprog / -xl</pre>
-XLFF	Prevents trailing form feed.

Flag	Description
-XMB	(Microsoft Windows) Disables the error message display so that you can run a program without interruption from error message boxes. Error messages are sent to an .err file. See the -E flag.
-XNAV	Prevents SQR for PeopleSoft from creating the navigation bar in .htm files generated with -PRINTER:HT. This occurs when only a single .htm file is produced. Multiple .htm files generated from a single report always contain the navigation bar.
-XP	(SYBASE) Prevents SQR for PeopleSoft from creating temporary stored procedures. See “SQR Command Reference,” BEGIN-SELECT.
-XTB	Preserves the trailing blanks in a .lis file at the end of a line.
-XTOC	Prevents SQR for PeopleSoft from generating the table of contents for the report. This flag is ignored when either -PRINTER:EH or -PRINTER:HT is also specified.
-ZIF {file}	Sets the full path and name of the SQR initialization file, SQR.INI.
-ZIV	Invokes the SPF Viewer after generating the <i>program.spf</i> file. This flag implicitly invokes the -KEEP flag to create <i>program.spf</i> . In the case of multiple output files, only the first report file is passed to the viewer.
-ZMF {file}	Specifies the full path and name of the SQR error message file, sqrrr.dat.
-ZRF {file}	Sets the full path and name of an alternate <i>registry.properties</i> file. The following path is a common default path to the <i>registry.properties</i> file on an NT system: <code>c:\PeopleTools\properties\registry.properties</code> The <i>registry.properties</i> file lists Data Sources that SQR for PeopleSoft can access. The information in the <i>registry.properties</i> file makes it possible for SQR for PeopleSoft to access Data Sources for which DDO drivers have been loaded and configured.

Understanding SQR Data Elements

Each SQR data element begins with a special character that denotes the type of data element. SQR data elements include:

- Columns
- Variables
- Literals

Columns

Columns are fields defined in the database.

The character & begins a database column or expression name. It can be any type of column, such as character, number, or date. Except for dynamic columns and database or aggregate functions, it is declared automatically for columns defined in a query.

Variables

Variables are storage places for text or numbers that you define and manipulate.

- \$ begins a text or date variable.
- # begins a numeric variable.
- % begins a list variable.
- @ begins a variable name for a marker location.

Marker locations identify positions to begin for printing in a BEGIN-DOCUMENT paragraph.

Variable Rules

The following rules govern the use of variables in SQR:

- Variables can be almost any name of almost any length.

For example, \$state_name or #total_cost.

- Do not use an underscore (_) or colon (:) as the first character of a two-variable name.

See *SQR for PeopleSoft Builder User's Guide*, "Hyphens and Underscores."

- Variable names are *not* case-sensitive.

That is, you can use a name in uppercase on one line and lowercase on the next; both refer to the same variable.

- SQR for PeopleSoft initializes variables to null (text and date) or zero (numeric).
- A command can grow to whatever length the memory of your computer can accommodate.
- Numeric variables can be one of three types: FLOAT, INTEGER, or DECIMAL.

See "SQR Command Reference," DECLARE-VARIABLE.

- Variables and columns are known globally throughout a report, except when used in a local procedure (one with arguments or declared with the LOCAL argument), in which case they are known in that procedure only.

See the "SQR Command Reference," BEGIN-PROCEDURE.

SQR Reserved Variables

When you create multiple reports, the variables apply to the current report. SQR for PeopleSoft reserves a library of predefined variables for general use.

The following table describes the SQR reserved variables.

Variable	Description
<code>#current-column</code>	Current column on the page.
<code>\$current-date</code>	Current date and time on the local machine when SQR for PeopleSoft starts running the process.
<code>#current-line</code>	Current line on the page. This value is the physical line on the page, not the line in the report body. See <i>SQR for PeopleSoft Developer's Guide</i> . Line numbers are referenced in PRINT and other SQR commands used for positioning the data on the page. Optional page headers and footers, defined with BEGIN-HEADING and BEGIN-FOOTING commands, have their own line sequences. Line 2 of the heading is different from line 2 of the report body or footing.
<code>#end-file</code>	Set to 1 if the end of the file occurs when reading a flat file. See "SQR Command Reference," READ.
<code>#page-count</code>	Current page number.
<code>#return-status</code>	Value to return to the operating system when SQR for PeopleSoft exits. Can be set in your report. #return-status is initialized to the "success" return value for the operating system.
<code>#sql-count</code>	Count of the rows affected by a DML statement (INSERT, UPDATE, or DELETE). This is equivalent to ROWCOUNT in Oracle and Sybase.
<code>\$sql-error</code>	Text message from the database explaining an error. This variable is rewritten when a new error is encountered.
<code>#sql-status</code>	The value of #SQL-STATUS is set whenever a BEGIN-SELECT command is run. Normally this variable is checked from inside an ON-ERROR procedure, so its value describes the error condition (whereas the \$\$SQL-ERROR variable contains the error message). The actual meaning of #SQL-STATUS is database dependent. Therefore, consult the proper database manual to fully interpret its meaning.
<code>\$sqr-encoding-console</code> <code>{sqr-encoding-console}</code>	Name of encoding for character data written to the log file or console.
<code>\$sqr-encoding-database</code> <code>{sqr-encoding-database}</code>	Character data retrieved from and inserted into the database.

Variable	Description
<i>\$sqr-encoding-file-input</i> <i>{sqr-encoding-file-input}</i>	Name of encoding for character data read from files used with the OPEN command.
<i>\$sqr-encoding-file-output</i> <i>{sqr-encoding-file-output}</i>	Name of encoding for character data written to files used with the OPEN command.
<i>\$sqr-encoding-report-output</i> <i>{sqr-encoding-report-output}</i>	Report generated by SQR for PeopleSoft (for example, an LIS file or a Post-Script file).
<i>\$sqr-encoding-source</i> <i>{sqr-encoding-source}</i>	Name of encoding for SQR source files and include files.
<i>\$sqr-database</i> , <i>{sqr-database}</i>	Database type for which SQR was compiled. Valid values are DB2, ODBC, SYBASE, INFORMIX, and ORACLE.
<i>\$sqr-dbc</i> , <i>{sqr-dbc}</i>	Specifies whether SQR for PeopleSoft recognizes double-byte character strings. Valid values are YES and NO.
<i>sqr-encoding</i> , <i>{sqr-encoding}</i>	Name of the default encoding as defined by the ENCODING environment variable when SQR for PeopleSoft is invoked. Valid values are ASCII, JEUC, and SJIS.
<i>sqr-hostname</i> , <i>{sqr-hostname}</i>	This variable contains the name of the computer on which SQR for PeopleSoft is currently running.
<i>\$sqr-locale</i>	Name of the current locale being used. A plus symbol (+) at the end of the name indicates that an argument used in the locale has changed.
<i>#sqr-max-lines</i>	Maximum number of lines, as determined by the layout. When a new report is selected, this variable is automatically updated to reflect the new layout.
<i>#sqr-max-columns</i>	Maximum number of columns, as determined by the layout. When a new report is selected, this variable is automatically updated to reflect the new layout.
<i>#sqr-pid</i>	Process ID of the current SQR process. #sqr-pid is unique for each run of SQR. This variable is useful in creating unique temporary names.
<i>\$sqr-platform</i> , <i>{sqr-platform}</i>	The hardware or operating system type for which SQR was compiled. Valid values are VMS, MVS, WINDOWS-NT, and UNIX.
<i>\$sqr-program</i>	Name of the SQR process file.
<i>\$sqr-ver</i>	Text string shown with the -ID flag. SQR version.

Variable	Description
<i>\$username</i>	Database username specified on the command line.
<i>\$sqr-report</i>	Name of the report output file. \$sqr-report reflects the actual name of the file to use, as specified by the -F flag or NEW-REPORT command.

List Variables

List variables contain an ordered collection of SQR variables and are nonrecursive; that is, you cannot nest lists inside lists.

Indicate list variables with the percent (%) symbol. Create list variables with the LET command and a list of variables. For example:

```
LET %LIST1 = LIST (num_var1|str_var1, num_var2|str_var2,...)
```

Working with list variables includes the following tasks:

- Defining a list variable:

You can use a list variable to hold multiple rows of information. Before you assign a list variable, define it using the following syntax:

```
let %listname=LIST(col_var|num_var|str_var|str_lit|num_lit[,...])
```

or

```
let %listname[num_lit]=list (NUMBER|DATE|TEXT$colname  
|'.colname'[,...])
```

- Assigning a list variable:

Assign a list variable using the following syntax:

```
let %listname|%listname[num_var|num_lit]=list (col_var|str_var  
|num_var|str_lit|num_lit[,...])
```

- Accessing a list variable:

Access a list variable using the following syntax:

```
let str_var|num_var=%listname[num_var|num_lit].#colname
```

List Variable Arguments

The value between the brackets indicates either the number of rows in the list for the definition case or the row in the list to modify or assign.

If there are no brackets, there is no need to predefine; assign the types based on the given variable types. For multirow lists, the assignment must be compatible with the types given in the definition.

A NUMBER field has the same characteristics as an undeclared #var. The underlying storage depends on the contents, and the DEFAULT-NUMERIC setting applies.

The usual SQR rules for variable assignment apply to list access. Assignment is prohibited only between Date and Numeric types. Assignment of a numeric column to a string variable returns the string representation of the numeric value; assignment of a date variable to a string variable returns the default-edit-mask representation of the date.

Literals

Literals are text or numeric constants.

- A single quote begins and ends a text literal. For example:

```
'Hello'
```

- Numerals 0–9 begin numeric literals.

Numerals that include digits with an optional decimal point and leading sign are acceptable numeric literals. For example, `-543.21`. Numeric literals can also be expressed in scientific form. For example, `1.2E5`.

Viewing Sample Reports

To see a quick overview of how an SQR report looks, view the sample reports listed in “SQR Samples.” These programs are stored in the SQR for PeopleSoft directory `SAMPLE`. You can modify these reports to meet your needs.

See Also

SQR for PeopleSoft Developer's Guide

CHAPTER 2

SQR Command Reference

This chapter describes and demonstrates each command in the SQR lexicon. The commands in this section follow the conventions listed in the table from the Introduction, and use the abbreviations described in the following table.

Warning! If you are copying codes directly from the examples in the PDF file, make sure you change the slanted quotes to regular quotes or you will receive an error message.

<i>Abbreviation</i>	<i>Description</i>	<i>Example</i>
txt_col	Text column retrievable from a database.	&address
num_col	Numeric column retrievable from a database.	&price
date_col	Date or datetime column retrievable from a database.	&date1
txt_var	String variable defined in a program.	\$your_name
num_var	Numeric variable defined in a program.	#total_cost
date_var	A variable explicitly defined as a date variable.	\$date1
any_lit	A literal of any type.	'abc' 12
any_var	A variable of any type.	\$string #number \$date
any_col	A column of any type.	&string &number &date
txt_lit	Text literal defined in a program.	'Company Confidential'
num_lit	Numeric literal defined in a program.	12345.67
int_lit	Integer literal defined in a program.	12345
nn	Integer literal used as an argument to a command.	123
position	The position qualifier, which consists of the line, column, and length specification. The minimum position, (), means to use the current line and column position on the page for the length of the field being printed.	(5,10,30)

ADD

Syntax

```
ADD{src_num_lit | _var | _col} TO dst_num_var [ROUND=nn]
```

Description

Adds one number to another.

The source value is added to the destination variable and the result is placed in the destination. The source is always first and the destination is always second.

When dealing with money-related values (dollars and cents), use decimal variables rather than float variables. Float variables are stored as double precision floating point numbers, and small inaccuracies can appear when adding many numbers in succession. These inaccuracies can appear due to the way floating point numbers are represented by different hardware and software implementations and also due to inaccuracies that can be introduced when converting between floating point and decimal.

Parameters

Parameter	Description
<i>src_num_lit _var _col</i>	A numeric source value added to the contents of <i>dst_num_var</i> .
<i>dst_num_var</i>	A numeric destination variable that contains the result after execution.
ROUND	Rounds the result to the specified number of digits to the right of the decimal point. For float variables, this value can be from 0 to 15. For decimal variables, this value can be from 0 to the precision of the variable. For integer variables, this argument is not appropriate.

Example

```
add 10 to #counter
add #counter to #new_count
add &price to #total round=2
```

See Also

The LET command for information about complex arithmetic expressions.

ALTER-COLOR-MAP

Syntax

```
ALTER-COLOR-MAP
```

```

NAME = {color_name_lit | _var | _col}
VALUE = ({color_name_lit | _var | _col} | {rgb})

```

Description

Dynamically alters a defined color.

The ALTER-COLOR-MAP command is allowed wherever the PRINT command is allowed. This command enables you to dynamically alter a defined color. You cannot use this command to define a new color.

Parameters

Parameter	Description
NAME	Defines the name of the color you want to alter. For example, <i>light blue</i> .
VALUE	Defines the RGB value of the color you want to alter. For example, (193, 233, 230).
{color_name_lit _var _col}	A color_name is composed of the alphanumeric characters (A-Z, 0-9), the underscore (_) character, and the hyphen (-) character. It must start with an alpha (A-Z) character. It is case insensitive. The name 'none' is reserved and cannot be assigned a value. A name in the format (RGBredgreenblue) cannot be assigned a value. The name 'default' is reserved and may be assigned a value. 'Default' is used during execution when a referenced color is not defined in the runtime environment.
{rgb}	red_lit _var _col, green_lit _var _col, blue_lit _var _col where each component is a value in the range of 000 to 255. In the BEGIN-SETUP section, only literal values are allowed.

The default colors implicitly installed with SQR include:

- black = (0,0,0)
- white=(255,255,255)
- gray=(128,128,128)
- silver=(192,192,192)
- red=(255,0,0)
- green=(0,255,0)
- blue=(0,0,255)
- yellow=(255,255,0)

- purple=(128,0,128)
- olive=(128,128,0)
- navy=(0,0,128)
- aqua=(0,255,255)
- lime=(0,128,0)
- maroon=(128,0,0)
- teal=(0,128,128)
- fuchsia=(255,0,255)

Example

```

begin-setup
  declare-color-map
    light_blue = (193, 222, 229)
  end-declare
end-setup

begin-program
  alter-color-map name = 'light_blue' value = (193, 233, 230)

  print 'Yellow Submarine' ()
    foreground = ('yellow')
    background = ('light_blue')

  get-color print-text-foreground = ($print-foreground)
  set-color print-text-foreground = ('purple')
  print 'Barney' (+1,1)
  set-color print-text-foreground = ($print-foreground)
end-program

```

See Also

The DECLARE-COLOR-MAP, SET-COLOR AND GET-COLOR commands in this section.

ALTER-CONNECTION

Syntax

```

ALTER-CONNECTION
NAME=connection_name
[DSN={uq_txt_lit|_var}]
[USER={uq_txt_lit|_var}]

```

```
[PASSWORD={uq_txt_lit|_var}]
[PARAMETERS=keyword_str=attr_str;
[, keyword_str=attr_str;...]]
```

Description

Alters the data source logon parameters prior to logon. You can use this command to override the default connection logon parameters.

Parameters

Parameter	Description
<i>connection_name</i>	A user-defined name for describing a datasource connection.
DSN	The logical datasource name as recorded in the Registry.properties file.
USER, PASSWORD	Traditional logon semantics.
PARAMETERS = <i>keyword_str=attr_str</i> ;	Defines a list of keyword-attribute pairs required by a datasource driver for logon. There is no syntax restriction on these entries apart from the delimiting semi-colons (;) and equal signs (=). The keywords must match the logon property names listed for a Data Source in its property file.
NO-DUPLICATE=TRUE FALSE (default is FALSE)	This optional keyword prevents SQR from automatically creating additional logins to Data Sources that are busy handling a previous query. Creating a new login in such cases is the default behavior for SQR, allowing a single CONNECTION declaration to be used in a subquery. This behavior, while allowing dynamic logins as-needed, causes problems when doing both DDL (BEGIN-SQL) and DML (BEGIN-SELECT) against temporary tables in certain vendors Data Sources. In such cases, you must fetch from the temporary table using the same login in which it was created. Here, you should code the CONNECTION as NO-DUPLICATE=TRUE, and then use that connection in both the table creation logic of BEGIN-SQL and the row fetching logic of BEGIN-SELECT.

Example

```
alter-connection
  name=PT814
  password=psswd
parameters=logon.client=600;logon.ashost=starfish;logon.sysnr=
00;logon.language=EN;
```

Note. Do not wrap the lines in the 'parameters=' line. Space restrictions dictate the wrapped line in the preceding example.

See Also

The DECLARE-CONNECTIONS command.

ALTER-LOCALE

Syntax

```
ALTER-LOCALE
[LOCALE={txt_lit _var|DEFAULT|SYSTEM}]
[NUMBER-EDIT-MASK={txt_lit|_var|DEFAULT|SYSTEM}]
[MONEY-EDIT-MASK={txt_lit|_var|DEFAULT|SYSTEM}]
[DATE-EDIT-MASK={txt_lit|_var|DEFAULT|SYSTEM}]
[INPUT-DATE-EDIT-MASK={txt_lit|_var|DEFAULT|SYSTEM}]
[MONEY-SIGN={txt_lit|_var|DEFAULT|SYSTEM}]
[MONEY-SIGN-LOCATION={txt_var|DEFAULT|SYSTEM|LEFT
|RIGHT}]
[THOUSAND-SEPARATOR={txt_lit|_var|DEFAULT|SYSTEM}]
[DECIMAL-SEPARATOR={txt_lit|_var|DEFAULT|SYSTEM}]
[DATE-SEPARATOR={txt_lit|_var|DEFAULT|SYSTEM}]
[TIME-SEPARATOR={txt_lit|_var|DEFAULT|SYSTEM}]
[EDIT-OPTION-NA={txt_lit|_var|DEFAULT|SYSTEM}]
[EDIT-OPTION-AM={txt_lit|_var|DEFAULT|SYSTEM}]
[EDIT-OPTION-PM={txt_lit|_var|DEFAULT|SYSTEM}]
[EDIT-OPTION-BC={txt_lit|_var|DEFAULT|SYSTEM}]
[EDIT-OPTION-AD={txt_lit|_var|DEFAULT|SYSTEM}]
[DAY-OF-WEEK-CASE={txt_var|DEFAULT|SYSTEM|UPPER|LOWER
|EDIT|NO-CHANGE}]
[DAY-OF-WEEK-FULL=( {txt_lit1|_var1}...{txt_lit7
|_var7} ) ]
[DAY-OF-WEEK-SHORT=( {txt_lit1|_var1}...{txt_lit7
|_var7} ) ]
[MONTHS-CASE={txt_var|DEFAULT|SYSTEM|UPPER|LOWER|EDIT
|NO-CHANGE}]
[MONTHS-FULL=( {txt_lit1|_var1}...{txt_lit12|_var12} ) ]
[MONTHS-SHORT=( {txt_lit1|_var1}...{txt_lit12|_var12} ) ]
```

Description

Selects a locale or changes locale parameters used for printing date, numeric, and money data and for data accepted by the INPUT command. A locale is a set of preferences for language, currency, and the presentation of charts and numbers.

The SYSTEM locale represents the behavior of older versions of SQR prior to Version 4.0. When you install SQR for PeopleSoft Version 4.0 or later, the default locale is set to **SYSTEM**. This provides upwards compatibility for older SQR programs. The SYSTEM locale settings are described in the following table.

Keyword	Value
NUMBER-EDIT-MASK	The PRINT command prints two digits to the right of the decimal point and left justifies the number in the field. The MOVE, SHOW, and DISPLAY commands format the number with six digits to the right of the decimal point and left justifies the number.
MONEY-EDIT-MASK	SQR uses the same default as the NUMBER-EDIT- MASK keyword.
DATE- EDIT-MASK	SQR uses the default database date format. See the Date Time section for more details.
INPUT-DATE-EDIT-MASK	SQR uses a default date edit mask with the INPUT command. See the Sample Date Edit Masks table for a listing of the date edit mask.
MONEY-SIGN	'\$'
MONEY-SIGN-LOCATION	LEFT
THOUSAND-SEPARATOR	','
DECIMAL-SEPARATOR	'.'
DATE-SEPARATOR	'/'
TIME-SEPARATOR	':'
EDIT-OPTION-NA	'n/a'
EDIT-OPTION-AM	'am'
EDIT-OPTION-PM	'pm'
EDIT-OPTION-BC	'bc'
EDIT-OPTION-AD	'ad'
DAY-OF-WEEK-CASE	EDIT
DAY-OF-WEEK-FULL	('Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday')
DAY-OF-WEEK-SHORT	('Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat')
MONTHS-CASE	EDIT
MONTHS-FULL	('January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December')
MONTHS-SHORT	('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec')

Parameters

Note. Many of the settings can have a value of DEFAULT or SYSTEM. For a given setting, specifying DEFAULT retrieves the value from the corresponding setting of the *default* locale as identified in the [Default-Settings] section of the SQR.INI file. Similarly, specifying the keyword SYSTEM retrieves the value from the corresponding setting of the *system* locale. You can alter the *system* locale using the ALTER-LOCALE command; however, you cannot define it in the SQR.INI file.

Parameter	Description
LOCALE	Specifies the name of the locale to use. This name must be defined in the SQR.INI file. If this field is omitted, then the current locale is used. The locale name is case-insensitive and is limited to the following character set: A-Z, 0-9, underscore, or hyphen. The current locale can be determined by printing the reserved variable \$sqr-locale.
NUMBER-EDIT-MASK	Specifies the numeric edit mask to use with the keyword NUMBER in a PRINT, MOVE, SHOW, or DISPLAY command.
MONEY-EDIT-MASK	Specifies the numeric edit mask to use with the keyword MONEY in a PRINT, MOVE, SHOW, or DISPLAY command.
DATE-EDIT-MASK	The default date edit mask to used with the keyword DATE in the PRINT, MOVE, SHOW, or DISPLAY command, or the LET functions datetostr() or strtodate().
INPUT-DATE-EDIT-MASK	The default date format to use with the INPUT command when TYPE=DATE is specified with the command or the input variable is a date variable. For information about Edit Masks, see "PRINT".
MONEY-SIGN	Specifies the characters that replace the \$ or other currency symbol used in edit masks.
MONEY-SIGN-LOCATION	Specifies where to place the MONEY-SIGN characters. Valid values are LEFT and RIGHT.
THOUSAND-SEPARATOR	Specifies the character to replace the ',' edit character.
DECIMAL-SEPARATOR	Specifies the character to replace the '.' edit character.
DATE-SEPARATOR	Specifies the character to replace the '/' character.
TIME-SEPARATOR	Specifies the character to replace the ':' character.
EDIT-OPTION-NA	Specifies the characters to use with the 'na' option.

Parameter	Description
EDIT-OPTION-AM	Specifies the characters to replace 'AM'.
EDIT-OPTION-PM	Specifies the characters which to replace 'PM'.
EDIT-OPTION-BC	Specifies the characters to replace 'BC'.
EDIT-OPTION-AD	Specifies the characters to replace 'AD'.
DAY-OF-WEEK-CASE	Specifies how the case for the DAY-OF-WEEK-FULL or DAY-OF-WEEK-SHORT entries are affected when used with the format codes 'DAY' or 'DY'. Valid values are UPPER, LOWER, EDIT, and NO-CHANGE. UPPER and LOWER force the output to either all uppercase or lowercase, ignoring the case of the format code in the edit mask. Use EDIT to follow the case specified with the format code in the edit mask. Use NO-CHANGE to ignore the case of the format code and output the day of week explicitly listed in the DAY-OF-WEEK-FULL or DAY-OF-WEEK-SHORT entries.
DAY-OF-WEEK-FULL	Specifies the full names for the days of the week. SQR considers the first day of the week to be Sunday. You must specify all seven days.
DAY-OF-WEEK-SHORT	Specifies the abbreviated names for the days of the week. SQR considers the first day of the week to be Sunday. You must specify all seven abbreviations.
MONTHS-CASE	Specifies how the case for the MONTHS-FULL or MONTHS-SHORT entries are affected when used with the format codes 'MONTH' or 'MON'. Valid values are UPPER, LOWER, EDIT, and NO-CHANGE. UPPER and LOWER force the output to either all uppercase or lowercase, ignoring the case of the format code in the edit mask. Use EDIT to follow the case specified with the format code in the edit mask. Use NO-CHANGE to ignore the case of the format code and output the month explicitly listed in the MONTHS-FULL or MONTHS-SHORT entries.
MONTHS-FULL	Specifies the full names for the months of the year. SQR for PeopleSoft considers the first month of the year to be January. All 12 months must be specified.
MONTHS-SHORT	Specifies the abbreviated names for the months of the year. SQR for PeopleSoft considers the first month of the year to be January. All 12 abbreviations must be specified.

Example

```

!
! The following program segments will illustrate the various
! ALTER-LOCALE features.
!
begin-setup
  declare-variable
    date $date $date1 $date2 $date3
  end-declare
end-setup

!
! Set default masks
!
alter-locale
  number-edit-mask = '9,999,999.99'
  money-edit-mask  = '$999,999,999.99'
  date-edit-mask   = 'Mon DD, YYYY'

let #value = 123456
let $edit = 'Mon DD YYYY HH:MI:SS'
let $date = strtodate('Jan 01 1995 11:22:33', $edit)
show 'With NUMBER option    #Value = ' #value number
show 'With MONEY  option    #Value = ' #value money
show 'Without NUMBER option #Value = ' #value
show 'With DATE option      $Date = ' $date date
show 'Without DATE option   $Date = ' $date

```

Produces the following output:

```

With NUMBER option    #Value =    123,456.00
With MONEY  option    #Value = $    123,456.00
Without NUMBER option #Value = 123456.000000
With DATE option      $Date = Jan 01, 1995
Without DATE option   $Date = 01-JAN-95

!
! Reset locale to SQR defaults and assign a multi-character
! money-sign.
!
alter-Locale
  locale = 'System'
  money-sign = 'AU$'                                ! Australian dollars

let #value = 123456
show #value edit '$999,999,999,999.99'
show #value edit '$$$$,$$$999,999.99'

```

Produces the following output:

```
AU$      123,456.00
      AU$123,456.00

!
! Move the money-sign to the right side of the value.  Note
! the leading space.
!
alter-locale
  money-sign = ' AU$'           ! Australian dollars
  money-sign-location = right

let #value = 123456
show #value edit '$999,999,999,999.99'
show #value edit '$$$$,$$$$999,999.99'
```

Produces the following output:

```
      123,456.00 AU$
      123,456.00 AU$

!
! Reset locale to SQR defaults and flip the thousand and
! decimal separator characters.
!
alter-locale
  locale = 'System'
  thousand-separator = '.'
  decimal-separator = ','

let #value = 123456
show #value edit '999,999,999,999.99'
```

Produces the following output:

```
123.456,00

!
! Reset locale to SQR defaults and change the date and time
! separators
!
alter-locale
  locale = 'System'
  date-separator = '-'
  time-separator = '.'

let $edit = 'Mon/DD/YYYY HH:MI:SS'
let $date = strtodate('Jan/01/1995 11:22:33', $edit)
```

```
show $date edit :$edit
```

Produces the following output:

```
Jan-01-1995 11.22.33
```

```
!
! Reset locale to SQR defaults and change the text used with
! the edit options 'na', 'am', 'pm', 'bc', 'ad'
!
alter-locale
  locale = 'System'
  edit-option-na = 'not/applicable'
  edit-option-am = 'a.m.'
  edit-option-pm = 'p.m.'
  edit-option-bc = 'b.c.'
  edit-option-ad = 'a.d.'

let $value = ''
let $edit = 'Mon DD YYYY HH:MI'
let $date1 = strtodate('Jan 01 1995 11:59', $edit)
let $date2 = strtodate('Feb 28 1995 12:01', $edit)
show $value edit '999,999,999,999.99Na'
show $date1 edit 'Mon DD YYYY HH:MI:SS PM'
show $date2 edit 'Mon DD YYYY HH:MI:SS pm'
```

Produces the following output:

```
Not/Applicable
Jan 01 1995 11:59:00 A.M.
Feb 28 1995 12:01:00 p.m.
```

```
!
! Input some dates using the 'system' locale and
! output using other locales from the SQR.INI file.
!
alter-locale
  locale = 'System'
let $date1 = strtodate('Jan 01 1995', 'Mon DD YYYY')
let $date2 = strtodate('Feb 28 1995', 'Mon DD YYYY')
let $date3 = strtodate('Mar 15 1995', 'Mon DD YYYY')
show 'System:'
show
show $date1 edit 'Month DD YYYY' ' is ' $date1 edit 'Day'
show $date2 edit 'Month DD YYYY' ' is ' $date2 edit 'Day'
show $date3 edit 'Month DD YYYY' ' is ' $date3 edit 'Day'
alter-locale
  locale = 'German'
show
```

```

show 'German:'
show
show $date1 edit 'DD Month YYYY' ' ist ' $date1 edit 'Day'
  show $date2 edit 'DD Month YYYY' ' ist ' $date2 edit 'Day'
show $date3 edit 'DD Month YYYY' ' ist ' $date3 edit 'Day'
alter-locale
  locale = 'Spanish'
show
show 'Spanish:'
show
show $date1 edit 'DD Month YYYY' ' es ' $date1 edit 'Day'
show $date2 edit 'DD Month YYYY' ' es ' $date2 edit 'Day'
show $date3 edit 'DD Month YYYY' ' es ' $date3 edit 'Day'

```

Produces the following output:

System:

```

January 01 1995 is Sunday
February 28 1995 is Tuesday
March 15 1995 is Wednesday

```

German:

```

01 Januar 1995 ist Sonntag
28 Februar 1995 ist Dienstag
15 März 1995 ist Mittwoch

```

Spanish:

```

01 enero 1995 es domingo
28 febrero 1995 es martes
15 marzo 1995 es miércoles

```

See Also

The DISPLAY, LET, MOVE, PRINT, AND SHOW commands in this section.

PSSQR.INI in PSSQR.INI.

ALTER-PRINTER

Syntax

```

ALTER-PRINTER
[POINT-SIZE={point_size_num_lit|_var}]
[FONT-TYPE={font_type|txt_var}]
[SYMBOL-SET={symbol_set_id|txt_var}]

```

```
[FONT={font_int_lit|_var}]
[PITCH={pitch_num_lit|_var}]
```

Description

Alters printer parameters at runtime.

You can place the ALTER-PRINTER command in any part of an SQR program except the **SETUP** section.

ALTER-PRINTER attempts to change the attributes of the *current* printer for the *current* report. If an attribute does not apply to the *current* printer, it is ignored. For example, **ALTER-PRINTER** is ignored if it specifies proportional fonts for a report printed on a line printer. When your program is creating multiple reports and the printer is shared by another report, the attributes are changed for that report as well.

Parameters

Parameter	Description
POINT-SIZE	Specifies the new font point size.
FONT-TYPE	Specifies the new font type. Enter PROPORTIONAL or FIXED.
SYMBOL-SET	Specifies the new symbol set identifier.
FONT	Specifies the new font as a number. (For example, 3 = Courier and 4 = Helvetica.)
PITCH	Specifies the new pitch in characters per inch.

See the "DECLARE-PRINTER Command Arguments" Table for information about these arguments.

Example

Change the font and symbol set for the current printer.

```
alter-printer

font=4      ! Helvetica

symbol-set=12U    ! PC-850 Multilingual
```

If the output prints to a PostScript printer, the SYMBOL-SET argument is ignored; however, if the .SPF file is kept (see the **-KEEP** command line flag) and later printed on an HP LaserJet, the symbol set 12U can be used.

See Also

The DECLARE-PRINTER command.

ALTER-REPORT

Syntax

```
ALTER-REPORT
  [HEADING={heading_name_txt_lit|_var|_col}]
  [HEADING-SIZE={heading_size_int_lit|_var|_col}]
  [FOOTING={footing_name_txt_lit|_var|_col}]
  [FOOTING-SIZE={heading_size_int_lit|_var|_col}]
```

Description

Alters some of the report-specific functionality.

This command gives you the ability to dynamically change those heading and/or footing sections that are active for the current report. You can also change how much space the heading and/or footing sections occupy.

If the HEADING or FOOTING value is set to 'NONE' then that section is disabled for the current report.

If the HEADING or FOOTING value is set to 'DEFAULT' then that section reverts to whatever was in effect when the report was initiated.

If no HEADING or FOOTING value is set then the HEADING-SIZE and/or FOOTING-SIZE values affect the HEADING/FOOTING currently being used.

If the ALTER-REPORT command was not invoked from within a BEGIN- HEADING and/or BEGIN-FOOTING section and the page has not been written to, then the assignment takes effect immediately; otherwise it takes effect for the next page.

Parameters

<i>Parameter</i>	<i>Description</i>
HEADING	Specifies the name of the BEGIN-HEADING section to use.
HEADING-SIZE	Specifies the amount of space the BEGIN-HEADING section will occupies in the page.
FOOTING	Specifies the name of the BEGIN-FOOTING section to use.
FOOTING-SIZE	Specifies the amount of space the BEGIN-FOOTING section will occupies in the page.

Example

```
begin-footing 2 name=confidential
  print 'Company Confidential' (1,1,0) center
  page-number (2,37,0)
```

```

end-footing

begin-footing 2          name=proprietary
  print 'Company Proprietary' (1,1,0) center
  page-number (2,37,0)
end-footing

begin-report
  alter-report
    footing = 'Proprietary'
    footing-size = 6          ! Increase
depth
.
.
.
end-report

```

See Also

The BEGIN-FOOTING command.

The BEGIN-HEADING command.

ARRAY-ADD, ARRAY-DIVIDE, ARRAY-MULTIPLY, ARRAY-SUBTRACT

Syntax

```

ARRAY-ADD{src_num_lit|_var|_col}...TO
dst_array_name (element_lit|_var|_col) [field
[(occurs_lit|_var|_col)]]...
ARRAY-DIVIDE{src_num_lit|_var|_col}...INTO
dst_array_name (element_int_lit|_var|_col) [field
[(occurs_lit|_var|_col)]]...
ARRAY-MULTIPLY{src_num_lit|_var|_col}...TIMES
dst_array_name (element_int_lit|_var|_col) [field
[(occurs_lit|_var|_col)]]...
ARRAY-SUBTRACT{src_num_lit|_var|_col}...FROM
dst_array_name (element_int_lit|_var|_col) [field
[(occurs_lit|_var|_col)]]...

```

Description

These four commands perform arithmetic on one or more elements in an array.

The following information applies to the array arithmetic commands:

- The array must first be created using the CREATE-ARRAY command.

- The four array arithmetic commands perform on one or more source numbers, placing the result into the corresponding field in the array.
- Array element and field occurrence numbers can be numeric literals (123) or numeric variables (*#j*) and can be from zero (0) to one less than the size of the array.
- If fields are not listed, the results are placed into consecutively defined fields in the array. If fields are listed, results are placed into those fields, at the specified occurrence of the field. If an occurrence is not specified the zeroth (0) occurrence is used.
- All fields must be of the type **NUMBER**, **DECIMAL**, **FLOAT**, or **INTEGER**. They cannot be of type **DATE**, **CHAR**, or **TEXT**.
- If division by zero is attempted, a warning message appears, the result field is unchanged, and SQR continues executing.

Parameters

Parameter	Description
<i>src_num_lit/_var/_col</i>	Source values are added to, divided into, multiplied times, or subtracted from the respective destination array fields. All variables must be numeric in type.
<i>dst_array_name (element_int_lit/_var/_col) [field [(occurs_lit/_var/_col)]]</i>	Destination array fields contain the results after the operation. All variables must be numeric in type.

Example

The following example adds *&salary* and *#comm* to the first two fields defined in the *emps* array. The *#j*'th element of the array is used.

```
array-add &salary #comm to emps(#j)
```

The following example subtracts *#lost*, *#count*, and 1 from the fields *loses*, *tot* and sequence of the *#j2*'th element of the *stats* array.

```
array-subtract #lost #count 1 from stats(#j2) loses tot sequence
```

The following example multiplies occurrences 0 through 2 of the field *p* in the *#i*'th element of the *percentages* array by 2.

```
array-multiply 2 2 2 times percentages(#i) p(0) p(1) p(2)
```

The following example divides the *#i2*'th occurrence of the *salesman* field of the *#j*'th element of the *commissions* array by 100.

```
array-divide 100 into commissions(#j) salesman(#i2)
```

The following example uses the **ARRAY-ADD** command in an SQR program.

```

begin-setup
! declare arrays
create-array name=emps size=1                                ! one row
needed for this example
    field=Salary:number=35000                                ! initialize to
35,000
    field=Comm:number=5000                                    !
initialize to 5,000
end-setup

begin-program
do Main
end-program

begin-procedure Main local
! Show original contents of the arrays, then the modified arrays
! array-add
! retrieve values from the only row of array "emps"
get #sal #com FROM emps(0) Salary Comm
print 'Array-Add'          (+1, 1)

print 'Add 1000 to each column' (+1, 1)
print 'Salary' (+1, 3) bold underline
print 'Comm' (,25) bold underline

print #sal (+1, 1) money
print #com (,22) money

let #salary = 1000
let #commission = 1000
let #j = 0 ! address the array row with variable "#j"
! Add 1000 (in variables) to each column of row 0 (the 1st and only row)
array-add #salary #commission TO emps(#j)
! retrieved the new "added" values
get #sal #com FROM emps(0) Salary Comm
print #sal (+1,1) money
print #com (,22) money
end-procedure

```

See Also

The CREATE-ARRAY command for information about creating an array.

The CLEAR-ARRAY command for information about clearing or initializing an array.

The GET, PUT, and LET commands for information about using arrays.

ASK

Syntax

```
ASK substitution_variable [prompt]
```

Description

Retrieves values for a compile-time substitution variable. The retrieval can be by user input, command-line arguments, or as entries in the @file on the command line.

See "SQR Command-line Flags" in the Introduction section for more information.

The value of the substitution variable replaces the reference variable in the program. Variables are referenced by enclosing the variable name in braces, for example, '{state_name}'. If the substitution variable is text or date, surround the brackets by single quotes. Substitutions are made as the program is compiled and are saved in the SQT file. Each variable can be referenced multiple times.

ASK is used only in the SETUP section and must appear prior to any substitution variable references.

You cannot break the **ASK** command across program lines.

Parameters

Parameter	Description
<i>substitution_variable</i>	The variable to be used as the substitution variable.
<i>Prompt</i>	An optional literal text string to be displayed as a prompt if the value for the substitution variable is not entered on the command line or in an argument file.

Example

In the following example, state takes the value entered by the user in response to the prompt *Enter state for this report*.

```
begin-setup
  ask state 'Enter state for this report'
end-setup
...
begin-select
  name, city, state, zip
from customers where state = '{state}'
end-select
```

See Also

The **INPUT** command for information about input at runtime.

SQR for PeopleSoft Developer's Guide, “Compiling Programs and Using SQR Execute”.

BEGIN-DOCUMENT

Syntax

```
BEGIN-DOCUMENT position
END-DOCUMENT
```

Description

Begins a DOCUMENT paragraph. A document paragraph enables you to write free-form text to create form letters, invoices, and so on.

Database columns, SQR variables, and document markers can be referenced within a document. Their location in the document determines where they are printed on the page. You should not use tabs inside a document paragraph. To indent text or field use the spacebar. Note also that if the variables being printed inside a document paragraph are variable in length, you might want to manipulate the variable outside the **DOCUMENT** paragraph.

Note. A document must be executed before you can reference its document markers. Because documents can be printed at relative positions on the page, the actual location of a document marker may not be known by SQR until the document itself has been executed.

Parameters

<i>Parameter</i>	<i>Description</i>
<i>position</i>	The location on the page where the document begins. The position can be fixed or relative to the current position.

Example

```
begin-document (1,1)
.b
Dear $firstname
...
end-document
```

See Also

The END-DOCUMENT command.

SQR for PeopleSoft Builder Developer's Guide, “Creating Form Letters” (for a full example of the **BEGIN-DOCUMENT** command).

BEGIN-EXECUTE

Syntax

```

BEGIN-EXECUTE
[CONNECTION=uq_txt_lit]
[ON-ERROR=procedure[(arg1[,argi]...)]]
[RSV=num_var]
[STATUS=list_var|num_var|txt_var]
[SCHEMA=txt_lit|txt_var]
[PROCEDURE=txt_lit|txt_var]
[PARAMETERS=(arg1[IN|INOUT|NULL[,argi[IN|INOUT]]...]]]
|COMMAND=txt_lit|txt_var
|GETDATA=txt_lit|txt_var]
[BEGIN-SELECT[BEFORE=sqr_procedure[(arg1[,argi]...)]]]]
[AFTER=sqr_procedure[(arg1[,argi]...)]]]]
col-name type=char|text|number|date[edit-mask]
[on-break]...
FROM ROWSETS=( {m, -n, n-m, m- /all} )
|FROM PARAMETER=txt_lit|txt_var
END-SELECT]
END-EXECUTE

```

Description

Begins a new construct. In a BEGIN-EXECUTE paragraph, the syntax of **BEGIN-SELECT** varies as shown in the following syntax:

Parameters

Parameter	Description
CONNECTION	Identifies a name previously specified using the DECLARE-CONNECTION construct. If you do not specify a name, SQR Server uses the default connection. The default connection is defined by the command-line entries for datasource (DSN), username (USER), and password (PASSWORD). Name is not case-sensitive.
ON-ERROR	Declares a procedure to execute if an error occurs.
RSV	Row Set Variable. A global SQR variable containing the row set being retrieved.
STATUS	Identifies a list or scalar variable that receives the status of the stored procedure.
SCHEMA	Identifies the location in the datasource of the object being queried.

Parameter	Description
PROCEDURE	The name of the datasource-stored procedure to be executed. If the datasource is SAP R/3, this procedure is a BAPI. The name may include spaces.
PARAMETER_LIST	Scalar and/or list variables of the form list_var num_lit txt_lit txt_var num_var any_col. If you do not specify the keywords IN or INOUT, IN is the default. Specify all parameters in order; leaving any parameters unnamed causes a syntax error. To ignore a parameter, fill its position with the keyword NULL. This results in a Null value for that parameter position.
COMMAND	A text string you pass to the datasource without modification by SQR. This string can include embedded SQR variables.
BEFORE/AFTER	Names an SQR procedure to be executed before or after the row set. The procedure is not performed unless at least one row is returned from the specified rowsets.
FROM ROWSET	Special case addition to the BEGIN-SELECT syntax. Available for use with all datasource types, including SAP R/3 and JDBC. Names the rowsets from which to retrieve the column variables. If you specify more than one row set, use identical column name/type signatures. Row set numbers must be sequential from left-to-right within the parentheses, and they must not overlap as in this example: (1-3, 2-4). Numeric literals or #variables are allowed.
FROM PARAMETER	Special case addition to the BEGIN-SELECT syntax. Available only for SAP R/3 datasources. Use only in conjunction with PROCEDURE keyword. This argument names an output parameter containing one or more rows from which the column variables are to be retrieved.
GETDATA	Supports the Java (DDO) GetData paradigm for data access.
PROPERTIES = (txt_var strlit = txt_var strlit num_var num_lit any_col, ...)	Specifies a (set of) keyword/value pairs that represent modifications to be made to the Properties of the datasource (specified by the CONNECTION = statement). An arbitrary number of such pairs can be specified.

Note. This is a similar concept to the PARAMETERS = statement in Declare- and alter-connection, with the minor difference being that the properties here specified alter the flow of returned information, as opposed to just setting login properties. Can be used in conjunction with any data-access model (Procedure, Command, Getdata). An application of this statement would be in the MDB setting, where it might be used to specify such things as Level, Generation, or Include-Column. For example, PROPERTIES = ('SetColumn' = 5)

Example

```
begin-setup
```



```

        declare-variable
            date $when_ordered
            text $ship_method
            integer #theRow
            integer #theStatus
            integer #howMany
        end-declare
    end-setup

    input #howMany type=integer
    input $pword
    let %parm1 = list($when_ordered, $ship_method, #howMany)

    declare-connection SAPR3
    user=scott
    parameters=clientno=5;node=starfish;
    end-declare

    alter-connection
        name=SAPR3
        password=$pword

    Begin-Execute
        connection=SAPR3
        rsv=#theRow
        status=#theStatus
        on-error=it_failed(#theStatus)
        procedure='CreditHistory version 5'
        parameters=(%parm1,'recalculate')

        print 'proc ran OK, status is ' (+1,1)
        print #theStatus (,+5) edit 999

    Begin-Select before=do_eject after=cleanup
    city &col=char (1,1) on-break level=1 after=city-tot
    keyval type=number (1,+1)
    rcvd type=date (0,+2)
    from Rowsets=(1)
    End-Select

    End-Execute

```

See Also

The EXECUTE command.

BEGIN-FOOTING

Syntax

```
BEGIN-FOOTING footing_lines_int_lit
[FOR-REPORTS=(report_name1 [, report_namei] ...)]
[FOR-TOCS=(toc_name1 [, toc_namei] ...)]

[NAME={ footing_name }]

END-FOOTING
```

Description

Begins the **FOOTING** section.

The **FOOTING** section defines and controls information to be printed at the bottom of each page.

You must define the *report_name* in a DECLARE-REPORT paragraph. If you do not use **DECLARE-REPORT**, the footing is applied to all reports. You can also specify **FOR-REPORTS=(ALL)**. Note that the parentheses are required.

You can specify more than one **BEGIN-FOOTING** section; however, there can be only one for each report. A **BEGIN-FOOTING** section with **FOR-REPORTS=(ALL)** can be followed by other **BEGIN-FOOTING** sections for specific reports, which override the **ALL** setting.

You must define the *toc_name* in a DECLARE-TOC paragraph. You can also specify **FOR-TOCS=(ALL)**. Note that the parentheses are required.

You can specify more than one **BEGIN-FOOTING** section; however, there can be only one for each Table of Contents. A **BEGIN-FOOTING** section with **FOR-TOCS=(ALL)** can be followed by other **BEGIN-FOOTING** sections for a specific Table of Contents, which override the **ALL** setting.

The **BEGIN-FOOTING** section can be shared between reports and Table of Contents.

You can print outside the Footing area of the report, that is into the body area, from the Footing, but you cannot print into the Footing area from the body.

Parameters

Parameter	Description
<i>footing_lines_int_lit</i>	The number of lines to be reserved at the bottom of each page.
FOR-REPORTS	Specifies the reports to which this footing applies. This argument is required only for a program with multiple reports. If you are writing a program that produces a single report, you can ignore this argument.

Parameter	Description
FOR-TOCS	Specifies the Table of Contents to which this heading applies.
NAME	Specifies the name to be associated with this footing section. This is used in conjunction with the ALTER-REPORT command. The name cannot be NONE or DEFAULT.

Example

```

begin-footing 2 for-reports=(customer, summary)
  print 'Company Confidential' (1,1,0) center
  page-number (2,37,0)
end-footing
begin-footing 2                                ! For all reports
  print 'Division Report' (1,1,0) center
  page-number (2,37,0)
end-footing
begin-footing 2 for-tocs=(all)
  print 'Table of Contents' (2,1)
  let $page = roman(#page-count)
               ! ROMAN numerals
  print $page (,64)
end-footing

```

See Also

The ALTER-REPORT command for information about dynamic headings/footings.

The DECLARE-LAYOUT command for information about page layout.

The DECLARE-REPORT command for information about programs with multiple reports.

The DECLARE-TOC command for information about Table of Contents.

The END-FOOTING command.

BEGIN-HEADING

Syntax

```

BEGIN-HEADING heading_lines_int_lit
[FOR-REPORTS=(report_name1 [, report_namei] ...)]
[FOR-TOCS=(toc_name1 [, toc_namei] ...)]

[NAME={footing_name}]

END-HEADING

```

Description

Begins a **HEADING** section.

The **HEADING** section defines and controls information to be printed at the top of each page.

You must define the *report_name* in a DECLARE-REPORT paragraph. If you do not use **DECLARE-REPORT**, the heading is applied to all reports. You can also specify **FOR-REPORTS=(ALL)**. Note that the parentheses are required.

You can specify more than one **BEGIN-HEADING** section; however, there can be only one for each report. A **BEGIN-HEADING** section with **FOR- REPORTS= (ALL)** can be specified followed by other **BEGIN-HEADING** sections for specific reports, which override the **ALL** setting.

You must define the *toc_name* in a DECLARE-TOC paragraph. You can also specify **FOR-TOCS= (ALL)**. Note that the parentheses are required.

You can specify more than one **BEGIN-HEADING** section; however, there can be only one for each Table of Contents. A **BEGIN-HEADING** section with **FOR- TOCS=(ALL)** can be specified followed by other **BEGIN-HEADING** sections for specific Table of Contents, which override the **ALL** setting.

The **BEGIN-HEADING** section can be shared between reports and Table of Contents.

You can print outside the heading area of the report, that is into the body area, from the heading, but you cannot print into the heading area from the body.

Parameters

Parameter	Description
<i>heading_lines_int_lit</i>	The number of lines to be reserved at the top of each page.
FOR-REPORTS	Specifies the reports to which this heading applies. This is required only for a program with multiple reports. If you are writing a program that produces a single report, you can ignore this argument.
FOR-TOCS	Specifies the Table of Contents to which this heading applies.
NAME	Specifies the name to be associated with this heading section. This option cannot be used if FOR-REPORTS or FOR-TOCS is also specified. This is used in conjunction with the ALTER-REPORT command. The name cannot be NONE or DEFAULT.

Example

```
begin-heading 2                                ! Use 2 lines for
print $current-date (1,1) edit MM/DD/YY        ! heading,
print 'Sales for the Month of ' (1,30)         ! 2nd is blank.
print $month ()
```

```

end-heading
begin-heading 2 for-tocs=(all)
  print 'Table of Contents' (1,1) bold center
end-heading

```

See Also

The ALTER-REPORT command for information about dynamic headings/footings.

The **DECLARE-LAYOUT** command for information about page layout.

The DECLARE-REPORT command for information about programs with multiple reports.

The **DECLARE-TOC** command for information about Table of Contents.

The END-HEADING command.

BEGIN-PROCEDURE

Syntax

```

BEGIN-PROCEDURE procedure_name [LOCAL] (arg1
[, argi]...)
END-PROCEDURE

```

Description

Begins a procedure. A procedure is one of the most powerful parts of the SQR language. It modularizes functions and provides standard execution control.

The procedure name must be unique. The name is referenced in DO commands. Procedures contain other commands and paragraphs (for example, **SELECT**, **SQL**, **DOCUMENT**).

By default, procedures are global. That is, variables or columns defined within a procedure are known and can be referenced outside the procedure.

A procedure is local when the word **LOCAL** appears after the procedure name or when the procedure is declared with arguments. That is, variables declared within the procedure are available only within the procedure, even when the same variable name is used elsewhere in the program. In addition, any query defined in a local procedure will have local database column variable names assigned that do not conflict with similarly named columns defined in queries in other procedures.

SQR procedure can be called recursively. However, unlike C or Pascal, SQR maintains only one copy of the local variables and they are persistent.

Arguments passed by a **DO** command to a procedure must match in number:

- Database text or date columns, string variables, and literals can be passed to procedure string arguments. If passing a date string to a date argument, the date string must be in the format specified by the SQR_DB_DATE_FORMAT setting, or a database dependent format (see the Default Database Formats table), or the database-independent format `SYYYMMDD [HH24 [MI [SS [NNNNNN] ] ] ]`.
- Database numeric columns, numeric variables, and numeric literals can be passed to procedure numeric arguments.
- Numeric variables (DECIMAL, `INTEGER`, FLOAT) can be passed to procedure numeric arguments without regard to the argument type of the procedure. SQR automatically converts the numeric values upon entering and leaving the procedure as required.
- Date variables or columns can be passed to procedure date or string arguments. When passing a date variable or column to a string argument, the date is converted to a string according to the following rules:
 - For DATETIME columns and SQR DATE variables SQR uses the format specified by the SQR_DB_DATE_FORMAT setting. If this has not been set, the first database-dependent format as listed in the Default Database Formats table.
 - For DATE columns the format specified by the SQR_DB_DATE_ONLY_FORMAT setting is used. If this has not been set, the format listed in the Default Database Formats table is used.
 - For TIME columns the format specified by the SQR_DB_TIME_ONLY_FORMAT setting is used. If this has not been set, the format listed in the TIME Column Formats table is used.

To reference or declare global variables from a local procedure, add a leading underscore to the variable name, after the initial \$, #, or &. (Example: `#_amount`)

Note. All the SQR reserved variables, such as `#sql-status` and `$sql-error`, are global variables. Within a local procedure, they must be referenced using the leading underscore: `#_sql-status` or `$_sql-error`.

Parameters

Parameter	Description
<code>procedure_name</code>	Specifies a unique name for this procedure. Procedure names are not case-sensitive.
LOCAL	Specifies that this is a local procedure.

Parameter	Description
<i>arg1</i> [, <i>argi</i>]...	Specifies arguments to be passed to or returned from the procedure. Arguments can be either string variables (\$arg), numeric variables (#arg), or date variables (\$arg). If you want to return a value passed back to the calling DO command, place a colon (:) before the variable name. The arguments of the BEGIN-PROCEDURE and DO commands must match in number, order, and type.

Example

The following example shows a procedure, `main`, that also executes the procedure `print-list`, for each row returned from the SELECT statement. No parameters are passed to `print-list`.

```
begin-procedure main
begin-select
name
address
phone
  do print_list
from custlist order by name
end-select
end-procedure                                ! main
```

In the following example, five arguments are passed to the `Calc` procedure:

```
do Calc (&tax, 'OH', &county_name, 12, #amount)

begin-procedure Calc(#rate, $state, $county, #months, :#answer)
.
.
.
let #answer = ...
end-procedure
```

In the preceding example the value for `:#answer` is returned to `#amount` in the calling DO command.

The following example references global variables:

```
begin-procedure print-it ($a, $b)
print  $_deptname (+2,5,20)                                ! $deptname is
print  $a          (+1)                                     ! declared outside
print  $b          (+1)                                     ! this procedure
end-procedure
```

See Also

The `DO` and `END-PROCEDURE` commands.

BEGIN-PROGRAM

Syntax

```
BEGIN-PROGRAM
END-PROGRAM
```

Description

Begins the program section of an SQR program.

After processing the commands in the **SETUP** section, if any, SQR starts program execution at the BEGIN-PROGRAM section. The PROGRAM section typically contains a list of **DO** commands, though other commands can be used. This is the only required section in an SQR program.

Example

```
begin-program
  do startup
  do main
  do finish
end-program
```

See Also

The BEGIN-REPORT, **BEGIN-SETUP**, and END-PROGRAM command.

BEGIN-SELECT

Syntax

```
BEGIN-SELECT [DISTINCT] [-Cnn] [-Bnn] [-XP] [-NR] [-SORTnn]
[-LOCK{RR|CS|RO|RL|XX}] [-DBdatabase]
[-DBconnectionstring]
[LOOPS=nn] [ON-ERROR=procedure (arg1[,argi]...)]
{column} [&synonym]
{expression &synonym}
{[$columnname] &synonym = (char | number | date)}
[SQR commands]
FROM {table,...| [table:$tablename]}
      [additional SQL]
      [$variable]
END-SELECT
```


Description

Begins a **SELECT** paragraph. A SELECT paragraph is the principal means of retrieving data from the database and printing it in a report. A SELECT paragraph must be inside a **PROCEDURE** or **BEGIN-PROGRAM** section.

Note that **SELECT * FROM** is not a valid SQR SQL statement. **BEGIN- SELECT** can be placed inside a **BEGIN-PROGRAM** section.

Parameters

Note. The arguments can span multiple lines; however, do not use the first character position unless the continuation character terminated the previous line. Otherwise, the argument will be misconstrued as a **SELECT** column.

Parameter	Description
DISTINCT	Specifies that you want to eliminate duplicate rows from your query.
-Cnn	(Ingres, Oracle, SQLBase) Sets the context area size (buffer size for query) to larger or smaller than the default; this option is rarely needed.
-Bnn	(ODBC, Oracle, SYBASE CT-Lib) Sets the number of rows to retrieve at one time. This is for performance purposes only. Regardless of this setting, all rows are selected. The default, without using -B, is 10 rows. An overall setting for a program can be indicated on the SQR command line with -B, which can be overridden by a separate -B flag on each BEGIN-SELECT command.

Parameter	Description
-XP	(SYBASE) Prevents the creation of a stored procedure for the SELECT paragraph. When -XP is specified, SQR generates a new SQL statement using the current value of any bind variables each time the BEGIN-SELECT is executed. This disables the potential performance optimization created by stored procedures. If you change the variables frequently during execution and you do not want SQR to automatically create stored procedure, use this flag. You can also use -XP if the username/password to your program does not have permission to create stored procedures. However, if you do not change variables frequently during execution, the use of stored procedures may optimize your program's performance. In that case, do not use this argument. Note also that -XP is available as a command-line flag. -XP improves performance when you use bind variables and dynamic query variables in the same query. Each time the dynamic query variable changes in value, a new stored procedure is created. If the dynamic query variable changes often and the query contains bind variables, you create many stored procedures if you do not use -XP.
-NR	(SQLBase) Specifies that No Recovery mode is to be used when connecting to the database for this SELECT paragraph.
-SORT <i>nn</i>	(SQLBase) Specifies the size of the sort buffer in characters.
-LOCK	(SQLBase) Defines the types of locking (isolation level) for the session. Use XX to inhibit setting the default level (RR). See the SQLBase documentation for a full explanation of the different isolation levels.
-DB <i>database</i>	(SQLBase) Specifies the default database name for this SELECT paragraph only.

Parameter	Description
<code>-DBconnectionstring</code>	(ODBC) Specifies the ODBC connection string for this SELECT paragraph only. A connection string has the following syntax: DSN=data_source_name[;keyword=value[;keyword=value [;...]]] This option enables you to combine data from multiple databases in one program. For example, a connection string for an Oracle database named "ora8" might look like the following: 'DSN=ora7;UID=scott;PWD=tiger' where DSN, UID, and PWD are keywords common to all drivers (representing: name, user ID, and password, respectively). Connection string options are always separated by a semicolon (;). Other driver-specific options may be added to the connection string using driver-defined keywords. See your ODBC driver documentation for available options.
LOOPS	Specifies the number of rows to retrieve. After the specified number has been processed, the SELECT loop exits.
ON-ERROR	Declares a procedure to execute if an error occurs due to incorrect SQL syntax. Error trapping should be used in conjunction with dynamic query variables. SELECT paragraphs without dynamic variables are checked for errors before the program is processed and therefore do not require a special error procedure. You can optionally specify arguments to be passed to the ON-ERROR procedure. Arguments can be any variable, column, or literal.

Example

In this example, duplicate rows are not selected for the city, state, and zip columns because of the "distinct" keyword. The numbers within parentheses accompanying City, State, and Zip define the column positions of these rows. Column names can not have spaces in front of them.

See "Column Variables" on in the *SQR for PeopleSoft Developer's Guide*.

```
begin-select distinct
city          (1,1,30)
```

```

state          (0,+2,2)
zip            (1,+3,6)
from custlist order by city
end-select

```

In this example, the first two columns may, or may not, be present when the statement is compiled. The column `cust_id` is declared to be a number. A runtime error is produced if the database table, as identified by the variable `$table_name`, declares it to be something other than a number.

```

begin-select          loops=100
[$col_var_char]      &col1=char
[$col_var_num]       &col2=number
cust_id              &id=number
from [$table_name]
[$where clause]
[$order_by_clause]
end-select

```

See Also

SQR for PeopleSoft Developer's Guide, “Selecting Data From the Database” and “Dynamic SQL and Error Checking.”

The END-SELECT and `EXIT-SELECT` commands.

BEGIN-SETUP

Syntax

```

BEGIN-SETUP
END-SETUP

```

Description

Begins a SETUP section. This section is optional, but if included, it is processed prior to the `BEGIN-PROGRAM`, `BEGIN-HEADING`, and `BEGIN-FOOTING` section.

The SETUP section should be the first section in the program.

The `SETUP` section contains commands that determine the overall characteristics of the program. The commands used in the SETUP section cannot be used elsewhere unless specified. The `SETUP` section can include the following commands:

```

ASK
  BEGIN-SQL

```

(This command can also be used in `BEGIN-PROCEDURE` paragraphs.)

```

CREATE-ARRAY

```

(This command can also be used in the other sections of an SQR program.)

```
DECLARE-CHART
DECLARE-IMAGE
DECLARE-LAYOUT
DECLARE-PRINTER
DECLARE-PROCEDURE
DECLARE-REPORT
DECLARE-VARIABLE
```

(This command can also be used in **LOCAL** procedures.)

```
DECLARE-TOC
LOAD-LOOKUP
```

(This command can also be used in the other sections of an SQR program.)

```
USE
```

(SYBASE and Microsoft SQL Server only)

Example

```
begin-setup
  declare-layout customer_list
    paper-size=(8.5, 11)
    left-margin=1.0
    right-margin=1.0
  end-declare
end-setup
```

See Also

The ASK, **BEGIN-SQL**, CREATE-ARRAY, **LOAD-LOOKUP**, and USE commands.

BEGIN-SQL

Syntax

```
BEGIN-SQL [-Cnn] [-XP] [-NR] [-SORTnn]
[-LOCK{RR|CS|RO|RL|XX}]
[-DBdatabase] [-DBconnectionstring]
  [ON-ERROR=procedure[ (arg1[, argi] . . . ) ] (non-setup)
  | [ON-ERROR={STOP|WARN|SKIP}] (insetup)
END-SQL
```

Description

Begins an SQL paragraph. This paragraph can reside in a **BEGIN-PROCEDURE**, **BEGIN-SETUP**, or **BEGIN-PROGRAM** section.

BEGIN-SQL starts all SQL statements except SELECT, which has its own BEGIN-SELECT paragraph. If a single paragraph contains more than one SQL statement, each statement except, of course, the last must be terminated by a semicolon (;).

If a single paragraph contains more than one SQL statement, and the **-C** flag is used, all are assigned the same context area size or logical connection number.

Only non-SELECT statements can be used (except **SELECT INTO** for SYBASE and Microsoft SQL Server). Columns and variables can be referenced in the SQL statements.

Stored Procedures

For Ingres, SYBASE, and Microsoft SQL Server, SQR supports stored procedures with the EXECUTE command. For Oracle, stored procedures are implemented using PL/SQL in the **BEGIN-SQL** paragraph.

For some databases such as ORACLE, using DDL statements within a BEGIN- SQL paragraph causes a commit of outstanding inserts, updates, and deletes and releases cursors. For this reason, ensure that these are done in the proper order or unpredictable results may occur.

Oracle PL/SQL

For Oracle, PL/SQL is supported in a **BEGIN-SQL** paragraph. This requires an additional semicolon at the end of each PL/SQL statement. See the following Oracle example.

For Oracle PL/SQL:

```
begin-sql
declare
    varpl varchar2 (25);;
    var2 number (8,2);;
begin
    varpl := 'abcdefg';;
    $v1 := varpl;;
    $v2 := '1230894asd';;
    var2 := 1234.56;;
    #v := var2;;
end;;
end-sql
```

For Oracle stored procedures:

```
begin-sql
begin
    #dept_number := get_dept_no($dept_name);;
```

```
end;;
end-sql
```

Parameters

Parameter	Description
-Cnn	(Ingres, Oracle, SQLBase) Sets the context area size (buffer size for query) to larger or smaller than the default; this option is rarely needed.
-XP	(SYBASE) Prevents the creation of a stored procedure for the SQL paragraph. When -XP is specified, SQR generates a new SQL statement using the current value of the bind variables each time the BEGIN-SQL is executed. This disables the performance optimization created by stored procedures. If you change the variables frequently during execution and you do not want SQR to automatically create stored procedures, use this flag. You may also use it if your program does not have permission to create stored procedures. However, if you do not change variables frequently during execution, the use of stored procedures optimizes the performance of the program. In that case, do not use this argument. -XP improves performance when you use bind variables and dynamic query variables in the same query. Each time the dynamic query variable changes in value, a new stored procedure is created. If the dynamic query variable changes often and the query contains bind variables, you create many stored procedures if you do not use -XP.
-NR	(SQLBase) Specifies that No Recovery mode is to be used when connecting to the database for this SQL paragraph.
-SORTnn	(SQLBase) Specifies the size of the sort buffer in characters.
-LOCK	(SQLBase) Defines the types of locking (isolation level) for this SQL paragraph. Use XX to inhibit setting the default level (RR). See the SQLBase documentation for a full explanation of the different isolation levels.
-DBdatabase	(SQLBase) Specifies the default database name for this SQL paragraph only.

Parameter	Description
<code>-DBconnectionstring</code>	(ODBC) Specifies the ODBC connection string for this SQL paragraph only. A connection string has the following syntax: <pre>DSN=data_source_name[;keyword=value[;keyword=value[...]]]</pre> This option enables you to combine data from multiple databases in one program. For example, a connection string for an Oracle named "ora8" might look like the following: <pre>'DSN=ora8;UID=scott;PWD=tiger'</pre> where DSN, UID, and PWD are keywords common to all drivers (representing name, user ID, and password, respectively). Connection string options are always separated by a semicolon (;). Other driver-specific options may be added to the connection string using driver-defined keywords. See your ODBC driver documentation for available options.
<code>Connection=connstr</code>	This option is used with SQR for DDO. It specifies the name of a datasource previously declared using the DECLARE-CONNECTION construct. If it is not specified, the default connection is used.

Parameter	Description
ON-ERROR	Declares a procedure to execute if an error occurs due to incorrect SQL syntax except when executed in a BEGIN-SETUP section. By default, SQR reports any error and then halts; if an error procedure is declared, you can trap errors, report or log them, and continue processing. The procedure is invoked when an error occurs in any SQL statement in the paragraph. After the error procedure ends, control returns to the next SQL statement, if any. You can optionally specify arguments to be passed to the ON-ERROR procedure. Arguments can be any variable, column, or literal. If ON-ERROR is used in the SETUP section, it is a condition flag supporting the following conditions: STOP Do not run the program. WARN Run the program but with a warning message. SKIP Ignore any errors and run the program.

Example

```

begin-sql
update orders set invoice_num = #next_invoice_num

where order_num = &order_num
end-sql

begin sql
delete orders

where order_num = &order_num;
insert into orders values ($customer_name, #order_num,...)
end-sql

```

See Also

SQR for PeopleSoft Developer's Guide, “Dynamic SQL and Error Checking” and “Using DML and DDL.”

The END-SQL, **BEGIN-PROCEDURE**, and EXECUTE commands.

The **-S** command-line flag.

BREAK

Syntax

```
BREAK
```

Description

Causes an exit from within an EVALUATE or WHILE command. Execution then continues to the command immediately following the END-WHILE or END-EVALUATE.

This command is used inside a WHILE ... END-WHILE loop or within an EVALUATE command.

See Also

The WHILE and EVALUATE commands.

CALL, CALL SYSTEM

Syntax

```
CALL subroutine USING {src_txt_lit|_var|_col}|{ src_num_lit|_var|_col}
{dst_txt_var|_num_var} [param]
```

To issue operating system commands from within an SQR program, use the following syntax:

```
CALL SYSTEM USING command status [ WAIT | NOWAIT ]
```

Description

Issues an operating system command or calls a subroutine that you have written in another language such as C or COBOL and passes the specified parameters.

You can write your own subroutines to perform tasks that are awkward in SQR. Subroutines can be written in any language.

Warning! PeopleSoft recommends that the *ucall* function not use any database calls as it may cause erroneous results.

Used in an SQR program, CALL has the following format:

```
CALL your_sub USING source destination [param_literal]
CALL SYSTEM USING command status [WAIT|NOWAIT]
```

The CALL SYSTEM is a special subroutine which is provided as part of SQR to enable the program to issue operating system commands. Its arguments, *command*, *status*, and *WAIT|NOWAIT* are described previously.

The values of the source and destination variables and the parameter's literal value are passed to your subroutine. Upon return from the subroutine, a value is placed in the destination variable.

You must write the subroutine and call it in one of the supplied ucall routines. (Optionally, you could rewrite `ucall` in another language instead).

The source file UCALL.C contains sample subroutines written in C. The ucall function takes the following arguments:

Argument	Description	How Passed
<i>callname</i>	Name of the subroutine.	By reference with a maximum of 31 characters, null terminated.
<i>strsrc</i>	Source string.	By reference with a maximum of 255 characters, null terminated.
<i>strdes</i>	Destination string.	By reference with a maximum of 255 characters.
<i>dblsrc</i>	Source double floating point.	By reference.
<i>dbldes</i>	Destination double floating point.	By reference.
<i>param</i>	Subroutine parameter string. It must be a literal.	By reference with a maximum of 80 characters, null terminated.

When you use the CALL command, your arguments are processed as follows:

- Calling arguments are copied into the variables depending on the type of argument. Strings are placed into *strsrc*, and numerics are placed into *dblsrc*.
- Return values are placed into *strdes* or *dbldes* depending on whether your destination argument for CALL is a string or numeric variable.

The destination arguments can also be used to pass values to your subroutine.

To access your subroutine, add a reference to it in UCALL, passing along the arguments you need.

You must relink SQR to `CALL` after compiling a user defined function that becomes another SQR function.

You must add your subroutine to the link command file—in UNIX it is called SQRMAKE, in NT it is called SQREXT.MAK—if you have created a new object file. (Alternatively, you could add your routine to the bottom of the UCALL source module that is already included in the link).

Your subroutine and calling SQR program are responsible for passing the correct string or numeric variables and optional parameter string to the subroutine. No checking is performed.

Parameters

Parameter	Description
<i>subroutine</i>	Specifies the name of your subroutine.
<i>src_txt_lit/_var/_col</i>	Specifies a text column, variable, or literal which is to be input to the called subroutine.
<i>src_num_lit/_var/_col</i>	Specifies a numeric column, variable (decimal, float, or integer), or literal that is to be input to the called subroutine.
<i>dst_txt_var/_num_var</i>	Specifies a text or numeric variable (decimal, float, or integer) into which the called subroutine is to place the return result. See the "UCALL Subroutine Arguments" table for more information.
<i>param</i>	Specifies an optional alphanumeric string of characters to be passed as a parameter to the subroutine.
SYSTEM	Specifies that this CALL command issues an operating system command.
<i>command</i>	Specifies the operating system command to execute. The command can be a quoted string, string variable, or column.
<i>status</i>	Contains the status returned by the operating system. The status must be a numeric variable. The value returned in status is system-dependent as shown in the following table.

System	Value Returned
OS/2, and UNIX	Zero (0) indicates success. Any other value is the system error code.
PC/Windows	A value less than 32 indicates an error.
VAX/VMS	If the SPAWNING of the subordinate DCL session is successful, then the value returned is the status back from the DCL session (for example, the DCL \$STATUS value). If the SPAWNING of the subordinate DCL session is unsuccessful the failing status from the LIB\$SPAWN function is returned.

Parameter	Description
WAIT NOWAIT	(Windows and VMS only) - WAIT specifies that SQR suspend its execution until the CALL SYSTEM command has finished processing. NOWAIT specifies that SQR start the CALL SYSTEM command but continue its own processing while that command is in progress. For Windows, the default is NOWAIT. For VMS, the default is WAIT. On UNIX operating systems the behavior is always WAIT.

Example

See these sample subroutines included in the **UCALL** source file:

- TODASH shows how strings can be manipulated.
- **SQROOT** demonstrates how to access numerics.
- SYSTEM invokes a secondary command processor.

The following code calls these subroutines:

```

call todash using $addr $newaddr '/.',
                                ! Convert these to
                                !
dashes
call sqroot using #n #n2
                                ! Put square root of
                                ! #n
into #n2
call sqroot using &hnvr #j
                                ! Hnvr is numeric
                                !
database column
call system using 'dir' #s
                                ! Get directory listing

```

The following example uses the **SYSTEM** argument to issue an operating system command. Some operating systems let you invoke a secondary command processor to enter one or more commands and then return to SQR.

```

! VAX/VMS                                (Type 'logout' to return to SQR)
!
call system using 'spawn' #vms_status

! Unix                                    (Type 'exit' to return to SQR)
!
let $shell = getenv('SHELL')
if isblank($shell)
    let $shell = '/bin/sh'
end-if

```

```

call system using $shell #unix_status

!Windows 98/NT (Type 'exit' to return to SQR)
!
let $comspec = getenv('COMSPEC')
let $cmd = comspec || '/c' || $comspec || ' /k'
call system using $cmd #win_status wait

```

The following step-by-step example shows how to add a user-defined subroutine to SQR so that it can be invoked from SQR using the CALL command. For this example, the C function `initcap`, which makes the first letter of a string uppercase, is added. The function accepts two parameters. The first parameter is the string to which the `initcap` function is applied. The second is the resultant string.

To add `initcap` function to SQR, the following modifications are needed to the `UCALL.C` file, which was provided with SQR:

1. Add the prototype for the `initcap` function:

```
static void initcap CC_ARGS((char *, char *));
```

2. Modify the `ucall` routine in the `UCALL.C` file. Specifically, add an `else if` statement at the end of the `if` statement to check for the `initcap` function:

```

void ucall CC_ARGL((callname, strsrc, strdes, dblsrc, dbldest, params))
...

/* If other subroutines, add "else if..." statement for each */
else if (strcmp(callname,"initcap") == 0)
    initcap(strsrc, strdes);
else
    sq999("Unknown CALled subroutine: %s\n", callname);
return;
}

```

3. At the end of the `ucall.c` file, add the `initcap` routine listed in the following example. The routine name must be lowercase; however, in your SQR program it can be referenced either uppercase or lowercase.

```

static void initcap CC_ARGL((strsrc, strdes))
CC_ARG(char *, strsrc) /* Pointer to source
string */
CC_LARG(char *, strdes) /* Pointer to destination
string */
{
    int nIndex;
    int nToUpCase;
    char cChar;

```

```

nToUpCase = 1;
for (nIndex = 0; cChar = strsrc[nIndex]; nIndex++)
{
    if (isalnum(cChar))
    {
        if (nToUpCase)
            strdes[nIndex] = islower(cChar) ? toupper(cChar) :
cChar;
        else
            strdes[nIndex] = isupper(cChar) ? tolower(cChar) :
cChar;
        nToUpCase = 0;
    }
    else
    {
        nToUpCase = 1;
        strdes[nIndex] = cChar;
    }
}
strdes[nIndex] = '\0';
}

```

Note. The CC_ARG macros are defined in the UCALL.C source module. The macros give the programmer the ability to define a fully prototyped function without having to worry if the C compiler supports the feature.

After these modifications, recompile UCALL.C and relink SQR. See the programming manual for your particular machine for details.

Finally, the following is an example of a simple SQR program which uses the `initcap` function:

```

begin-program
    input $name 'Enter the first name ' ! Get the first name
                                         ! from the user
    lowercase $name                      ! Set the first
name to                                ! all lower case

    call initcap using $name $capname
    ! Now set the first

                                         ! character to upper case
    input $last 'Enter the last name '
    ! Get the last name from the
                                         ! user
    lowercase $last                      ! Set the last
name to all

```

```
! lower case
call initcap using $last $caplast
! Now set the first

! character to upper case

.
.
.
```

See Also

The LET command for information about user-defined functions using `ufunc.c` that can be used in the context of an expression and that can pass and/or return any number of arguments.

CLEAR-ARRAY

Syntax

```
CLEAR-ARRAY NAME=array_name
```

Description

Resets each field of an array to its initial value.

The CLEAR-ARRAY command resets each field of the named array to the initial value specified for that field in the `CREATE-ARRAY` command. If no initial value was specified, numeric fields are reset to zero, text fields are reset to null, and date fields are reset to null. CLEAR-ARRAY also releases all memory used by the specified array and returns it to its pristine state.

Parameters

Parameter	Description
NAME	Specifies the name of the array to be cleared.

Example

```
clear-array name=custs
```

See Also

The `CREATE-ARRAY` command.

CLOSE

Syntax

```
CLOSE {filenum_lit|_var_col}
```


Description

Closes a file, specified by its file number.

Closes a flat file that has been previously opened using the **OPEN** command.

Parameters

Parameter	Description
<i>filenum_lit/_var_col</i>	Specifies the number assigned to the file in the OPEN command.

Example

```
close 5
close #j
```

See Also

The OPEN, **READ**, and WRITE commands.

COLUMNS
Syntax

```
COLUMNS {int_lit/_var/_col} [int_lit/_var/_col] ...
```

Description

Defines logical columns to be used for **PRINT** commands.

COLUMNS defines the left-most position of one or more columns within the current page layout. It sets the first column as current.

COLUMNS can be used for printing data either down the page or across the page, depending on how you use the NEXT-COLUMN and **USE-COLUMN** commands.

The COLUMNS command applies only to the current report. If you want to print columns in more than one report, you must specify the **COLUMNS** command for each report.

The USE-COLUMN 0 turns off columns. See **USE-COLUMN**.

Parameters

Parameter	Description
<i>int_lit/_var/_col</i>	Specifies the left margin position of each column.

See Also

The NEXT-COLUMN, **NEXT-LISTING**, NEW-PAGE, **USE-COLUMN**, and USE-REPORT commands.

COMMIT
Syntax

```
COMMIT
```

Description

Causes a database commit.

COMMIT is useful when you are doing many inserts, updates, or deletes in an SQL paragraph. A database commit releases the locks on the records that have been inserted, updated, or deleted. Used with some databases, it also has other effects. For this reason, it should not be used within the scope of an active SELECT paragraph or unpredictable results may occur.

When the application completes, a commit is performed automatically unless a **ROLLBACK** was done or, for callable SQR, the -XC flag was set.

Other commands or options, such as the **CONNECT** command and the use of DDL statements for some databases with a BEGIN-SQL paragraph, can also cause the database to do a commit.

COMMIT is an SQR command and *should not* be used within an SQL paragraph. If used in an SQL paragraph, unpredictable errors can occur.

Note. The COMMIT command can be used with SQR servers for DB2, Informix, Ingres, ODBC, ORACLE, and SQLBase. For SYBASE and Microsoft SQL Server, use BEGIN TRANSACTION and **COMMIT TRANSACTION** within SQL paragraphs as in the following code segment.

Example

```
add 1 to #updates_done
if #updates_done > 50
    commit
    move 0 to #updates_done
end-if
```

For SYBASE:

```
...      ! Begin Transaction occurred previously
begin-sql
    insert into custlog values (&cust_num, &update_date)
end-sql
```

```

add 1 to #inserts
  if #inserts >= 50
    begin-sql
      commit transaction;                                ! Commit every 50
rows
      begin transaction                                ! Begin next
transaction
    end-sql
    move 0 to #inserts
  end-if

...      ! One more Commit Transaction is needed

```

Warning! Any data being changed by a current transaction is locked by the database and cannot be retrieved in a **SELECT** paragraph until the transaction is completed by a **COMMIT** or **ROLLBACK** statement (or **COMMIT TRANSACTION** or **ROLLBACK TRANSACTION** statement for SYBASE or Microsoft SQL Server).

CONCAT

Syntax

```

CONCAT {src_any_lit|_var|_col} WITH
dst_txt_var[:$]edit_mask

```

Description

Concatenates a variable, column, or literal with a string variable.

The contents of the source field are appended to the end of the destination field.

CONCAT can optionally edit the source field before appending it. Edit masks can be changed dynamically by placing them in a string variable and referencing the variable name preceded by a colon (:).

See the command, "PRINT" for information about the use of edit masks.

Also, the source can be a date variable or column. If an edit mask is not specified, the date is converted to a string according to the following rules:

- For DATETIME columns and SQR DATE variables, the format specified by the SQR_DB_DATE_FORMAT setting. If this has not been set, the first database-dependent format as listed in the Default Database Formats table is used.
- For DATE columns, the format specified by the SQR_DB_DATE_ONLY_FORMAT setting is used. If this has not been set, the format listed in the Default Database Formats table is used.

- For TIME columns, the format specified by the `SQR_DB_TIME_ONLY_FORMAT` setting is used. If this has not been set, the format as listed in the TIME Column Formats table is used.

Parameters

Parameter	Description
<i>src_any_lit/_var/_col</i>	Specifies the source field to be concatenated with the <i>dst_txt_var</i> field.
<i>dst_txt_var</i>	Contains the result after execution.
<i>edit_mask</i>	Specifies an optional edit mask.

Example

```
concat &zip_plus_4 with $zip '-xxxx'           ! Edit zip plus 4.
concat &descrip with $rec :$desc_edit          ! Edit mask in
variable.
concat $date1 with $string                     !
Concatenate a date.
```

See Also

The PRINT command for information about edit masks.

The LET command for string functions.

The STRING and UNSTRING commands.

CONNECT

Syntax

```
CONNECT {txt_lit|_var|_col} [ON-ERROR=procedure[(arg1
[, argi]...)]]
```

Description

Logs off the database and logs on under a new username and password.

The new username and password can be stored in a string variable, column, or literal.

Warning! The username/password is not encrypted, so beware of security issues.

After each CONNECT, the reserved variable *\$username* is set to the new username.

All database cursors or logons are closed before the **CONNECT** occurs. You should not issue a CONNECT within a **SELECT** or an SQL paragraph while a query is actively fetching or manipulating data from the database.

Parameters

Parameter	Description
<i>txt_lit/_var/_col</i>	Specifies a username and password for the logon.
ON-ERROR	Specifies a procedure to be executed if the logon fails. If no ON-ERROR procedure is specified and the logon fails, SQR halts with an error message.

Note. You can optionally specify arguments to be passed to the **ON-ERROR** procedure. Arguments can be any variable, column, or literal.

Example

```
connect $new-user on-error=bad-logon($new_user)
connect 'sqr/test'
```

CREATE-ARRAY**Syntax**

```
CREATE-ARRAY NAME=array_name SIZE=nn
{FIELD=name:type[:occurs]
[={init_value_txt_lit|_num_lit}]}...
```

Description

Creates an array of fields to store and process data.

You can define arrays to store intermediate results or data retrieved from the database. For example, a **SELECT** paragraph can retrieve data, store it in an array, and gather statistics at the same time. When the query finishes, a summary could be printed followed by the data previously stored in the array.

SQR creates arrays before a program starts to execute. The CREATE-ARRAY command can be used in any section of a program.

Commands to process arrays include the following:

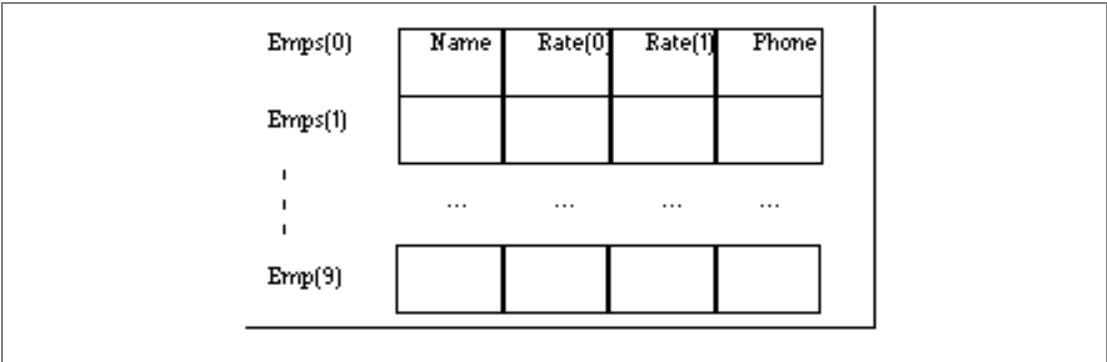
```
CREATE-ARRAY
CLEAR-ARRAY
GET
PUT
ARRAY-ADD
ARRAY-SUBTRACT
ARRAY-MULTIPLY
ARRAY-DIVIDE
LET
```

The maximum number of arrays in a program is 128; the maximum number of fields per array is 200.

Figure is a representation of an array *emps* with three fields. The following CREATE-ARRAY command defines the array:

```
create-array name=emps size=10
field=name:char='Unknown'
    field=rate:number:2=10.50
    field=phone:char='None'
```

The *name* is a simple field (one occurrence), *rate* has two occurrences, and *phone* is a simple field. Both array elements and field occurrences are referenced beginning with zero (0). The *rate* is referenced by rate(0) or rate(1). The *emps* array will contain 10 elements, 0 through 9. All *name* fields are initialized to "Unknown", all *phone* fields are initialized to "None", and all *rate* fields are initialized to 10.50.



Sample array

Parameters

Parameter	Description
NAME	Names the array. The name is referenced in other array commands.
SIZE	Defines the number of elements in the array.

Parameter	Description
FIELD	Defines each field or column in the array. Each field must be defined as type: DECIMAL[(p)] Decimal numbers with an optional precision (p). FLOAT Double precision floating point numbers. INTEGER Whole numbers. NUMBER Uses the DEFAULT-NUMERIC type. See the DECLARE-VARIABLE command. CHAR (or TEXT) Character string. DATE Same as date variable. You can specify an initialization value for each field. Each field is set to this value when the array is created and when the CLEAR-ARRAY command is executed. If no initialization value is specified, numeric fields (DECIMAL, FLOAT, INTEGER) are set to zero, character fields are set to null, and date fields are set to null. All occurrences of a multiply occurring field are set to the same value. For dates, the initialization string must be formatted as 'SYYYYMMDD[HH24[MI[SS[NNNNNN]]]]'. See the Date Edit Format Codes table for a description of the format codes.
OCCURS	Fields can optionally have a number of occurrences (occurs), that is, they can be repeated any number of times.

Example

```

create-array  name=custs size=100
    field=name:char
    field=no:number
    field=state:char
    field=zip:char
    field=contacts:char:5
    field=last-contacted:date

```

See Also

The sample report CUSTOMR4.SQR included with SQR.

The **DECLARE-VARIABLE**, **ARRAY-ADD**, **ARRAY-DIVIDE**, **ARRAY-MULTIPLY**, **ARRAY-SUBTRACT**, **GET**, **PUT**, **LET**, and **CLEAR- ARRAY** commands.

The **LOAD-LOOKUP** command for an alternative way to store database tables in memory.

CREATE-COLOR-PALETTE

Syntax

```
CREATE-COLOR-PALETTE
NAME = {palette_name_txt_lit}
COLOR_1 = {rgb_value}
COLOR_2 = {rgb_value}
[COLOR_n] = {rgb_value}
```

Description

Create a color palette.

This command enables you to create a palette of colors. There is no limit to the number of palettes that can be defined in a program. No gaps are permitted in the palette.

Parameters

<i>Parameter</i>	<i>Description</i>
NAME	Specifies the name of the color palette.
COLOR_1	Specifies the first color in the palette.
COLOR_2	Specifies the second color in the palette.
COLOR_n	Specifies the n'th color in the palette. You can specify up to 64 colors in the palette.
{rgb}	Designates a color reference. This can be expressed as (<i>r,g,b</i>) where <i>r</i> , <i>g</i> , and <i>b</i> are either a numeric literal (0 to 255), a numeric variable, or a numeric column. It can also be expressed as a (<i>c</i>) where <i>c</i> is a string literal, column, or variable that is the name of a color.

Example

```
begin-report
create-color-palette
name = 'funky'
color_1 = ('blue')
color_2 = ('red')
color_3 = ('orange')
```



```
Print-Chart Groovy
  Color-Palette = 'Funky'
end-report
```

See Also

The **DECLARE-CHART** command.

The **PRINT-CHART** command.

#DEBUG

Syntax

```
#DEBUG [x...] SQR_Command
```

Description

Causes the current command to be processed during a debugging session.

A **-DEBUG**[xx] flag in the SQR command line enables conditional compilation of SQR commands. When this flag is used, any command (including other compiler directives) preceded by the word **#DEBUG** is processed; other commands are ignored.

This is useful for placing **DISPLAY**, **SHOW**, **PRINT** or other commands in your program for testing and for deactivating them when the report goes into production.

The **-DEBUG** flag can contain a suffix up to 10 letters or digits. These characters are used to match any letters or digits appended to the **#DEBUG** preprocess command inside the program. **#DEBUG** commands with one or more matching suffix characters are processed; other commands are ignored. Commands without any suffix always match.

In addition, for each **-DEBUG**xx letter, a substitution variable is defined. For example, if the flag **-DEBUGab** is used on the command line, three substitution variables are defined: *debug*, *debuga*, and *debugb*. These variables can be referenced in **#IFDEF** commands to turn whole sections of code on or off for debugging.

Parameters

Parameter	Description
x	Is any letter or digit.

Example

The following SQR command line contains the **-DEBUG** flag with no suffixes:

```
sqr myprog sammy/baker -debug
```

The following **SHOW** command in the program executes if invoked with the previous command line because the **-DEBUG** flag was used:

```
#debug show 'The total is ' #grand-tot 999,999,999
```

In the following example, the command line contains the **-DEBUG** flag with the suffixes **a**, **b**, and **c**:

```
sqr myprog sammy/baker -debugabc
```

In the following program segment, the first three **#DEBUG** commands are compiled, but the fourth, beginning **"#debug"**, is not because its suffix does not match any of the suffixes on the **-DEBUG** flag:

```
#debuga show 'Now selecting rows...'
#debug  show 'Finished query.'
#debugb show 'Inserting new row.'
#debug  show 'Deleting row.'
```

The following example shows the use of an **#IF** with a **#DEBUG**:

```
#debuga #if {platform}='unix'
#debuga show 'Platform is UNIX'
#debuga #endif
```

See Also

The **#IF**, **#IFDEF**, and **#IFDEFDEF** commands.

DECLARE-CHART

Syntax

```
DECLARE-CHART chart_name
[DATA-LABELS=data_labels_lit]
[COLOR-PALETTE=color_palette_lit]
[ITEM-COLOR=(chart_item_keyword_lit, color_value_lit | (r,g,b))]
[CHART-SIZE=(chart_width_int_lit,chart_depth_int_lit)]
[TITLE=title_txt_lit]
[SUB-TITLE=subtitle_txt_lit]
[FILL=fill_lit]
[3D-EFFECTS=3d_effects_lit]
[BORDER=border_lit]
[POINT-MARKERS=point_markers_lit]
[TYPE=chart_type_lit]
[LEGEND=legend_lit]
```

```

[LEGEND-TITLE=legend_title_txt_lit]
[LEGEND-PLACEMENT=legend_placement_lit]
[LEGEND-PRESENTATION=legend_presentation_lit]
[PIE-SEGMENT-QUANTITY-DISPLAY=
pie_segement_quantity_display_lit]
[PIE-SEGMENT-PERCENT-DISPLAY=
pie_segement_percent_display_lit]
[PIE-SEGMENT-EXPLODE=pie_segement_explode_lit]
[X-AXIS-LABEL=x_axis_label_txt_lit]
[X-AXIS-MIN-VALUE={x_axis_min_value_lit|_num_lit}]
[X-AXIS-MAX-VALUE={x_axis_max_value_lit|_num_lit}]
[X-AXIS-SCALE=x_axis_scale_lit]
[X-AXIS-MAJOR-TICK-MARKS=x_axis_major_tick_marks_lit]
[X-AXIS-MINOR-TICK-MARKS=x_axis_minor_tick_marks_lit]
[X-AXIS-MAJOR-INCREMENT=
{x_axis_major_increment_lit|_num_lit}]
[X-AXIS-MINOR-INCREMENT=
x_axis_minor_increment_num_lit]
[X-AXIS-TICK-MARK-PLACEMENT=
x_axis_tick_mark_placement_lit]
[X-AXIS-GRID=x_axis_grid_lit]
[Y-AXIS-LABEL=y_axis_label_lit]
[Y-AXIS-MIN-VALUE={y_axis_min_value_lit|_num_lit}]
[Y-AXIS-MAX-VALUE={y_axis_max_value_lit|_num_lit}]
[Y-AXIS-SCALE=y_axis_scale_lit]
[Y-AXIS-MAJOR-TICK-MARKS=y_axis_major_tick_marks_lit]
[Y-AXIS-MINOR-TICK-MARKS=y_axis_minor_tick_marks_lit]
[Y-AXIS-MAJOR-INCREMENT=
{y_axis_major_increment_lit|_num_lit}]
[Y-AXIS-MINOR-INCREMENT=
y_axis_minor_increment_num_lit]
[Y-AXIS-TICK-MARK-PLACEMENT=
y_axis_tick_mark_placement_lit]
[Y-AXIS-GRID=y_axis_grid_lit]
END-DECLARE

```

Note. If CHART-SIZE is not defined, it must be defined in **PRINT-CHART**.

Description

Defines the attributes of a chart that can later be displayed using PRINT-CHART.

The **DECLARE-CHART** command can define the attributes of a chart to be printed as part of a report.

This command may only appear in the SETUP section.

A chart defined using DECLARE-CHART is printed by referencing its name in the **PRINT-CHART** command. Some or all of the chart attributes may be overridden at runtime

with the PRINT-CHART command. As such, **DECLARE-CHART** is useful when the basic properties of a chart are common to many PRINT-CHART commands.

Note. All DECLARE-CHART attributes can be overridden as part of the **PRINT-CHART** command. Columns are not supported within the DECLARE-CHART command or the **PRINT-CHART** command. Attributes that are specified more than once produce a warning, and the first instance is regarded as the actual value. Attributes can be used in any order, with the exception of *chart-name*, which must follow the DECLARE-CHART keyword.

Also, the **FILL** specification in the DECLARE-PRINTER command can influence the appearance of the chart. The following table lists the final appearance of the chart with a combination of values for PRINTER.COLOR and CHART.FILL options.

CHART.FILL=	PRINTER.COLO R=Y	PRINTER.COL OR=N
GRAYSCALE	GRAYSCALE	GRAYSCALE
COLOR	COLOR	GRAYSCALE
CROSSHATCH	COLOR-CROSSHATCH	CROSSHATCH
NONE	NONE	NONE

Specifying Chart Data Series Colors

Color palettes are used in the New Graphics to set the colors of each data point in a data series. Specifying the color palette to be used in a Business Chart is done by creating a SQR COLOR-PALETTE using the **CREATE-COLOR- PALETTE** command. The following code demonstrates how to create the color palette:

```
Create-Color-Palette
  Name = 'Test-Palette'
  Color_1 = (100,133,238)
  Color_2 = (0, 0, 255)
  Color_3 = (0,255,0)
  Color_4 = (0,0,255)
  Color_5 = (0,0,0)
```

Users can specify any number of palettes, with up to 64 colors defined in each palette. If there are more data points in the data sets than defines colors in the palette, the palette resets and continues to set the data point colors from Color_1 to Color_n.

After a color palette has been defined, it can be used within the DECLARE- CHART and **PRINT-CHART** commands to specify the color palette to be used. The following code demonstrates the use of a color palette:

```
Print-Chart test_Chart
  COLOR-PALETTE = 'Test-Palette'
```

Note. For now, SQR for UNIX requires a x-windows server running when NewGraphics=TRUE. This is a requirement of the New Graphics, because there is currently no method to run Java-based graphics on a UNIX system without a graphic display, either local or remote.

Specifying Chart Item Colors

Users can specify the foreground and background colors of the individual areas within a Business Chart using ITEM-COLOR = (*rgb-value*) within the **DECLARE-CHART** and **PRINT-CHART** commands. The following is a list of chart item keywords that are valid for **ITEM-COLOR**:

- ChartBackground - Background color of entire chart area
- ChartForeground - Text and Line color of chart area
- HeaderBackground - Area within the text box specified for the Title and SubTitle
- HeaderForeground - Text color of the Title and Subtitle
- FooterBackground - Area within the text box specified for the X Axis label
- FooterForeground - Text color of the X Axis label
- LegendBackground - Area within the box defining the legend
- LegendForeground - Text and outline color of the legend
- ChartAreaBackground - Area that includes the body of the chart
- ChartAreaForeground - Text and line colors of the chart area
- PlotAreaBackground - Area within the X and Y Axis of a chart
- PlotAreaForeground - Text and line colors of the plot area

Parameters

Parameter	Description
<i>chart_name</i>	A unique name to be used for referencing a chart.
CHART-SIZE	The size of the chart frame in standard SQR coordinate units.

The following DECLARE-CHART Command Arguments table describes other arguments for the DECLARE-CHART command. The default values are underlined.

Argument	Choices	Description
DATA-LABELS	Yes No	NewGraphics—If YES, SQR prints the numeric value above the individual data points. If NO, no numeric values are displayed.
COLOR-PALETTE	palette_name	NewGraphics—Sets the color palette to be used to color the individual data points in each chart (example: Bar, slice, point). A valid SQR color-palette must be defined to use COLOR-PALETTE.
ITEM-COLOR	ChartBackground ChartForeground HeaderBackground HeaderForeground FooterBackground FooterForeground LegendBackground LegendForeground ChartAreaBackground ChartAreaForeground PlotAreaBackground	NewGraphics—Set to color an individual item in a chart. Specify a chart item and a valid (r,g,b) color to set the color of the chart item.
TITLE	<u>NONE</u> text	Specifies a title for the chart. That text is placed at the top of the chart.
SUB-TITLE	<u>NONE</u> text	Specifies a subtitle for the chart. That text is placed below the title regardless of whether TITLE is specified.
FILL	<u>GRAYSCALE</u> COLOR CROSS-HATCH NONE	Specifies the type of filling that is applied to the shapes (bars, pie-segments, and so on.) that represent the data loaded in the chart. GRAYSCALE varies the density of black dots. COLOR sends color instructions to the current printer. If the current printer does not support color, then it could appear in a GRAYSCALE fashion. CROSSHATCH uses patterns to fill the shapes representing each data set. With NONE, all graph shapes are filled with white.
3D-EFFECTS	YES <u>NO</u>	3D-EFFECTS gives your chart depth if set to YES. If set to NO, the chart is displayed in the default 2D mode.
BORDER	<u>YES</u> NO	If YES, then a border is drawn around the chart. If NO, then no border is displayed around the chart.

Argument	Choices	Description
POINT-MARKERS	<u>YES</u> NO	If YES, then point markers are displayed for line charts. If NO, then point markers are not displayed.
TYPE	LINE PIE <u>BAR</u> , STACKED-BAR 100%-BAR OVERLAPPED-BAR FLOATING-BAR HISTOGRAM AREA STACKED-AREA 100%-AREA XY-SCATTER-PLOT HIGH-LOW-CLOSE	Specifies the type of chart. See the "Business Charts" section in the <i>SQR for PeopleSoft Developer's Guide</i> .
LEGEND	<u>YES</u> NO	YES, displays a legend. NO does not display a legend.
LEGEND-TITLE	<u>NONE</u> text	Specifies the title for the legend. If NONE, then no title is displayed in the legend box.
LEGEND-PLACEMENT	<u>CENTER-RIGHT</u> CENTER-LEFT UPPER-RIGHT UPPER-LEFT UPPER-CENTER LOWER-RIGHT LOWER-LEFT LOWER-CENTER	Places the legend in the specified location on the chart. The first portion of the placement parameter (CENTER, UPPER, or LOWER) is the vertical position, and the second portion is the horizontal.
LEGEND-PRESENTATION	INSIDE <u>OUTSIDE</u>	If INSIDE, then the legend is presented inside the area defined by the two axes. If OUTSIDE, then the legend is presented within the chart border, but outside of the region represented by the two axes.
PIE-SEGMENT-QUANTITY-DISPLAY	<u>YES</u> NO	If YES, then the quantity is presented for each pie segment.
PIE-SEGMENT-PERCENT-DISPLAY	<u>YES</u> NO	If YES, then percent-of-total figures is presented for each pie segment.
PIE-SEGMENT-EXPLODE	<u>NONE</u> MAX MIN USE-3RD-DATA-COLUMN	Controls what pie segments are exploded (selected) within the pie chart. MAX selects the largest segment. MIN selects the smallest segment. If USE-3RD-DATA-COLUMN, then the third field in the DATA-ARRAY is used to determine which pie segments are exploded. This third field should be a CHAR and values of 'YES' or 'Y' indicate that the segment should be exploded.

Argument	Choices	Description
X-AXIS-LABEL or Y-AXIS-LABEL	<u>NONE</u> text	Specifies a line of text to be displayed below (or alongside) the tick-mark labels on the axis.
X-AXIS-MIN-VALUE	<u>AUTOSCALE</u> number	The minimum value on the axis. If there are data values that are less than X-AXIS- MIN-VALUE, then they are not displayed. AUTOSCALE directs SQR to calculate an appropriate minimum value.
Y-AXIS-MIN-VALUE	<u>AUTOSCALE</u> number	The minimum value on the axis. If there are data values that are less than Y-AXIS- MIN-VALUE, then they are not displayed. AUTOSCALE directs SQR to calculate an appropriate minimum value.
X-AXIS-MAX-VALUE	<u>AUTOSCALE</u> number	The maximum value on the axis. If there are data values that are greater than X- AXIS-MAX-VALUE, then they are not displayed. AUTOSCALE directs SQR to calculate an appropriate maximum value.
Y-AXIS-MAX-VALUE	<u>AUTOSCALE</u> number	The maximum value on the axis. If there are data values that are greater than Y- AXIS-MAX-VALUE, then they are not displayed. AUTOSCALE directs SQR to calculate an appropriate maximum value.
X-AXIS-SCALE or Y-AXIS-SCALE	LOG <u>LINEAR</u>	LOG specifies a logarithmic scale for the axis. Otherwise, the scale is LINEAR.
X-AXIS-MAJOR-TICK- MARKS	<u>YES</u> NO	YES specifies the display tick-marks along the axis between X-AXIS-MIN-VALUE and X-AXIS-MAX-VALUE, according to the X-AXIS-SCALE setting spaced by X-AXIS-MAJOR-INCREMENT.
Y-AXIS-MAJOR-TICK- MARKS	<u>YES</u> NO	YES displays the tick-marks along the axis between Y-AXIS-MIN-VALUE and Y-AXIS-MAX-VALUE, according to the Y-AXIS-SCALE setting spaced by Y-AXIS-MAJOR-INCREMENT.

Argument	Choices	Description
X-AXIS-MINOR-TICK- MARKS	YES <u>NO</u>	YES displays the tick-marks along the axis between X-AXIS-MIN-VALUE and X-AXIS-MAX-VALUE, according to the X-AXIS-SCALE setting spaced by X-AXIS-MINOR-INCREMENT.
Y-AXIS-MINOR-TICK- MARKS	YES <u>NO</u>	YES displays the tick-marks along the axis between Y-AXIS-MIN-VALUE and Y-AXIS-MAX-VALUE, according to the Y-AXIS-SCALE setting spaced by Y-AXIS-MINOR-INCREMENT.
X-AXIS-MAJOR-INCREMENT or Y-AXIS-MAJOR-INCREMENT	<u>AUTOSCALE</u> number	Specifies for SQR the increment used for spacing the major tick-marks on the axis. AUTOSCALE directs SQR to determine an appropriate increment.
X-AXIS-MINOR-INCREMENT or Y-AXIS-MINOR-INCREMENT	number	Specifies for SQR the increment used for spacing the minor tick-marks on the axis. These must be set for the X-AXIS- MINOR-TICK-MARKS and the Y-AXIS- MINOR-TICK-MARKS to be displayed.
X-AXIS-TICK-MARK-PLACEMENT or Y-AXIS-TICK-MARK-PLACEMENT	INSIDE <u>OUTSIDE</u> BOTH	INSIDE (or OUTSIDE) directs SQR to place the tick-marks on the inside (or outside) of the axis only. BOTH directs SQR to draw the tick-marks such that they appear on both sides of the axis.
X-AXIS-GRID or Y-AXIS-GRID	<u>YES</u> NO	If YES, then a dashed grid line is drawn for each major tick-mark on the axis. If NO, the no grid line is drawn on this axis.

Example

This example declares a basic sales chart using **DECLARE-CHART**. Then, for each region, the SUB-TITLE, **DATA-ARRAY**, and other elements are overridden to provide the chart with the specific features desired.

```
begin-setup

declare-chart base_sales_chart
chart-size                = (30, 20 )
title                    = 'Quarterly Sales'
sub-title                = none
fill                    = color
```

```

3d-effects                = yes
type                      = stacked-bar
legend-title              = 'Product'
x-axis-grid               = yes
end-declare

end-setup

begin-program

print-chart base_sales_chart
sub-title                 = 'Region I'
data-array                = reg1_sales
data-array-row-count      = #rows_reg1
data-array-column-count   = 2
y-axis-max-value          = #max_of_all_regions
y-axis-min-value          = #min_of_all_regions
legend                    = no

print-chart base_sales_chart
sub-title                 = 'Region II'
data-array                = reg2_sales
data-array-row-count      = #rows_reg2
data-array-column-count   = 2
y-axis-max-value          = #max_of_all_regions
y-axis-min-value          = #min_of_all_regions
legend                    = no

end-program

begin-procedure chart_region_sales ($sub, $ary,
                                   #rows, #cols,
                                   #max_of_all_regions,
                                   #min_of_all_regions)

    print-chart base_sales_chart (20, 15 )
sub-title                 = $sub
data-array                = all sales
data-array-row-count      = #rows
data-array-column-count   = #cols
data-array-column-labels  = ('Q1', 'Q2',
                             'Q3', 'Q4' )

```

```

y-axis-max-value           = #max_of_all_regions
y-axis-min-value          = #min_of_all_regions
chart-size                 = (50, 30)

```

```
end-procedure
```

See Also

The PRINT-CHART command.

DECLARE-COLOR-MAP

Syntax

In the SETUP section:

```

DECLARE-COLOR-MAP
    color_name = ({rgb})
color_name = ({rgb})
.
.
.
END-DECLARE

```

Description

Defines colors in an SQR report.

The DECLARE-COLOR-MAP command in the BEGIN-SETUP section defines or redefines colors in an SQR report. You can define an endless number of entries.

Parameters

Parameter	Description
<i>color_name</i>	A <i>color_name</i> is composed of the alphanumeric characters (A-Z, 0-9), the underscore (_) character, and the hyphen (-) character. It must start with an alpha (A-Z) character. It is case insensitive. The name 'none' is reserved and cannot be assigned a value. A name in the format (RGBredgreenblue) cannot be assigned a value. The name 'default' is reserved and can be assigned a value. 'Default' is used during execution when a referenced color is not defined in the runtime environment.

Parameter	Description
<i>{rgb}</i>	<i>red_lit _var _col, green_lit _var _col, blue_lit _var _col</i> where each component is a value in the range of 000 to 255. In the BEGIN-SETUP section, only literal values are allowed. The default colors implicitly installed with SQR include: black = (0,0,0) white=(255,255,255) gray=(128,128,128) silver=(192,192,192) red=(255,0,0) green=(0,255,0) blue=(0,0,255) yellow=(255,255,0) purple=(128,0,128) olive=(128,128,0) navy=(0,0,128) aqua=(0,255,255) lime=(0,128,0) maroon=(128,0,0) teal=(0,128,128) fuchsia=(255,0,255)

Example

```

begin-setup
  declare-color-map
    light_blue = (193, 222, 229)
  end-declare
end-setup

```

See Also

The ALTER-COLOR-MAP, SET-COLOR, and GET-COLOR commands in this section.

DECLARE-CONNECTION
Syntax

In the SETUP section:

```
DECLARE-CONNECTION connection_name_txt_lit
```

```

DSN={uq_txt_lit}
[USER={uq_txt_lit}]
[PASSWORD={uq_txt_lit}]
[PARAMETERS=keyword_str=attr_str;[,keyword_str=attr_str ;...]]
END-DECLARE

```

In the body of the report:

```

DECLARE-CONNECTION connection_name
DSN={uq_txt_lit|_var}
[USER={uq_txt_lit|_var}]
[PASSWORD={uq_txt_lit|_var}]
[PARAMETERS=keyword_str=attr_str;[, keyword_str=attr_str;...]]
END-DECLARE

```

Description

Defines the datasource logon parameters prior to logon. Can be used to override the default connection logon parameters.

Parameters

Parameter	Description
<i>connection_name</i>	A user-defined name for describing a datasource connection.
DSN	The logical datasource name as recorded in the DDO Registry (Registry.properties file).
USER, PASSWORD	Traditional logon semantics.
PARAMETERS = <i>keyword_str=attr_str</i> ;	Defines a list of keyword-attribute pairs required by a datasource driver for logon. There is no syntax restriction on these entries apart from the delimiting semi-colons (;) and equal signs (=). The keywords must match the logon property names listed for a datasource.

Parameter	Description
NO-DUPLICATE=TRUE FALSE (default is FALSE)	This optional keyword prevents SQR from automatically creating additional logins to datasources that are busy handling a previous query. Creating a new login in such cases is the default behavior for SQR, which allows a single CONNECTION declaration to be used in a subquery. This behavior, while allowing dynamic logins as-needed, causes problems when doing both DDL (BEGIN-SQL) and DML (BEGIN-SELECT) against temporary tables in certain vendors datasources. In such cases, you must fetch from the temporary table using the same login in which it was created. Here, you should code the CONNECTION as NO-DUPLICATE=TRUE, and then use that connection in both the table creation logic of BEGIN-SQL and the row fetching logic of BEGIN-SELECT.

Example

```

declare-connection SAPR3-1
    dsn=SAPR3
    username=guest
    password=guest
end-declare

```

See Also

The ALTER-CONNECTION command.

DECLARE-IMAGE

Syntax

```

DECLARE-IMAGE image_name
    [TYPE=image_type_lit]
    [IMAGE-SIZE=(width_num_lit,height_num_lit)]
    [SOURCE=file_name_lit]
END-DECLARE

```

Note. If **TYPE**, **IMAGE-SIZE**, and **SOURCE** are not defined in **DECLARE-IMAGE**, they must be defined in **PRINT-IMAGE**.

Description

Declares the type, size, and source of an image to be printed.

The DECLARE-IMAGE command defines and names an image. This image can then be placed in a report at the position specified in the **PRINT-IMAGE** command.

Note. If the image file is unrecognizable, or has incomplete header information, then a box (either shaded, for HP printers, or having a diagonal line through it in the case of POSTSCRIPT) appears where the image is expected.

Parameters

Parameter	Description
<i>image_name</i>	Specifies a unique name for referencing the image declaration.
TYPE	Specifies the image type. Types can be EPS-FILE, HPGL-FILE, GIF-FILE, JPEG-FILE, or BMP-FILE (for Windows).
IMAGE-SIZE	Specifies the width and height of the image in SQR coordinates.
SOURCE	Specifies the name of a file containing the image. The file must be in the SQRDIR directory or you must specify the full path.

Note. If the file is not in the SQRDIR directory, the full path or no path should be given. A relative path will not do, because you must know where you execute the file from.

Example

```
declare-image officer-signature
    type           = eps-file
    source          = 'off_sherman.eps'
    image-size     = (40, 5)
end-declare
```

See Also

The PRINT-IMAGE command.

DECLARE-LAYOUT

Syntax

```
DECLARE-LAYOUT layout_name
[PAPER-SIZE=({paper_width_num_lit[uom],
paper_depth_num_lit[uom]}|{paper_name})]
[FORMFEED=form_feed_lit]
[ORIENTATION=orientation_lit]
```

```

[LEFT-MARGIN=left_margin_num_lit[uom]]
[TOP-MARGIN=top_margin_num_lit[uom]]
[RIGHT-MARGIN=right_margin_num_lit[uom]]
|LINE-WIDTH=line_width_num_lit[uom]
|MAX-COLUMNS=columns_int_lit]
[BOTTOM-MARGIN=bottom_margin_num_lit[uom]]
|PAGE-DEPTH=page_depth_num_lit[uom]
|MAX-LINES=lines_int_lit]
[CHAR-WIDTH=char_width_num_lit[uom]]
[LINE-HEIGHT=line_height_num_lit[uom]]
END-DECLARE

```

Description

Defines the attributes for the layout of an output file.

The **DECLARE-LAYOUT** command describes the characteristics of a layout to be used for an output file. A layout can be shared by more than one report. If no **DECLARE-LAYOUT** is defined or if a **DECLARE-REPORT** does not reference a defined layout, a layout named **DEFAULT** is created with the default attribute values shown in the **DECLARE-LAYOUT Command Arguments** table. For an example of how **DECLARE-LAYOUT** relates to **DECLARE-REPORT**, see the **DECLARE-REPORT "Examples"** in this document.

You can define as many layouts as are necessary for the requirements of the application. You can override the **DEFAULT** layout attributes by defining a layout called **DEFAULT** in your program. Each layout name must be unique.

SQR maps its line and column positions on the page by using a grid determined by the **LINE-HEIGHT** and **CHAR-WIDTH** arguments. That is, SQR calculates the number of columns per row by dividing the **LINE-WIDTH** by the **CHAR-WIDTH** and calculates the number of lines by dividing the **PAGE-DEPTH** by the **LINE-HEIGHT**. Each printed segment of text is placed on the page using this grid. Because the characters in proportional fonts vary in width, it is possible that a word or string is wider than the horizontal space you have allotted, especially in words containing uppercase letters or bolded characters. To account for this behavior, you can either move the column position in the **PRINT** or **POSITION** statements or indicate a larger **CHAR-WIDTH** in the **DECLARE-LAYOUT** command.

The **ORIENTATION** parameter selects the proper fonts. In addition, the parameter interacts with **PAPER-SIZE** as follows:

- When you do not specify **ORIENTATION=LANDSCAPE** or the **PAPER-SIZE dimensions**, then SQR creates a page with the dimensions set to 11 inch by 8.5 inch. This results in a page of 100 columns by 45 lines with 0.5 inch margins.
- When you specify **PAPER-SIZE=(paper_name)** the page orientation is set according to the *paper_name* specified. If you also specify **ORIENTATION** and the value differs from the **PAPER-SIZE** value, then the **ORIENTATION** value overrides the **PAPER-SIZE** value.

- When you specify **PAPER-SIZE=**(*page_width*, *page_depth*) is specified then SQR *does not* swap the page width and page depth if ORIENTATION=LANDSCAPE.

Parameters

Parameter	Description
<i>layout_name</i>	A unique layout name to be used for referencing the layout and its attributes.
<i>uom</i>	An optional suffix which denotes the unit of measure to apply to the preceding value. The following lists valid uom suffixes.

Suffix	Meaning	Definition
dp	decipoint	0.001388 inch
pt	point	0.01388 inch
mm	millimeter	0.03937 inch
cm	centimeter	0.3937 inch
in	inch	1.0000 inch

Parameter	Description
<i>paper_name</i>	An option of PAPER-SIZE. This name is associated with predefined dimensions. The following table lists valid paper names.

Name	Width	Depth	Orientation
Letter	8.5 in	11 in	Portrait
Legal	8.5 in	14 in	Portrait
A4	8.27 in	11.69 in	Portrait
Executive	7.25 in	10.5 in	Portrait
B5	7.17 in	10.12 in	Portrait
Com-10	4.125 in	9.5 in	Landscape
Monarch	3.875 in	7.5 in	Landscape
DL	4.33 in	8.66 in	Landscape
C5	6.378 in	9.016 in	Landscape

The following table describes the arguments of the DECLARE-LAYOUT command

Argument	Choice or Default uom	Default Value	Description
PAPER-SIZE	inches	8.5 in, 11 in	Physical size of the page. The first parameter is the width of the page. The second parameter is the depth or length. It may also be a predefined name. (See the Valid Paper Names table.) Note. When ORIENTATION= LANDSCAPE the default values are 11 in, 8.5 in.
FORMFEED	YES, NO	YES	Specifies whether formfeeds are to be written at the end of each page.
ORIENTA-TION	PORTRAIT, LANDSCAPE	PORTRAIT	Portrait pages are printed vertically. Landscape pages are printed horizontally. Printing in landscape for the printer type HPLASERJET requires landscape fonts.
LEFT-MARGIN	inches	0.5 in	Amount of blank space to leave at the left side of the page.
TOP-MARGIN	inches	0.5 in	Amount of blank space to leave at the top of the page.
RIGHT-MARGIN	inches	0.5 in	Amount of blank space to leave at the right side of the page. If you specify LINE-WIDTH or MAX-COLUMNS, you cannot use this parameter.
LINE-WIDTH	inches	7.5 in	Length of the line. If you specify RIGHT-MARGIN or MAX-COLUMNS, you cannot use this parameter.
MAX-COLUMNS		75	Maximum number of columns in a line. If you specify RIGHT-MARGIN or LINE-WIDTH, you cannot use this parameter.
BOTTOM-MARGIN	inches	0.5 in	Amount of blank space to leave at the bottom of the page. If you specify PAGE-DEPTH or MAX-LINES, you cannot use this parameter.
PAGE-DEPTH	inches	10 in	Depth of the page. If you specify BOTTOM-MARGIN or MAX-LINES, you cannot use this parameter.

Argument	Choice or Default uom	Default Value	Description
MAX-LINES		60	Maximum number of lines printed on the page. If you specify PAGE-DEPTH or BOTTOM-MARGIN, you cannot use this parameter.
LINE- HEIGHT	points	12 pt	Size of each SQR line on the page. There are 72 points per inch. If LINE-HEIGHT is not specified, it follows the value for POINT-SIZE, if specified. The default value of 12 points yields 6 lines per inch. For the printer type LINEPRINTER, this value is used only to calculate the TOP-MARGIN and BOTTOM-MARGIN (for example, not in computing the position on the page).
CHAR-WIDTH	points	7.2 pt	Size of each SQR horizontal character column on the page (for example, the distance between the locations (1, 12) and (1, 13)). For the printer type LINEPRINTER, this value is used only to calculate the TOP-MARGIN and BOTTOM-MARGIN (not in computing the position on the page).

Example

The following example illustrates the ability to specify the ! parameters using a different measurement system (that is, metric).

```

!
declare-layout my-layout                                ! Results in:

paper-size=(a4)                                         !      paper-size=(210mm, 297mm)

left-margin=12.7 mm                                   !      top-
margin=12.7mm                                           !

right-margin=25.4 mm                                   !      left-margin=12.7mm
end-declare                                             !      right-margin=25.4mm

!      bottom-margin=12.7mm
!      orientation=portrait
!      columns=67
!      lines=64

```

The following example changes the page dimensions and also changes the left and right margins to be 1 inch.

```

!
declare-layout large-paper                                ! Results
in:

paper-size=(14, 11)                                     !      paper-size=(14in,
11in)

left-margin=1                                           !      top-margin=0.5in

right-margin=1                                           !      left-margin=1.0in
end-declare                                             !      right-margin=1.0in

!      bottom-margin=0.5in

!      orientation=portrait

!      columns=120

!      lines=60

```

The following example retains the default page dimensions and changes the left and right margins to be 1 inch.

```

declare-layout default                                ! Results in:

left-margin=1                                           !      paper-size=(8.5in, 11in)

right-margin=1                                           !      top-margin=0.5in
end-declare                                             !      left-margin=1.0in

!      right-margin=1.0in

!      bottom-margin=0.5in

!      orientation=portrait

!      columns=65

!      lines=60

```

The following example changes the orientation to landscape. The default page dimensions of (8.5in and 11in) are swapped. The columns and rows are recalculated. All other values remain the same.

```

declare-layout default                                ! Results in:

orientation=landscape                                   !      paper-size=(11in,
8.5in)

```

```

end-declare                                !      top-margin=0.5in

                                           !      left-margin=0.5in

                                           !      right-margin=0.5in

                                           !      bottom-margin=0.5in

                                           !      orientation=landscape

                                           !      columns=100

                                           !      lines=45

```

The following example changes the orientation to landscape. The default page dimensions of (8.5in and 11in) are swapped. In addition the top margin is set to 1 inch.

```

declare-layout my_landscape                ! Results
in:

orientation=landscape                      !      paper-size=(11in,
8.5in)                                     !

top-margin=1                              !      top-margin=1.0in
end-declare                               !      left-margin=0.5in

                                           !      right-margin=0.5in

                                           !      bottom-margin=0.5in

                                           !      orientation=landscape

                                           !      columns=100

                                           !      lines=43

```

The following example illustrates how to specify the page dimensions using one of the predefined names. Note that the orientation has also changed because this example is an envelope.

```

declare-layout envelope                    ! Results in:

paper-size=(com-10)                       !      paper-
size=(4.125in, 9.5in)

end-declare                               !      top-margin=0.5in

                                           !      left-margin=0.5in

                                           !      right-margin=0.5in

```

```

!      bottom-margin=0.5in
!
!      orientation=landscape
!
!      columns=85
!
!      lines=18

```

See Also

The **DECLARE-REPORT** command.

DECLARE-PRINTER

Syntax

```

DECLARE-PRINTER printer_name
[FOR-REPORTS=(report_name1 [, report_namei] ...)]
[TYPE=printer_type_lit]
[INIT-STRING=initialization_string_txt_lit]
[RESET-STRING=reset_string_txt_lit]
[COLOR=color_lit]
[POINT-SIZE=point_size_num_lit]
[FONT-TYPE=font_type_int_lit]
[SYMBOL-SET=symbol_set_id_lit]
[STARTUP-FILE=file_name_txt_lit]
[PITCH=pitch_num_lit]
[FONT=font_int_lit]
[BEFORE-BOLD=before_bold_string_txt_lit]
[AFTER-BOLD=after_bold_string_txt_lit]
END-DECLARE

```

Description

Overrides the printer defaults for the specified printer type.

Each printer has a set of defaults as listed in the DECLARE-PRINTER Command Arguments table. The DECLARE-PRINTER command overrides these defaults.

Use the DECLARE-PRINTER command in the SETUP section to define the characteristics of the printer or printers to be used. If you need to change some of the arguments depending on the runtime environment, you can use the ALTER-PRINTER command in any part of the program except the PROGRAM and SETUP sections.

A program can contain no more than one DECLARE-PRINTER command for each printer type for each report. If you do not provide a printer declaration, the default specifications are used. The default printer attributes can be overridden by providing a DECLARE-PRINTER specification for each printer. Their names are: DEFAULT-LP for line printer, DEFAULT-HP for HP LaserJet, DEFAULT-HT for HTML, and DEFAULT-PS for PostScript.

The following table describes each of the arguments, the possible choices, and the default values.

<i>Argument</i>	<i>Choice or Measure</i>	<i>Default</i>	<i>Description</i>
FOR- REPORTS		ALL	The name of the reports that use this printer definition. The default is ALL reports. This argument is required only for a program with multiple reports. If you are writing a program that produces only a single report, you can ignore this argument.
TYPE	LINEPRINTER, POSTSCRIPT, HPLASERJET, HTML, LP, PS, HP, HT	LP	SQR creates output specific to each printer. LINEPRINTER (LP) files generally consist of ASCII characters and can be viewed by a text editor. POSTSCRIPT (PS) files consist of ASCII characters, but you need to know PostScript to understand what will be shown on the printer. HPLASERJET (HP) files are binary files and cannot be edited or viewed. HTML (HT) files consist of ASCII characters and can be viewed by a browser.
INIT-STRING		(none)	Sends control or other characters to the printer at the beginning of the report. This parameter is designed primarily for the LINEPRINTER and has limited use with other printer types. Specify non-display characters by placing their decimal values inside angle brackets. For example, <27> is the ESC or escape character.

Argument	Choice or Measure	Default	Description
RESET- STRING		(none)	Sends control or other characters to the printer at the end of the report. This parameter is designed primarily for the LINE-PRINTER and has limited use with other printer types. Specify non-display characters by placing their decimal values inside angle brackets. For example, <27> is the ESC or escape character.
COLOR	Yes, No	No	Specifies whether this printer can print in color.
POINT-SIZE	points	12	This argument does not apply to LINEPRINTER printers. This is the beginning size of the selected font.
FONT-TYPE	PROPORTIONAL, FIXED	Depends on the font	This argument applies only to HP LASERJET printers and needs to be specified only for font types not defined in the Fonts Available for HP LaserJet Printers in SQR table.
SYMBOL- SET	HP defined sets	0U	This argument applies only to HP LASERJET printers. The default value of "0U" is for the ASCII symbol set. For a complete list of the symbol sets, see the HP LaserJet <i>Technical Reference Manual</i> .
STARTUP- FILE	filename	POSTSC RI.STR	This argument applies only to POSTSCRIPT printers. This argument is used to specify an alternate startup file. Unless otherwise specified, the default startup file is located in the directory pointed to by the environment variable SQDIR.
PITCH	characters/inch	10	This argument is for HPLASERJET printers only. If you specify a fixed pitch font, you should also indicate the pitch.

Argument	Choice or Measure	Default	Description
FONT	font_number	3	This is the font number of the typeface to use. For HP LASERJET printers, this is the typeface value as defined by Hewlett-Packard. For a complete list of the typeface numbers, see the HP LaserJet <i>Technical Reference Manual</i> . For POSTSCRIPT printers, SQR supplies a list of fonts and arbitrary font number assignments in the file POSTSCRI.STR. The font numbers are the same as those for HP LaserJet printers, wherever possible, so that you can use the same font number for reports to be printed on both types of printers. You can modify the font list in POSTSCRI.STR to add or delete fonts. Read the POSTSCRI.STR file for instructions. The Fonts Available for HP LaserJet Printers in SQR table lists the fonts available in SQR internally. The Fonts Available for PostScript Printers table lists the fonts available in the SQR POSTSCRI.STR file.
BEFORE-BOLD	any string	(none)	The BEFORE-BOLD and AFTER-BOLD arguments are for LINEPRINTER printers only. They specify the character string to turn bolding on and off. If the string contains blank characters, enclose it in single quote marks ('...'). To specify non-printable characters, such as ESC, enclose the decimal value inside angle brackets as follows: BEFORE-BOLD=<27>[r ! Turn on bold AFTER-BOLD=<27>[u ! Turn it off These arguments work in conjunction with the BOLD argument of the PRINT command.

Argument	Choice or Measure	Default	Description
AFTER-BOLD	any string	(none)	See BEFORE-BOLD.

The following table lists the fonts available in SQR for use with the **FONT** argument for HP LaserJet printer types.

Value	Typeface	Style
0	Line printer	Fixed
1	Pica	Fixed
2	Elite	Fixed
3	Courier	Fixed
4	Helvetica	Proportional
5	Times Roman	Proportional
6	Letter Gothic	Fixed
8	Prestige	Fixed
11	Presentations	Fixed
17	Optima	Proportional
18	Garamondi	Proportional
19	Cooper Black	Proportional
20	Coronet Bold	Proportional
21	Broadway	Proportional
22	Bauer Bodini Black Condensed	Proportional
23	Century Schoolbook	Proportional
24	University Roman	Proportional

The font you choose—in orientation, typeface, and point size—must be an internal font, available in a font cartridge, or downloaded to the printer.

For fonts not listed in the Fonts Available for HP LaserJet Printers in SQR table, you must indicate the font style using the FONT-TYPE argument, or the correct typeface cannot be selected by the printer.

The following table lists the fonts available in SQR for use with the FONT argument for PostScript printer types. Those for which bold face types are available are indicated by a "Y" in the *Bold* column.

Value	Typeface	Bold
3	Courier	Y
4	Helvetica	Y
5	Times Roman	Y
6	Avant Garde Book	
8	Palatino Roman	Y
11	Symbol	
12	Zapf Dingbats	
17	Zapf Chancery Medium Italic	
18	Bookman Light	
23	New Century Schoolbook Roman	Y
30	Courier Oblique	Y
31	Helvetica Oblique	Y
32	Times Italic	Y
33	Avant Garde Demi	
34	Avant Garde Book Oblique	
35	Avant Garde Demi Oblique	
36	Palatino Oblique	Y
37	New Century Schoolbook Italic	Y
38	Helvetica Narrow	Y
39	Helvetica Narrow Oblique	Y
40	Bookman Demi	
41	Bookman Light Italic	
42	Bookman Demi Italic	

Other type faces can be added to the POSTSCRIPT.STR file.

The Fonts Available for Windows Printers table lists the fonts available in SQR for Windows when printing on Windows printer drivers (using the -PRINTER:WP command-line flag). When you use the -PRINTER:WP flag, your report is sent directly to the default Windows printer. To specify a non-default Windows printer, enter **-PRINTER:WP:{Printer Name}**. The {Printer Name} is the name assigned to your printer. For example, to send output to a Windows printer named *NewPrinter*, you would use **-PRINTER:WP:NewPrinter**. If your printer name has spaces, enclose the entire argument in quotes.

Fonts are specified in the FONT qualifier of the ALTER-PRINTER command by their number.

Value	Windows Font/Name	Style
3	Courier New	Fixed
300	Courier New	Bold
4	Arial	Proportional
400	Arial	Bold
5	Times New Roman	Proportional
500	Times New Roman	Bold
6	AvantGarde	Proportional
8	Palatino	Proportional
800	Palatino	Bold
11	Symbol	Proportional

Note. Fonts 6, 8, and 800 are not supplied with Windows. You can get these fonts by purchasing the ADOBE Type Manager (ATM). The advantage of using ATM fonts is the compatibility for PostScript printer fonts. The Symbol font uses the SYMBOL_CHARSET instead of the usual ANSI_CHARSET character set. You can add more fonts by editing the appropriate [Fonts] section in the SQR.INI file. See the [Fonts] section in the SQR.INI file for more information.

Parameters

Parameter	Description
<i>printer_name</i>	A unique name to be used for referencing a printer definition and its attributes.

Note. The DECLARE-PRINTER Command Arguments table describes the other arguments of the DECLARE-PRINTER command. The table lists the options, default values, and description of each of the arguments.

Example

```

declare-printer HP-definition      ! Default HP definition

type=HP                          ! for all reports

font=4                           ! Helvetica

symbol-set=12U                   ! PC-850 Multilingual

end-declare

```

```

declare-printer PS-Sales ! PS definition

for-reports=(sales)      ! for the Sales report

type=PS

font=5                  ! Times-Roman
end-declare

```

See Also

The ALTER-PRINTER and **DECLARE-REPORT** commands.

DECLARE-PROCEDURE

Syntax

```

DECLARE-PROCEDURE
[FOR-REPORTS=(report_name1 [, report_namei] ...)]
[BEFORE-REPORT=procedure_name [(arg1 [, argi] ...)] ]
[AFTER-REPORT=procedure_name [(arg1 [, argi] ...)] ]
[BEFORE-PAGE=procedure_name [(arg1 [, argi] ...)] ]
[AFTER-PAGE=procedure_name [(arg1 [, argi] ...)] ]
END-DECLARE

```

Description

Declares procedures that are triggered when a specified event occurs.

The DECLARE-PROCEDURE command can be used to define SQR procedures that are to be invoked before or after a report is printed or before the beginning or end of each page.

Issue the **DECLARE-PROCEDURE** in the SETUP section. For multiple reports, you can use the command as often as required to declare procedures required by all the reports. If you issue multiple DECLARE-PROCEDURE commands, the last one takes precedence. In this way, you can use one command to declare common procedures for **ALL** reports and others to declare unique procedures for individual reports. The referenced procedures can accept arguments.

If no FOR-REPORTS is specified, **ALL** is assumed. Initially, the default for each of the four procedure types is NONE. If a procedure is defined in one **DECLARE-PROCEDURE** for a report, that procedure is used unless NONE is specified.

Use the **USE-PROCEDURE** command to change the procedures to be used at runtime. To turn a procedure off, specify NONE in the **USE-PROCEDURE** statement.

Parameters

Parameter	Description
FOR-REPORTS	Specifies one or more reports that use the given procedures. This argument is required only for a program with multiple reports. If you are writing a program that produces only a single report, you can ignore this argument.
BEFORE-REPORT	Specifies a procedure to be executed at the time of the execution of the first command which causes output to be generated (PRINT). It can be used, for example, to create a report heading.
AFTER-REPORT	Specifies a procedure to be executed just before the report file is closed at the end of the report. It can be used to print totals or other closing summary information. If no report was generated, the procedure does not execute.
BEFORE-PAGE	Specifies a procedure to be executed at the beginning of every page, just before the first output command for the page. It can be used, for example, to set up page totals.
AFTER-PAGE	Specifies a procedure to be executed just before each page is written to the file. It can be used, for example, to display page totals.

You can optionally specify arguments to be passed to any of the procedures. Arguments can be any variable, column, or literal.

Example

```

declare-procedure ! These procedures will

before-report=report_heading      ! be used by all reports

after-report=report_footing
end-declare
declare-procedure ! These procedures will

for-reports=(customer)          ! be used by the customer

before-page=page_setup          ! report

after-page=page_totals
end-declare

```

See Also

The USE-PROCEDURE command.

DECLARE-REPORT

Syntax

```
DECLARE-REPORT report_name
[TOC=toc_name]
[LAYOUT=layout_name]
[PRINTER-TYPE=printer_type]
END-DECLARE
```

Description

Defines reports and their attributes.

Issue the DECLARE-REPORT in the SETUP section.

You can use the **DECLARE-REPORT** command to declare one or more reports to be produced in the application.

You must use this command when developing applications to produce more than one report.

Multiple reports can share the same layout and the same printer declarations or each report can use its own layout or printer definitions if the report has unique characteristics.

When you are printing multiple reports, unless report names are specified using the -F command-line flag, the first report declared is generated with the name of *program.lis*, where *program* is the application name.

Additional reports are generated with names conforming to the rules dictated by the SQR.INI OUTPUT-FILE-MODE setting.

When the **-KEEP** or **-NOLIS** flags are used, the first intermediate print file (SPF file) is generated with a name of *program.spf* and additional reports are generated with names conforming to the rules dictated by the SQR.INI OUTPUT-FILE-MODE setting.

Parameters

Parameter	Description
<i>report_name</i>	Specifies the name of the report.
TOC	Specifies the name of the Table of Contents for this report.
LAYOUT	Specifies the name of the layout for this report. If none is specified, the default layout is used.

Parameter	Description
PRINTER-TYPE	Specifies the type of printer to be used for this report. If none is specified, the default is the LINEPRINTER for this report. If no DECLARE-PRINTER is specified, DEFAULT-LP is used. Valid values for PRINTER-TYPE are HT, HP, PS, LP, HTML, HPLASERJET, POSTSCRIPT, and LINEPRINTER.

Example

```

declare-layout customer_layout
    left-margin
    right-margin
end-declare

declare-layout summary_layout
    orientation=landscape
end-declare

declare-report customer_detail

toc=detailed

layout=customer_layout

printer-type=postscript
end-declare
declare-report customer_summary

layout=summary_layout

printer-type=postscript
end-declare
.
.
.
use-report customer_detail

...print customer detail...
use-report customer_summary

...print customer summary...

```

See Also

The [USE-REPORT](#), [DECLARE-LAYOUT](#), [DECLARE-PRINTER](#), and [DECLARE-TOC](#) commands.

DECLARE-TOC

Syntax

```
DECLARE-TOC toc_name
[FOR-REPORTS=(report_name1 [, report_namei] ...)]
[DOT-LEADER=YES|NO]
[INDENTATION=position_count_num_lit]
[BEFORE-TOC=procedure_name [(arg1 [, argi] ...)] ]
[AFTER-TOC=procedure_name [(arg1 [, argi] ...)] ]
[BEFORE-PAGE=procedure_name [(arg1 [, argi] ...)] ]
[AFTER-PAGE=procedure_name [(arg1 [, argi] ...)] ]
[ENTRY=procedure_name [(argi [, argi] ...)] ]
END-DECLARE
```

Description

Defines the Table of Contents and its attributes.

Use DECLARE-TOC in the **SETUP** section.

You can use DECLARE-TOC command to declare one or more Table of Contents for the application.

A Table of Contents can be shared between reports.

Parameters

Parameter	Description
<i>toc_name</i>	Specifies the name of the Table of Contents.
FOR-REPORTS	Specifies one or more reports that uses this Table of Contents.
DOT-LEADER	Specifies whether a dot leader precedes the page number. The default setting is NO.
INDENTATION	Specifies the number of spaces by which each level is indented. The default setting is 4.
BEFORE-TOC	Specifies a procedure to be executed before generating the Table of Contents. If no Table of Contents is generated, the procedure does not execute.
AFTER-TOC	Specifies a procedure to be executed after generating the Table of Contents. If no Table of Contents is generated, the procedure does not execute.
BEFORE-PAGE	Specifies a procedure to be executed at the start of every page.

Parameter	Description
AFTER-PAGE	Specifies a procedure to be executed at the end of each page.
ENTRY	Specifies a procedure that is executed to process each Table of Contents entry (instead of SQR doing it for you). When this procedure is invoked, the following SQR-reserved variables are populated with data about the TOC entry: #SQR-TOC-LEVEL Contains the level #SQR-TOC-TEXT Contains the text #SQR-TOC-PAGE Contains the page number. These are global variables. If the procedure is local, you must precede it with an underscore (for example, #_sqr-toc-page). These three SQR-reserved variables are only valid within the scope of the ENTRY procedure. They can be referenced outside the scope, but their contents are undefined.

Example

```

begin-setup
  declare-toc common
    for-reports=(all)
    dot-leader=yes
    indentation=2
end-declare
end-setup
.
.
.
toc-entry level=1 text=$Chapter
toc-entry level=2 text=$Heading
.
.

```

See Also

The **BEGIN-FOOTING**, **BEGIN-HEADING**, **DECLARE-REPORT**, and **TOC-ENTRY** commands.

DECLARE-VARIABLE

Syntax

```
DECLARE-VARIABLE
[DEFAULT-NUMERIC={DECIMAL[(prec_lit)]|FLOAT|INTEGER}]
[DECIMAL[(prec_lit)] num_var[(prec_lit)] [num_var
[(prec_lit)]] ...]
[FLOAT num_var[num_var] ...]
[DATE date_var[date_var] ...]
[INTEGER num_var[num_var] ...]
[TEXT string_var[string_var] ...]
END-DECLARE
```

Description

Enables the user to explicitly declare a variable type.

You can set the default numeric type externally, using the `-DNT` command-line flag or the `DEFAULT-NUMERIC` setting in the [Default-Settings] section of the SQR.INI file. However, the setting in the `DECLARE-VARIABLE` command takes precedence over all other settings. If the command has not been used, then the `-DNT` command-line flag takes precedence over the setting in the SQR.INI file.

Besides `FLOAT`, `INTEGER`, or `DECIMAL`, the `DEFAULT-NUMERIC` setting in the SQR.INI file and `-DNT` command-line flag can be set to `V30`. With `V30`, the program acts in the same manner as in pre-version 4.0 releases; that is, all variables are `FLOAT`. Incidentally, `V30` is not a valid setting for the `DEFAULT-NUMERIC` setting in the `DECLARE-VARIABLE` command.

The `DECLARE-VARIABLE` command enables the user to determine the type of variables to use. This command can only appear in the `SETUP` section or as the first statement of a local procedure. The placement of the command affects its scope. When used in the `SETUP` section, it affects all variables in the entire program. Alternately, when it is placed in a local procedure, its effect is limited to the scope of the procedure. If the command is in both places, the local declaration takes precedence over the `SETUP` declaration.

In addition to declaring variables, the command enables the default numeric type to be specified using the `DEFAULT-NUMERIC` setting as `FLOAT`, `INTEGER`, or `DECIMAL`. When dealing with money or where more precision is required, you can use the `DECIMAL` qualifier.

The `DECLARE-VARIABLE` command, the `-DNT` command-line flag, and the `DEFAULT-NUMERIC` setting in the SQR.INI file affects the way numeric literals are typed. If `V30` is specified, then all numeric literals are `FLOAT` (just as in pre-version 4.0 releases); otherwise, the use or lack of a decimal point determines the type of the literal as either `FLOAT` or `INTEGER`, respectively. Finally, not specifying `DECLARE-VARIABLE` command, the `-DNT` command-line flag, and the `DEFAULT-NUMERIC` setting in the SQR.INI file is the same as specifying `V30`.

Note. In SQR for DDO, list variables should not be declared using this construct.

Parameters

<i>Parameter</i>	<i>Description</i>
DEFAULT-NUMERIC	Specifies the default type for numeric variables. Unless explicitly declared otherwise, a numeric variable assumes the variable type. This qualifier overrides any setting from the command-lineflag -DNT or the DEFAULT- NUMERIC entry in the [Default-Settings] section of the SQR.INI file. If -DNT was not specified on the command line and the SQR.INI file entry has no DEFAULT-NUMERIC entry, then the default numeric type is FLOAT.
DECIMAL	Specifies that the numeric variables that follow are decimal variables with a precision specified with prec_lit. The precision can be assigned to the group of variables or to each individual variable. The precision is the total number of digits used to represent the number. This precision can range from 1 to 38. The default value is 16. The range of decimal numbers is from -9.99999999999999999999999999999999 E□4096 to +9.99999999999999999999999999999999 E□4096
FLOAT	Specifies that the numeric variables that follow are used as double precision floating point. The range and precision of these numbers are machine-dependent.
DATE	Specifies that the date variables that follow can contain a date in the range of January1,4713 BC to December 31, 9999 AD.
INTEGER	Specifies that the numeric variables that follow are used as integers with a range of -2147483648 to +2147483647.
TEXT	Specifies that the string variables that follow are text variables.

Example

```
begin-setup
  declare-variable
    default-numeric=float
    decimal    #decimal(10)
    integer    #counter
    date       $date
```

```

    end-declare
end-setup
.
.
let $date = strtodate('Jan 01 1995','Mon DD YYYY')
print $date (1,1)
position (+2,1)

let #counter = 0
while #counter < 10
    let #decimal = sqrt(#counter)
    add 1 to counter
    print #decimal (+1,1) 9.999999999
end-while

do sub1($date, 'day', 10)

do sub2

.
.
begin-procedure sub1(:$dvar, $units, #uval)
declare-variable
    date $dvar
    integer #uval
end-declare
let $dvar = dateadd($dvar, $units, #uval)
print $dvar (+1,1)
position (+2,1)
end-procedure

.
.
begin-procedure sub2 LOCAL
declare-variable
    date $mydate
end-declare
let $mydate = dateadd($_date, 'year', 5)
print $mydate (+1,1)
position (+2,1)
end-procedure

.
.

```

See Also

The -DNT command-line flag, described in the Introduction.

Using the “PSSQR.INI File,” [Default-Settings] Section

#DEFINE

Syntax

```
#DEFINE substitution_variable value
```

Description

Declares a value for a substitution variable within the body of the report (rather than using the **ASK** command).

#DEFINE is useful for specifying constants such as column locations, printer fonts, or any number or string that is used in several locations in the program. When the value of the number or string must be changed, you need only change your **#DEFINE** command. All references to that variable change automatically, which makes modifying programs much simpler.

If the **ASK** command is used to obtain the value of a substitution variable that has already been defined, **ASK** uses the previous value and the user is not prompted. This enables you to predefine some variables and not others. When the report runs, **ASK** requests values for only those variables that have not had a value assigned.

You can use **#DEFINE** commands inside an include file. This is a method of gathering commonly used declarations into one place, and reusing them for more than one report.

The *value* in the **#DEFINE** command can have embedded spaces, and needs no enclosing quotes. The entire string is used as is.

The **#DEFINE** command cannot be broken across program lines.

Parameters

Parameter	Description
<i>substitution_variable</i>	The variable to be used as the substitution variable. The substitution variable is used to substitute any command, argument, or part of a SQL statement at compile time.
<i>value</i>	The value to be substituted.

Example

The following code defines several constants:

```
#define page_width 8.5
#define page_depth 11
#define light LS^10027
#define bold LS^03112
#define col1 1
#define col2 27
#define col3 54
#define order_by state, county, city, co_name
```

The following excerpt from a report uses the preceding definitions:

```

begin-setup

declare-printer contacts

    type=hp

    paper-size=({page_width}, {page_depth})

end-declare
end-setup
begin-heading 5
    print 'Company Contacts'                                (1,1)  center
    print 'Sort: {order_by}'                                (2,1)  center
    print 'Company'                                          (4,{col1})
    print 'Contact'                                          (4,{col2})
    print 'Phone'                                            (4,{col3})
end-heading
begin-procedure main
begin-select
company                (1,{col1})
    print '{bold}'      (0,{col2})      ! Print contact in
boldface.
contact                ()
    print '{light}'     ()              ! Back to lightface.
phone                  (0,{col3})      ! Note:          There must
be enough
    next-listing                               ! space between col2
from customers                               ! and col3 for both
order by {order_by}                               ! font changes and
the
end-select                                ! contact field.
end-procedure

```

See Also

The ASK command.

DISPLAY

Syntax

```

DISPLAY {any_lit|_var|_col}
[[:$]edit_mask|NUMBER|MONEY|DATE] [NOLINE]

```

Description

Displays the specified column, variable, or literal.

The **DISPLAY** command can display data to a terminal. The data is displayed to the current location on the screen. If you want to display more than one field on the same line, use **NOLINE** on each display except the last.

Dates can be contained in a date variable or column, or a string literal, column, or variable. When a date variable or column is displayed without an edit mask, the date appears in the following manner:

- For DATETIME columns and SQR DATE variables, the format specified by the **SQR_DB_DATE_FORMAT** setting is used. If this has not been set, SQR uses the first database-dependent format as listed in the Default Database Formats table.
- For DATE columns, the format specified by the **SQR_DB_DATE_ONLY_FORMAT** setting is used. If this has not been set, the format listed in the DATE Column Formats table is used.
- For TIME columns, the format specified by the **SQR_DB_TIME_ONLY_FORMAT** setting is used. If this has not been set, the format listed in the TIME Column Formats table is used.

When displaying a date in a string literal, column, or variable using **EDIT** or **DATE**, the string uses the format specified by the **SQR_DB_DATE_FORMAT** setting, one of the database-dependent formats as listed in the Default Database Formats table, or the database-independent format **SYMMDD[HH24[MI[SS[NNNNN]]]]**.

If you require more control over the display, use the **SHOW** command.

Parameters

Parameter	Description
<i>any_lit _var _col</i>	The text, number, or date to be displayed.
<i>edit_mask</i>	Causes the field to be edited before being displayed. For additional information regarding edit masks, see the command "PRINT".
NUMBER	Indicates that any_lit _var _col is to be formatted using the NUMBER-EDIT-MASK of the current locale. This option is not legal with date variables.
MONEY	Indicates that any_lit _var _col is to be formatted using the MONEY-EDIT-MASK of the current locale. This option is not legal with date variables.
DATE	Indicates that any_lit _var _col is to be formatted using the DATE-EDIT-MASK of the current locale. This option is not legal with numeric variables. If DATE-EDIT-MASK has not been specified, then the date is displayed using the default format for that database (see the Default Database Formats table).

Parameter	Description
NOLINE	Suppresses the carriage return after the field is displayed.

Example

The following segments illustrate the various features of the DISPLAY command:

```
!
! Display a string using an edit mask
!
display '123456789' xxx-xx-xxxx
```

Produces the following output:

```
123-45-6789
```

```
!
! Display a number using an edit mask
!
display 1234567.89 999,999,999.99
```

Produces the following output:

```
1,234,567.89
```

```
!
! Display a number using the default edit mask (specified in SQR.INI)
!
display 123.78
```

Produces the following output:

```
123.780000
```

```
!
! Display a number using the locale default numeric edit mask
!
alter-locale number-edit-mask = '99,999,999.99'
display 123456.78 number
```

Produces the following output:

```
123,456.78
```

```

!
! Display a number using the locale default money edit mask
!
alter-locale money-edit-mask = '$$, $$$, $$9.99'
display 123456.78 money

```

Produces the following output:

```
$123,456.78
```

```

!
! Display a date column using the locale default date edit mask
!
begin-select
dcol
  from tables
end-select
alter-locale date-edit-mask = 'DD-Mon-YYYY'
display &dcol date

```

Produces the following output:

```
01-Jan-1999
```

```

!
! Display two values on the same line
!
display 'Hello' noline
display ' World'

```

Produces the following output:

```
Hello World
```

```

!
! Display two values on the same line with editing of the values
!
alter-locale money-edit-mask = '$$, $$$, $$9.99'
let #taxes = 123456.78
display 'You owe ' noline
display #taxes money noline
display ' in back taxes.'

```

Produces the following output:

```
You owe $123,456.78 in back taxes.
```

See Also

The **SHOW** command for information about screen control.

The **LET** command for information about copying, editing, or converting fields.

The **EDIT** parameter of the **PRINT** command for a description of the edit masks.

The **ALTER-LOCALE** command for a description of the arguments **NUMBER-EDIT-MASK**, **MONEY-EDIT-MASK**, and **DATE-EDIT-MASK**.

DIVIDE**Syntax**

```
DIVIDE {src_num_lit|_var|_col} INTO dst_num_var
      [ON-ERROR={HIGH|ZERO}] [ROUND=nn]
```

Description

Divides one number into another.

The source field is divided into the destination field and the result is placed in the destination. The source is always first, the destination always second.

When dealing with money-related values (dollars and cents), use decimal variables rather than float variables. Float variables are stored as double precision floating point numbers, and small inaccuracies can appear when dividing many numbers in succession. These inaccuracies can appear due to the way different hardware and software implementations represent floating point numbers.

Parameters

Parameter	Description
<i>src_num_lit _var _col</i>	Divided into the contents of <i>dst_num_var</i> .
<i>dst_num_var</i>	Contains the result after execution.
ON-ERROR	Sets the result to the specified number when a division by zero is attempted. If ON-ERROR is omitted and a division by zero is attempted, SQR halts with an error message.
ROUND	Rounds the result to the specified number of digits to the right of the decimal point. For float variables, this value can be from 0 to 15. For decimal variables, this value can be from 0 to the precision of the variable. For integer variables, this argument is not appropriate.

Example

```
divide 37.5 into #price ! #price / 37.5
```

```

divide &rate into #tot on-error=high
divide #j into #subtot on-error=zero

```

Note. *High* in the preceding example is the "Maximum Value," while zero is the "Lowest Value."

See Also

The ADD command.

The LET command for a discussion of complex arithmetic expressions.

DO

Syntax

```
DO procedure_name[(arg1[, arg]...)]
```

Description

Invokes the specified procedure.

When the procedure ends, processing continues with the command following the DO command. You can use arguments to send values to or receive values from a procedure.

Arguments passed by a DO command to a procedure must match in number:

- Database text columns, string variables, and literals can be passed to procedure string or date arguments.
- Database numeric columns, numeric variables, and numeric literals can be passed to procedure numeric arguments.
- Numeric variables (DECIMAL, INTEGER, FLOAT) can be passed to procedure numeric arguments without regard to the argument type of the procedure. SQR automatically converts the numeric values upon entering and leaving the procedure as required.
- Date variables can be passed to procedure date or string arguments.

When a field in a DO command receives a value back from a procedure (a colon indicates it is a back value—that is, a value that's being returned), it must be a string, numeric, or date variable, depending on the procedure argument; however, a date can be returned to a string variable and vice versa.

When a date is passed to a string, the date is converted to a string according to the following rules:

- For DATETIME columns and SQR DATE variables, the format specified by the SQR_DB_DATE_FORMAT setting is used. If this has not been set, the first database-dependent format as listed in the Default Database Formats table is used.
- For DATE columns, the format specified by the SQR_DB_DATE_ONLY_FORMAT setting is used. If this has not been set, the format listed in the DATE Column Formats table is used.
- For TIME columns the format specified by the SQR_DB_TIME_ONLY_FORMAT setting is used. If this has not been set, the format as listed in the TIME Column Formats table is used.

Parameters

Parameter	Description
<i>procedure_name</i>	Specifies the name of the procedure to be executed.
<i>arg1</i> [, <i>argi</i>]	Specifies the arguments to be passed to the procedure. Arguments can be any type of variable or constant value.

Example

```
do get_names
do add_to_list ($name)
do print_list ('A', #total, &co_name, $name)
```

See Also

The **BEGIN-PROCEDURE** command for information about passing arguments.

#ELSE

Syntax

```
#ELSE
```

Description

Compiles the code following the #ELSE command when a preceding #IF, #IFDEF, or #IFDEF command is FALSE. (#ELSE is a compiler directive that works with the #IF, #IFDEF, and #IFDEF compiler directives.)

See Also

The #IF, #IFDEF, and #IFDEF commands for a description of each compiler directive.

ELSE

Syntax

```
ELSE
```

Description

ELSE is an optional command in an IF command.

See Also

The **IF** command for a description and example.

ENCODE

Syntax

```
ENCODE src_code_string_lit INTO dst_txt_var
```

Description

Assigns a non-display or display character to a string variable.

The ENCODE command can define nondisplay characters or escape sequences sent to an output device. These characters or sequences can perform complex output device manipulations. The **ENCODE** command also displays characters not in the keyboard. If your keyboard does not have the Euro symbol, use the Encode feature to create a string variable for it.

The encode characters can be included in a report at the appropriate location using a PRINT or **PRINT-DIRECT** command.

Only values <001> to <255> can be defined in the ENCODE command.

Parameters

Parameter	Description
<i>src_code_string_lit</i>	Specifies a string of characters to be encoded and placed in <i>dst_txt_var</i> .
<i>dst_txt_var</i>	Contains the result after execution.

Example

```

encode '<27>L11233' into $bold           ! Code sequence to turn
bold on.
print $bold () code-printer=lp
```

See Also

The chr function described in the Miscellaneous Functions table under the **LET** command.

The PRINT and PRINT-DIRECT commands.

END-DECLARE, END-DOCUMENT, END-EVALUATE, END-FOOTING, END-HEADING

Syntax

```
END-DECLARE
END-DOCUMENT
END-EVALUATE
END-FOOTING
END-HEADING
```

Description

Completes a section or paragraph.

The END-DECLARE command completes a paragraph started with:

```
DECLARE-CHART
DECLARE-IMAGE
DECLARE-LAYOUT
DECLARE-PRINTER
DECLARE-PROCEDURE
DECLARE-REPORT
DECLARE-VARIABLE
```

Other END-*section* commands complete the corresponding **BEGIN-** *section* command:

```
BEGIN-DOCUMENT
EVALUATE
BEGIN-FOOTING
BEGIN-HEADING
```

Each command must begin on its own line.

Example

```
begin-footing 2
  print 'Company Confidential' (1) center
end-footing
```

See Also

The DECLARE-*paragraph* command.

The **BEGIN**-*section* command.

#END-IF, #ENDIF

Syntax

```
#END-IF
```

Description

Ends an #IF, #IFDEF, or #IFNDEF command. (**#END-IF** is a compiler directive.)

#ENDIF (without the dash) is a synonym for **#END-IF**.

Example

```
#ifdef debuga
  show 'DebugA: #j = ' #j edit 9999.99
  show 'Cust_num = ' &cust_num
#end-if
```

See Also

The #IF, #IFDEF, and #IFNDEF commands for a description of each compiler directive.

END-IF

Syntax

```
END-IF
```

Ends an **IF** command.

See Also

The IF command for a description and example.

END-PROCEDURE, END-PROGRAM, END-SELECT, END-SETUP, END-SQL, END-WHILE, END-EXECUTE

Syntax

```
END-PROCEDURE
END-PROGRAM
END-SELECT
END-SETUP
END-SQL
```


END-WHILE

Description

Completes the corresponding section or paragraph.

Each **END**-*section* command completes the corresponding BEGIN- *section* command:

```
BEGIN-PROCEDURE
BEGIN-PROGRAM
  BEGIN-SELECT
BEGIN-SETUP
BEGIN-SQL
WHILE
```

Each command must begin on its own line.

Example

```
begin-program
  do main
end-program
```

See Also

The BEGIN-*section* command.

The **WHILE** command.

EVALUATE

Syntax

```
EVALUATE {any_lit|_var|_col}
```

This command is equivalent to case/switch in C or Java. The general format of an EVALUATE command is the following:

```
EVALUATE {any_lit|_var|_col}
WHEN comparison_operator {any_lit|_var|_col}
  SQR_Commands...
  [BREAK]
[WHEN comparison_operator {any_lit|_var|_col}
  SQR_Commands...
  [BREAK] ]
[WHEN-OTHER
  SQR_Commands...
  [BREAK] ]
END-EVALUATE
```

Description

Determines the value of a column, literal, or variable and takes action based on that value.

The **EVALUATE** command is useful for branching to different commands depending on the value of a specified variable or column.

EVALUATE commands can be nested.

Evaluating a date variable or column with a string results in a date comparison (chronological, not a byte by byte comparison as is done for strings). The string must be in the proper format as follows:

- For DATETIME columns and SQR DATE variables the format specified by the SQR_DB_DATE_FORMAT setting, SQR uses one of the database-dependent formats (see the Default Database Formats table), or the database-independent format 'YYYYMMDD[HH24[MI[SS[NNNNN]]]]'.
- For DATE columns, SQR uses the format specified by the SQR_DB_DATE_ONLY_FORMAT setting, or the format listed in the DATE Column Formats table.
- For TIME columns, SQR uses the format specified by the SQR_DB_TIME_ONLY_FORMAT setting, or the format as listed in the TIME Column Formats table.

Parameters

Parameter	Description
<i>any_lit/_var/_col</i>	Specifies a text or numeric column; a text, numeric, or date variable; or a text or numeric literal to be used in the evaluation. In short, an evaluation argument.
<i>comparison_operator</i>	Any valid comparison operator. See "comparison operators" in the Operators table.
WHEN	Specifies the evaluation expression. The evaluation argument is compared with the argument, beginning from the first WHEN. If the expression is TRUE, SQR processes the commands after the WHEN. If the expression is FALSE, SQR processes the next WHEN expression. Each WHEN must be on its own line. If more than one WHEN expression appears directly before a set of commands, any one of them, if TRUE, causes the commands to execute.
BREAK	Causes an immediate exit of the EVALUATE command. Use BREAK at the end of a set of commands.

Parameter	Description
WHEN-OTHER	Signifies the start of default commands to be processed if all other WHEN expressions are FALSE. WHEN-OTHER must appear after all other WHEN expressions.

Example

```

evaluate &code
  when = 'A'
    move 1 to #j
    break
  when = 'B'
  when = 'C'
    move 2 to #j ! Will happen if &code is B or C.
    break
  when > 'D'
    move 3 to #j ! Move 3 to #j and continue checking.
  when > 'H'
    add 1 to #j ! Add 1 to #j and continue checking.
  when > 'W'
    add 2 to #j
    break
  when-other
    if isnull (&code)
      do null_code
    else
      move 0 to #j ! Unknown code.
    end-if
    break
end-evaluate

```

See Also

The commands "IF" and "LET" for comparison operators.

EXECUTE (SYBASE and MS-SQL Server)
Syntax

```

EXECUTE [-XC] [ON-ERROR=procedure[(arg1[,argi]...)]]
[DO=procedure[(arg1[,argi]...)]]
[@#status_var=]stored_procedure_name
[@param=]{any_col|_var|_lit}[,...]
[INTO any_coldata_type[(length_int_lit)]
[,...]] [WITH RECOMPILE]

```

The syntax of this command generally follows that of the SYBASE Transact-SQL EXECUTE command, with the exception of optional arguments and the **INTO** argument.

Description

Executes a stored procedure in Sybase/MS database.

If the stored procedure specified in *stored_procedure_name* contains a SELECT query, the EXECUTE command must specify an **INTO** argument to process the values from the query. If no INTO argument is specified, then the values from the query are ignored.

EXECUTE retrieves just the first row when the following instances are true:

- The DO procedure is not specified.
- The stored procedure, *stored_procedure_name* selects one or more rows.
- An **INTO** argument is specified.

This is useful for queries returning a single row.

Parameters

Parameter	Description
-XC (SYBASE only)	Specifies that the EXECUTE command shares the same connection as the DO=procedure it can invoke. This argument is required to share Sybase temporary tables.
ON-ERROR	Declares an SQR procedure to execute if an error occurs. If ON-ERROR is omitted and an error occurs, SQR halts with an error message. For severe errors (for example, passing too few arguments) SQR halts, even if an error procedure is specified. You can specify arguments to be passed to the ON-ERROR procedure. Arguments can be any variable, column, or literal.
DO	Specifies an SQR procedure to execute for each row selected in the query. Processing continues until all rows have been retrieved. You can specify arguments to be passed to the procedure. Arguments can be any variable, column, or literal.
@# <i>status_variable</i>	Returns the procedure status in the specified numeric variable. The status is returned only after selected rows are retrieved.
<i>stored_procedure_name</i>	Names the stored procedure to execute.

Parameter	Description
<i>@param</i>	Names the parameter to pass to the stored procedure. Parameters can be passed with or without names. If used without names, they must be listed in the same sequence as defined in the stored procedure.
<i>any_lit _var _col</i>	Specifies the value passed to the stored procedure. It can be a string, numeric, or date variable, a previously selected column, a numeric literal, or a string literal.
OUT[PUT]	Indicates that the parameter receives a value from the stored procedure. The parameter must be a string, numeric, or date SQR variable. Output parameters receive their values only after rows selected have been retrieved. If you specify multiple output parameters, they must be in the same sequence as defined in the stored procedure.
INTO	Indicates where to store rows that are retrieved from the stored procedure's SELECT statement. The INTO argument contains the names of the columns with data types and lengths (if needed). You must specify the columns in the same sequence and match the data type used in the stored procedure's SELECT statement. If the stored procedure contains more than one SELECT query, only the first query is described with the INTO argument. Rows from subsequent queries are ignored. WITH RECOMPILE Causes the query to recompile each time it is executed rather than using the plan stored with the procedure. Normally, this is not required or recommended.

Example

The following example invokes the stored procedure `get_total` with two parameters: a string literal and a string variable. The result from the stored procedure is stored in the variable `#total`.

```
execute get_total 'S. Q. Reporter' $State #Total Output
```

The following example invokes the stored procedure `get_products` with two parameters. The stored procedure selects data into five column variables. The SQR procedure `print_products` is called for each row retrieved. The return status from the stored procedure is placed in the variable `#proc_return_status`.

```
execute do=print_products
```

```

    @#proc_return_status=
    get_products
    @prodcode=&code,
    INTO &prod_code
        &description
        &discount
        &restock
        &expire_date

                                @max=#maximum
                                int,
                                char (45),
                                float,
                                char,
                                datetime

begin-procedure print_products
print &prod_code                (+1,1)
print &description              (+5,45)
print &discount                  (+5) edit 99.99
print &restock                   (+5) match Y 0 5 Yes N 0 5 No
print &expire_date               (+5,) edit 'Month dd, yyyy'
end-procedure

```

EXIT-SELECT

Syntax

```
EXIT-SELECT
```

Description

Exits a SELECT paragraph immediately.

EXIT-SELECT causes SQR to jump to the command immediately following the **END-SELECT** command.

Use **EXIT-SELECT** when you need to end a query before all rows have been retrieved.

Example

```

begin-select
cust_num, co_name, contact, city, state, zip, employees
add &employees to #tot_emps
if #tot_emps >= 5000
    exit-select                ! Have reached required total emps.
end-if
do print_company
from customers order by employees desc
end-select

```

See Also

The **BEGIN-SELECT** command.

EXTRACT

Syntax

```
EXTRACT {dst_txt_var|date_var} FROM
{{src_txt_lit|_var|_col}|{src_date_var|_col}}
{start_num_lit|_var}{length_num_lit|_var}
```

Description

Copies a portion of a string into a string variable.

You must specify the starting location of the string as an offset from the beginning of the string and its length. An offset of zero (0) begins at the left- most character; an offset of 1 begins one character beyond that, and so on.

If the source is a date variable or column, it is converted to a string before the extraction according to the following rules:

- For DATETIME columns and SQR DATE variables SQR specifies the **SQR_DB_DATE_FORMAT** setting. If this has not been set, the first database-dependent format as listed in the Default Database Formats table is used.
- For DATE columns SQR uses the format specified by the **SQR_DB_DATE_ONLY_FORMAT** setting is used. If this has not been set, the format listed in the DATE Column Formats table is used.
- For TIME columns the format specified by the **SQR_DB_TIME_ONLY_FORMAT** setting is used. If this has not been set, the format as listed in the TIME Column Formats table is used.

If the destination is a date variable, the string extracted from the source must be in one of the following formats:

- The format specified by the **SQR_DB_DATE_FORMAT** setting
- One of the database-dependent formats (see the Default Database Formats table)
- The database-independent format 'SYYYMMDD [HH24 [MI [SS [NNNNNN]]]]'.

Parameters

Parameter	Description
<i>dst_txt_var / date_var</i>	Specifies a text or date variable into which the extracted string is placed.
<i>{ src_txt_lit/_var/_col } { src_date_var / _col }</i>	Specifies a text or date variable, column, or literal from which the string is to be extracted.
<i>start_num_lit/_var</i>	Specifies starting location of the string to be extracted.
<i>length_num_lit/_var</i>	Specifies length of the string to be extracted.

Example

```

extract $state from $record 45 2
extract $foo from "SQR Rocks" 0 4 ! $foo='SQR'

code from &phone 0 3
extract $zip_four from &zip 5 4
extract $rec from $tape_block #loc #rec_len

```

Note. PeopleSoft recommends that you do not use the EXTRACT command when processing strings.

See Also

The substr function described in Miscellaneous Functions table under the **LET** command.

The FIND command.

FIND
Syntax

```

FIND {{obj_txt_lit|_var|_col}}|{date_var|_col}} IN
{{src_txt_var|_col}|{date_var|_col}}
{start_int_lit|_var} dst_location_int_var

```

Description

Determines the location of a character sequence within a string.

FIND searches the specified string for a character sequence and, if the string is found, returns its location as an offset from the beginning of the specified string. If the sequence is not found, **FIND** returns -1 in *dst_location_int_var*.

You must specify an offset from which to begin the search and supply a numeric variable for the return of the location.

If the source or search object is a date variable or column, it is converted to a string before the search according to the following rules:

- For DATETIME columns and SQR DATE variables, the format specified by the SQR_DB_DATE_FORMAT setting is used. If this has not been set, the first database-dependent format as listed in the Default Database Formats table is used.
- For DATE columns, the format specified by the SQR_DB_DATE_ONLY_FORMAT setting is used. If this has not been set, the format listed in the DATE Column Formats table is used.

- For TIME columns, the format specified by the SQR_DB_TIME_ONLY_FORMAT setting is used. If this has not been set, the format as listed in the TIME Column Formats table is used.

Parameters

Parameter	Description
<code>{obj_txt_lit/_var/_col} {date_var/_col}</code>	Specifies a text variable, column, or literal that is to be sought in <code>src_txt_var _col</code> .
<code>{src_txt_var/_col} {date_var/_col}</code>	Specifies a text variable or column to be searched.
<code>start_int_lit/_var</code>	Specifies the starting location of the search.
<code>dst_location_int_var</code>	Specifies the returned starting location of the left-most character of the matching text in <code>{src_txt_var _col date_var _col}</code> .

Example

```
find 'aw.2' in &code5 0 #loc
find ',' in &name 0 #comma_loc
if #comma_loc = -1
...comma not found...
```

See Also

The `instr` function described in the Miscellaneous Functions table under the LET command.

The `EXTRACT` command.

GET

Syntax

```
GET dst_any_var...FROM src_array_name(element)
[field[(occurs)]]...
```

Description

Retrieves data from an array and places it into a date, string, or numeric variable.

Parameters

Parameter	Description
<i>dst_any_var</i>	A date, string, or numeric variable (not database columns) can be destination variables. Numeric variables (decimal, float, integer) are copied from number fields. String variables are copied from char, text, or date fields. Date variables are copied from char, text, or date fields. When a date field is copied to a string variable, SQR converts the date to a string in the format specified by the SQR_DB_DATE_FORMAT setting. If this has not been set, SQR uses the first database-dependent format listed in the Default Database Formats table. If the destination is a date variable, the string extracted from the source must be in the format specified by the SQR_DB_DATE_FORMAT setting, or one of the database-dependent formats (see the Default Database Formats table), or the database-independent format 'SYYYYMMDD[HH24[MI[SS[NNNNNN]]]]'.
<i>src_array_name(element)</i>	If the array's field names are listed, SQR takes the values from the fields and occurrences specified. If the array's field names are not listed, the values are taken from consecutively defined fields in the array.
<i>field[(occurs)]</i>	Array element and field occurrence numbers can be numeric literals (such as 123) or numeric variables (such as #j). If no field occurrence is stated, occurrence zero is used.

Example

The following example copies *\$name*, *\$start_date*, and *#salary* from the first three fields in the #j'th element of the *emps* array.

```
get $name $start_date #salary from emps(#j)
```

The following example copies *#city_tot* and *#county_tot* from the fields *cities* and *counties* in the #j'th element of the *states* array.

```
get #city_tot #county_tot from states(#j) cities counties
```

The following example copies *\$code* from the #j'th occurrence of the *code* field in the #n'th element of the *codes* array.

```
get $code from codes(#n) code(#j)
```

See Also

The PUT command for information about moving data into an array.

GET-COLOR

Syntax

```
GET-COLOR
[PRINT-TEXT-FOREGROUND=({color_name_var | {rgb}})]
[PRINT-TEXT-BACKGROUND=({color_name_var | {rgb}})]
```

Description

Retrieves the current colors.

The GET-COLOR command is allowed wherever the PRINT command is allowed. If the requested color settings does not map to a defined name, then the name is returned as *RGBredgreenblue*, where each component is a three digit number. For example, *RGB127133033*. You can use this format wherever you use a color name. The color name 'none' is returned if no color is associated with the specified area.

Parameters

Parameter	Description
PRINT-TEXT- FOREGROUND	Defines the color in which the text prints.
PRINT-TEXT-BACKGROUND	Defines the color to print as a background behind the text.
{color_name_var}	A color_name is composed of the alphanumeric characters (A-Z, 0-9), the underscore (_) character, and the hyphen (-) character. The name must start with an alpha (A-Z) character and it is case insensitive. The name 'none' is reserved and cannot be assigned a value. A name in the format (RGBredgreenblue) cannot be assigned a value. The name 'default' is reserved and can be assigned a value. 'Default' is used during execution when a referenced color is not defined in the runtime environment.

Parameter	Description
<i>{rgb}</i>	<i>red_lit _var _col, green_lit _var _col, blue_lit _var _col</i> where each component is a value in the range of 000 to 255. In the BEGIN-SETUP section, only literal values are allowed. The default colors implicitly installed with SQR include: black = (0,0,0) white=(255,255,255) gray=(128,128,128) silver=(192,192,192) red=(255,0,0) green=(0,255,0) blue=(0,0,255) yellow=(255,255,0) purple=(128,0,128) olive=(128,128,0) navy=(0,0,128) aqua=(0,255,255) lime=(0,128,0) maroon=(128,0,0) teal=(0,128,128) fuchsia=(255,0,255)

Example

```

begin-setup
  declare-color-map
    light_blue = (193, 222, 229)
  end-declare
end-setup

begin-program
  alter-color-map name = 'light_blue' value = (193, 233, 230)

  print 'Yellow Submarine' ()
    foreground = ('yellow')
    background = ('light_blue')

  get-color print-text-foreground = ($print-foreground)
  set-color print-text-foreground = ('purple')

```

```

    print 'Barney' (+1,1)
    set-color print-text-foreground = ($print-foreground)
end-program

```

See Also

The DECLARE-COLOR-MAP, ALTER-COLOR_MAP, and SET-COLOR commands in this section.

GOTO

Syntax

```
GOTO label
```

Description

Skips to the specified label.

Labels must end with a colon (:) and can appear anywhere within the same section or paragraph as the GOTO command.

Parameters

Parameter	Description
<i>label</i>	Specifies a label within the same section or paragraph.

Example

```

begin-select
price
  if &price < #old_price
    goto next
  end-if
  print &price (2,13,0) edit 999,999.99
  ...
next:
  add 1 to #count
from products
end-select

```

GRAPHIC BOX, GRAPHIC HORZ-LINE, GRAPHIC VERT-LINE

Syntax

The **GRAPHIC** commands have the following syntax:

```
GRAPHIC ({line_int_lit|_var},{column_int_lit|_var},
{width_int_lit|_var})
```

```
BOX {depth_int_lit|_var}
[rule_width_int_lit|_var[shading_int_lit|_var]]
```

```
GRAPHIC ({line_int_lit|_var},{column_int_lit|_var},
{length_int_lit|_var})
```

```
HORZ-LINE [rule_width_int_lit|_var]
```

```
GRAPHIC ({line_int_lit|_var},{column_int_lit|_var},
{length_int_lit|_var})
```

```
VERT-LINE [rule_width_int_lit|_var]
```

Description

Draws a box or line.

After GRAPHIC commands execute, SQR changes the current print location to the starting location of the graphic. (This is different than the way the **PRINT** command works.)

The GRAPHIC command has the following variations:

- BOX
- HORZ-LINE
- VERT-LINE

Parameters

The following sections describe the individual **GRAPHIC** commands:

Parameter	Description
BOX	BOX draws a box of any size at any location on the page. Boxes can be drawn with any size rule and can be shaded or left empty.
<i>width</i> and <i>depth</i>	The width is the horizontal size in character columns; depth is the vertical size in lines. The top left corner of the box is drawn at the line and column specified. The bottom right corner is calculated using the width and depth. You can specify relative placement with (+), (-), or numeric variables, as with regular print positions.
<i>rule_width</i>	The default rule width is 2 decipoints (there are 720 decipoints per inch). The top horizontal line is drawn just below the base of the line above the the starting point. The bottom horizontal line is drawn just below the base of the ending line. Therefore, a one-line deep box surrounds a single line.

Parameter	Description
<i>shading</i>	A number between 1 and 100, specifying the percentage of shading to apply. 1 is very light, and 100 is black. If no shading is specified, the box is blank. Specify a rule-width of zero, if a border is not desired.
HORZ-LINE	HORZ-LINE draws a horizontal line from the location specified, for the length specified. Horizontal lines are drawn just below the base.
<i>rule_width</i>	The default rule width is 2 decipoints.
VERT-LINE	VERT-LINE draws a vertical line from the location specified for the length (in lines) specified. Vertical lines are drawn just below the base line of the line position specified to just below the base line of the line reached by the length specified. To draw a vertical line next to a word printed on line 27, position the vertical line to begin on line 26, for a length of 1 line.
<i>rule_width</i>	The default rule width is 2 decipoints.

Example

The following example shows the **GRAPHIC BOX** command:

```

graphic (1,1,66) box 58 20                                ! Draw box
around page
graphic (30,25,10) box 10                                  ! Draw a 10-
characters-wide-by-10- characters-long box
graphic (1,1,66) box 5 0 8                                  ! Draw 5
line shaded box (without                                     ! border)
! border)
graphic (50,8,30) box 1                                     ! Draw box around
1 line
```

The following example shows the **GRAPHIC HORZ-LINE** command:

```

graphic (4,1,66) horz-line 10                               ! Put line
under page heading
graphic (+1,62,12) horz-line                                ! Put line
under final total
```

The following example shows the **GRAPHIC VERT-LINE** command:

```

graphic (1,27,54) vert-line                                  ! Draw
lines between columns
graphic (1,52,54) vert-line
graphic (3,+2,4) vert-line 6                                 ! Red line
the paragraph
```

See Also

The **ALTER-PRINTER** and **DECLARE-PRINTER** commands for information about setting and changing the **FONT**, **FONT-TYPE**, **POINT-SIZE**, and **PITCH**.

#IF

Syntax

```
#IF {txt_lit|num_lit} comparison_operator
{txt_lit|num_lit}
```

Description

Indicates that the commands following are to be compiled when the expression is TRUE. (#IF is a compiler directive.)

SQR has five compiler directives that enable different pieces of SQR code to be compiled, depending on the existence or value of substitution variables (not program variables, such as, string, numeric, or date).

Substitution variables defined automatically for each -DEBUGxxx letter can also be used with the **#IF**, **#IFDEF**, and **#IFDEF** directives. They can turn entire sections of an SQR program on or off from the command line, depending on the -DEBUGxxx flag.

You can nest **#IF**, **#IFDEF**, or **#IFDEF** directives to a maximum of 10 levels.

The **#IF**, **#IFDEF**, or **#IFDEF** directives cannot be broken across program lines.

The following table lists the compiler directives.

Directive	Example	Description
#IF	#IF {option}='A'	Compiles the commands following the #IF directive if the substitution variable option is equal to 'A'. The test is case-insensitive. Only one simple expression is allowed per #IF command.
#ELSE	#ELSE	Compiles the commands following the #ELSE directive when the #IF expression is FALSE.
#ENDIF	#ENDIF	Ends the #IF directive. #ENDIF can also be typed #END-IF (with a hyphen).
#IFDEF	#IFDEF option	Compiles the commands following the #IFDEF directive if the substitution variable option is defined.
#IFDEF	#IFDEF option	Compiles the command following the #IFDEF directive if the substitution variable option is not defined.

Parameters

Parameter	Description
txt_lit / num_lit	Any text or numeric literal.

Parameter	Description
<i>comparison_operator</i>	Any of the comparison operators as follows: = Equal != Not Equal <> Not Equal < Less than > Greater than <= Less than or equal >= Greater than or equal

Example

```

begin-setup
  ask type 'Use Male, Female or Both (M,F,B) '
end-setup

begin-procedure Main
  #if {type} = 'M'
    ...code for M here
  #else
    #if {type} = 'F'
      ...code for F here
    #else
      #if {type} = 'B'
        ...code for B here
      #else
        show 'M, F or B not selected. Report not created.'
        stop
      #endif      ! for B
    #endif      ! for F
  #endif      ! for M

  #ifdef  debug
    show 'DEBUG:  Cust_num = '  &cust_num  edit  099999
  #endif

  #ifndef debugB                                ! DebugB turned on with -DEBUGB on
    do test_procedure                          ! SQR command line.
  #endif

```

See Also

The **#DEBUG** command for information about the -DEBUG command-line flag.

IF

Syntax

```
IF logical_expression
```

IF commands have the following structure:

```
IF logical_expression
  SQR Commands...
[ELSE
  SQR Commands...]
END-IF
```

Description

Executes commands depending on the value of a condition.

The expression is evaluated as a logical TRUE or FALSE. A value or expression that evaluates to nonzero is TRUE.

Each IF command must have a matching END-IF.

IF commands can be nested.

Comparing a date variable or column with a string, results in a date comparison (chronological, not a byte by byte comparison as is done for strings). The string must be in the proper format as follows:

- For DATETIME columns and SQR DATE variables, the format specified by the SQR_DB_DATE_FORMAT setting, one of the database-dependent formats (see the Default Database Formats table), or the database-independent format 'SYYYYMMDD [HH24 [MI [SS [NNNNNN]]]]' is used.
- For DATE columns, the format specified by the SQR_DB_DATE_ONLY_FORMAT setting, or the format listed in the DATE Column Formats table is used.
- For TIME columns, the format specified by the SQR_DB_TIME_ONLY_FORMAT setting, or the format as listed in TIME Column Formats table is used.

Parameters

Parameter	Description
<i>logical_expression</i>	Any valid logical expression. See the LET command for a description of logical expressions.

Example

```
if &price > &old_price and instr(&code, 'M', 1) > 0
  add 1 to #price_count
if #price_count > 50
```

```

        show 'More than 50 prices found.' noline
        input $x 'Continue? (Y/N)'
        if upper($x) = 'N'
            stop
        end-if
    end-if
else
    add 1 to #old_price_count
end-if
if #rows      ! Will be TRUE if #rows is non-zero.
    do print-it
end-if

if $date1 > 'Apr 21 1996 23:59'
    do past_due
end-if

```

See Also

The LET command for a description of logical expressions.

The **EVALUATE** command.

#IFDEF
Syntax

```
#IFDEF substitution_variable
```

Description

Indicates that the following commands are to be compiled when the substitution variable has been declared by an ASK or **#DEFINE** command, or by the -DEBUG flag on the SQR command line. (**#IFDEF** is a compiler directive.)

Parameters

Parameter	Description
<i>substitution_variable</i>	Is the variable to be used as the substitution variable.

See Also

The #IF command for a description of each compiler directive.

#IFDEF

Syntax

```
#IFDEF substitution_variable
```

Description

Indicates that the following commands are to be compiled when the substitution variable has not been declared by an **ASK** or **#DEFINE** command, or by the **-DEBUG** flag on the SQR command line. (**#IFDEF** is a compiler directive.)

Parameters

Parameter	Description
<i>substitution_variable</i>	Is the variable to be used as the substitution variable.

See Also

The **#IF** command for a description of each compiler directive.

#INCLUDE

Syntax

```
#INCLUDE filename_lit
```

Parameters

Parameter	Description
<i>filename_lit</i>	A file name that is valid for the platform on which this application is to be compiled.

Description

Includes an external source file into the SQR report specification.

You may want to keep commonly used routines in a single file and reference or "include" that file in programs that use the routine. For example, you might have a set of **#DEFINE** commands for different printers to control initialization, font changes, and page size declarations. You can reference the appropriate include file depending on which printer you want to use.

INCLUDE files can be nested up to four levels.

Variable substitution scanning takes place before the **#INCLUDE** command is processed. This enables you to substitute all or part of the **INCLUDE** file name at runtime, adding flexibility to controlling which file is included for the run.

Example

```

#include 'gethours.dat'                ! Common procedure.
#include 'XYZheader.dat'              ! Common report heading
for                                  ! XYZ Company.

#include 'printer{num}.dat'           ! Include printer
definitions for                      ! printer {num}, which is passed
                                     ! on the command line:
                                     ! SQR REP1A SAM/JOE 18
                                     ! where 18 is the arbitrary
                                     ! number assigned your printer
                                     ! definition file, 'printer18.dat'.
                                     ! The report would contain the
                                     ! command: ASK num
                                     ! in the SETUP section, preceding
                                     ! this #include statement.

```

INPUT**Syntax**

```

INPUT input_var [MAXLEN=nn] [prompt]
[TYPE={CHAR|TEXT|NUMBER|INTEGER|DATE}]
[STATUS=num_var] [NOPROMPT] [BATCH-MODE]
[FORMAT={txt_lit|_var|_col}]

```

Description

Accepts data entered by the user at a terminal.

Use MAXLEN to prevent the user from entering data that is too long. If an INSERT or UPDATE command references a variable whose length is greater than that defined in the database, the SQL is rejected and SQR halts. If the maximum length is exceeded, the terminal beeps (on some systems, this may cause the screen to flash instead).

If *prompt* is omitted, SQR uses the default prompt, **Enter** [**\$|**#] *var*:. In any case, a colon (:) and two spaces are added to the prompt.

Specifying **TYPE** causes data type checking to occur. If the string entered is not the type specified, the terminal beeps and an error message is displayed. The INPUT command is then re-executed. If **TYPE**=DATE is specified, then *input_var* can be a date or text variable; however, **TYPE**=DATE is optional if *input_var* is a date variable. If a numeric variable is used, it is validated as a numeric variable. The types CHAR, **TEXT**, and DATE are invalid types. The datatypes supported are described in the following table.

Datatype	Description
CHAR, TEXT	Any character. This is the default datatype.

Datatype	Description
NUMBER	A floating point number in the format [+/-]9999.999[E[+/-]99]
INTEGER	An integer in the format [+/-]99999
DATE	A date in one of the following formats:
	MM/DD/YYYY [BC AD] [HH:MI[:SS[.NNNNNN]] [AM PM]]
	MM-DD-YYYY [BC AD] [HH:MI[:SS[.NNNNNN]] [AM PM]]
	MM.DD.YYYY [BC AD] [HH:MI[:SS[.NNNNNN]] [AM PM]]
	SYYYMMDD[HH24[MI[SS[NNNNNN]]]]

Specifying **STATUS** causes the INPUT command to complete regardless of what the user enters. No error message is displayed. A nonzero error code is stored in the indicated numeric variable if the length or datatype entered is incorrect.

The following table lists the values of the INPUT command's status argument.

Status Value	Indicates
0	Successful.
1	Bad type (did not match the datatype of TYPE).
2	Too long (longer than MAXLEN or the input for an INTEGER variable is < -2147483648 or > +2147483647).
3	No arguments remain on the command line. The command was ignored.

By using **NOPROMPT** and STATUS with the SHOW command, you can write a sophisticated data entry routine.

FORMAT can only be used with dates. It can be a date edit mask or the keyword **DATE**. Use the keyword DATE if the date must be in the format as specified with **INPUT-DATE-EDIT-MASK** setting for the current locale. If FORMAT has not been set, use a database-independent format for the data as listed in the Datatypes Supported by the INPUT CommandExamples table.

Parameters

Parameter	Description
<i>input_var</i>	Specifies a text, numeric, or date variable for the input data.
MAXLEN	Specifies the maximum length for the data.

Parameter	Description
<i>prompt</i>	Specifies the prompt displayed to the user.
TYPE	Specifies the datatype required for the input.
STATUS	Specifies a numeric variable for a return status code.
NOPROMPT	Prevents the prompt from being displayed before the INPUT command is processed.
BATCH-MODE	If BATCH-MODE is specified and no more arguments are in the command line, a value of 3 is returned in the STATUS variable and the user is not prompted for input.
FORMAT	Specifies the format for entering a date. The Date Edit Format Codes table lists date edit format codes.

Example

The following example shows several **INPUT** commands:

```
input $state maxlen=2 'Please enter state abbreviation'
input #age 'Enter lower age boundary' type=integer
input $start_date 'Enter starting date for report' type=date
input $date_in format='Mon dd yyyy'
input $date format=date
```

The following example shows another INPUT command:

```
show clear-screen (5,32) reverse 'CUSTOMER SUMMARY' normal
Try_again:
show (12,20) 'Enter Start Date: ' clear-line
input $start-date noprompt status=#istat type=date
if #istat != 0
    show (24,1) 'Please enter date in format DD-MON-YY' beep
    goto try_again
end-if
show (24,1) clear-line                                     ! Clear error
message line.
```

The following example illustrates the use of the **BATCH-MODE** option:

```
begin-program
while (1)
    input $A status=#stat batch-mode
    if #stat = 3
        break
    else
        do procedure ($a)
```

```

end-if
end-while
end-program

```

See Also

The ALTER-LOCALE command in this section.

The INPUT-DATE-EDIT-MASK setting in the "SQR.INI" section.

LAST-PAGE

Syntax

```
LAST-PAGE position [pre_txt_lit [post_txt_lit]]
```

Description

Places the last page number on each page, as in "page *n* of *m*".

The text strings specified in *pre_txt_lit* and *post_txt_lit* are printed immediately before and after the number.

Using LAST-PAGE causes SQR and SQRT to delay printing until the last page has been processed so that the number of the last page is known.

Parameters

Parameter	Description
<i>position</i>	Specifies the position for printing the last page number.
<i>pre_txt_lit</i>	Specifies a text string to be printed before the last page number.
<i>post_txt_lit</i>	Specifies a text string to be printed after the last page number.

Example

```

begin-footing 1
  page-number          (1,37)  'Page '          ! Will
  appear as
  last-page            ( )    ' of ' '. '        ! "Page 12 of 25."
end-footing

```

See Also

The **PAGE-NUMBER**, **BEGIN-HEADING**, and **BEGIN-FOOTING** commands.

LET

Syntax

```
LET dst_var=expression
```

Description

Assigns the value of an expression to a string, numeric, or date variable.

Valid expressions are formed as a combination of operands, operators, and functions. String, numeric, date, and array field operands can be used in an expression and embedded functions. SQR supports a standardized set of mathematical operators and logical comparison operators working within a carefully defined set of precedence rules. SQR also provides a rich set of mathematical, string, date, and file manipulation functions along with a number of special purpose utility functions. All combined, the SQR expression provides a powerful tool that can be tailored to suit any information processing need. The following detail outlines the specific behavior of each expression component: (1) the operand, (2) the operator, and (3) the function.

Parameters

Parameter	Description
<i>dst_var</i>	A string, numeric, or date variable or array field to which the result of the expression is assigned.
<i>expression</i>	The expression to evaluate.

Operands

Operands form the backbone of an SQR expression. Operands do not have to be the same type. You can combine string, numeric and array field operands to form a valid expression. SQR performs a sequence of automatic operand conversions as it evaluates expressions that contain dissimilar operand types. As the expression is evaluated, operands of lower precision are converted to match the operand of higher precision. Consider the following example:

```
let #answer = #float * #decimal / #integer
```

Because the *multiply* and *divide* operators are equal in precedence, the expression is evaluated as $(\#float * \#decimal) / \#integer$. Working from the inside out, the *#float* variable is converted to a decimal type where a multiply is performed yielding the simplified expression, $(\#decimal) / \#integer$. SQR now converts the *#integer* operand to a decimal type before performing the final divide. When finished with the expression evaluation, SQR converts the result to match the type of the *#answer* variable.

Converting operands of lower precision to operands of higher precision preserves the number of significant digits. The number of significant digits is not lost when an integer is converted to float or decimal. In a similar manner, the number of significant digits is preserved when floating point operands are converted to the decimal type. The number of significant digits is only sacrificed when the final result is converted to match the type of the *#answer* variable and this variable is less precise than the highest of the operands being evaluated. In the

example, precision is not lost if the *#answer* is declared as a decimal type. SQR considers integer variables as the lowest in the precision hierarchy, followed by float and then decimal.

Here are a few simple expression examples:

```
let #discount = round (&price * #rate / 100, 2)
let $name = $first_name || ' ' || $last_name
let customer.total (#customer_id) =

    customer.total (#customer_id) + #invoice_total
if not range(upper($code), 'A', 'G')
    ...processing when out of range...
let store.total (#store_id, #qtr) =

    store.total (#store_id, #qtr) + #invoice_total
let $date1 = strtodate ('Apr 10 1996', 'MON DD YYYY')
```

The following sections list operators and functions supported in expressions.

Operators

The following table lists operators in descending order of precedence. Operators listed in the same row within the table have the same precedence (the operators *,/,% are equal in precedence).

Operators of the same precedence are processed in the sequence they appear in the expression, from left to right. Use parentheses to override the normal precedence rules. All numeric types (decimal, float, integer) are supported for all operators.

Operator	Explanation
	Concatenate two strings or dates
+, -	Sign prefix (positive or negative)
^	Exponent
*, /, %	Multiply, divide, remainder: a % b = mod(a,b) for integers
+, -	Plus, minus Note: SQR distinguishes between a sign prefix and arithmetic operation by the context of the expression.
>, <, >=, <=, <>, !=, =	Comparison operators: greater than, less than, greater or equal to, less than or equal to, not equal (!= or <>), equal
not	Logical NOT
and	Logical AND
or, xor	Logical OR, XOR (exclusive OR)

Functions

This section lists numeric, file-related, and miscellaneous functions. The functions are listed in alphabetical order.

Function arguments are enclosed in parentheses and can be nested. Arguments referenced as x, y, or z indicate the first, second, or third argument of a function. Otherwise, functions take a single argument or no arguments. All arguments are evaluated before a function is evaluated.

Not all functions support all numeric types (decimal, float, integer). Certain functions do not support the decimal type directly, but convert input decimal operands to the float type before the function is evaluated. The following table annotates the functions that directly support the decimal type and which ones do not.

Use parentheses to override the normal precedence rules.

The following table lists numeric functions.

Function	Description
abs	Returns the absolute value of <i>num_value</i>. This function returns a value of the same type as <i>num_value</i>. Syntax: <pre><i>dst_var</i> = abs(<i>num_value</i>)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #dabsvar = abs(#dvar)</pre>
acos	Returns the arccosine of <i>num_value</i>. In the range of 0 to π radians. The value of <i>num_value</i> must be between -1 and 1. This function returns a float value. Syntax: <pre><i>dst_var</i> = acos(<i>num_value</i>)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #facosvar = acos(#fvar)</pre>

Function	Description
asin	Returns the arcsine of <i>num_value</i> in the range of $-\pi/2$ to $\pi/2$ radians. The value of <i>num_value</i> must be between -1 and 1. This function returns a float value. Syntax: <pre><i>dst_var</i> = asin(<i>num_value</i>)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #fasinvar = asin(#fvar)</pre>
atan	Returns the arctangent of <i>num_value</i> in the range of $-\pi/2$ to $\pi/2$ radians. The value of <i>num_value</i> must be between -1 and 1. This function returns a float value. Syntax: <pre><i>dst_var</i> = atan(<i>num_value</i>)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #fatanvar = atan(#fvar)</pre>
ceil	Returns a value representing the smallest integer that is greater than or equal to <i>num_value</i>. This function returns a value of the same type as <i>num_value</i>. Syntax: <pre><i>dst_var</i> = ceil(<i>num_value</i>)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #fceilvar = ceil(#fvar)</pre>

Function	Description
cos	Returns the cosine of <i>num_value</i>. This function returns a float value. Syntax: <pre>dst_var = cos(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #fcosvar = cos(#fvar)</pre>
cosh	Returns the hyperbolic cosine of <i>num_value</i>. This function returns a float value. Syntax: <pre>dst_var = cosh(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #fcoshvar = cosh(#fvar)</pre>
deg	Returns a value expressed in degrees of <i>num_value</i> which is expressed in radians. This function returns a float value. Syntax: <pre>dst_var = deg(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #fdegvar = deg(#fvar)</pre>

Function	Description
e10	Returns the value of 10 raised to <i>num_value</i>. This function returns a float value. Syntax: <pre>dst_var = e10(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #fe10var = e10(#fvar)</pre>
exp	Returns the value of e raised to <i>num_value</i>. This function returns a float value. Syntax: <pre>dst_var = exp(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #fexpvar = exp(#fvar)</pre>
floor	Returns a value representing the largest integer that is less than or equal to <i>num_value</i>. This function returns a value of the same type as <i>num_value</i>. Syntax: <pre>dst_var = floor(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #ffloorvar = floor(#fvar)</pre>

Function	Description
log	Returns the natural logarithm of <i>num_value</i>. This function returns a float value. Syntax: <pre>dst_var = log(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #flogvar = log(#fvar)</pre>
log10	Returns the base-10 logarithm of <i>num_value</i>. This function returns a float value. Syntax: <pre>dst_var = log10(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #flog10var = log10(#fvar)</pre>
mod	Returns the fractional remainder, <i>f</i>, of <i>x_value</i>/<i>y_value</i> such that $x\_value = i * y\_value + f$, where <i>i</i> is an integer, <i>f</i> has the same sign as <i>x_value</i>, and the absolute value of <i>f</i> is less than the absolute value of <i>y_value</i>. The arguments are promoted to the type of the greatest precision and the function returns a value of that type. Syntax: <pre>dst_var = mod(x_value, y_value)</pre> <i>x_value</i> = decimal, float, or integer literal, column, variable, or expression. <i>y_value</i> = decimal, float, or integer literal, column, variable, or expression. <i>dst_var</i> = decimal, float, or integer variable. Example: let #fmodvar = mod(#fxvar, #fyvar)

Function	Description
power	Returns the value of <i>x_value</i> raised to the power of <i>y_value</i>. This function returns a float value. Syntax: <pre>dst_var = power(x_value, y_value)</pre> <i>x_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>y_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: let #fpowervar = power(#fxvar, #fyvar)
rad	Returns a value expressed in radians of <i>num_value</i> which is expressed in degrees. This function returns a float value. Syntax: <pre>dst_var = rad(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>place_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #fradvar = rad(#fvar)</pre>
round	Returns a value that is <i>num_value</i> rounded to <i>place_value</i> digits after the decimal separator. This function returns a value of the same type as <i>num_value</i>. Syntax: <pre>dst_var = round(num_value, place_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. <i>place_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #frndvar = round(#fvar, #fplace) (#x, #y)</pre>

Function	Description
sign	Returns a -1, 0, or +1 depending on the sign of <i>num_value</i>. This function returns a float value. Syntax: <pre>dst_var = sign(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #fsignvar = sign(#fvar)</pre>
sin	Returns the sine of <i>num_value</i>. This function returns a float value. Syntax: <pre>dst_var = sin(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #fsinvar = sin(#fvar)</pre>
sinh	Returns the hyperbolic sine of <i>num_value</i>. This function returns a float value. Syntax: <pre>dst_var = sinh(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #fsinhvar = sinh(#fvar)</pre>

Function	Description
sqrt	Returns the square root of <i>num_value</i>. This function returns a float value. Syntax: <pre>dst_var = sqrt(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #fsqrtvar = sqrt(#fvar)</pre>
tan	Returns the tangent of <i>num_value</i>. This function returns a float value. Syntax: <pre>dst_var = tan(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #ftanvar = tan(#fvar)</pre>
tanh	Returns the hyperbolic tangent of <i>num_value</i>. This function returns a float value. Syntax: <pre>dst_var = tanh(num_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #ftanhvar = tanh(#fvar)</pre>

Function	Description
Trunc	Returns a value that is <i>num_value</i> truncated to <i>place_value</i> digits after the decimal separator. This function returns a value of the same type as <i>num_value</i>. Syntax: <pre>dst_var = trunc(num_value, place_value)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. <i>place_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let #ftruncvar = trunc(#fvar, #fplace)</pre>

The transcendental functions `sin`, `cos`, `tan`, `sinh`, `cosh`, and `tanh` take their arguments in radians. The functions `asin`, `acos`, and `atan` return radian values. To convert from radians to degrees or degrees to radians, use the `rad` or `deg` functions as follows:

```
let #x = sin(rad(45))           ! Sine of 45 degrees.
let #y = deg(asin(#x))         ! Convert back to degrees.
```

If arguments or intermediate results passed to a numeric function are invalid for that function, SQR halts with an error message.

For example, passing a negative number to the `sqrt` function causes an error. Use the `cond` function described in the Miscellaneous Functions table to prevent division by zero or other invalid function or operator argument values.

The following table lists file-related functions. These functions return zero (0) if successful; otherwise, they return the system error code.

Function	Description
delete	Deletes the file <i>filename</i>. The function returns either a zero (0) to indicate success or the value returned from the operating system to indicate an error. Syntax: <pre>stat_var = delete(filename)</pre> <i>filename</i> = text literal, column, variable, or expression. <i>stat_var</i> = decimal, float, or integer variable. Example: <pre>let #fstatus = delete(\$filename)</pre>

Function	Description
exists	Determines if the file, <i>filename</i>, exists. The function returns either a zero (0) to indicate success or the value returned from the operating system to indicate an error. Syntax: <pre>stat_var = exists(filename)</pre> <i>filename</i> = text literal, column, variable, or expression. <i>stat_var</i> = decimal, float, or integer variable. Example: <pre>let #fstatus = exists(\$filename)</pre>
Rename	Renames <i>old_filename</i> to <i>new_filename</i>. The function returns either a zero (0) to indicate success or the value returned from the operating system to indicate an error. Syntax: <pre>stat_var = rename(old_filename, new_filename)</pre> <i>old_filename</i> = text literal, column, variable, or expression. <i>new_filename</i> = text literal, column, variable, or expression. <i>stat_var</i> = decimal, float, or integer variable. Example: <pre>let #fstatus = rename(\$old_filename, \$new_filename)</pre>

The following table lists miscellaneous functions. These functions return a string value unless otherwise indicated.

In these functions where a string argument is expected and a date variable, column, or expression is entered, SQR converts the date to a string according to the following rules:

- For DATETIME columns and SQR DATE variables, the format specified by the `SQR_DB_DATE_FORMAT` setting is used. If this has not been set, the first database-dependent format as listed in the Default Database Formats table is used.
- For DATE columns, the format specified by the `SQR_DB_DATE_ONLY_FORMAT` setting is used. If this has not been set, the format listed in the DATE Column Formats table is used.
- For TIME columns, the format specified by the `SQR_DB_TIME_ONLY_FORMAT` setting is used. If this has not been set, the format as listed in the TIME Column Formats table is used.

On the other hand, except where noted in an individual function, if a string variable, column, or expression is entered where a date argument is expected, then the string must be in the format specified by the `SQR_DB_DATE_FORMAT` setting, one of the database-dependent

formats listed in the Default Database Formats table, or the database-independent format 'SYYYMMDD[HH24[MI[SS[NNNNNN]]]]'

Function	Explanation
array	Returns a pointer to the starting address of the specified array field. The value returned from this function can only be used by a user-defined function. See the routine printarray in the file UFUNC.C for complete instructions on how to use this function. Syntax: <pre>array_var = array(array_name, field_name)</pre> <i>array_name</i> = text literal, column, variable, or expression <i>field_name</i> = text literal, column, variable, or expression <i>array_var</i> = text variable Example: <pre>let #fstatus = printarray(array('products', 'name'), 10, 2, 'c')</pre>
ascii	Returns the ASCII value for the first character in <i>str_value</i>. This function returns a float value. Syntax: <pre>ascii_var = ascii(str_value)</pre> <i>str_value</i> = date or text literal, column, variable, or expression <i>ascii_var</i> = decimal, float, or integer variable Example: <pre>let #fascii = ascii(\$filename)</pre>
asciic	Returns the numeric value for the first character (rather than byte) of the specified string. Syntax: <pre>ascii_var = asciic(str_value)</pre> <i>str_value</i> = date or text literal, column, variable, or expression <i>ascii_var</i> = decimal, float, or integer variable Example: <pre>let #fascii = asciic(\$filename)</pre>

Function	Explanation
chr	Returns a string that is composed of a character with the ASCII value of <i>num_value</i>. Syntax: <pre><i>dst_var</i> = chr(<i>num_value</i>)</pre> <i>num_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>dst_var</i> = text variable. Example: <pre>let \$svar = chr(#num)</pre>
cond	Returns <i>y_value</i> if the <i>x_value</i> is nonzero (0) otherwise returns <i>z_value</i>. If y-value is numeric, the <i>z_value</i> must also be numeric; otherwise, date and textual arguments are compatible. If either the <i>y_value</i> or <i>z_value</i> is a date variable, column, or expression, a date is returned. The return value of the function depends on which value is returned. Syntax: <pre><i>dst_var</i> = cond(<i>x_value</i>, <i>y_value</i>, <i>z_value</i>)</pre> <i>x_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. <i>y_value</i> = Any literal, column, variable, or expression <i>z_value</i> = Any literal, column, variable, or expression <i>dst_var</i> = Any variable Example: <pre>let #avg = #total / cond(&rate != 0, &rate, 1)</pre>

Function	Explanation
dateadd	Returns a date after adding (or subtracting) the specified units to the <i>date_value</i>. Syntax: <pre>date_var = dateadd(<i>date_value</i>, <i>units_value</i>, <i>quantity_value</i>)</pre> <i>date_value</i> = date variable or expression <i>units_value</i> = text literal, column, variable, or expression. Valid units are 'year', 'quarter', 'week', 'month', 'day', 'hour', 'minute', and 'second' <i>quantity_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to float. date_var = date variable Example: <pre>let \$date = dateadd(\$startdate, 'day', 7.5)</pre>
datediff	Returns the difference between the specified dates expressed in <i>units_value</i>. The function returns a float value. The result can be negative if the first date is earlier than the second date. Syntax: <pre>dst_var = datediff(<i>date1_value</i>, <i>date2_value</i>, <i>units_value</i>)</pre> <i>date1_value</i> = date variable or expression <i>date2_value</i> = date variable or expression <i>units_value</i> = text literal, column, variable, or expression. Valid units are 'year', 'quarter', 'week', 'month', 'day', 'hour', 'minute', and 'second' <i>dst_var</i> = decimal, float, or integer variable Example: <pre>let #diff = datediff(\$date1, \$date2, 'hour')</pre>
datenow	Returns the current local date and time from the client machine. Syntax: <pre>dst_var = datenow()</pre> <i>dst_var</i> = date variable Example: <pre>let \$date = datenow()</pre>

Function	Explanation
datetostr	Converts the date <i>date_value</i> to a string in the format <i>format_mask</i>. Syntax: <pre>dst_var = datetostr(date_value [, format_mask])</pre> <i>date_value</i> = date variable or expression <i>format_mask</i> = text literal, column, variable, or expression. The keyword DATE can be used to specify the DATE-EDIT-MASK setting from the current locale. If this argument is not specified, then the format specified by the SQR_DB_DATE_FORMAT setting is used. If this has not been set, then the first database-dependent format listed in the Default Database Formats table is used. <i>dst_var</i> = text variable Example: <pre>let \$formdate = datetostr(\$date, 'Day Mon DD, YYYY') let \$localedate = datetostr(\$date, DATE)</pre>
edit	Formats <i>source_value</i> according to <i>edit_mask</i> and returns a string containing the result. Syntax: <pre>dst_var = edit(source_value, edit_mask)</pre> <i>source_value</i> = Any literal, column, variable, or expression <i>edit_mask</i> = text literal, column, variable, or expression <i>dst_var</i> = text variable Example: <pre>let \$phone = edit(&phone, '(xxx) xxx-xxxxx') let \$price = edit(#price, '999.99') let \$today = edit(\$date, 'DD/MM/YYYY')</pre>
getenv	Returns the value of the specified environment variable. If the environment variable does not exist, an empty string is returned. Syntax: <pre>dst_var = getenv(env_value)</pre> <i>env_value</i> = text literal, column, variable, or expression <i>dst_var</i> = text variable Example: <pre>let \$myuser = getenv('USER')</pre>

Function	Explanation
instr	Returns the numeric position of <i>sub_value</i> in <i>source_value</i> or zero (0) if not found. The search begins at offset <i>offset_value</i>. This function returns a float value. Syntax: <pre>dst_var = instr(source_value, sub_value, offset_value)</pre> <i>source_value</i> = date or text literal, column, variable, or expression <i>sub_value</i> = text literal, column, variable, or expression <i>offset_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to integer. <i>dst_var</i> = decimal, float, or integer variable Example: <pre>let #offset = instr(&description, 'auto', 10)</pre>
instrb	Performs the same functionality as the instr function except that the starting point and returned value are expressed in bytes rather than in characters. Syntax: <pre>dst_var = instrb(source_value, sub_value, offset_value)</pre> <i>source_value</i> = date or text literal, column, variable, or expression <i>sub_value</i> = text literal, column, variable, or expression <i>offset_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to integer. <i>dst_var</i> = decimal, float, or integer variable Example: <pre>let #offset = instrb(&description, 'auto', 10)</pre>

Function	Explanation
isblank	Returns a value of one (1) if source_val is an empty string, null string, or composed entirely of whitespace characters; otherwise, returns a value of zero (0). Syntax: <pre>dst_var = isblank(source_value)</pre> source_value = date or text literal, column, variable, or expression dst_var = decimal, float, or integer variable Example: <pre>let #blank = isblank(&description)</pre>
isnull	Returns a value of one (1) if source_val is null; otherwise, returns a value of zero (0). Syntax: <pre>dst_var = isnull(source_value)</pre> source_value = date or text literal, column, variable, or expression dst_var = decimal, float, or integer variable Example: <pre>let #null = isnull(\$date)</pre>
length	Returns the number of characters in source_value. Syntax: <pre>dst_var = length(source_value)</pre> source_value = date or text literal, column, variable, or expression dst_var = decimal, float, or integer variable Example: <pre>let #length = length(&description)</pre> Note. PeopleSoft recommends that you use either the lengthp or lengtht function instead of the length function.

Function	Explanation
lengthb	(Multi-byte versions of SQR only) Has the same functionality as the length function except that the return value is expressed in bytes, rather than in characters. Syntax: <pre><i>dst_var</i> = lengthb(<i>source_value</i>)</pre> <i>source_value</i> = date or text literal, column, variable, or expression <i>dst_var</i> = decimal, float, or integer variable Example: <pre>let #length = lengthb(&description)</pre> Note. PeopleSoft recommends that you use either the lengthp or lengtht function instead of the lengthb function.
lengthp	Returns the length of a given string in print positions. Syntax: <pre><i>dst_var</i> = lengthp(<i>source_value</i>)</pre> <i>source_value</i> = date or text literal, column, variable, or expression <i>dst_var</i> = decimal, float, or integer variable Example: <pre>let #printlen = lengthp(&string)</pre>
lengtht	Returns the length of a given string in bytes when converted (transformed) to a specified encoding. Syntax: <pre><i>dst_var</i> = lengtht(<i>source_value</i>, <i>encoding_value</i>)</pre> <i>source_value</i> = date or text literal, column, variable, or expression <i>encoding_value</i> = text literal, column, variable, or expression. <i>dst_var</i> = decimal, float, or integer variable Example: <pre>let #sjislen = lengtht(\$string, 'Shift-JIS')</pre>

Function	Explanation
lower	Converts the contents of <i>source_value</i> to lowercase and returns the result. Syntax: <pre>dst_var = lower(source_value)</pre> <i>source_value</i> = date or text literal, column, variable, or expression <i>dst_var</i> = text variable Example: <pre>let \$lower = lower(&description)</pre>
lpad	Pads the <i>source_value</i> on the left to a length of <i>length_value</i> using <i>pad_value</i> and returns the result. Syntax: <pre>dst_var = lpad(source_value, length_value, pad_value)</pre> <i>source_value</i> = date or text literal, column, variable, or expression <i>length_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to integer. <i>pad_value</i> = text literal, column, variable, or expression <i>dst_var</i> = text variable Example: <pre>let \$lpad = lpad(\$notice, 25, '.')</pre>
ltrim	Trims characters in <i>source_value</i> from the left until a character is not in <i>set_value</i> and returns the result. Syntax: <pre>dst_var = ltrim(source_value, set_value)</pre> <i>source_value</i> = date or text literal, column, variable, or expression <i>set_value</i> = text literal, column, variable, or expression <i>dst_var</i> = text variable Example: <pre>let \$ltrim = ltrim(&description, '.')</pre>

Function	Explanation
nvl	Returns <i>y_value</i> if the <i>x_value</i> is null otherwise returns <i>x_value</i>. If <i>x_value</i> is numeric, <i>y_value</i> must also be numeric; otherwise, date and textual arguments are compatible. In any case, the <i>x_value</i> determines the type of expression returned. The return value of the function depends on which value is returned. Syntax: <pre>dst_var = nvl(x_value, y_value)</pre> <i>x_value</i> = Any literal, column, variable, or expression <i>y_value</i> = Any literal, column, variable, or expression <i>dst_var</i> = Any variable Example: <pre>let \$city = nvl(&city, '-- not city --')</pre> If <i>x_value</i> is a date and <i>y_value</i> is textual, then <i>y_value</i> is validated according to the following rules: For DATETIME columns and SQR DATE variables, the format specified by the SQR_DB_DATE_FORMAT setting, one of the database-dependent formats (see the Default Database Formats table), or the database-independent format 'SYYYMDD[HH24[MI[SS[NNNNNN]]]]' is used. For DATE columns, the format specified by the SQR_DB_DATE_ONLY_FORMAT setting, or the format listed in the DATE Column Formats table is used. For TIME columns, the format specified by the SQR_DB_TIME_ONLY_FORMAT setting, or the format as listed in the TIME Column Formats table is used.

Function	Explanation
range	Returns a value of one (1) if <i>x_value</i> is between <i>y_value</i> and <i>z_value</i>; otherwise returns a value of zero (0). If the first argument is text or numeric, the other arguments must be of the same type. If the first argument is a date, the remaining arguments can be dates and/or text. It is also possible to perform a date comparison on a mix of date and text arguments, for example, where <i>x_value</i> is a date and <i>y_value</i> and <i>z_value</i> are text arguments. In a comparison of this sort, <i>y_value</i> must represent a date that is earlier than that of <i>z_value</i>. Syntax: <pre>dst_var = range(x_value, y_value, z_value)</pre> <i>x_value</i> = Any literal, column, variable, or expression <i>y_value</i> = Any literal, column, variable, or expression <i>z_value</i> = Any literal, column, variable, or expression <i>dst_var</i> = decimal, float, or integer variable Example: <pre>let #inrange = range(&grade, 'A', 'D') let #inrange = range(\$date, \$startdate, \$enddate) let #inrange = range(\$date, \$startdate, '15- Apr-97') let #inrange = range(#price, #low, #high)</pre> If <i>x_value</i> is a date and <i>y_value</i> and/or <i>z_value</i> is textual, then <i>y_value</i> and/or <i>z_value</i> is validated according to the following rules: For DATETIME columns and SQR DATE variables, the format specified by the SQR_DB_DATE_FORMAT setting, one of the database-dependent formats (see the Default Database Formats table), or the database-independent format 'SYYYMMDD[HH24[MI[SS[NNNNNN]]]]' is used. For DATE columns, the format specified by the SQR_DB_DATE_ONLY_FORMAT setting, or the format listed in the table DATE Column Formats is used. For TIME columns, the format specified by the SQR_DB_TIME_ONLY_FORMAT setting, or the format as listed in table TIME Column Formats is used.

Function	Explanation
replace	Inspects the contents of <i>source_value</i> and replaces all occurrences of <i>from_string</i> with <i>to_string</i> and returns the modified string. Syntax: <pre>dst_var = replace(source_value, from_string, to_string)</pre> <i>source_value</i> = date or text literal, column, variable, or expression <i>from_string</i> = text literal, column, variable, or expression <i>to_string</i> = text literal, column, variable, or expression <i>dst_var</i> = text variable Example: <pre>let \$replaced = replace(\$paragraph, 'good', 'excellent')</pre>
roman	Returns a string that is the character representation of <i>source_value</i> expressed in lowercase roman numerals. Syntax: <pre>dst_var = roman(source_value)</pre> <i>source_value</i> = text literal, column, variable, or expression. <i>dst_var</i> = text variable Example: <pre>let \$roman = roman(#page-count)</pre>
rpad	Pads the <i>source_value</i> on the right to a length of <i>length_value</i> using <i>pad_value</i> and returns the result. Syntax: <pre>dst_var = rpad(source_value, length_value, pad_value)</pre> <i>source_value</i> = date or text literal, column, variable, or expression <i>length_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to integer. <i>pad_value</i> = text literal, column, variable, or expression <i>dst_var</i> = text variable Example: <pre>let \$rpad = rpad(\$notice, 25, '.')</pre>

Function	Explanation
rtrim	Trims characters in <i>source_value</i> from the right until a character is not in <i>set_value</i> and returns the result. Syntax: <pre>dst_var = rtrim(source_value, set_value)</pre> <i>source_value</i> = date, or text literal, column, variable, or expression <i>set_value</i> = text literal, column, variable, or expression <i>dst_var</i> = text variable Example: <pre>let \$rtrim = rtrim(&description, '.')</pre>
strtodate	Converts the string <i>source_value</i> in the format <i>format_mask</i> to a date type. Syntax: <pre>dst_var = strtodate(source_value [, format_mask])</pre> <i>source_value</i> = text literal, column, variable, or expression <i>format_mask</i> = text literal, column, variable, or expression that describes the exact format of the <i>source_value</i>. The keyword DATE can be used to specify the DATE-EDIT-MASK setting from the current locale. If this argument is not specified, then <i>source_value</i> must be in the format specified by the SQR_DB_DATE_FORMAT setting, one of the database-dependent formats (see the Default Database Formats table), or the database-independent format 'YYYYMMDD[HH24[MI[SS[NNNNNN]]]]'. Valid format codes are specified in the Date Edit Format Codes table. <i>dst_var</i> = date variable Example: <pre>let \$date = strtodate(\$str_date, 'Mon DD, YYYY') let \$date = strtodate(\$str_date, DATE)</pre>

Function	Explanation
substr	Extracts the specified portion <i>source_value</i>. The extraction begins at <i>offset_value</i> (origin is 1) for a length of <i>length_value</i> characters. Syntax: <pre>dst_var = substr(source_value, offset_value, length_value)</pre> <i>source_value</i> = date or text literal, column, variable, or expression. <i>Offset_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to integer. <i>length_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to integer. <i>dst_var</i> = text variable. Example: <pre>let \$piece = substr(&record, 10, #len)</pre> Note. PeopleSoft recommends that you use either the substri or substri function instead of the substr function.
substrb	Has the same functionality as the substr function except that the starting point and length are expressed in bytes, rather than in characters. Syntax: <pre>dst_var = substrb(source_value, offset_value, length_value)</pre> <i>source_value</i> = date or text literal, column, variable, or expression. <i>offset_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to integer. <i>length_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to integer. <i>dst_var</i> = text variable. Example: <pre>let \$piece = substrb(&record, 10, #len)</pre> Note. PeopleSoft recommends that you use either the substri or substri function instead of the substrb function.

Function	Explanation
substrp	Returns a substring of a given string starting at a specified print position into the string and of a specified print length. Syntax: <pre><i>dst_var</i> = substrp(<i>source_value</i>, <i>offset_value</i>, <i>length_value</i>)</pre> <i>source_value</i> = date or text literal, column, variable, or expression. <i>offset_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to integer. <i>length_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to integer. <i>dst_var</i> = decimal, float, or integer variable. Example: <pre>let \$sub = substrp(&string, #printpos, #printlen)</pre>
substrt	Returns a substring of a given string starting at a specified byte and byte length in a given encoding. Syntax: <pre><i>dst_var</i> = substrt(<i>source_value</i>, <i>offset_value</i>, <i>length_value</i>, <i>encoding_value</i>)</pre> <i>source_value</i> = date or text literal, column, variable, or expression. <i>Offset_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to integer. <i>length_value</i> = decimal, float, or integer literal, column, variable, or expression. The value is always converted to integer. <i>encoding_value</i> = text literal, column, variable, or expression. <i>dst_var</i> = text variable. Example: <pre>let \$sjisPrep = substrt(&string, 1, 10, 'shift-JIS')</pre>

Function	Explanation
to_char	Converts <i>source_value</i> to a string, using maximum precision. Syntax: <pre><i>dst_var</i> = to_char(<i>source_value</i>)</pre> <i>source_value</i> = decimal, float, or integer literal, column, variable, or expression <i>dst_var</i> = text variable Example: <pre>let \$string = to_char(#number)</pre>
to_multi_byte	Converts the specified string as follows: Any occurrence of a double-byte character that also has a single-byte representation (numerals, punctuation, roman characters, and katakana) is converted. Syntax: <pre><i>dst_var</i> = to_multi_byte (<i>source_value</i>)</pre> <i>source_value</i> = date or text literal, column, variable, or expression Example: <pre>let \$multi = to_multi_byte (&text)</pre>
to_number	Converts <i>source_value</i> to a number. This function returns a float value. Syntax: <pre><i>dst_var</i> = to_number(<i>source_value</i>)</pre> <i>source_value</i> = decimal, float, or integer literal, column, variable, or expression <i>dst_var</i> = decimal, float, or integer variable Example: <pre>let #value = to_number(\$number)</pre>

Function	Explanation
to_single_byte	Converts the specified string as follows: For SJIS, EBCDIK290, and EBCDIK1027, any occurrence of a single-byte character that also has a multi-byte representation (numerals, punctuation, roman characters, and katakana) is converted. This function also converts a sequence of kana characters followed by certain grammatical marks into a single-byte character that combines the two elements. For all other encodings, the string is not modified. Syntax: <pre>dst_var = to_single_byte(source_value)</pre> <i>source_value</i> = date or text literal, column, variable, or expression Example: <pre>let \$single = to_single_byte (&text)</pre>
translate	Inspects the contents of <i>source_value</i> and converts characters that match those in <i>from_set</i> to the corresponding character in <i>to_set</i> and returns the translated string. Syntax: <pre>dst_var = translate(source_value, from_set, to_set)</pre> <i>source_value</i> = date or text literal, column, variable, or expression <i>from_set</i> = text literal, column, variable, or expression <i>to_set</i> = text literal, column, variable, or expression <i>dst_var</i> = text variable Example: <pre>let \$translated = translate(edit(&price, '999,999.99'), ',.', ',.')</pre>
transform	Returns a Unicode string which is a specified transform of a given string. Syntax: <pre>dst_var = transform(source_value, transform_value)</pre> <i>source_value</i> = date or text literal, column, variable or expression <i>transform_value</i> = text literal, column, variable, or expression <i>dst_var</i> = text variable Example: <pre>let \$hiragana = transform(\$string, 'ToHiragana')</pre>

Function	Explanation
unicode	Returns a Unicode string from the string of hexadecimal values provided. Syntax: <pre><i>dst_var</i> = unicode(<i>source_value</i>)</pre> <i>source_value</i> = text literal, column, variable or expression <i>dst_var</i> = text variable Example: <pre>let \$unistr = unicode('u+5e73 u+2294')</pre>
upper	Converts the contents of <i>source_value</i> to uppercase and returns the result. Syntax: <pre><i>dst_var</i> = upper(<i>source_value</i>)</pre> <i>source_value</i> = date or text literal, column, variable, or expression <i>dst_var</i> = text variable Example: let \$upper = upper(&description)
wrapdepth	Returns the number of print lines required by <i>source_value</i>. See the PRINTWRAP command for detailed descriptions of the parameters to this function. This function returns a float value. Syntax: <pre><i>dst_var</i> = wrapdepth(<i>source_value</i>, <i>wrap_width</i>, <i>line_height</i>, <i>on</i>, <i>strip</i>)</pre> <i>source_value</i> = text literal, column, variable, or expression <i>wrap_width</i> = decimal, float, or integer literal, column, variable, or expression <i>line_height</i> = decimal, float, or integer literal, column, variable, or expression <i>on</i> = text literal, column, variable, or expression. <i>strip</i> = text literal, column, variable, or expression <i>dst_var</i> = decimal, float, or integer variable Example: <pre>let #depth = wrapdepth(&description,40,1,'<13>','')</pre>

Writing Custom Functions

In addition to using the preceding built-in functions, you can write your own functions in C, using the supplied source file UFUNC.C.

You can pass any number of arguments to your function and values can be returned by the function or passed back in variables.

After editing and recompiling UFUNC.C, you must relink SQR.

The following is a step-by-step example of how to add a user-defined function to SQR so that it can be invoked using the **LET**, **IF**, or **WHILE** command.

The example adds the C function `random`, which returns a random number. The function accepts a parameter that is used as the seed to start a new sequence of numbers. If the seed is zero, then the same sequence is used.

When adding functions to UFUNC, keep in mind the following considerations:

- String functions require the following arguments:
 - (int) Number of arguments.
 - (`char *`) or (`double *`) Array of argument pointers, to either `char[ ]` or `double`.
 - (`char *`) Address for result string. If unchanged, function returns a NULL string.
 - (int) Maximum length of result string, in bytes.
- Numeric functions require the following arguments:
 - (`int`) Number of arguments.
 - (`char *`) or (`double *`) Array of argument pointers, to either `char[ ]` or `double`.
 - (`double *`) Address for result numeric value. If unchanged, function returns zero.

To add the `random` function to SQR, add the following modifications to the UFUNC.C file that was provided with SQR:

- Add the prototype for the `random` function:
`static void random CC_ARGS((char *, char *));`
- Add the function name to the declaration list. The name of the function called from `sqr` is `random`. The return type is `n` for numeric. The number of arguments passed is "1," and the argument type is `n` for numeric. The function name in UFUNC.C is `random`. The characters "PVR" must be entered before the function name.

Name	Return_type	Number of Arguments	Arg_Types	Function
"max",	'n',	0,	"n",	PVR max,
"max",	'n',	0,	"n",	PVR max,

<i>Name</i>	<i>Return_type</i>	<i>Number of Arguments</i>	<i>Arg_Types</i>	<i>Function</i>
"split",	'n',	0,	"C",	PVR split,
"printarray",	'n',	4,	"cnnc",	PVR printarray,
"random",	'n',	1,	"n",	PVR random,

```
/* Last entry must be NULL -- do not change */
"", '\0', 0, "", 0
};
```

- At the end the of UFUNC.C file, add the following random routine. The routine name must be lowercase; however, in your SQR program it can be referenced in either uppercase or lowercase.

```
static void random CC_ARGL((argc, argv, result))
CC_ARG(int, argc) /* The number arguments passed
                  */
CC_ARG(double *, argv[]) /* The argument list
                          */
CC_LARG(double *, result) /* Where to store result
                          */
{
  if (*argv[0] != 0)
    srand(*argv[0]);
  *result = rand();
  return;
}
```

After these modifications, recompile UFUNC.C and relink SQR. See the programmer's reference manual for details about your particular machine.

Finally, the following is an example of a simple SQR program that uses the random function:

```
begin-program
  do get-random-number
  do process-calculations
end-program

begin-procedure
  let #seed = 44
  let #ran = random(#seed)
end-procedure

begin-procedure process-calculations
  .
  .
  .
```

Example

The following examples show some complex expressions:

```

let #j = ((#a + #b) * #c) ^ 2
if #j > 2 and sqrt(#j) < 20 or #i + 2 > 17.4
while upper(substr(&descrip,1,#j+2)) != 'XXXX'

and not isnull(&price)
let #len = length(&fname || &initial || &lname) + 2
let $s = edit(&price * &rate, '99999.99')
let summary.total(#j) = summary.total(#j) + (&price * &rate)
if summary.total(#j) > 1000000
let store.total (#store_id, #dept)

    = store.total (#store_id, #dept) + #total
let #diff = datediff(datenow(), strtodate('1995','YYYY'),'day')
let $newdate = dateadd(datenow(),'month',50)
let $date1 = datetostr(strtodate(&sale_date), 'Day Month DD, YYYY')
```

SQR analyzes LET, IF, and WHILE expressions when it compiles your code and saves the result in an internal format so that repetitive execution is at maximum speed.

LOAD-LOOKUP

Syntax

In the SETUP section:

```

LOAD-LOOKUP
NAME=lookup_table_name
TABLE=database_table_name
KEY=key_column_name
RETURN_VALUE=return_column_name
[ROWS=initial_row_estimate_int_lit]
[EXTENT=size_to_grow_by_int_lit]
[WHERE=where_clause_txt_lit]
[SORT=sort_mode]
[QUIET]
[SCHEMA=schema_name]
[PROCEDURE=procedure_name]
[COMMAND=command_string]
[GETDATA=getdata_string]
[PARAMETERS=(...)]
[FROM-ROWSET=(m,n,-n,m-,all)]
[FROM-PARAMETER=rowset_name]
```

In the body of the report:


```

LOAD-LOOKUP
NAME=lookup_table_name
TABLE=database_table_name
KEY=key_column_name
RETURN_VALUE=return_column_name
[ROWS=initial_row_estimate_lit|_var|_col]
[EXTENT=size_to_grow_by_lit|_var|_col]
[WHERE=where_clause_txt_lit|_var|_col]
[SORT=sort_mode]
[QUIET]
[SCHEMA=schema_name]
[PROCEDURE=procedure_name]
[COMMAND=command_string]
[GETDATA=getdata_string]
[PARAMETERS=(...)]
[FROM-ROWSET=(m,n,-n,m-,all)]
[FROM-PARAMETER=rowset_name]

```

Description

Loads an internal table with columns from the database. Enables a quick search using the **LOOKUP** command.

Use the LOAD-LOOKUP command in conjunction with one or more **LOOKUP** commands.

LOAD-LOOKUP retrieves two columns from the database, the **KEY** field and the **RETURN_VALUE** field. Rows are ordered by **KEY** and stored in an array.

LOAD-LOOKUP commands specified in the SETUP section are always loaded and cannot reference variables for the **ROWS**, **EXTENT**, and **WHERE** arguments.

When you use the LOOKUP command SQR searches the array (with a "binary" search) to find the **RETURN_VALUE** corresponding to the **KEY** referenced in the lookup.

Usually this type of lookup can be done with a database join, but joins take substantially longer. However, if your report is small and the number of rows to be joined is small, a lookup table cannot be warranted, and in fact can be slower, because the entire table has to be loaded and sorted for each report run.

By default, SQR lets the database sort the data. This works fine if the database and SQR both use the same character set and collating sequence. The **SORT** argument enables you to specify the sorting method if this is not true. Additionally, if the machine that SQR is running on is faster than the machine the database is running on, letting SQR perform the sort could decrease the execution time of the report.

The only limit to the size of a lookup table is the amount of memory your computer has available. You could conceivably load an array with many thousands of rows. The binary search is performed quickly regardless of how many rows are loaded.

Except for the amount of available memory, there is no limit to the number of lookup tables that can be defined.

Parameters

Parameter	Description
NAME	The name of the lookup table. The array name is referenced in the LOOKUP command.
TABLE	The name of the table in the database, where the KEY and RETURN_VALUE columns or expressions are stored.
KEY	The name of the column that is used as the key in the array that is used for looking up the information. Keys can be character, date, or numeric data types. If numeric, SQR permits only integers 12 digits or less for the KEY column. Keys can be any database-supported expression. See the RETURN_VALUE argument, next.
RETURN_VALUE	The name of the column (expression) that is returned for each corresponding key. Note. You can combine several columns into an expression if you need several fields returned for each lookup. You can do this by concatenating columns. This is not supported for DDO. The following example is for ORACLE. See your database manual for the correct syntax. RETURN_VALUE='name "-" country "-" population'
ROWS	The initial size of the lookup table. This argument is optional, and if not specified, a value of 100 is used.
EXTENT	The amount to increase the array when it becomes full. This argument is optional, and if not specified, a value of 25% of the ROWS value is used.
WHERE	A WHERE clause used to select a subset of all the rows in the table. If specified, the selection begins after the word WHERE. The WHERE clause is limited to 255 characters.

Parameter	Description
SORT	The sorting method to be used. The following values are permitted: DC—Database sorts data, case-sensitive sort (not supported for DDO) DI—Database sorts data, case-insensitive sort (not supported for DDO) SC—SQR sorts data, case-sensitive sort SI—SQR sorts data, case-insensitive sort The default is SC or the method specified by the -LL command-line flag. The DI method is applicable only to databases that provide this feature and have been installed in that manner.
QUIET	Suppresses the message Loading lookup array... when the command executes. The warning message stating the number of duplicate keys found is also suppressed.

Example

The following command loads the array *states* with the columns *abbr* and *name* from the database table *stateabbrs* where *country* is "USA."

```
load-lookup

name=states

rows=50

table=stateabbrs

key=abbr

return_value=name

where=country='USA'
```

The preceding array is used in the example for the LOOKUP command to retrieve the full text of a state name from the abbreviation.

The following example uses the **LOOKUP** command to validate data entered by a user using an INPUT command:

```
get_state:
input $state 'Enter state abbreviation'
uppercase $state
lookup states $state $name
if $name = ''                                ! Lookup didn't find a match
show 'No such state.'
```

```
        goto get_state
    end-if
```

Surround any command argument with embedded spaces by single quotes, as shown in the following example:

```
where='country='USA' and region = 'NE''
```

The entire **WHERE** clause is surrounded by quotes. The two single quotes around USA and NE are translated to one single quote in the SQL statement.

The following example uses joins in a LOAD-LOOKUP command by including two tables in **TABLE** and the join in WHERE:

```
load-lookup

name=states

rows=50

sort=sc

table='stateabbrs s, regions r'

key=abbr

return_value=name

where='s.abbr = r.abbr and r.location = 'ne''
```

LOOKUP

Syntax

```
LOOKUP lookup_table_name{key_any_lit|_var|_col}
{ret_txt_var|_date_var}
```

Description

Searches a lookup table (an array) for a key value and returns the corresponding text string.

Speeds up processing for long reports. For example, if you want to print the entire state name rather than the abbreviation, you could use LOAD-LOOKUP and then **LOOKUP** to do this.

Parameters

Parameter	Description
<i>lookup_table_name</i>	Specifies the lookup table. This table must be previously loaded with a LOAD- LOOKUP command.

Parameter	Description
<i>key_any_lit / _var / _col</i>	The key used for the lookup.
<i>ret_txt_var / _date_var</i>	A string variable into which the corresponding value is returned.

Example

The following example works in conjunction with the example for the LOAD- LOOKUP command:

```
lookup states &state_abbr $state_name
```

This example searches the *states* lookup table for a matching *&state_abbr* value; if found, it returns the corresponding state name in *\$state_name*. If not found, a null is placed in *\$state_name*.

See Also

The **LOAD-LOOKUP** command.

LOWERCASE

Syntax

```
LOWERCASE txt_var
```

Description

Converts the contents of a text variable to lowercase.

Parameters

Parameter	Description
<i>txt_var</i>	Specifies a text variable to be converted to lowercase.

Example

```
input $answer 'Type EXIT to stop'
lowercase $answer                                ! Allows user to enter
                                                ! upper or lowercase.

if $answer = 'exit'
...etc...
```

See Also

The *lower* function listed in the Miscellaneous Functions table.

MBTOSBS

Syntax

```
MBTOSBS {txt_var}
```

Description

Converts a double-byte string to its single-byte equivalent.

This command converts the specified string as follows: Any occurrence of a double-byte character that also has a single-byte representation (numerals, punctuation, roman characters, and katakana) is converted.

Parameters

Parameter	Description
<i>txt_var</i>	Specifies the string to be converted.

See Also

The TO_SINGLE_BYTE function of the **LET** command.

MOVE

Syntax

```
MOVE {src_any_lit|_var|_col} TO dst_any_var  
[[:$] format_mask|NUMBER|MONEY|DATE]
```

Description

Moves the source field to the destination field. Optionally, you can reformat the field using the *format_mask* argument. Source and destination fields can be different types, numeric, text, or date. MOVE is also useful for converting from one type to another; however, date and numeric variables are incompatible.

When a date variable or column is moved to a string variable, the date is converted according to the following rules:

- For DATETIME columns and SQR DATE variables, the format specified by the **SQR_DB_DATE_FORMAT** setting is used. If this has not been set, the first database-dependent format as listed the Default Database Formats table is used.
- For DATE columns, the format specified by the **SQR_DB_DATE_ONLY_FORMAT** setting is used. If this has not been set, the format listed in the DATE Column Formats table is used.

- For TIME columns, the format specified by the `SQR_DB_TIME_ONLY_FORMAT` setting is used. If this has not been set, the format as listed in the TIME Column Formats table is used.

Finally, as this example shows, the edit mask can be contained in a string variable.

Parameters

Parameter	Description
<i>src_any_lit _var _col</i>	Specifies any source column, variable, or literal. Note. A date can be stored in a date variable or column, or a string literal, column, or variable. When using a date format_mask or the keyword DATE with the MOVE command, the source, if a string literal, column, or variable, must be in the format specified by the <code>SQR_DB_DATE_FORMAT</code> setting, one of the database-dependent formats as listed in the Default Database Formats table, or the database-independent format 'YYYYMMDD[HH24[MI[SS[NNNNNN]]]]'.
<i>dst_any_var</i>	Specifies a destination variable.
<i>format_mask</i>	Specifies an optional format mask. For additional information regarding edit masks, see the command, "PRINT".
NUMBER	Indicates that <code>src_any_lit _var _col</code> is to be formatted using the NUMBER-EDIT-MASK from the current locale. This option is not legal with date variables. (See the command "ALTER_LOCALE".)
MONEY	Indicates that <code>src_any_lit _var _col</code> is to be formatted using the MONEY-EDIT-MASK from the current locale. This option is not legal with date variables. (See the command, "ALTER_LOCALE".)
DATE	Indicates that <code>src_any_lit _var _col</code> is to be formatted using the DATE-EDIT-MASK from the current locale. This option is not legal with numeric variables. (See the command, ""ALTER_LOCALE".)

Example

The following example illustrates the various features of the MOVE command:

```
!
! Convert a string in place
!
move '123456789' to $ssn
move $ssn to $ssn xxx-xx-xxxx
show '$SSN = ' $ssn
```

Produces the following output:

```

$SSN = 123-45-6789

!
! Convert a number to a string using an edit mask
!
move 1234567.89 to #value
move #value to $value 999,999,999.99
show '$Value = ' $value

```

Produces the following output:

```

$Value = 1,234,567.89

!
! Convert a number to a string using a variable edit mask
!
move 123 to #counter
move '099999' to $mask
move #counter to $counter :$mask
show '$Counter = ' $counter

```

Produces the following output:

```

$Counter = 000123

!
! Convert a number to a string using the default edit mask
!
! SQR, by default, outputs six digits of precision.
! If you require more or less precision, specify an edit mask.
!
move 123.78 to #defvar
move #defvar to $defvar
show '$DefVar = ' $defvar

```

Produces the following output:

```

$DefVar = 123.780000

!
! Convert the number to a string using the locale default
! numeric edit mask
!
alter-locale number-edit-mask = '99,999,999.99'
move 123456.78 to #nvar
move #nvar to $nvar number
show '$NVar = ' $nvar

```


Produces the following output:

```
$NVar =      123,456.78

!
! Convert the money value to a string using the locale default
! money edit mask
!
alter-locale money-edit-mask = '$9,999,999.99'
move 123456.78 to #mvar
move #mvar to $mvar money
show '$MVar = ' $mvar
```

Produces the following output:

```
$MVar = $  123,456.78

!
! Convert the date column to a string using the locale default
! date edit mask
!
begin-select
dcol
  from tables
end-select
alter-locale date-edit-mask = 'Mon-DD-YYYY'
move &dcol to $dvar date
show '$DVar = ' $dvar
```

Produces the following output:

```
$DVar = Jan-01-1999

!
! Reset date to first day of the month
! ($date1 and $date2 have been defined as date variables)
!
let $date1 = datenow()
move $date1 to $date2 'MMYYYY'
show '$Date2 = ' $date2 edit 'MM/DD/YY HH:MI'
```

Produces the following output if the report was run in October of 1995.

```
$Date2 = 10/01/95 00:00

!
```

```

! Convert date to a string
! ($date1 has been defined as a date variable)
!
move $date1 to $str_date 'DD-MON-YYYY'
show '$Str_Date = ' $str_date

```

Produces the following output.

```
$Str_Date = 01-DEC-1995
```

```

!
! Convert string (in partial format of SYYYMMDDHHMISSNNN) to a
! date
!
move '19951129' to $date1
show '$Date1 = ' $date1 edit 'Mon DD YYYY HH:MI'

```

Produces the following output.

```
$Date1 = Nov 29 1995 00:00
```

See Also

The LET command for information about copying, editing, or converting fields.

The **EDIT** parameter of the PRINT command for a description of the edit masks.

The ALTER-LOCALE command for a description of the arguments NUMBER-EDIT-MASK, MONEY-EDIT-MASK, and DATE-EDIT-MASK.

The **PRINT** command regarding the default date-time components as a result of moving an incomplete date to a date variable.

MULTIPLY

Syntax

```
MULTIPLY {src_num_lit/_var/_col} TIMES dst_num_var
[ROUND=nn]
```

Description

Multiplies the first field by the second and places the result into the second field.

When dealing with money-related values (dollars and cents), use decimal variables rather than float variables. Float variables are stored as double precision floating point numbers, and small inaccuracies can appear when multiplying many numbers in succession. These inaccuracies can appear due to the way different hardware and software implementations represent floating point numbers.

Parameters

Parameter	Description
<i>src_num_lit / _var / _col</i>	Specifies a numeric source column, variable, or literal.
<i>dst_num_var</i>	Specifies a destination numeric variable.
ROUND	Rounds the result to the specified number of digits to the right of the decimal point. For float variables, this value can be from 0 to 15. For decimal variables, this value can be from 0 to the precision of the variable. For integer variables, this argument is not appropriate.

Example

```
multiply &quantity times #cost
multiply 1.5 times #result
```

See Also

The **ADD** command for more information.

The **LET** command for a discussion of complex arithmetic expressions.

NEW-PAGE

Syntax

```
NEW-PAGE [erase_from_line_num_lit/_var/_col]
```

Description

Writes the current page and begins a new one.

For line printers, this command can optionally erase the old page starting at a specified line. After this action is performed, the location on the page is unchanged—that is, the value of **#CURRENT-LINE** is the same. The default action is to erase the entire page and reset **#CURRENT-LINE** to its initial value for the page.

In reports where an overflow page is needed, sometimes it is useful to retain information from the first page on succeeding pages.

Each **NEW-PAGE** occurrence adds a form feed character to the output file unless you specify **FORMFEED=NO** in the **DECLARE-LAYOUT** for this program in the **SETUP** section.

Note. A NEW-PAGE automatically occurs if page overflow is detected. Tabular reports do not require explicit **NEW-PAGE** commands; use NEXT-LISTING instead.

Parameters

Parameter	Description
<i>erase_from_line_num_lit / _var / _col</i>	Specifies a numeric column, variable, or literal for line printers.

Example

```
! Write current page, then erase it
! beginning at line 5.
new-page 5
```

NEW-REPORT

Syntax

```
NEW-REPORT {report_filename_txt_lit/_var/_col}
```

Description

Closes the current report output file and opens a new one with the specified file name.

This command is normally used with single reports only. When used with multiple report declarations, this command affects the *current* report only.

The internal page counter is reset to 1 when NEW-REPORT is executed.

Note. SQR does not create a report output file until the first page is completed. It is possible that **NEW-REPORT** will not create a new file, for example, if no data is selected and nothing is printed on the page.

Parameters

Parameter	Description
<i>{report_filename_txt_lit / _var / _col}</i>	Specifies a new file name.

Example

The following example shows the **NEW-REPORT** command:

```
new-report 'rep2a.lis'
new-report $next-file
```

Assign the report file name within an SQR report by issuing the NEW-REPORT command before printing. You might even prompt for the file name to use, as shown in the following example:

```
begin-report
  input $file 'Enter report file name'
  new-report $file
  ...
```

After execution of this command, the reserved variable *\$sqr-report* is updated to reflect the new report name.

See Also

The **DECLARE-REPORT** and **USE-REPORT** commands.

The **-F** command-line flag.

NEXT-COLUMN

Syntax

```
NEXT-COLUMN [AT-END={NEWLINE|NEWPAGE}]
  [GOTO-TOP={num_lit|_var|_col}]
  [ERASE-PAGE={num_lit|_var|_col}]
```

Description

Sets the current position on the page to the next column defined with the COLUMNS command.

Parameters

Parameter	Description
AT-END	Takes effect if the current column is the last one defined when NEXT-COLUMN is invoked.
GOTO-TOP	Causes the current line in the next column to be <i>num_lit _var _col</i> . This argument is useful when printing columns down the page.
ERASE-PAGE	Specifies where to begin erasing the page when an AT-END=NEWPAGE occurs.

Example

The following example prints columns across the page:

```
columns 10 50                                ! Define two
columns
```

```

begin-select
name      (0,1,20)
phone     (0,+3,0) edit (xxx)xxxx-xxxx
  next-column at-end=newline
Print names

! across the page
! within two

from phonelist
columns.
order by name
end-select

```

The following example prints columns down the page:

```

columns 10 50
move 55 to #bottom_line
begin-select
name      (0,1,20)
phone     (0,+3,0) edit (xxx)xxxx-xxxx
  if #current-line >= #bottom_line
    next-column goto-top=1 at-end=newpage
  else
    position (+1,1)
  end-if
from phonelist
order by name
end-select

```

See Also

The **COLUMNS** and **USE-COLUMN** commands.

NEXT-LISTING

Syntax

```

NEXT-LISTING [NO-ADVANCE]
[SKIPLINES={num_lit|_var|_col}]
[NEED={num_lit|_var|_col}]

```

Description

Ends the current set of detail lines and begins another.

NEXT-LISTING is used in tabular reports. This command causes a new vertical offset in the page to begin.

After **NEXT-LISTING** is executed, line 1 is reset one line below the deepest line previously printed in the page body. That is, if you then write **PRINT (1, 5)**, the string is printed on

the next available line starting in column 5. Note that the SQR reserved variable `#current-line` still reflects the actual line number within the page body.

The value of `SKIPLINES` must be a nonnegative integer. If it is less than 0, then 0 is assumed.

The value of `NEED` must be an integer greater than 0. If it is less than or equal to 0, then 1 is assumed.

Parameters

Parameter	Description
NO-ADVANCE	Suppresses any line movement when no printing has occurred since the previous NEXT-LISTING or NEW-PAGE. The default increments the line position even when nothing was printed.
SKIPLINES	Causes the specified number of lines to be skipped before setting up the new offset.
NEED	Specifies the minimum number of lines needed to begin a new listing or set of detail lines. If this number of lines does not exist, a new page is started. You can use NEED to prevent a group of detail lines from being broken across two pages.

Example

```
begin-select
cust_num  (1,1)          edit 099999
              ! Each detail group prints

city  (+3)                                ! starting
on line 1 since

name  (2,10,30)                                !
NEXT-LISTING keeps

address  (+2)                                !
moving line 1 down the

  next-listing  skiplines=1  need=2
                  ! page. NEED=2 keeps 2

from customers order by cust_num
                  ! line detail groups from

end-select
breaking across
! pages.
```

Note. The NEXT-LISTING command automatically issues a Use-Column 1 command if columns are active.

OPEN

Syntax

```
OPEN {filename_lit|_var|_col} AS
{filenum_num_lit|_var|_col}
{FOR-READING|FOR-WRITING|FOR-APPEND}
{RECORD=length_num_lit[:FIXED|:FIXED_NOLF|:VARY]}
[STATUS=num_var]
```

Parameters

Parameter	Description
{filename_lit _var _col}	Specifies the file name. The file name can be literal, variable, or column. This makes it easy to prompt for a file name at runtime.
{filenum_num_lit _var _col}	Specifies a number that identifies the file in the application. All file commands use the file number to reference the file. File numbers can be numeric variables and literals. The number can be any positive integer less than 64,000.
FOR-READING	When a file is opened for reading, SQR procures all data sequentially. SQR does not allow for random access of information.
FOR-WRITING	When a file is opened for writing, a new file is created. If a file of the same name already exists, it can be overwritten (this depends on the operating system).
FOR-APPEND	When a file is opened in append mode, the current file contents are preserved. All data written is placed at the end of the file. SQR creates the file if one does not already exist. For existing files, make sure the attributes used are the same as those used when the file was created. Failure to do this can produce unpredictable results.
RECORD	For the VARY file type, this is the maximum size for a record. For the FIXED file type, this is the size of each record without the line terminator. For the FIXED_NOLF file type, this is the size of each record.
FIXED	This file type assumes that all records contained within the file are the same length. Terminate each record by a line terminator (system dependent). You can use this file type when writing or reading binary data.

Parameter	Description
FIXED_NOLF	This file type specifies that all records contained within the file are the same length with no line terminators. When writing records, SQR pads short records with blank characters to ensure each record is the same length. This file type can be used when writing or reading binary data.
VARY	This file type specifies that the records can be of varying length. Each record is terminated by a line terminator (system-dependent). Only records containing display characters (no binary data) can be used safely. When reading records, any data beyond the maximum length specified is ignored. This is the default file type.
STATUS	Sets the numeric variable to zero if the OPEN succeeds and to -1 if it fails. Without the STATUS argument, a failure on OPEN causes SQR to halt. By using a STATUS variable, you can control what processing should occur when a file cannot be opened.

Description

Opens an operating system file for reading or writing.

After a file is opened, it remains open until explicitly closed by the CLOSE command. A maximum of 256 files can be opened at one time.

Example

```
open 'stocks.dat' as 1 for-reading record=100
open 'log.dat'    as 5 for-writing record=70
open $filename   as #j for-append  record=80:fixed
open $filename   as 2 for-reading record=80:fixed_nolf

open $filename   as 6 for-reading record=132:vary status=#filestat
if #filestat != 0
  ... error processing ...
end-if
```

Expanded OPEN Command Syntax

An encoding directive added to the OPEN command allows differently encoded files to be managed in a single run of SQR. When no ENCODING is specified on the OPEN command, SQR uses the file input or output encoding specified in the INI file unless the file is *UCS-2* encoded and auto-detection of *UCS-2* files is enabled.

The complete syntax of the OPEN command is:

```
OPEN {filename_lit | _var | _col} AS {filenum_num_lit |
```

```
_var | _col}
{ FOR-READING | FOR-WRITING | FOR-APPEND }
{ RECORD = length_num_lit[:FIXED | :FIXED_NOLF |
:VARY] }
[ STATUS = num_var ]
[ ENCODING = { _var | _col | ASCII | ANSI | SJIS | JEUC
| EBCDIC |
EBCDIK290 | EBCDIK1027 | UCS-2 | UTF-8 | others... }
```

The ENCODING directive is only allowed when converting to Unicode internally.

See Also

The **READ**, **WRITE**, and **CLOSE** commands for information about using files.

PAGE-NUMBER

Syntax

```
PAGE-NUMBER position[pre_txt_lit[post_txt_lit]]
```

Description

Places the current page number on the page.

The text specified in *pre_txt_lit* and *post_txt_lit* are printed immediately before and after the number.

Parameters

Parameter	Description
<i>position</i>	Specifies the position of the page number.
<i>pre_txt_lit</i>	Specifies a text string to be printed before the page number.
<i>post_txt_lit</i>	Specifies a text string to be printed after the page number.

Example

```
begin-footing 1
  page-number          (1,37)  'Page '          ! Will
  appear as
  last-page            ( )  ' of '  '.'          ! "Page 12 of 25."
end-footing
```

See Also

The **LAST-PAGE** command.

POSITION

Syntax

```
POSITION position
[@document_marker[COLUMNS{num_lit|_var|_col}
[num_lit|_var|_col]...]]
```

Description

Sets the current position on a page.

Parameters

Parameter	Description
@document_marker	References a location defined in a DOCUMENT paragraph. In this case, the position used is the location of that marker in the text of the document.
COLUMNS	Defines columns beginning at the location of the document marker. The columns defined are relative to the position of the document marker. When COLUMNS is used, the entire command cannot be broken across more than one program line.

Example

```
position (12,5)           ! Set current position to line 12, column 5.
position (+2,25)          ! Set position 2 lines down, at 25th column.
position () @total_location !
Set position to document
print #total () edit 999,999,999
! marker @total_location.
position () @name_loc columns 1 30
print name ()             ! Columns are defined at @name_loc and
next-column               ! 29 characters to the right of @name_loc
print title ()
```

See Also

The **COLUMNS** command for more information.

The examples with the description of a DOCUMENT paragraph *SQR for PeopleSoft Report Developer's Guide*, "Creating Form Letters."

PRINT

Syntax

```
PRINT {any_lit|_var|_col} position[format_command[format_cmd_params]...]...
```

Description

Puts data on the page at a specified position.

See the "Changing Fonts" in the *SQR for PeopleSoft Developer's Guide*.

Parameters

Parameter	Description
<i>{any_lit _var _col}</i>	Specifies the data to be printed. Note. Dates can be contained in a date column or variable or, in a string literal, column, or variable. When using EDIT or DATE with the PRINT command, a date in a string literal, column, or variable must be in an acceptable format. See the description for "EDIT" for further details.
<i>position</i>	Specifies the position where the data is to be printed.
<i>format_command[format_cmd_params]</i>	Specifies optional formatting commands and parameters.

Format Commands

The PRINT command has the following format commands:

BACKGROUND	MATCH
BOLD	MONEY
BOX	NOP
CENTER	NUMBER
CODE-PRINTER	ON-BREAK
DATE	SHADE
EDIT	UNDERLINE
FILL	WRAP
FOREGROUND	

Some of these format commands can be used in combination with others and some are mutually exclusive. The following tables shows which can be used together. An "X" indicates that they can be used together.

Note. SQR for DDO does not support printing of List variables.

	BOXD	BOX	CENTER	CODE- PRINTER	EDIT NUMBER MONEY DATE	FIXX
BOXD		X	X		X	X
BOX	X		X		X	X
CENTER	X	X			X	X
CODE-						
PRINTE R						
EDIT						
NUMBE R						
MONEY						
DATE	X	X	X			
FIXX	X	X	X			
F/B	X		X		X	X
MATCH	X	X	X			
NOP	X	X	X	X	X	X
ON- BREAK	X	X	X		X	X
SHADE	X	X	X		X	X
UNDER-						
XINE	X		X		X	X
WRAP	X	X				X

Table, Part 1

	F / B	MATCH	NOP	ON-BREAK	SHADE	UNDER- XINE	WRAP
BOXD	X	X	X	X	X	X	X
BOX	X	X	X	X	X		X
CENTER	X	X	X	X	X	X	
CODE- PRINTER			X				

	<i>F / B</i>	<i>MATCH</i>	<i>NOP</i>	<i>ON-BREAK</i>	<i>SHADE</i>	<i>UNDER-XINE</i>	<i>WRAP</i>
EDIT NUMBE R MONEY DATE	X		X	X	X	X	
FIXX	X		X	X	X	X	X
F/B		X	X	X	X		X
MATCH	X		X	X	X	X	
NOP	X	X		X	X	X	X
ON- BREAK	X	X	X		X	X	X
SHADE	X	X	X	X		X	X
UNDER- XINE	X	X	X	X	X		X
WRAP	X		X	X	X	X	

Table, Part 2

Note. In the previous tables, F/B stands for FOREGROUND/BACKGROUND.

The following sections describe these format commands.

BOLD

BOLD causes the string or number to print in **bold** type.

For HP LaserJet printers, the appropriate boldface font must be loaded in the printer.

For PostScript printers, the appropriate boldface must be defined in the PostScript startup file, POSTSCRIPT.STR.

See the DECLARE-PRINTER Command Arguments table for information about which font can be bolded.

For line printers, when the **BEFORE-BOLD** and **AFTER-BOLD** arguments on the **DECLARE-PRINTER** command are used, the specified strings are added before and after the data to be bolded. If **BEFORE-BOLD** and **AFTER-BOLD** are not specified, then BOLD has no effect.

For example:

```
print &name (+1, 20) bold
print 'Your account is in arrears' (1,1) bold
```

BOX

BOX draws a one-line deep graphical box around the printed data. This option has no effect for line printers.

For example:

```
print &grand_total (+5, 20) box
print 'Happy Birthday !!' (1,1) box
```

Note. For HP LaserJet printers using proportional fonts, **BOX** and **SHADE** are not able to determine the correct length of the box because it varies with the width of the characters printed. **BOX** and **SHADE** work well with fixed-pitch fonts and with all PostScript fonts.

CENTER

CENTER centers the field on a line. The position qualifier for column is ignored. For example:

```
print 'Quarterly Sales' (1) center
```

CODE-PRINTER

CODE-PRINTER has the following syntax:

```
CODE-PRINTER = printer_type
```

Valid values for *printer_type* are HT, **HP**, PS, **LP**, HTML, **HPLASERJET**, **POSTSCRIPT**, and **LINEPRINTER**.

CODE-PRINTER

Adds nondisplay characters to the program for the purpose of sending a sequence to the printer. **CODE-PRINTER** causes the string to be placed "behind" the page buffer, rather than within it, so alignment of printed data is not affected by the white space consumed by the nondisplay characters. Only strings can be printed using **CODE-PRINTER**.

Because the report might be printed on different types of printers, you should specify for which type this data is to be used. The report is ignored if printed to a different type. If necessary, you can send a different sequence to another type with a second **PRINT** statement.

For example:

```
encode '<27>[5U' into $big_font
encode '<27>[6U' into $normal_font
...
print $big_font (0, +2) code-printer=lp
print &phone () edit '(xxx) xxx-xxxx'
print $normal_font () code-printer=lp
```

In the previous example, the two CODE-PRINTER arguments put the *\$big_font* and *\$normal_font* sequences into the output, without overwriting any data in the page buffer. Sequences printed with the **CODE- PRINTER** argument are positioned using the regular line and column positioning. However, unlike the PRINT command, the current print location after execution is the beginning location where the **CODE-PRINTER** string was placed. Multiple coded strings printed using the same line and column location appear in the output in the same sequence in which they were printed.

DATE

You cannot use DATE with numeric columns or variables. **DATE** indicates that the field is to be formatted using the DATE-EDIT-MASK from the current locale. (See the command, "ALTER_LOCALE".) If this entry is not set, then the date is printed according to the following rules shown in the Date table.

Column Type	Default Mask	If not set
DATETIME	SQR_DB_DATE_FORMAT	See the Default Database Formats table for the format that is used.
DATE	SQR_DB_DATE_ONLY_FORMAT	See the DATE Column Formats table for the format that is used.
TIME	SQR_DB_TIME_ONLY_FORMAT	See the TIME Column Formats table for the format that is used.

EDIT

EDIT has the following syntax:

```
EDIT edit_format
```

EDIT causes each field to be edited before it is printed. The three types of edits are:

- Text edit
- Numeric edit
- Date edit

The following table lists the text edit format characters.

Character	Description
X	Use character in field.
B	Insert blank.
~ (tilde)	Skip character in field.

Character	Description
R[n]	Reverse sequence of string, for languages such as Hebrew. The optional number indicates right justification within length indicated.

Any other character (for example, punctuation) in a text edit mask is treated as a constant and is included in the edited field.

The characters 8, 9, 0, V, and \$ are illegal in a text edit mask because they are used to indicate that the mask is for a numeric edit.

The following table lists the numeric edit format characters.

Character	Description
8	Digit, zero fill to the right of the decimal point, trim leading blanks (left justify the number).
9	Digit, zero fill to the right of the decimal point, space fill to the left.
0	Digit, zero fill to the left.
\$	Dollar sign, optionally floats to the right.
B	Treated as a "9", but if a value is zero, the field is converted to blanks.
C	Entered at the end of the mask, causes the comma and period characters to be transposed when the edit occurs. This is to support monetary values where periods delimit thousands and commas delimit decimals. (Example: 1.234,56).
E	Scientific format. The number of 9s after the decimal point determines the number of significant digits displayed. The "E" can be uppercase or lowercase; the display follows the case of the mask.
V	Implied decimal point.
MI	Entered at the end of the mask, causes a minus to display at the right of the number.
PR	Entered at the end of the mask, causes angle brackets (<>) to display around the number if the number is negative.
PS	Entered at the end of the mask, causes parentheses to display around the number if the number is negative.
PF	Entered at the end of the mask, causes floating parentheses to display around the number if the number is negative.
NA	Entered at the end of the mask, causes "N/A" to display if the numeric column variable is null. The case of N/A follows that of the mask.

Character	Description
NU	Entered at the end of the mask, causes blanks to display if the numeric column variable is null.
.	Decimal point.
,	Comma.

Characters other than those listed in the Numeric Edit Format Characters table are illegal for numeric edit masks and cause errors during processing.

The following table shows sample edit masks and resulting fields.

Mask	Value	Display
999.99	34.568	34.57
9,999,999V9999	123,456.789	123,456789
9	0	0
8,888,888.888	123,456.789	123,456.789
9,999	1234	1,234
9,999	123	123
09999	1234	01234
9999	-123	-123
9999	-1234	****
9999	12345	****
9999mi	-123	123-
9999pr	-123	< 123>
999999ps	-123	(123)
999999pf	-123	(123)
9999na	(null)	n/a
9999nu	(null)	(blank)
\$\$9,999.99c	1234.56	\$1.234,56
\$\$9,999.99	1234.56	\$1,234.56
\$\$9,999.99	12.34	\$ 12.34
\$\$\$,\$\$9.99	12.34	\$12.34
9.999e	123456	1.235e+05
B9,999	0	(blank)
B9,999	12345	12,345
(xxx)bxxx-xxxx	2169910551	(216) 991-0551

Mask	Value	Display
xxx-xx-xxxx	123456789	123-45-6789
~x~xx~xx	ABCDEFGH HIJ	CDFG
r10	ABCDEFGH	GFEDCBA

The following example shows some uses of edit masks:

```

print #total (7,55,0) edit $999,999.99
                                ! $ 12,345.67
print #total (7,55,0) edit $$$9,999.99
                                ! $12,345.67
print #total (7,55,0) edit 999,999.99pr
                                ! < 12,345.67>(if
                                ! neg)
print #comm (7,55,0) edit b99,999.99
                                ! Blank if zero
print &cnum (16,1,0) edit 099999
                                ! 001234
print #cat (5,10,0) edit 9.999E
                                ! 1.235E+04
print #phone (16,60,0) edit (xxx)bxxx-xxxx
                                ! (216) 397-0551
print #total (7,55,0) edit £££9,999.99
                                ! Dollar-Symbol £

```

The following table lists the date edit format codes.

Character	Description
YYY YY Y	Last 3, 2, or 1 digits of the year. On input, for calculating the 4-digit year, the current century and/or decade are used. For example, a '9' using the 'Y' mask would result in 1999 as the year if the current year is in the 1990s.
YYYY SYYY Y	4 digit year, "S" prefixes BC dates with "-".
RR	Last 2 digits of year; for years in other centuries. See the Date Edit Format Code-RR table.
CC or SCC	Century; "S" prefixes BC dates with "-".
BC AD	BC/AD indicator.
Q	Quarter of year (1,2,3,4; JAN-MAR=1).
RM	Roman numeral month (I-XII; JAN=I).
WW	Week of year (1-53) where week 1 starts on the first day of the year and continues to the seventh day of the year.
W	Week of the month (1-5) where week 1 starts on the first day of the month and ends on the seventh.
DDD	Day of year (1-366).

Character	Description
DD	Day of month (1 - 31).
D	Day of week (1-7). Sunday is first day of week.
DAY	Name of day.
DY	Abbreviated name of day.
ER	Japanese Imperial Era. Returns the name of the of the Japanese Imperial Era in the appropriate kanji ('Heisei' is the current era).
EY	Year of the Japanese Imperial Era. Returns the current year within the Japanese Imperial Era. Note: The common Japanese date format is: 'YYYY<nen>MM<gatsu>DD<nichi>' where <nen>, <gatsu>, and <nichi> are the kanji strings for year, month, and day respectively.
J	Julian day; the number of days since Jan 1, 4713 BC. Numbers specified with 'J' must be integers.
AM PM	Meridian indicator.
HH	Assumes 24 hour clock unless meridian indicator specified.
HH12	Hour of day (1-12).
HH24	Hour of day (0-23).
SSSSS	Seconds past midnight (0-86399).
N NN NNN NNNN NNNNN NNNNNN	Fractions of a second. Precise to microseconds; however, for most hardware and databases, this much accuracy will not be attainable.
MONTH	Name of month.
MON	Abbreviated name of month.
MM	Month (01-12; JAN=01).
MI	Minute (0-59).
SS	Second (0-59).
	Used to concatenate different masks.

Last 2 digits of current year	2-digit year is 00 – 49	2-digit year is 50 – 99
00 - 49	The return date is in the current century.	The return date is in the century before the current one.
50 - 99	The return date is in the century after the current one.	The return date is in the current century.

All masks can be used by the `strtodate` function except for CC, SCC, Q, W, and WW.

A backslash forces the next character into the output from the mask. For example, a mask of *"The cu\rre\nt \mo\nth is Month"* results in the output string of *"The current month is January"*. Without the backslashes the output string would be *"The cu95e7t january is January"*.

A vertical bar can be used as a delimiter between format codes; however, in most cases the bar is not necessary. For example, the mask 'YYYY|MM|DD' is the same as 'YYYYMMDD'.

Any other character (for example, punctuation) in a date edit mask is treated as a constant and is included in the edited field. If the edit mask contains spaces, it must be enclosed in single quotes (').

The masks MON, MONTH, DAY, DY, AM, PM, BC, AD, and RM are case-sensitive and follow the case of the mask entered. For example, if the month is January, the mask Mon yields "Jan" and MON yields "JAN".

All other masks are case-insensitive and can be entered in either uppercase or lowercase. In addition, National Language Support is provided for the following masks: MON, MONTH, DAY, DY, AM, PM, BC, and AD.

See the command ALTER-LOCALE in the "SQR Samples" section for additional information.

If the value of the date field being edited is "Mar 14 1996 9:35", the edit masks produce the results in the following table.

Edit Mask	Result
dd/mm/yy	14/03/96
DD-MON-YYYY	14-MAR-1996
'Month dd, YYYY'	March 14, 1996
MONTH-YYYY	MARCH-1996
HH:MI	09:35
'HH:MI PM'	09:35 AM

<i>Edit Mask</i>	<i>Result</i>
YYYYMMDD	19960314
MM.DD.YYYY	03.14.1996
Mon	Mar
DD D DDD	143073

In addition to being used with the **EDIT** argument, edit masks can also be used with the **MOVE**, **CONCAT**, **DISPLAY**, and **SHOW** commands, and with the edit function of the **LET** command. You edit the field using the supplied mask before storing or displaying it.

When a date with missing date and/or time components is displayed or printed, the defaults are as follows:

- The default year is the current year.
- The default month is the current month.
- The default day is one.
- The default time is zero (00:00:00.000000).

For example, assuming today is September 7, 1996, the following assignment would produce an equivalent date-time of September 1, 1996 13:21:00.000000:

```
let $date1 = strtodate('13:21','HH:MI')
```

Edit masks can be changed dynamically by storing them in a string variable and referencing the variable name preceded by a colon (:). For example:

```
move '$999,999.99' to $mask
print #total (5,10) edit :$mask
show #total edit :$mask
```

When a date that is stored in a string literal, column, or variable is printed with an edit mask, it must be in one of the following formats:

- The format specified by the environment variable **SQR_DB_DATE_FORMAT**, or the corresponding setting in the SQR.INI file.
- One of the database-dependent formats, as listed in the Default Database Formats table.
- The database-independent format, 'SYYYYMMDD [HH24 [MI [SS [NNNNNN]]]]'.

When a date column or variable is printed without an edit mask, the date is printed in the format specified by the environment variable **SQR_DB_DATE_FORMAT** or the corresponding setting in the SQR.INI file. If this has not been set, then the date is printed in the primary database format (the first entry) listed in the Default Database Formats table.

This applies to **DISPLAY**, **MOVE**, and **SHOW** commands as well as **PRINT**.

The following table lists default date formats for each database.

Database	Default Database Formats
DB2	YYYY-MM-DD- HH:MI:SS.NNNNNN YYYY-MM-DD
Informix	YYYY-MM-DD HH:MI:SS.NNN MM/DD/YYYY MM-DD-YYYY MM.DD.YYYY
Ingres	DD-MON-YYYY HH:MI:SS MM/DD/YYYY HH:MI:SS MM-DD-YYYY HH:MI:SS
ODBC	MM-DD-YYYY
Oracle	DD-MON-YY
Sybase	MON DD YYYY HH:MIPM MON DD YYYY [HH:MI[:SS[:NNN]]][PM] MON DD YYYY [HH:MI[:SS[:NNN]]][PM] YYYYMMDD [HH:MI[:SS[:NNN]]PM] YYYYMMDD [HH:MI[:SS[:NNN]]PM]
SQLBase	YYYY-MM-DD- HH.MI.SS.NNNNNN YYYY-MM-DD HH.MI.SS.NNNNNN

Database	DATE Column Formats
DB2	YYYY-MM-DD
Informix	MM/DD/YYYY
ODBC	DD-MON-YYYY

Database	TIME Column Formats
DB2	HH24.MI.SS
ODBC	HH24:MI:SS

FILL

FILL fills the page with the specified character or string as indicated by the print position and length.

The following example prints a line of stars and then a line of dashes followed by stars:

```
print  '*'   (1,1,79)   fill           ! Fill line with *'s
print  '-*'   (+1,20,40) fill         ! Fill with '-*' characters.
```

FOREGROUND/BACKGROUND

When you specify a color on the PRINT command its scope is that of the PRINT command. If you do not define the specified color name, then the setting for "default" is used. Use the color name "none" to turn off color for the specified area.

Syntax

```
PRINT {any_lit|_var|_col}
[FOREGROUND = ({color_name_lit|_var|_col} | {rgb})]
[BACKGROUND = ({color_name_lit|_var|_col} | {rgb})]
```

MATCH

MATCH has the following syntax:

```
MATCH match_text { line_num_lit|_var|_col }
{ column_num_lit|_var|_col } print_text ...
```

MATCH compares a field to a list of key values and if a match is found, prints the corresponding string at the specified line and column.

If the *match_text* contains white space, it must be enclosed in single quotes (').

Any number of match text(s) can be tested, but each must have its own line, column, and *print_text*.

If a match is not found, the unmatched field is printed at the position specified in the parentheses.

Line and column positions for each matched string are treated as fixed or relative positions depending on the type of positioning used in the position qualifier for the **PRINT** command.

For example:

```
print  &type_buyer (20,12) match
      A 20 12 Casual
      B 20 22 Impulsive
      C 21 12 Informed
      D 21 22 Choosey
```


To use relative line and fixed column positioning, for example, you could use the following example:

```
print $state (0,25) match
    OH 0 25 Ohio
    MI 0 37 Michigan
    NY 0 25 'New York'
```

The column positions are treated as fixed locations due to the fixed "25" position declared in parentheses.

MONEY

MONEY indicates that the column or variable is to be formatted using the **MONEY-EDIT-MASK** from the current locale. (See the command, "ALTER_LOCALE".) This can only be used with a numeric column or variable.

NOP

NOP suppresses the print command, causing "no operation" to be executed. This argument is useful for temporarily preventing a field from printing.

For example:

```
print &ssn (1,1) nop                                     !
Hide the social security number.
```

NUMBER

NUMBER indicates that the column or variable is to be formatted using the **NUMBER-EDIT-MASK** from the current locale. (See the **ALTER-LOCALE** command.) This argument can only be used with a numeric column or variable.

ON-BREAK

ON-BREAK has the following syntax:

```
ON-BREAK [PRINT={ALWAYS|CHANGE|CHANGE/TOP-PAGE|NEVER}]
[SKIPLINES={num_lit|_var|_col}]
[PROCEDURE=procedure_name[(arg1[ ,argi]...)]
[AFTER=procedure_name[(arg1[ ,argi]...)]
[BEFORE=procedure_name[(arg1[ ,argi]...)]
[SAVE=txt_var]
[LEVEL=nn]
[SET=nn]
```

ON-BREAK causes the specified action in a tabular report when the value of a field changes (a break occurs). The default action prints the field only when its value changes (**PRINT=CHANGE**).

ON-BREAK has the following qualifiers:

- **PRINT**—Specifies when the break field is printed.
 - **ALWAYS** duplicates the break field for each detail group.
 - **CHANGE** prints the value only when it changes. This is the default.
 - **CHANGE/TOP-PAGE** prints the value both when it changes and at the top of each new page.
 - **NEVER** suppresses printing.
- **SKIPLINES**—Specifies how many lines to skip when the value changes.
- **PROCEDURE** procedure—Specifies the procedure to be invoked when the value changes. This qualifier cannot be used with either the **AFTER** or **BEFORE** qualifiers.
- **AFTER** and **BEFORE** procedures—Specify procedures to invoke either after or before the value changes. If no rows are fetched, neither procedure is executed. The **AFTER** and **BEFORE** can be used only within a **SELECT** paragraph.
- The sequence of events is shown here:
 - **SAVE**—Indicates a string variable where the previous value of a break field is stored.
 - **LEVEL**—Specifies the level of the break for reports containing multiple breaks. For example, a report sorted by state, county, and city might have three break levels: state is level 1 (the most major), and city is level 3 (the most minor). When a break occurs, other breaks with equal or higher level numbers are cleared. The level number also affects the sequence in which **AFTER** and **BEFORE** procedures are processed.
 - **SET**—Assigns a number to the set of leveled breaks in reports with more than one set of independent breaks.

Following is the sequence of events for a query containing **ON-BREAK** fields:

1. Any **BEFORE** procedures are processed in ascending **LEVEL** sequence before the first row of the query is retrieved.
2. When a break occurs in the query, the following happens:
 - a. **AFTER** procedures are processed in descending sequence from the highest level to the level of the current break field.
 - b. **SAVE** variables are set with the new value.
 - c. **BEFORE** procedures are processed in ascending sequence from the current level to the highest level break.
 - d. Any breaks with the same or higher level numbers are cleared so they do not break on the next value.

- e. If a **PROCEDURE** has been declared, the procedure is invoked.
 - f. If **SKIPLINES** was specified, the current line position is advanced.
 - g. The value is printed (unless **PRINT=NEVER** was specified).
3. After the query finishes (at **END-SELECT**) any **AFTER** procedures are processed in descending level sequence.

For example:

```
begin-select
state  (+1,1,2)  on-break level=1  after=state-tot  skiplines=2
county (+,2,14)  on-break level=2  after=county-tot  skiplines=1
city   (+,2,14)  on-break level=3  after=city-tot
...
end-select
```

The breaks are processed as follows:

- When city breaks, the **city-tot** procedure is executed.
- When county breaks, first the **city-tot** procedure is executed, then the **county-tot** procedure is executed.
- When state breaks, the **city-tot**, **county-tot**, and **state-tot** procedures are processed in that sequence.

If any **BEFORE** breaks were indicated, they would also be processed automatically, after all of the **AFTER** breaks and in sequence from lower to higher level numbers.

For example:

```
begin-select
state  (+1,1,2)  on-break level=1  before=bef-state after=state- tot
county (+,2,14)  on-break level=2  before=bef-cnty  after=cnty-tot
city   (+,2,14)  on-break level=3  before=bef-city  after=city-tot
...
end-select
```

Now when state breaks, the sequence of procedures executed is as follows:

1. City-tot
2. Cnty-tot
3. State-tot
4. Bef-state

5. Bef-cnty
6. Bef-city

Upon entering the query at BEGIN-SELECT, the three **BEFORE** procedures are executed in sequence:

1. Bef-state
2. Bef-cnty
3. Bef-city

After the last row is retrieved, at END-SELECT, the three **AFTER** procedures are executed in sequence:

1. City-tot
2. Cnty-tot
3. State-tot

The **SAVE** qualifier saves the previous break value in the specified string variable for use in an **AFTER** procedure. You may want to print the previous break field with a summary line:

```
print &state (+1,1) on-break after=state-tot save=$old- state
```

The **SET** qualifier enables you to have sub-reports with leveled breaks. By separating the **ON-BREAKS** into sets, the associated leveled breaks in each set do not interfere with each other.

```
begin-select
state (+1,1,2) on-break set=1 after=state-tot level=1
```

SET=1 associates this leveled break with other breaks having the same set number.

SHADE

Draws a one-line deep, shaded graphical box around the printed data. For line printers this argument has no effect.

```
print 'Company Confidential' (1,1) shade
print &state (+2, 40) shade
```

Note. For HP LaserJet printers using proportional fonts, **BOX** and **SHADE** are not able to determine the correct length of the box because it varies with the width of the characters printed. **BOX** and **SHADE** work well with fixed pitch fonts and with all PostScript fonts.

UNDERLINE

UNDERLINE prints the specified data with underlined characters. For line printers, UNDERLINE causes backspace and underscore characters to be output, which emulates underlining.

For example:

```
print &name (+1, 45) underline
print 'Your account is in arrears' (1,1) underline
```

WRAP

WRAP wraps text at word spaces. Additional text is moved to a new line. WRAP has the following syntax:

```
WRAP {line_length_lit|_var|_col}
{max_lines_lit|_var|_col} [KEEP-TOP]
[STRIP=strip_chars] [ON=break_chars] [R]
[LINE-HEIGHT={line_height_lit|_var|_col}]

line_length_lit|_var|_col
```

Specifies the maximum paragraph width in characters.

Note. After a string wraps, the current position is one character to the right of the last character in the column. When a string ends on the last position of a line, an implicit line feed causes the new current position to be the first character of the following line. In the SETUP section, use the **DECLARE-LAYOUT** command to make the page width one character wider than the right edge of the wrapped text to avoid generating an implicit line feed.

For example:

```
print &comment (48,20,0) wrap 50 3
                                ! Paragraph is 50
                                ! characters wide,
                                ! with a maximum
                                ! depth of 3 lines.

print &note1 (1,20,30) wrap 30 4
print &note2 (1,+2,30) wrap 30 4
print &note3 (1,+2,30) wrap 30 4
```

In this example, the line position is 1 for each of the three wrapped fields: *note1*, *note2*, and *note3*. The current print position after a wrap occurs at the bottom right edge of the wrapped paragraph. To continue printing on the same line, you must use a fixed line number for the next field.

```
max_lines_lit|_var|_col
```

Specifies the maximum paragraph depth in lines. Usually, the line length and maximum lines are indicated with numeric literals. However, **WRAP** can also reference numeric variables or columns. This is useful when you want to change the width or depth of a wrapped paragraph during report processing. The numeric variable can optionally be preceded by a colon (:).

For example:

```
print $comments (1,30) wrap #wrap_width 6
print $message (5,45) wrap #msg_wid #msg_lines
```

KEEP-TOP retains the current line position except if a page break occurs, in which case, line 1 is used as the current line position. The default action is to set the next print position at the bottom of the wrapped data.

In the following example, the column *&resolution* prints on the same line as the first line of the column *&instructions*:

```
print &phone (+1,10) edit '(xxx) xxx-xxxx'
print &instructions (+1,10,30) wrap 6 30 keep-top
print &resolution (0,+3,25)
```

The **STRIP** and **ON** arguments affect which characters are to be converted before wrapping, and which characters force a wrap to occur.

- Characters in the **STRIP** string argument are converted to spaces before the wrap occurs.
- Characters in the **ON** string argument cause a wrap at each **ON** character found. The **ON** character is not printed.

Both arguments accept regular characters and nondisplay characters whose ASCII values are surrounded by angled brackets, *<nn>*.

For example, to print a long data type that contains embedded carriage returns, the setup would be:

```
print &long_field (5,20) wrap 42 30 on=<13>
```

The paragraph wraps at each carriage return, rather than at the usual word boundaries. If the **ON** character is not found within the width specified for the paragraph, the wrap occurs at a word space.

The following example converts the **STRIP** characters to spaces before wrapping on either a line feed *<10>* or a space (the default):

```
print &description (20,10) wrap 50 22 strip=/\^@<13> on=<10>
```

WRAP can also be used to print reversed characters, for support of languages such as Hebrew. An R after the length and *max_lines* arguments causes the field to be reversed before the wrap takes place. In addition, the entire paragraph is right-justified within the length indicated.

```
! Reverse wrap, in 30 character field.
print &comment (2,35) wrap 30 5 r
print $notes (1,50) wrap 50 7 r
```

LINE-HEIGHT specifies the number of lines to skip between each line of the wrapped data. By default a value of 1 (single space) is assumed.

The following example prints the *comment* column with one blank line between each printed line for a maximum of four printed lines:

```
print &comment (1,1) wrap 40 4 line-height = 2
! Double space text
```

See Also

The LET command for information about copying, editing, or converting fields.

The ALTER-LOCALE command for a description of the arguments NUMBER-EDIT-MASK, MONEY-EDIT-MASK, and DATE-EDIT-MASK.

The DISPLAY and SHOW commands.

PRINT-BAR-CODE

Syntax

```
PRINT-BAR-CODE position
{TYPE={bar_code_type_num_lit|_var|_col}}
```

```
{HEIGHT={bar_code_height_num_lit|_var|_col}}
```

```
{TEXT={bar_code_txt_lit|_var|_col}}
```

```
[CAPTION={bar_code_caption_txt_lit|_var|_col}]
```

```
[CHECKSUM={checksum_lit}]
```

Description

PRINT-BAR-CODE prints industry standard bar codes. SQR supports the bar code types listed in the following table.

<i>Type</i>	<i>Description</i>	<i>Text Length</i>	<i>Text Type¹</i>	<i>CHECKSUM RECOGNIZED</i>
1	UPC-A	11, 13, or 16	9	
2	UPC-E	11, 13, or 16	9	
3	EAN/JAN-13	12, 14, or 17	9	
4	EAN/JAN-8	7, 9, or 12	9	
5	3 of 9 (Code 39)	1 to 30	9, X, p	y
6	Extended 3 of 9	1 to 30	9, X, x, p, c	y
7	Interleaved 2 of 5	2 to 30	9	y
8	Code 128	1 to 30	9, X, x, p, c	
9	Codabar	1 to 30	9	y
10	Zip+4 Postnet	5, 9, or 11	9	
11	MSI Plessey	1 to 30	9	y
12	Code 93	1 to 30	9, X, p	y
13	Extended 93	1 to 30	9, X, x, p	y
14	UCC-128	19	9	
15	HIBC	1 to 30	9	y

Parameters

<i>Parameter</i>	<i>Description</i>
<i>position</i>	Specifies the position of the upper left corner. Position parameters can be relative. See the POSITION command for examples of relative positioning. Document markers are not allowed. After execution, the current position is returned to this location; however, the next listing line is the next line below the bottom of the bar code. (This is different than the way the PRINT command works.)
TYPE	Specifies the type of bar code to be printed. Types are shown in the Bar Code Types table.

1 Text Type characters:
 9- Numbers (0-9)
 X- Upper Case Letters (A-Z)
 x- Lower Case Letters (a-z)
 p- Punctuation
 c- Control Characters

<i>Parameter</i>	<i>Description</i>
HEIGHT	Specifies the height of the bar code in inches. The height must be between 0.1 and 2 inches. The code prints to the nearest one-tenth of an inch. For Zip+4 Postnet, the height of the bar code is fixed. The height should be between 0.2 and 2.0 for Zip+4 Postnet. If it is less than 0.2, the bar code extends above the position specified.
TEXT	Specifies the text to be encoded and printed. The number and type of text characters permitted or required depends on the bar code type. See the Bar Code Types table for specifications.
CAPTION	Specifies optional text to be printed under the bar code in the current font. SQR attempts to center the caption under the bar code; however, for proportional fonts this may vary slightly. CAPTION is not valid for Zip+4 Postnet. If specified, it is ignored.
CHECKSUM	Specifies an optional check sum to be computed and printed within the bar code. Valid values are YES and NO, where NO is the default. Note. Some barcode types ignore the CHECKSUM qualifier. See the Bar Code Types table for those barcode types for which CHECKSUM is relevant.

Example

This example shows how to use the **PRINT-BAR-CODE** command to create a UPC-A barcode as shown in the following illustration:

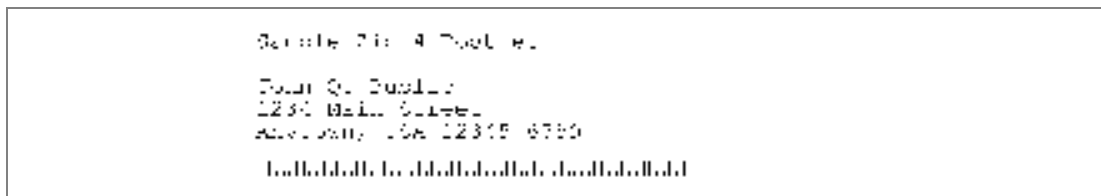
```
begin-program
  print 'Sample UPC-A Barcode' (1,1)
  print-bar-code (3,1)
    type=1                                ! UPC-A
    height=0.3
    text='01234567890'
    caption='0 12345 67890'
end-program
```



Sample UPC-A Bar Code

This example shows how to use the PRINT-BAR-CODE command to create a ZIP+4 Postnet code as shown in the following illustration:

```
begin-program
  print 'Sample Zip+4 Postnet'
    (1,1)
  print 'John Q. Public'
    (3,1)
  print '1234 Main Street'
    (4,1)
  print 'AnyTown, USA 12345-6789'
    (5,1)
  print-bar-code                                     (7,1)
    type=10
    height=0.2
    text='12345678934'
end-program
```



Sample Zip+4 Postnet Bar Code

Note. SQR does not check bar code syntax. See your bar code documentation for the proper formatting of certain bar codes.

PRINT-CHART

Syntax

```
PRINT-CHART[chart_name]position
DATA-ARRAY-ROW-COUNT={x_num_lit|_var|_col}
DATA-ARRAY-COLUMN-COUNT={x_num_lit|_var|_col}
DATA-ARRAY=array_name
[DATA-LABELS=data_labels_lit | _var | _col]
[COLOR-PALETTE=color_palette_lit | _var | _col]
[ITEM-COLOR=(Chart_item_keyword, txt_lit | var | (r,g,b))]
[DATA-ARRAY-COLUMN-LABELS={NONE|array_name|
  {(txt_lit|_var)[, (txt_lit|_var)]...}}]
[CHART-SIZE=(chart_width_num_lit|_var,
  chart_depth_num_lit|_var)]
[TITLE={title_txt_lit|_var|_col}]
[SUB-TITLE={subtitle_txt_lit|_var|_col}]
[FILL={fill_lit|txt_var|_col}]
[3D-EFFECTS={3d_effects_lit|txt_var|_col}]
[BORDER={border_lit|txt_var|_col}]
[POINT-MARKERS={point_markers_lit|txt_var|_col}]
```

```

[TYPE={chart_type_lit|txt_var|_col}]
[LEGEND={legend_lit|txt_var|_col}]
[LEGEND-TITLE={legend_title_txt_lit|_var|_col}]
[LEGEND-PLACEMENT={legend_placement_lit|txt_var|_col}]
[LEGEND- PRESENTATION={legend_presentation_lit|txt_var|_col}]
[PIE-SEGMENT-QUANTITY-DISPLAY=
{pie_segment_quantity_display_lit|txt_var|_col}]
[PIE-SEGMENT-PERCENT- DISPLAY={pie_segment_percent_display_lit
|txt_var|_col}]
[PIE-SEGMENT-EXPLODE={pie_segment_explode_lit
|txt_var|_col}]
[X-AXIS-LABEL={x_axis_label_txt_lit|_var|_col}]
[X-AXIS-MIN-VALUE={x_axis_min_value_lit|_num_lit|_var
|_col}]
[X-AXIS-MAX-VALUE={x_axis_max_value_lit|_num_lit|_var
|_col}]
[X-AXIS-SCALE={x_axis_scale_lit|txt_var|_col}]
[X-AXIS-MAJOR-TICK-MARKS={x_axis_major_tick_marks_lit
|txt_var|_col}]
[X-AXIS-MINOR-TICK-MARKS={x_axis_minor_tick_marks_lit
|txt_var|_col}]
[X-AXIS-MAJOR-INCREMENT={x_axis_major_increment_lit
|_num_lit|_var|_col}]
[X-AXIS-MINOR-INCREMENT={x_axis_minor_increment_lit
|_num_lit|_var|_col}]
X-AXIS-TICK-MARK-PLACEMENT=
{x_axis_tick_mark_placement_lit|txt_var|_col}]
[X-AXIS-GRID={x_axis_grid_lit|txt_var|_col}]
[Y-AXIS-LABEL={y_axis_label_lit|txt_var|_col}]
[Y-AXIS-MIN-VALUE={y_axis_min_value_lit|_num_lit
|_var|_col}]
[Y-AXIS-MAX-VALUE={y_axis_max_value_lit|_num_lit
|_var|_col}]
[Y-AXIS-SCALE={y_axis_scale_lit|txt_var|_col}]
[Y-AXIS-MAJOR-TICK-MARKS={y_axis_major_tick_marks_lit
|txt_var|_col}]
[Y-AXIS-MINOR-TICK-MARKS={y_axis_minor_tick_marks_lit
|txt_var|_col}]
[Y-AXIS-MAJOR-INCREMENT={y_axis_major_increment_lit
|_num_lit|_var|_col}]
[Y-AXIS-MINOR-INCREMENT={y_axis_minor_increment_lit
|_num_lit|_var|_col}]
[Y-AXIS-TICK-MARK-PLACEMENT=
{y_axis_tick_mark_placement_lit|txt_var|_col}]
[Y-AXIS-GRID={y_axis_grid_lit|txt_var|_col}]

```

Note. If you do not define CHART-SIZE with this command, you must define it with **DECLARE-CHART**.

Description

Prints a chart. Only PostScript printers or HP printers that support HPGL (generally, this is HPLaserJet 3 and higher) render chart output.

The **PRINT-CHART** command directs SQR to output a chart according to the named chart, if any, and the overridden attributes, if any.

Note. PRINT-CHART can be used without referencing a named chart if all required attributes for the DECLARE-CHART are supplied in addition to all its required parameters. **Note.** The PRINT-CHART command directs SQR to display the chart on the current page using the attribute values at the moment the command is executed. Manipulation of chart attribute values has no effect on the appearance of the chart after the PRINT-CHART command has been executed. For example, if you execute a PRINT-CHART with TITLE=\$ttl and \$ttl='Encouraging Results', and then change the value of \$ttl to 'Discouraging Results' immediately afterward, then the chart is printed with first value, 'Encouraging Results'. **Note.** PRINT-CHART expects the DATA-ARRAY to be organized in a particular way. See the Chart Array Field Types (fewer than four fields) table for details. **Note.** PRINT-CHART fills the area defined by CHART-SIZE as much as possible while maintaining an aesthetically pleasing ratio of height to width. In cases where the display area is not well suited to the chart display, the chart is centered within the specified region, and the dimensions are scaled to accommodate the region. Therefore, do not be alarmed if the chart does not fit exactly inside the box you have specified. It means that SQR has accommodated the shape of the region to provide the best looking chart possible. **Note.** Chart commands used to send output to a line printer are ignored. **Note.** Only PostScript printers or HP printers that support Hewlett Packard's HPGL (generally, this is HP LaserJet model 3 and higher) render chart output. If you attempt to print a chart to a LASERJET printer that does not support HPGL, the HPGL command output will likely become part of your output, leaving one or more lines of meaningless data across your report.

All the attributes defined for DECLARE-CHART are valid for the PRINT-CHART command. PRINT-CHART has five additional parameters. The position of the chart is described using the first parameter. The data that supports the chart is defined in the additional attributes: DATA-ARRAY, DATA-ARRAY-ROW-COUNT, DATA-ARRAY-COLUMN-COUNT, and DATA-ARRAY-COLUMN-LABELS.

As mentioned previously, each chart type meets a specific organizational requirement. The Chart Array Field Types (fewer than four fields) table describes these requirements.

Note. If the first field in the array designated by DATA-ARRAY is of type CHAR, then the value on the x-axis is the contents of that column. If the first field is not of type CHAR, then the value of the x-axis is the row number of the array designated by DATA-ARRAY, beginning with 1. Pie charts show the character value in the legend area. Histograms show the character value on the y-axis. XY-Scatter charts do not use the character value and none is needed in the array.

Note. If a PIE chart contains many small slices, the user must set the PIE-SEGMENT-QUANTITY-DISPLAY and/or PIE-SEGMENT-PERCENT-DISPLAY arguments to NO to prevent the values from one slice overwriting the values of another slice.

Chart Type	Field 0	Field 1	Field 2	Field 3
PIE	Type=char Pie segment labels, the names associated with each segment	Type=num The value associated with each pie segment	(Optional) Type=char Pie segment explode flag setting, 'Y' or 'N'	
LINE BAR STACKED-BAR 100%-BAR OVER-LAPPED-BAR HISTOGRAM AREA STACKED-AREA 100%-AREA	Type=char X-Axis values	Type=num Series 1 Y-Axis values	(Optional) Type=num Series 2 Y-Axis values	(Optional) Type=num Series 3... Y-Axis values
XY-SCATTER-PLOT	Type=num Series 1 X-Axis values	Type=num Series 1 Y-Axis values	(Optional) Type=num Series 2 X-Axis values	(Optional) Type=num Series 2 ... Y-Axis values
FLOATING-BAR	Type=char X-Axis values	Type=num Series 1 Y-Axis offset	Type=num Series 1 Y-Axis duration	(Optional) Type=Num Series 2 ... Y-Axis offset

Chart Type	Field 0	Field 1	Field 2	Field 3	Field 4
HIGH-LOW-CLOSE	Type=char X-Axis values	Type=num High value	Type=num Low value	Type=num Closing value	(Optional) Type=num Opening value

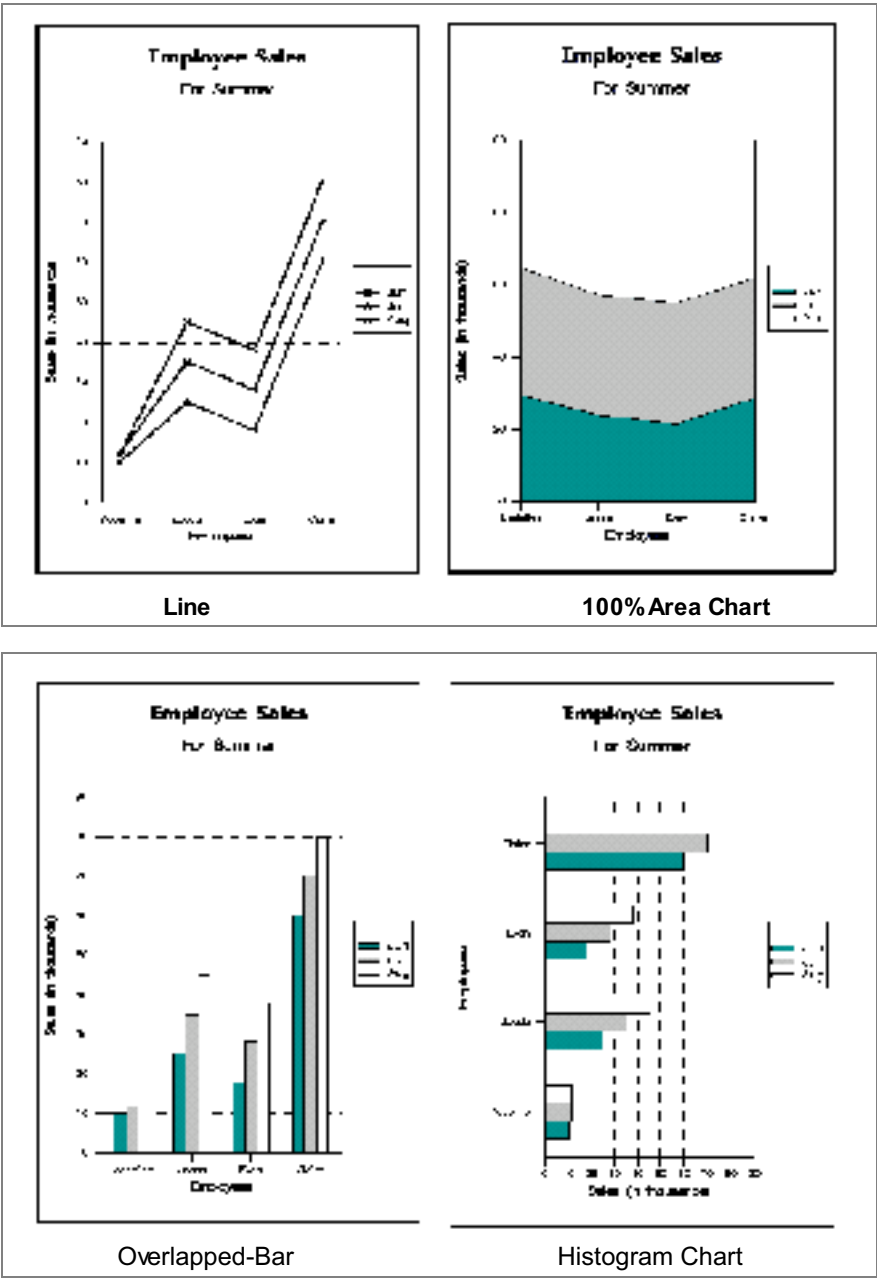
Parameters

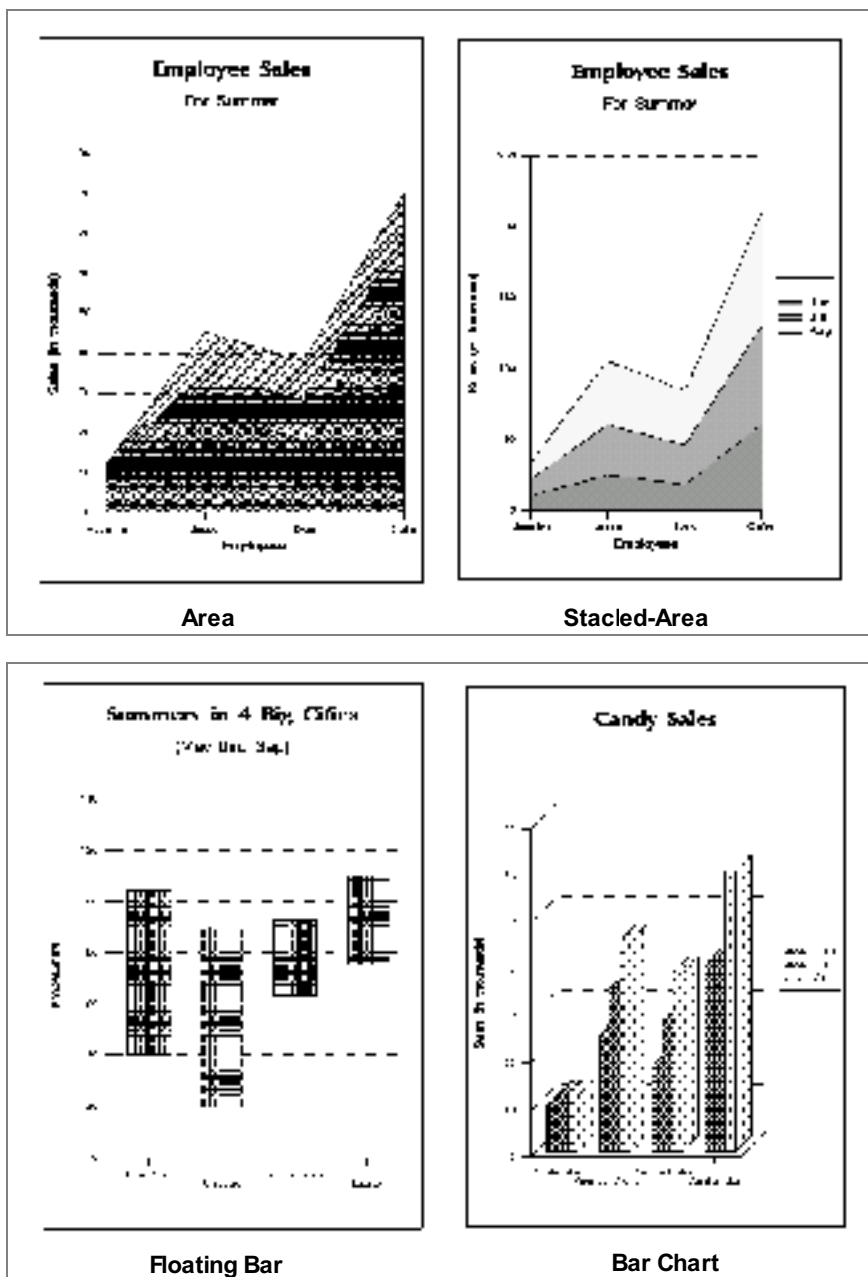
Parameter	Description
<i>chart_name</i>	Specifies the name of the chart from the DECLARE-CHART command. This name is not necessary if you specify the CHART-SIZE and all other pertinent attributes in the PRINT-CHART command.

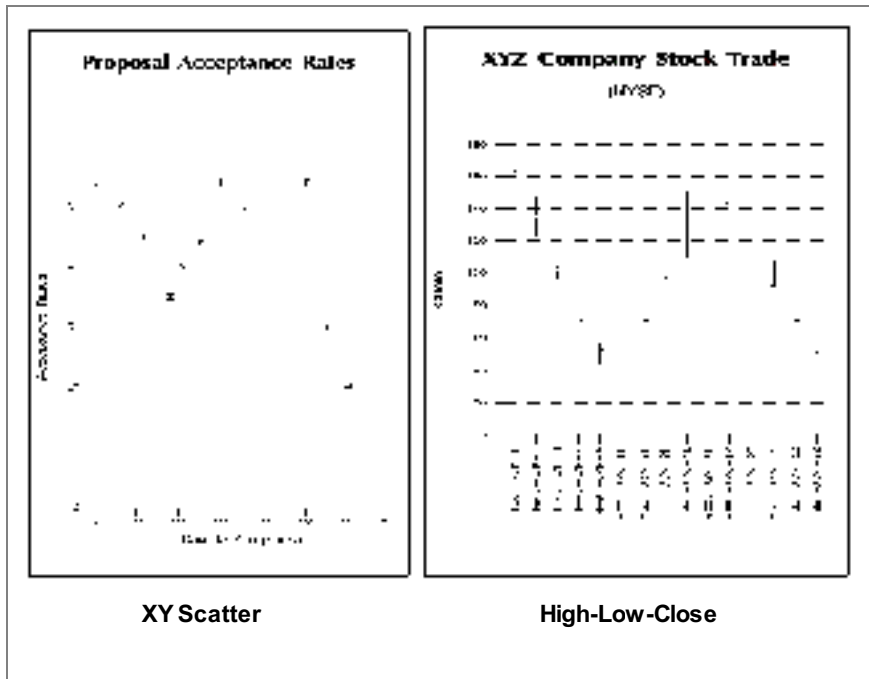
Parameter	Description
<i>position</i>	(row, column) Specifies the position of the upper left corner. Position parameters can be relative. See the POSITION command for examples of relative positioning. Document markers are not allowed. After execution, the current position is returned to this location; however, the next listing line is the next line below the bottom of the chart area. (This is different than the way the PRINT command works.)
DATA-ARRAY	Specifies the name of the array containing the data to be plotted. This must be the name of an array defined with CREATE-ARRAY.
DATA-ARRAY-ROW-COUNT	Specifies the number of rows or sets of data to be used from the DATA-ARRAY. If the DATA-ARRAY has a greater number of rows, only DATA-ARRAY-ROW-COUNT is included in the chart.
DATA-ARRAY-COLUMN-COUNT	Specifies the number of columns to be used from the DATA-ARRAY. If the DATA-ARRAY has a greater number of columns, only DATA-ARRAY-COLUMN-COUNT is included in the chart.
DATA-ARRAY-COLUMN-LABELS	Specifies labels for each Y-Axis value of the data set (fields) in DATA-ARRAY. These labels are displayed in the legend box. Column labels are ignored for pie charts. See the Chart Array Field Types (fewer than four fields) table for applicable fields for each type of chart.

For definitions of the other arguments, see the DECLARE-CHART Command Arguments table.

Example

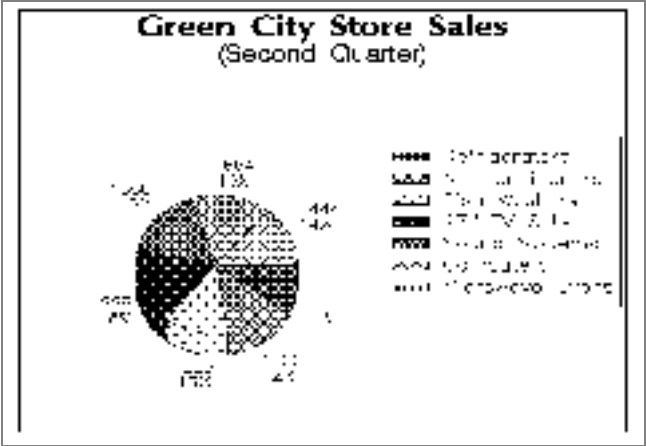






In this example, a pie chart is printed without explicit reference to a chart declared with **DECLARE-CHART**. All necessary arguments must be supplied in **PRINT-CHART**.

```
.
.
create-array
    name=unit_sales
    size=12
    field=product:char
    field=units:number
    field=explode:char
.
.
print-chart (15, 20)
    title                                = 'Green City Store Sales'
    sub-title                            = '(Second Quarter)'
    chart-size                           = (50, 28)
    type                                 = pie
    data-array                           = unit_sales
    data-array-column-count              = 3
    data-array-row-count                 = 7
    3d-effects                           = yes
    fill                                 = color
```



See Also

The `DECLARE-CHART` command.

PRINT-DIRECT

Syntax

```
PRINT-DIRECT
[NOLF]
[PRINTER={LINEPRINTER|POSTSCRIPT|HPLASERJET|HTML|LP|PS|HP|HT}]
{txt_lit|_var|_col}...
```

Description

Writes directly to the print output file without using the SQR page buffer.

PRINT-DIRECT can be used for special applications that cannot be accomplished directly with `PRINT` commands, such as initializing a page with graphics or other special sequences. Because this text is often printer-dependent and because the report can be printed on different types of printers that require different control characters, you can use the `PRINTER` qualifier to specify the printer type. If no `PRINTER` qualifier is specified, the command applies to all printer types.

When using PRINT-DIRECT in conjunction with `PRINT` commands, be aware that the SQR page buffer is copied to the output file only when each page is full or when a `NEW-PAGE` command is issued. One approach is to use `PRINT-DIRECT` commands inside a `BEFORE-PAGE` or `AFTER-PAGE` procedure (declared with the `DECLARE-PROCEDURE` command), so they are coordinated with the information coming out of the page buffer.

Parameters

Parameter	Description
NOLF	Specifies that no carriage return and line feed is to be printed. By default, printed text is followed by a carriage return and line feed character.

Parameter	Description
PRINTER	Specifies the type of printer to which this text applies.
{ <i>txt_lit</i> <i>_var</i> <i>_col</i> }	The text to be printed.

Example

```
print-direct printer=ps '%%Page: ' $page-number
print-direct nolf printer=lp reset
```

PRINT-IMAGE**Syntax**

```
PRINT-IMAGE [image_name] position
[TYPE={image_type_lit | _var | _col}]
[IMAGE-SIZE=(width_num_lit | _var | _col, height_num_lit
| _var | _col)]
[SOURCE={file_name_txt_lit | _var | _col}]
```

Note. If TYPE, IMAGE-SIZE, and SOURCE are not defined in PRINT-IMAGE, they must be defined in DECLARE-IMAGE.

Description

Prints an image.

The PRINT-IMAGE command can be placed in any section of a report with the exception of the SETUP section. The image file pointed to can be any file of the proper format.

PRINT-IMAGE may be used without referencing a named image if all required attributes for the DECLARE-IMAGE are supplied in addition to all its required parameters.

Parameters

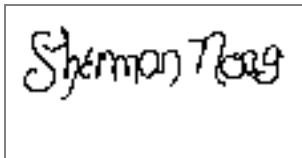
Parameter	Description
<i>image_name</i>	Specifies the name of an image specified by a DECLARE-IMAGE.
<i>position</i>	(row, column) Specifies the position of the upper left corner. Position parameters can be relative. See the POSITION command for examples of relative positioning. Document markers are not allowed. After execution, the current position is returned to this location; however, the next listing line below the bottom of the image area. (This is different from the way the PRINT command works.)

Parameter	Description
TYPE	Specifies the image type. Types can be EPS-FILE, HPGL-FILE, GIF-FILE, JPEG-FILE, or BMP-FILE (for Windows).
IMAGE-SIZE	Specifies the width and height of the image.
SOURCE	Specifies the name of a file containing the image.

Example

For PostScript:

```
print-image office-signature (50, 20)
print-image      (50, 20)
    type                = eps-file
    source               = 'sherman.eps'
    image-size          = (10, 3)
```



For Windows:

```
print-image company-logo (+21, 25)
    type=bmp-file
    source='m:\logos\gustavs.bmp'
    image-size=(75,50)
```

**See Also**

The **DECLARE-IMAGE** command.

PUT
Syntax

```
PUT {src_any_lit|_var|_col}...
INTO dst_array_name(element) [field[(occurs)]]...
```

Description

Moves data into an array.

Columns retrieved from the database and SQR variables or literals can be moved into an array. The array must have been created previously using the **CREATE-ARRAY** command.

Considerations using PUT

When a date variable or column is moved into a text or char array field, the date is converted to a string according to the following rules:

- For DATETIME columns and SQR DATE variables, the format specified by the SQR_DB_DATE_FORMAT setting is used. If this has not been set, the first database-dependent format as listed the Default Database Formats table is used.
- For DATE columns, the format specified by the SQR_DB_DATE_ONLY_FORMAT setting is used. If this has not been set, the format listed in the DATE Column Formats table is used.
- For TIME columns the format specified by the SQR_DB_TIME_ONLY_FORMAT setting is used. If this has not been set, the format as listed in the TIME Column Formats table is used.

When a string variable, column, or literal is moved to a date array field, the string must be in the format specified by the SQR_DB_DATE_FORMAT setting, one of the database-dependent formats as listed in the DATE Column Formats table is used, or the database-independent format 'SYYYYMMDD [HH24 [MI [SS [NNNNNN]]]]'.

```
dst_array_name(element)
```

If array fields are listed, data is placed into each field in the sequence it is listed, in the occurrence specified of that field.

If array fields are not listed, data is placed into consecutive fields in the order they were defined in the CREATE-ARRAY command; data is copied into occurrence zero of each field of the element specified in the array.

```
field [ ( occurs ) ]
```

Array element and field occurrence numbers can be numeric literals (123) or numeric variables (#j).

If no occurrence is specified, occurrence zero is used.

Parameters

Parameter	Description
<i>src_any_var</i>	The source variable or literal to be moved into the array. Numeric variables, literals, and database columns can be put into number (decimal, float, integer) fields. String variables, literals, and database columns can be put into char, text, or date fields. Date variables can be put into date, char, or text fields.

Example

In the following example, the four variables *&name*, *#count*, *\$date1*, and *\$code* are placed into the first four fields defined in the *names* array. The data is put into the *#j*'th element of the array.

```
put &name #count $date1 $code into names(#j)
```

The following command places *#j2*, *#j3*, and *#j4* into the zero through 2nd occurrences of the *tot* field in the *#j*'th element of the *totals* array.

```
put #j2 #j3 #j4 into totals(#j) tot(0) tot(1) tot(2)
```

The following command copies *#count* into the *#j2*'th occurrence of the *count* field in the *#j*'th element of the *states* array.

```
put #count into states(#j) count(#j2)
```

READ**Syntax**

```
READ {filenum_lit|_var|_col} INTO  
{any_var:length_int_lit}...[STATUS=status_num_var]
```

Description

Reads the next record of a file into the specified variables.

Text and binary data is parsed according to the following criteria:

- Text data is any string of characters. The length of the variable name indicates how many characters to place into the variable. After being transferred, trailing blanks in the variable are omitted.
- If the field was written as a date variable, then it may be read into a date variable or text variable. When reading a date into a date variable, it must be in the format specified by the *SQR_DB_DATE_FORMAT* setting, one of the database-dependent formats as listed in the *DATE Column Formats* table is used, or the database-independent format 'SYYYYMMDD[HH24[MI[SS[NNNNNN]]]'.

- Binary numbers, may be 1, 2, or 4 bytes in length. They must be read into numeric variables. Note that the bytes making up the binary number must be in the standard sequence expected by your operating system.
- When reading binary data the file must be opened with the FIXED or FIXED-NOLF qualifier.
- Only the integer portion of the number is represented with binary numbers. To maintain the decimal portion of the number convert the number to a string variable.
- If you use binary numbers, the file is not portable across platforms. This is because different hardware represents binary numbers differently.

The total length indicated for the variables must be less than or equal to the length of the record being read.

If there are no more records to read, the *#end-file* reserved variable is set to 1; otherwise, it is set to 0 (zero). Your program should check this variable after each READ command.

If **STATUS** is specified, SQR returns 0 if the read is successful; otherwise, it returns the value of *errno*, which is system-dependent.

Parameters

Parameter	Description
<i>filenum_lit/_var / _col</i>	Specifies the number assigned in the OPEN command to the file to be read.
<i>any_var :length_int_lit</i>	Specifies one or more variables into which data from the record read are to be put. <i>length_int_lit</i> specifies the length of each field of data.
STATUS	Specifies an optional variable into which a read status is returned.

Example

The following example shows several READ commands:

```
read 1 into $name:30 $addr:30 $city:20 $state:2 $zip:5
read 3 into $type:2 #amount:2 #rate:1 $code:5 $date:11
read #j into #sequence:2 $name:20 $title:15
```

The following example shows a **READ** command that reads two dates. One is loaded into a date variable; the other is loaded into a string variable, which is then converted to a date using the *strtodate* function.

```
.
.
.
declare-variable
date $date1 $date2
```

```

    text $text
end-declare
.
.
.
read 4 into $date1:18 $text1:18
let $date2 = strtodate($text1,'YYYYMMDDHHMISSNNN')
           or
let $date2 = strtodate($text1)

```

The following example shows a **READ** command with an INSERT loop:

```

begin-sql
  begin transaction
end-sql

while 1      ! Infinite loop, exited by BREAK, below.
  read 10 into $company:40 $parent:30 $location:50
  if #end-file
    break    ! End of file reached.
  end-if
  begin-sql
    insert into comps (name, parent, location)
      values ($company, $parent, $location)
  end-sql
  add 1 to #inserts
  if #inserts >= 100
    begin-sql
      end transaction;
      begin transaction
    end-sql
    move 0 to #inserts
  end-if
end-while

begin-sql
  end transaction
end-sql

```

See Also

The **OPEN**, **CLOSE**, and **WRITE** commands for information about files.

ROLLBACK

Syntax

```
ROLLBACK
```

Description

An automatic rollback is performed whenever SQR ends due to program errors. ROLLBACK is useful in testing or in certain error conditions.

ROLLBACK is an SQR command and should not be used inside an SQL paragraph.

Note. The ROLLBACK command can be used with DB2, Informix, Ingres, ODBC, ORACLE, and SQLBase. (Microsoft SQL Server is accessible only with SQR Server for ODBC.) For SYBASE and Microsoft SQL Server, use **BEGIN TRANSACTION** and **ROLLBACK TRANSACTION** within SQL paragraphs as in the following example. See the **COMMIT** command for an example of ROLLBACK.

Example

```
if #error-status = 1
    rollback
    stop
end-if
```

See Also

The COMMIT command.

SBTOMBS

Syntax

```
SBTOMBS { txt_var }
```

Description

Converts a single-byte character into a multi-byte equivalent.

This command converts the specified string as follows: Any occurrence of a single-byte character that also has a multi-byte representation (numerals, punctuation, roman characters and katakana) is converted. This command also converts a sequence of a kana character followed by certain grammatical marks into a single multi-byte character, which combines the two elements.

Parameters

Parameter	Description
<i>txt_var</i>	Specifies the string to be converted.

See Also

The `TO_MULTI_BYTE` function of the LET command.

SECURITY

Syntax

```
SECURITY
[SET=(sid [,sid] ...)]
[APPEND=(sid [,sid] ...)]
[REMOVE=(sid [,sid] ...)]
[MODE=mode]
```

Description

Enables you to mark sections of a report for security purposes.

The SECURITY command can be repeated as many times as desired for the current report. After the SECURITY command is executed, all subsequent commands for the current report are constrained by the designated *sids* until the report ends or another SECURITY command executes.

You can use the SECURITY command wherever you use the PRINT command.

Parameters

Parameter	Description
SET	Sets the list of security IDs for subsequent commands. The previous list of security IDs is replaced by the specified security IDs. This argument is optional and can only be used once.
<i>sid</i>	Can be any string literal, column, or variable. The value is case-sensitive.
APPEND	Appends the specified security IDS to the current list. This argument is optional and can be used multiple times.
REMOVE	Removes the specified security IDS from the current list. This argument is optional and can be used multiple times.
MODE	Used to turn on (reactivate) or turn off (suspend) the security feature for the current report. This argument is optional and can only be used once.
<i>mode</i>	Can be any string literal, column, or variable. The value is not case-sensitive and can be either ON or OFF.

Example

```
Begin-Report
```

```

Security Set=('Directors', 'Vice-Presidents')
.
.           ! Only Directors and VPS can see this
.
Security Remove=('Directors')
.
.           ! Only VPS can see this
.
Security Mode='Off'
.
.           ! Anybody can see this
.
Security Mode='On' Append=('Managers')
.
.           ! Only VPs and Managers can see this
.
Security Append=('Engineers')
.
.           ! Only VPs, Managers, and Engineers can see
this
.
End-report

```

SET-COLOR

Syntax

```

SET-COLOR
[PRINT-TEXT- FOREGROUND=({color_name_lit|_var|_col|{rgb}})]
[PRINT-TEXT- BACKGROUND=({color_name_lit|_var|_col|{rgb}})]

```

Description

Defines default colors.

The SET-COLOR command is allowed wherever the PRINT command is allowed. If the specified color name is not defined, SQR uses the settings for the color name 'default.' Use the color name 'none' to turn off color for the specified area.

Parameters

Parameter	Description
PRINT-TEXT- FOREGROUND	Defines the color in which the text is printed.
PRINT-TEXT-BACKGROUND	Defines the color to print as a background behind the text.

Parameter	Description
<i>{color_name_lit _var _col}</i>	A <i>color_name</i> is composed of the alphanumeric characters (A-Z, 0-9), the underscore (_) character, and the hyphen (-) character. It must start with an alpha (A-Z) character. It is case-insensitive. The name 'none' is reserved and cannot be assigned a value. A name in the format (RGBredgreenblue) cannot be assigned a value. The name 'default' is reserved and may be assigned a value. 'Default' is used during execution when a referenced color is not defined in the runtime environment.
<i>{rgb}</i>	<i>red_lit _var _col, green_lit _var _col, blue_lit _var _col</i> where each component is a value in the range of 000 to 255. In the BEGIN-SETUP section, only literal values are allowed. The default colors implicitly installed with SQR include: black = (0,0,0) white=(255,255,255) gray=(128,128,128) silver=(192,192,192) red=(255,0,0) green=(0,255,0) blue=(0,0,255) yellow=(255,255,0) purple=(128,0,128) olive=(128,128,0) navy=(0,0,128) aqua=(0,255,255) lime=(0,128,0) maroon=(128,0,0) teal=(0,128,128) fuchsia=(255,0,255)

Example

```

begin-setup
  declare-color-map
  light_blue = (193, 222, 229)
end-declare
end-setup

```

```

begin-program
  alter-color-map name = 'light_blue' value = (193, 233, 230)

  print 'Yellow Submarine' ()
    foreground = ('yellow')
    background = ('light_blue')

  get-color print-text-foreground = ($print-foreground)
  set-color print-text-foreground = ('purple')
  print 'Barney' (+1,1)
  set-color print-text-foreground = ($print-foreground)
end-program

```

See Also

The DECLARE-COLOR-MAP, ALTER-COLOR-MAP, and GET-COLOR commands in this section.

SET-GENERATIONS

Syntax

```
SET-GENERATIONS=(dimension, hierarchy, dimension, hierarchy, dimension,
hierarchy,...,...)
```

Description

Specifies dimension hierarchy for the previously declared dimension.

Returns the set of members in the dimension 'product' that are at the 5th generation in the dimension's hierarchy. (Returns all 'Brand Name' members (Generation Level 5) under the product hierarchy of 'all products.drink.alcoholic beverages.beer and wine.' This would increase the result set to a list of beers and wines.) Returns the set of members in the dimension 'time' that are at the 1st generation deep into the dimension. (Returns all 'Year' members (Generation Level 1) under the time hierarchy of '1997.Q1.2'. This reduces result set to '1997'.)

Example

```
set-generations=('product',5,'time',1 )
```

SET-LEVELS

Syntax

```
Set-levels=(dimension, level, dimension, level,...,...)
```

Description

Extends the dimension hierarchy for the previously declared dimension.

Set-levels used with only the previous 'set-members', returns all members under the product hierarchy and the next two generations (Product SubCategory and Brand Name) for the product hierarchy of all products.drink.alcoholic beverages.beer and wine'. Set-levels used with the previous 'set-members' and 'set-generations', returns all members for generation levels 5 through 7 under the product hierarchy of all products.drink.alcoholic beverages.beer and wine.'

Example

```
set-levels=('product',2 )
```

SET-MEMBERS
Syntax

```
set-members=(dimension, hierarchy, dimension, hierarchy,...,...)
```

Description

Returns the set of members in a dimension, level, or hierarchy whose name is specified by a string.

Returns the set of members in the dimension 'product' at the specific hierarchy of 'all products', at a specific level of 'drink', at a specific level of 'alcoholic beverages', at a specific level of 'beer and wine'. Returns the set of members in the dimension 'time' at the specific hierarchy of '1997', at the specific level of 'Q1', at the specific level of '2'.

Example

```
set-members=('product','all products.drink.alcoholic beverages.beer and wine','time','1997.Q1.2' )
```

SHOW
Syntax

```
SHOW[cursor_position]  
[CLEAR-SCREEN|CS|CLEAR-LINE|CL] [any_lit|_var|_col]  
[EDITedit_mask|NUMBER|MONEY|DATE] [BOLD] [BLINK]  
[UNDERLINE] [REVERSE] [NORMAL] [BEEP] [NOLINE] ...
```

Description

Displays one or more variables or literals on the screen. In addition, cursor control is supported for ANSI terminals.

Any number of variables and screen positions can be used in a single command. Each one is processed in sequence.

Screen locations can be indicated by either fixed or relative positions in the format (A,B), where A is the line and B is the column on the screen. A and/or B can also be numeric variables. Relative positions depend on where the previous SHOW command ended. If the line was advanced, the screen cursor is usually immediately to the right of the previously displayed value and one line down.

Fixed or relative cursor positioning can be used only within the boundaries of the terminal screen. Scrolling off the screen using relative positioning, for example (+1,1), is not supported. Instead, use a **SHOW** command without any cursor position when you want to scroll. Also, you cannot mix **SHOW** and **DISPLAY** commands while referencing relative cursor positions.

The **SHOW** command does not advance to the next line if a cursor location (...), **CLEAR-SCREEN**, **CLEAR-LINE**, or **BEEP** is used. (A **SHOW** command without any of these arguments automatically advances the line.) To add a line advance, add (+1,1) to the end of the line or use an extra empty **SHOW** command.

Only ANSI terminals are supported for cursor control, screen blanking, line blanking, and display characteristics.

Dates can be contained in a date variable or column, or a string literal, column, or variable. When displaying a date variable or column, without an edit mask, the date is displayed according to the following rules:

- For DATETIME columns and SQR DATE variables, the format specified by the **SQR_DB_DATE_FORMAT** setting is used. If this has not been set, the first database-dependent format as listed in the Default Database Formats table is used.
- For DATE columns, the format specified by the **SQR_DB_DATE_ONLY_FORMAT** setting is used. If this has not been set, the format listed in the Default Database Formats table is used.
- For TIME columns, the format specified by the **SQR_DB_TIME_ONLY_FORMAT** setting is used. If this has not been set, the format as listed in the TIME Column Formats table is used.

When displaying a date in a string literal, column, or variable using **EDIT** or **DATE**, the string must be in the format specified by the **SQR_DB_DATE_FORMAT** setting, one of the database-dependent formats as listed in the Default Database Formats table, or the database-independent format 'YYYYMMDD[HH24[MI[SS[NNNNNN]]]]'.

Parameters

Parameter	Description
<i>cursor_position</i>	Specifies the position on the screen to begin the display.

Parameter	Description
{CLEAR-SCREEN CS}	Clears the screen and sets the cursor position to (1,1).
{CLEAR-LINE CL}	Clears a line from the current cursor position to the end of the line.
{ <i>any_lit</i> / <i>_var</i> / <i>_col</i> }	Specifies the information to be displayed.
EDIT	Shows variables under an edit mask. If the mask contains spaces, enclose it in single quotes. For additional information regarding edit masks, see the PRINT command.
NUMBER	Indicates that any <i>_lit</i> <i>_var</i> <i>_col</i> is to be formatted using the NUMBER-EDIT- MASK from the current locale. (See the ALTER-LOCALE command.) This option is not legal for date variables.
MONEY	Indicates that any <i>_lit</i> <i>_var</i> <i>_col</i> is to be formatted using the MONEY-EDIT- MASK from the current locale. (See the ALTER-LOCALE command.) This option is not legal for date variables.
DATE	Indicates that any <i>_lit</i> <i>_var</i> <i>_col</i> is to be formatted using the DATE-EDIT-MASK from the current locale. (See the ALTER-LOCALE command.) This option is not legal for numeric variables. If DATE-EDIT-MASK has not been specified, then the date is displayed using the default format for that database (see the Default Database Formats table).
BOLD, BLINK, UNDERLINE, and REVERSE	Changes the display of characters on terminals that support those characteristics. Some terminals support two or more characteristics at the same time for the same text. To turn all special display characteristics off, use NORMAL.
NORMAL	Turns off all special display characteristics set with BOLD, BLINK, UNDERLINE, and REVERSE.
BEEP	Causes the terminal to beep.
NOLINE	Inhibits a line advance.

Example

The following program segments illustrate the various features of the **SHOW** command:

```
!
! Show a string using an edit mask
!
```



```
let $ssn = '123456789'
show $ssn edit xxx-xx-xxxx
```

Produces the following output:

```
123-45-6789

!
! Show a number using an edit mask
!
show 1234567.89 edit 999,999,999.99
```

Produces the following output:

```
1,234,567.89

!
! Show a number using the default edit mask
!
show 123.78
```

Produces the following output:

```
123.780000

!
! Show a number using the locale default numeric edit mask
!
alter-locale number-edit-mask = '99,999,999.99'
show 123456.78 number
```

Produces the following output:

```
123,456.78

!
! Show a number using the locale default money edit mask
!
alter-locale money-edit-mask = '$$,$$$,$$8.99'
show 123456.78 money
```

Produces the following output:

```
$123,456.78

!
```

```

! Show a date column using the locale default date edit mask
!
begin-select
dcol
  from tables
end-select
alter-locale date-edit-mask = 'DD-Mon-YYYY'
show &dcol date

```

Produces the following output:

```
01-Jan-1999
```

```

!
! Show two values on the same line
!
show 'Hello' ' World'

```

Produces the following output:

```
Hello World
```

```

!
! Show two values on the same line with editing of the values
!
let #taxes = 123456.78
show 'You owe ' #taxes money ' in back taxes.'

```

Produces the following output:

```
You owe $123,456.78 in back taxes.
```

The following program illustrates the usage of additional options of the SHOW command. Only terminals that support the ANSI escape characters can use the cursor control, screen blanking, line blanking and display attributes.

```

begin-program
!
! Produces a menu for the user to select from
!
show clear-screen
  (3,30) bold 'Accounting Reports for XYZ Company' normal
  (+2,10) '1. Monthly Details of Accounts'
  (+1,10) '2. Monthly Summary'
  (+1,10) '3. Quarterly Details of Accounts'
  (+1,10) '4. Quarterly Summary'

```

```

!
! Show a line of text and numerics combined
!
show (+2,1)
    'The price is ' #price edit 999.99

    '    Total = ' #total edit 99999.99

!
! Put an error message on a particular line
!
show (24,1) clear-line 'Error in SQL.  Please try again.' beep
end-program

```

See Also

The **LET** command for information about copying, editing, or converting fields.

The EDIT parameter of the **PRINT** command for a description of the edit masks.

The ALTER-LOCALE command for a description of the arguments **NUMBER-EDIT-MASK**, **MONEY-EDIT-MASK**, and **DATE-EDIT-MASK**.

The **DISPLAY** command.

STOP

Syntax

```
STOP [QUIET]
```

Description

The STOP command halts SQR and executes a ROLLBACK command (not in SYBASE, Microsoft SQL Server, or Informix). All report page buffers are flushed, if they contain data; however, no headers or footers are printed and the **AFTER-PAGE** and AFTER-REPORT procedures are not executed.

STOP is useful in testing.

Parameters

Parameter	Description
QUIET	Causes the report to complete with the "SQR: End Of Run" message, instead of ending with an error message.

Example

```

if #error-status = 1
    rollback
    stop
else
    commit
    stop quiet
end-if

```

STRING**Syntax**

```

STRING {src_any_lit|_var|_col}...BY {delim_txt_lit|_var|_col}
INTO dst_txt_var

```

Description

Concatenates a list of variables, columns, or literals into a single text variable. Each member of the list separated by the specified delimiter string.

The destination string must not be included in the list of source strings.

Parameters

Parameter	Description
<i>{src_any_lit _var _col}</i>	Specifies one or more fields to be concatenated, separated by the <i>delim_txt_lit _var _col</i> character or characters, and placed into the <i>dst_txt_var</i> variable. If the source is a date variable or column, it is converted to a string according to the following rules: For DATETIME columns and SQR DATE variables, the format specified by the SQR_DB_DATE_FORMAT setting is used. If this has not been set, the first database-dependent format as listed in the Default Database Formats table is used. For DATE columns, the format specified by the SQR_DB_DATE_ONLY_FORMAT setting is used. If this has not been set, the format listed in the Default Database Formats table is used. For TIME columns the format specified by the SQR_DB_TIME_ONLY_FORMAT setting is used. If this has not been set, the format as listed in the TIME Column Formats table is used.

Parameter	Description
<i>{delim_txt_lit/_var/_col}</i>	Specifies one or more characters to be used as separator characters between the source fields.
<i>dst_txt_var</i>	Specifies the destination field for the concatenated result.

Example

```

string &name &city &state &zip by ' - ' into $show-info
                                ! Result:  Sam Mann - New York - NY -
11287
string &cust_num &entry-date &total by ',' into $cust-data
                                ! Result:  100014,12-MAR-89,127
                                ! Use null delimiter.
string &code1 &code2 &code3 by '' into $codes123
                                ! Result:  AGL

```

See Also

The UNSTRING command for additional information.

The " | " concatenation operator in the Operators table under the LET command.

SUBTRACT

Syntax

```
SUBTRACT {src_num_lit/_var/_col} FROM dst_num_var[ROUND=nn]
```

Description

SUBTRACT subtracts the first value from the second and moves the result into the second field.

When dealing with money-related values (dollars and cents), use decimal variables rather than float variables. Float variables are stored as double precision floating point numbers, and small inaccuracies can appear when subtracting many numbers in succession. These inaccuracies can appear due to the way floating point numbers are represented by different hardware and software implementations.

Parameters

Parameter	Description
<i>{src_num_lit / _var / _col}</i>	Is subtracted from the contents of <i>dst_num_var</i> .
<i>dst_num_var</i>	Contains the result after execution.

Parameter	Description
ROUND	Rounds the result to the specified number of digits to the right of the decimal point. For float variables this value can be from 0 to 15. For decimal variables, this value can be from 0 to the precision of the variable. For integer variables, this argument is not appropriate.

Example

```
subtract 1 from #total      ! #total - 1
subtract &discount from #price ! #price - &discount
```

See Also

The ADD command for more information.

The LET command for information about complex arithmetic expressions.

TOC-ENTRY
Syntax

```
TOC-ENTRY
TEXT={src_txt_lit|_var|_col}
[LEVEL={level_num_lit|_var|_col}]
```

Description

Enter the text in the Table of Contents at the desired level.

Parameters

Parameter	Description
TEXT	Specifies the text to be placed in the Table of Contents.
LEVEL	Specifies the level at which to place the text. If this argument is not specified, the value of the previous level is used.

Example

```
toc-entry text = &heading
toc-entry text = &caption level=2
```

See Also

The DECLARE-TOC command.

UNSTRING

Syntax

```
UNSTRING {{src_txt_lit|_var|_col}}|{src_date_var|_col}}
BY {delim_txt_lit|_var|_col}
INTO dst_txt_var...
```

Description

Copies portions of a string into one or more text variables.

Each substring is located using the specified delimiter. The source string must not be included in the list of destination strings.

If more destination strings than substrings are found in the source strings, the extra destination strings are each set to an empty string.

If more substrings are found in the source string than in the destination strings, the extra substrings are not processed. It is up to the programmer to ensure that enough destination strings are specified.

If the source is a date variable or column, it is converted to a string according to the following rules:

- For DATETIME columns and SQR DATE variables, the format specified by the SQR_DB_DATE_FORMAT setting is used. If this has not been set, the first database-dependent format as listed in the Default Database Formats table is used.
- For DATE columns, the format specified by the SQR_DB_DATE_ONLY_FORMAT setting is used. If this has not been set, the format listed in the Default Database Formats table is used.
- For TIME columns, the format specified by the SQR_DB_TIME_ONLY_FORMAT setting is used. If this has not been set, the format as listed in the TIME Column Formats table is used.

Parameters

Parameter	Description
{src_txt_lit _var _col} {src_date_var _col}	Specifies the source field to be parsed.
delim_txt_lit _var _col	Specifies one or more characters to be used to delimit the fields within {src_txt_lit _var _col} {src_date_var _col}
Dst_txt_var	Specifies one or more destination fields to receive the results.

Example

```
unstring $show-info by ' - ' into $name $city $state $zip
```

```
unstring $cust-data by ',' into $cust_num $entry-date $total
```

See Also

The **STRING** and **EXTRACT** commands.

The **substr** and **instr** functions in the Miscellaneous Functions table under the **LET** command.

UPPERCASE

Syntax

```
UPPERCASE txt_var
```

Description

Converts a string variable to uppercase.

Parameters

Parameter	Description
<i>txt_var</i>	Specifies the field to be converted to uppercase.

Example

```
input $state 'Enter state abbreviation'
uppercase $state ! Force uppercase.
```

See Also

The upper function in the Miscellaneous Functions table under the **LET** command.

USE

Syntax

```
USE database
```

Description

Uses the named database, rather than the default database associated with your username. (SYBASE and Microsoft SQL Server only)

Use **USE** in the **SETUP** section only. When used, it must appear at the top of your report, before any queries are defined.

To reference more than one database in a program, specify secondary databases explicitly.
For example:

```
from sqdb.sqr.customers
```

You cannot issue the SYBASE or Microsoft SQL Server **USE** command from within an SQL paragraph.

Parameters

Parameter	Description
<i>database</i>	Specifies the name of the database to use.

Example

```
begin-setup
  use pubs
end-setup
```

See Also

The -DB command-line flag, described in “Invoking SQR Execute.”

USE-COLUMN

Syntax

```
USE-COLUMN {column_number_int_lit|_var|_col}
```

Description

Sets the current column.

The column must be previously defined with the COLUMNS command.

To stop printing within columns, use a column number of 0 (zero). Printing returns to normal; however, the columns remain defined for subsequent **NEXT-COLUMN** or **USE-COLUMN** commands.

Parameters

Parameter	Description
{ <i>column_number_int_lit</i> _var _col}	Specifies the number of the defined column (not the location on the page). For example, if five columns are defined, the <i>column_number_int_lit</i> _var _col can be 1 to 5.

Example

```

use-column 3      ! Print total in 3rd column.
print #total  ()  999,999
use-column 0      ! End of column printing.

```

USE-PRINTER-TYPE
Syntax

```
USE-PRINTER-TYPE printer-type
```

Description

Sets the printer type to be used for the current report.

The USE-PRINTER-TYPE command sets or alters the printer type to be used for the current report. The **USE-PRINTER-TYPE** command must appear before the first output is written to that report. If output has already been written to the report file, the USE-PRINTER-TYPE command is ignored.

Parameters

Parameter	Description
<i>printer-type</i>	Specifies the printer type to be used for the current report. See DECLARE-PRINTER for valid types.

Example

```

use-report customer_orders
use-printer-type PostScript
print (1, 1) 'Customer Name: '
print () $customer_name

```

See Also

The **DECLARE-PRINTER**, **DECLARE-REPORT**, and **USE-REPORT** commands.

USE-PROCEDURE
Syntax

```

USE-PROCEDURE
[FOR-REPORTS=(report_name1 [, report_namei] ...)]
[BEFORE-REPORT=procedure_name [(arg1 [, argi] ...)] ]
[AFTER-REPORT=procedure_name [(arg1 [, argi] ...)] ]
[BEFORE-PAGE=procedure_name [(arg1 [, argi] ...)] ]
[AFTER-PAGE=procedure_name [(arg1 [, argi] ...)] ]

```

Description

Changes the procedure usage.

The USE-PROCEDURE must be issued in the PROGRAM or PROCEDURE sections of an SQR program. **USE-PROCEDURE** is a runtime command; its compile-time equivalent is DECLARE-PROCEDURE. You can use the command as often as required to change to the necessary procedures required by the reports. If you issue multiple **USE-PROCEDURE** commands, each remains in effect for that report until altered by another USE-PROCEDURE command for that report. In this way, you can use one to change common procedures for ALL reports and others to change unique procedures for individual reports. The referenced procedures can accept arguments.

If no **FOR-REPORTS** is specified, ALL is assumed. Initially, the default for each of the four procedure types is **NONE**. If a procedure is defined in one DECLARE-PROCEDURE for a report, that procedure is used unless **NONE** is specified.

You can change the BEFORE-REPORT only before the first output is written to that report, because that causes the **BEFORE-REPORT** procedure to be executed.

Parameters

Parameter	Description
FOR-REPORTS	Specifies the reports that are to use these procedures. This argument is required only for a program with multiple reports. If you are writing a program that produces a single report, you can ignore this argument.
BEFORE-REPORT	Specifies a procedure to execute at the time of execution of the first command, which causes output to be generated. You can use the command, for example, to create a report heading.
AFTER-REPORT	Specifies a procedure to execute just before the report file is closed at the end of the report. This argument can be used to print totals or other closing summary information. If no report was generated, the procedure does not execute.
BEFORE-PAGE	Specifies a procedure to execute at the beginning of every page, just before the first output command for the page. It can be used, for example, to set up page totals.
AFTER-PAGE	Specifies a procedure to execute just before each page is written to the file. This argument can be used, for example, to display page totals. You can also specify arguments to be passed to the procedure. Arguments can be any variable, column, or literal.

Example

```

use-procedure                                ! These procedures will

for-reports=(all)                           ! be used by all reports

before-report=report_heading

after-report=report_footing
use-procedure                                ! These procedures will

for-reports=(customer)                      ! be used by the customer

before-page=page_setup                      ! report

after-page=page_totals
use-procedure                                ! The after-report

for-reports=(summary)                      ! procedure will be

after-report=none                          ! disabled for the

                                           ! summary report

```

See Also

The DECLARE-PROCEDURE command.

USE-REPORT

Syntax

```
USE-REPORT {report_name_lit|_var|_col}
```

Description

For programs with multiple reports, enables the user to switch between reports.

The USE-REPORT command specifies to which report files the subsequent report output is to be written. An application can contain several **USE- REPORT** statements to control several reports.

You must specify the report name and report characteristics in a DECLARE- REPORT paragraph and in the associated **DECLARE-LAYOUT** and DECLARE- PRINTER paragraphs.

Parameters

Parameter	Description
<code>{report_name_it/_var/_col}</code>	Specifies the report to become the "current" report. All subsequent PRINT and PRINT-DIRECT statements are written to this report until the next USE-REPORT is encountered.

Example

```

use-report customer_orders
use-printer-type PostScript
print (1, 1) 'Customer Name: '
print () $customer_name

```

See Also

The **DECLARE-REPORT**, **DECLARE-LAYOUT**, **DECLARE-PRINTER**, and **USE-PRINTER-TYPE** commands.

WHILE
Syntax

```
WHILE logical_expression
```

The general format of a WHILE command is as follows:

```

WHILE logical_expression
SQR_commands...
[BREAK]
SQR_commands...
END-WHILE

```

Description

Begins a **WHILE ... END-WHILE** loop.

The WHILE loop continues until the condition being tested is FALSE.

An expression returning 0 (zero) is considered FALSE; an expression returning nonzero is TRUE.

BREAK causes an immediate exit of the WHILE loop; SQR continues with the command immediately following END-WHILE.

WHILE commands can be nested to any level and can include or be included within **IF** and **EVALUATE** commands.

Parameters

Parameter	Description
<i>logical_expression</i>	A valid logical expression. See the LET command for a description of logical expressions.

Example

The following example shows an **IF** nested within a **WHILE**:

```
while #count < 50
  do get_statistics
  if #stat_count = 100
    break      ! Exit WHILE loop.
  end-if
  add 1 to #count
end-while
```

You can use single numeric variables in your expression to make your program more readable, for example when using flags:

```
move 1 to #have_data
...
while #have_data
  ...processing...
end-while
```

The following example sets up an infinite loop:

```
while 1
  ...processing...
  if ...
    break      ! Exit loop
  end-if
end-while
```

Any complex expression can be used in the **WHILE** command, as shown in the following example:

```
while #count < 100 and (not #end-file or isnull(&state))
  ...
end-while
```

See Also

The LET command for a description of expressions.

WRITE

Syntax

```
WRITE {filenum_lit|_var|_col} FROM
{{ {txt_lit|_var|_col} | {date_var|_col} | num_col }
[:len_int_lit] } | {num_lit|_var:len_int_lit} } ...
[STATUS=status_num_var]
```

Description

Writes a record to a file from data stored in variables, columns, or literals.

The file must already be opened for writing.

If length is specified, the variable is either truncated at that length or padded with spaces to that length. If length is not specified (for string variables or database columns), the current length of the variable is used.

When writing numeric variables, the length argument is required. Only 1, 2, or 4 byte binary integers are written. Floating point values are not supported directly in the WRITE command. However, you can first convert floating point numbers to strings and then write the string.

When writing binary data the file must be open using the **FIXED** or **FIXED-NOLF** qualifiers. The file is not portable across platforms because binary numbers are represented differently.

When writing a date variable or column, the date is converted to a string according to the following rules:

- For DATETIME columns and SQR DATE variables, the format specified by the **SQR_DB_DATE_FORMAT** setting is used. If this has not been set, the first database-dependent format as listed in the Default Database Formats table is used.
- For DATE columns, the format specified by the **SQR_DB_DATE_ONLY_FORMAT** setting is used. If this has not been set, the format listed in the DATE Column Formats table is used.
- For TIME columns, the format specified by the **SQR_DB_TIME_ONLY_FORMAT** setting is used. If this has not been set, the format as listed in the TIME Column Formats table is used.

Text literals take the length of the literal.

Files opened for writing are treated as having variable-length records. If you need a fixed-length record, specify a length for each variable written to the file.

The total length of the variables and literals being written must not be greater (but can be less) than the record length specified when the file was opened. Records are not padded, but are written with the total length of all variables in the WRITE command.

If STATUS is specified, SQR returns 0 if the write is successful; otherwise, it returns the value of *errno*, which is system-dependent.

Parameters

Parameter	Description
<i>filenum_lit/_var /_col</i>	Specifies the number assigned in the OPEN command to the file to be written.
<i>{{txt_lit/_var/_col} {date_var/_col}/num_col}</i> <i>[:len_int_lit] {num_lit/_var:len_int_lit}</i>	Specifies one or more variables to be written. <i>len_int_lit</i> specifies the length of each field of data.
STATUS	Specifies an optional variable into which a write status is returned.

Example

```

write 5 from $name:20 $city:15 $state:2
write 17 from $company ' - ' $city ' - ' $state ' ' $zip
write #j2 from #rate:2 #amount:4 #quantity:1
move #total to $tot 99999.99                ! Convert floating point to
                                           ! string.

write 1 from $tot
let $date1 = datenow()                      ! Put the current date and time
                                           ! into DATE variable

write 3 from $date1:20

```

See Also

The OPEN, CLOSE, and READ commands.

CHAPTER 3

Generating HTML Output

This section describes the procedures that enable SQR for PeopleSoft to generate HTML output. You can publish the output on an internet, intranet, or extranet website. An SQR program without HTML procedures has limited HTML capabilities. Adding HTML procedures to the SQR program enhances the appearance of the HTML output.

This chapter describes the following HTML procedures:

- General purpose
- Headings
- Highlighting
- Hypertext links
- Lists
- Tables

See Also

SQR for PeopleSoft Developer's Guide, “Working With HTML” for details on using these procedures.

HTML General Purpose Procedures

The following table lists the HTML general purpose procedures.

Procedure	Description
html_br	Produces the specified number of line breaks in a paragraph using the HTML
 tag. The paragraph continues onto the next line. Syntax: html_br(number count, string attributes) count = the number of HTML
 tags that are inserted. attributes = the HTML attributes that are incorporated inside the HTML
 tag. Example: Producing a line break: <pre>print 'Here is some text' () do html_br(3, '') print 'Here is some three lines down' ()</pre>
html_center	Marks the start of text to be centered on the HTML document using the HTML <CENTER> tag. You can also accomplish this using the SQR print statement with CENTER specified in the code. Syntax: html_center() attributes = the HTML attributes that are incorporated inside the HTML <CENTER> tag. Example: Centering text using the <CENTER> tag: <pre>do html_center('') print 'Here is some text' () do html_center_end</pre>
html_center_end	Marks the end of text previously specified to be centered. Syntax: html_center_end
html_hr	Produces a horizontal divider between sections of text using the HTML <HR> tag. Syntax: html_hr(string attributes) attributes = the HTML attributes that are incorporated inside the HTML <HR> tag. Example: Producing a horizontal divider: <pre>print 'Here is some text' () do html_hr('') print 'And some more text' ()</pre>

Procedure	Description
html_img	Inserts an image using the HTML tag. This can also be done using the command PRINT-IMAGE; however, the procedure html_img enables you to specify the full set of available HTML attributes. Syntax: html_img(string attributes) attributes = the HTML attributes that are incorporated inside the HTML tag. Some common attributes: src = URL of the image to be inserted. (Ex: src=/images/abc.gif) height = height of the image in pixels. (Ex: height=200) width = width of the image in pixels. (Ex: width=400) Example: Producing an image: <pre>do html_img('src="/images/stop.gif"')</pre>
html_nobr	Marks the start of text that cannot be wrapped using the HTML <NOBR> tag. Syntax: html_nobr Example: Preventing line wrapping: <pre>do html_nobr('') print 'Here's long text that should not wrap' () do html_nobr_end</pre>
html_nobr_end	Marks the end of text that cannot be wrapped using the HTML </NOBR> tag. Syntax: html_nobr_end
html_on	Turns on the HTML procedures. Call this procedure at the start of an SQR program; otherwise, the HTML procedures are not turned on. After the HTML procedures are turned on, the appearance of the web page must be formatted using the various HTML procedures. Syntax: html_on Example: Turning on the HTML procedures: <pre>do html_on</pre>

Procedure	Description
html_p	Marks the start of a new paragraph using the HTML <P> tag. Syntax: html_p(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <P> tag. Some common attributes: align = left right center Specifies the alignment of the paragraph Example: Displaying one paragraph with right-aligned text, then another paragraph with normal text: <pre>do html_p('ALIGN=RIGHT') print 'Right aligned text' (1,1) do html_p_end print 'Normal text' (+1,1)</pre>
html_p_end	Marks the end of a paragraph using the HTML </P> tag. The end of a paragraph is typically implied and not needed; however, we recommend that you specify it for completeness. Syntax: html_p_end
html_set_body_attributes	Specifies the attributes that are incorporated into the HTML <BODY> tag. This must be called at the start of the SQR program. Syntax: html_set_body_attributes(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <BODY> tag. Some common attributes: background = Specifies the image to display is the background of the web page. (Ex: background=/images/logo.gif) bgcolor=#rrggbb Specifies the background color of the Web page. (Ex: bgcolor=#80FFF) Example: Using BACKGROUND attribute to display image marble.gif in the background of the web page: <pre>do html_set_body_attributes('BACKGROUND="/images/x.gif"')</pre>

Procedure	Description
html_set_head_tags	Specifies the tags that are incorporated between the <HEAD> and </HEAD> HTML tags. By default these are empty by default. One common tag to set is <TITLE>, which specifies the title to display for the Web page. This must be called at the start of the SQR program. Syntax: html_set_head_tags(string attributes) attributes = defines the HTML attributes that are incorporated between the <HEAD> and </HEAD> HTML tags. Example: Displaying the title My Report for the Web page: <pre>do html_set_head_tags('<TITLE>My Report</TITLE>')</pre>

HTML Heading Procedures

The following table lists the HTML heading procedures.

Procedure	Description
html_h1	Marks the start of heading level one text using the HTML <H1> tag. Text under this heading is displayed more prominently than that of heading level two. Syntax: html_h1(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <H1> tag. Example: Displaying text as a heading level one: <pre>do html_h1('') print 'This is a heading' () do html_h1_end</pre>
html_h1_end	Marks the end of heading level one text using the HTML </H1> tag. Syntax: html_h1_end

Procedure	Description
html_h2	Marks the start of heading level two text using the HTML <H2> tag. Text under this heading is displayed less prominently than that of heading level one and more prominently than that of heading level three. Syntax: html_h2(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <H2> tag. Example: Displaying text as a heading level two: <pre>do html_h2('') print 'This is a heading' () do html_h2_end</pre>
html_h2_end	Marks the end of heading level two text using the HTML </H2> tag. Syntax: html_h2_end
html_h3	Marks the start of heading level three text using the HTML <H3> tag. Text under this heading displays less prominently than that of heading level two and more prominently than that of heading level four. Note. This heading level is the default value. Syntax: html_h3(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <H3> tag.
html_h3_end	Marks the end of heading level three text using the HTML </H3> tag. Syntax: html_h3_end
html_h4	Marks the start of heading level four text using the HTML <H4> tag. Text under this heading displays less prominently than that of heading level three and more prominently than that of heading level five. Syntax: html_h4(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <H4> tag.
html_h4_end	Marks the end of heading level four text using the HTML </H4> tag. Syntax: html_h4_end

Procedure	Description
html_h5	Marks the start of heading level five text using the HTML <H5> tag. Text under this heading displays less prominently than that of heading level four and more prominently than that of heading level six. Syntax: html_h5(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <H5> tag.
html_h5_end	Marks the end of heading level five text using the HTML </H5> tag. Syntax: html_h5_end
html_h6	Marks the start of heading level six text using the HTML <H6> tag. Text under this heading displays less prominently than that of heading level five and more prominently than that of heading level seven. Syntax: html_h6(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <H6> tag.
html_h6_end	Marks the end of heading level six text using the HTML </H6> tag. Syntax: html_h6_end

HTML Highlighting Procedures

The following table lists the HTML highlighting procedures.

Procedure	Description
html_blink	Marks the start of blinking style text using the HTML <BLINK> tag. Syntax: html_blink(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <BLINK> tag. Example: Displaying text with the address style: <pre>do html_blink('') print 'This is blinking' () do html_blink_end</pre>
html_blink_end	Marks the end of blinking style using the HTML </BLINK> tag. Syntax: html_blink_end

Procedure	Description
html_cite	Marks the start of citation style text using the HTML <CITE> tag. Syntax: html_cite(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <CITE> tag. Example: Displaying text with the citation style: <pre>do html_cite('') print 'This is a citation' () do html_cite_end</pre>
html_cite_end	Marks the end of citation style text using the HTML </CITE> tag. Syntax: html_cite_end
html_code	Marks the start of code style text using the HTML <CODE> tag. Syntax: html_code(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <CODE> tag. Example: Displaying text with the code style: <pre>do html_code('') print 'Here is code' () do html_code_end</pre>
html_code_end	Marks the end of code style text using the HTML </CODE> tag. Syntax: html_code_end
html_kbd	Marks the start of keyboard input style text using the HTML <KBD> tag. Syntax: html_kbd(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <KBD> tag. Example: Displaying text with the keyboard style: <pre>do html_kbd('') print 'Here is keyboard' () do html_kbd_end</pre>
html_kbd_end	Marks the end of keyboard style text using the HTML </KBD> tag. Syntax: html_kbd_end

Procedure	Description
html_samp	Marks the start of sample style text using the HTML <SAMP> tag. Syntax: html_samp(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <SAMP> tag. Example: Displaying text with the sample style: <pre>do html_samp('') print 'Here is sample' () do html_samp_end</pre>
html_samp_end	Marks the end of sample style text using the HTML </SAMP> tag. Syntax: html_samp_end
html_strike	Marks the start of strike-through style text using the HTML <STRIKE> tag. Syntax: html_strike(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <STRIKE> tag. Example: Displaying text with the strike-through style: <pre>do html_strike('') print 'Here is strike-through' () do html_strike_end</pre>
html_strike_end	Marks the end of strike-through style text using the HTML </STRIKE> tag. Syntax: html_strike_end
html_sub	Marks the start of subscript style text using the HTML <SUB> tag. Syntax: <i>html_sub(string attributes)</i> attributes = defines the HTML attributes that are incorporated inside the HTML <SUB> tag. Example: Displaying text with the subscript style: <pre>print 'Here is' () do html_sub('') print 'subscript text' () do html_sub_end</pre>
html_sub_end	Marks the end of subscript style text using the HTML </SUB> tag. Syntax: html_sub_end

<i>Procedure</i>	<i>Description</i>
html_sup	Marks the start of superscript style text using the HTML <SUP> tag. Syntax: html_sup(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <SUP> tag. Example: Displaying text with the superscript style: <pre>print 'Here is' () do html_sup('') print 'superscript text' () do html_sup_end</pre>
html_sup_end	Marks the end of superscript style text using the HTML </SUP> tag. Syntax: html_sup_end

HTML Hypertext Link Procedures

The following table lists the HTML hypertext link procedures.

Procedure	Description
html_a	Marks the start of a hypertext link using the HTML <A> tag. When the user clicks the area with the hypertext link, the web browser switches to the specified HTML document. Syntax: html_a(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <A> tag. At a minimum, the HREF attribute should be defined, which specifies the URL of an HTML document. Some common attributes: href = where the hypertext link points to. (Ex: href=home.html) name = an anchor to which a hypertext link can point. (Ex: name=marker1) Example: Creating an anchor with two hypertext links. The anchor is positioned at the top of the document. The first hypertext link points to the HTML document otherdoc.html. The second hypertext link points to the anchor named TOP. <pre>do html_a('NAME=TOP') do html_a_end print 'At the top of document' () do html_br(20, '') do html_a('HREF=otherdoc.html') print 'Goto other document' () do html_a_end do html_p('') do html_a('HREF=#TOP') print 'Goto top of document' () do html_a_end</pre>
html_a_end	Marks the end of a hypertext link using the HTML tag. Syntax: html_a_end

HTML List Procedures

The following table lists the HTML list procedures.

Procedure	Description
html_dd	Marks the start of a definition in a definition list using the HTML <DD> tag. Syntax: html_dd(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <DD> tag.
html_dd_end	Marks the end of a definition in a definition list using the HTML </DD> tag. The end of a definition in a definition list is typically implied and not needed; however, we recommend that you specify it for completeness. Syntax: html_dd_end
html_dir	Marks the start of a directory list using the HTML <DIR> tag. Syntax: html_dir(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <DIR> tag. Example: Displaying an ordered list with three items: <pre>do html_dir('') do html_li('') print 'First item' () do html_li('') print 'Second item' () do html_li('') print 'Last item' () do html_dir_end</pre>
html_dir_end	Marks the end of a directory using the HTML </DIR> tag. Syntax: html_dir_end

Procedure	Description
html_dl	Marks the start of a definition list using the HTML <DL> tag. A definition list displays a list of terms and definitions. The term displays above and to the left of the definition. Use the procedure html_dt to display a term. Use the procedure html_dd to display a definition. Syntax: html_dl(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <DL> tag. Example: Displaying a definition list with two terms and definitions: <pre>do html_dl('') do html_dt('') print 'A Daisy' () do html_dd('') print 'A sweet and innocent flower.' () do html_dt('') print 'A Rose' () do html_dd('') print 'A very passionate flower.' () do html_dl_end</pre>
html_dl_end	Marks the end of a definition list using the HTML </DL> tag. Syntax: html_dl_end
html_dt	Marks the start of a term in a definition list using the HTML <DT> tag. Syntax: html_dt(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <DT> tag.
html_dt_end	Marks the end of a term in a definition list using the HTML </DT> tag. Syntax: html_dt_end
html_li	Marks the start of a list item using the HTML tag. Syntax: html_li(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML tag.

Procedure	Description
html_li_end	Marks the end of a list item using the HTML tag. The end of a list item is typically implied and not needed; however, we recommend that you specify it for completeness. Syntax: html_li_end
html_menu	Marks the start of a menu using the HTML <MENU> tag. Use the procedure html_li to identify each item in the list. Syntax: html_menu(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <MENU> tag. Example: Displaying an ordered list with three items: <pre>do html_menu('') do html_li('') print 'First item' () do html_li('') print 'Second item' () do html_li('') print 'Last item' () do html_menu_end</pre>
html_menu_end	Marks the end of a menu using the HTML </MENU> tag. Syntax: html_menu_end

Procedure	Description
html_ol	Marks the start of an ordered list using the HTML tag. Each item in the list typically displays indented to the right with a number to the left. Use the procedure html_li to identify each item in the list. Syntax: html_ol(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML tag. Example: Displaying an ordered list with three items: <pre>do html_ol('') do html_li('') print 'First item' () do html_li('') print 'Second item' () do html_li('') print 'Last item' () do html_ol_end</pre>
html_ol_end	Marks the end of an ordered list using the HTML tag. Syntax: html_ol_end

Procedure	Description
html_ul	Marks the start of an unordered list using the HTML tag. Each item in the list typically is displayed indented to the right with a bullet to the left. Use the procedure html_li to identify each item in the list. Syntax: html_ul(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML tag. Example: Displaying an ordered list with three items: <pre>do html_ul('') do html_li('') print 'First item' () do html_li('') print 'Second item' () do html_li('') print 'Last item' () do html_ul_end</pre>
html_ul_end	Marks the end of an unordered list using the HTML tag. Syntax: html_ul_end

HTML Table Procedures

The following table lists the HTML table procedures.

Procedure	Description
html_caption	Marks the start of a table caption using the HTML <CAPTION> tag. Syntax: html_caption(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <CAPTION> tag.
html_caption_end	Marks the end of a table caption using the HTML </CAPTION> tag. The end of a table caption is typically implied and not needed; however, we recommend that specify it for completeness. Syntax: html_caption_end

<i>Procedure</i>	<i>Description</i>
html_table	Marks the start of a table using the HTML <TABLE> tag. Syntax: html_table(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <TABLE> tag. Some common attributes: border = Specifies that a border is displayed around each cell of the table. width = Specifies the width of the entire table in pixels. cols = Specifies the number of columns in the table. (Ex: COLS=4) Example: Displaying database records in a tabular format. The procedures html_caption_end, html_tr_end, html_td_end, and html_th_end are used for completeness; however, they are typically implied and not needed.

Procedure	Description
html_table (Continued)	<pre> ! start the table & display the column headings do html_table('border') do html_caption("") print 'Customer Records' (1,1) do html_caption_end do html_tr("") do html_th("") print 'Cust No' (+1,1) do html_th_end do html_th("") print 'Name" (,10) do html_th_end do html_tr_end ! display each record begin-select do html_tr("") do html_td("") cust_num (1,1,6) edit 099999 do html_td_end do html_td("") name (1,10,25) do html_td_end do html_tr_end next-listing skiplines=1 need=1 from customers end-select ! end the table do html_table_end </pre>
html_table_end	Marks the end of a table using the HTML </TABLE> tag. Syntax: html_table_end

Procedure	Description
html_td	Marks the start of a new column in a table row using the HTML <TD> tag. This specifies that the text that follows is displayed in the column. Syntax: html_td(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <TD> tag.
html_td_end	Marks the end of a column in a table using the HTML </TD> tag. The end of a column is typically implied and not needed; however, it is a good idea to specify it for completeness. Syntax: html_td_end
html_th	Marks the start of a new column header in a table row using the HTML <TH> tag. This specifies that the text that follows is displayed as the header of the column. Syntax: html_th(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <TH> tag.
html_th_end	Marks the end of a column header in a table using the HTML </TH> tag. The end of a column header is typically implied and not needed; however, we recommend that you specify it for completeness. Syntax: html_th_end
html_tr	Marks the start of a new row in a table using the HTML <TR> tag. Syntax: html_tr(string attributes) attributes = defines the HTML attributes that are incorporated inside the HTML <TR> tag.
html_tr_end	Marks the end of a list item using the HTML </TR> tag. The end of a column in a table is typically implied and not needed; however, we recommend that you specify it for completeness. Syntax: html_tr_end

CHAPTER 4

Invoking SQR Execute

This chapter describes how to invoke SQR Execute. SQR Execute is a runtime program that enables you to run a previously compiled SQR program.

Running SQR Execute

To begin running SQR Execute, enter the following command. (If you are running under Microsoft Windows, invoke SQRWT rather than SQR.)

```
SQR [program] [connectivity] [flags...] [args...] [@file...]
```

SQR Execute Flags

The following table describes *flags*. The table in the “SQR Command-Line Arguments” section describes *program*, *connectivity*, *args*, and *@file*.

Flag	Description
-A	Appends the output to an existing output file of the same name. If the file does not exist, it creates a new one. This is useful when you want to run the same report more than once, but want to create only a single output file.
-C	(Microsoft Windows) Specifies that the Cancel dialog box appears while the program is running so you can easily terminate the program execution.
-BURST:{xx}	Specifies the type of bursting to be performed. -BURST:T generates the Table of Contents file only. -BURST:S generates the report output according to the symbolic Table of Contents entries set in the program with the TOC- ENTRY command's "level" argument. In -BURST:S[{l}], {l} is the level at which to burst upon. The setting -BURST:S is equivalent to -BURST:S1 -BURST:P generates the report output by report page numbers. In -BURST:P[{l} , {s} [, {s}] ...] , {l} is the number of logical report pages that each .htm file contains and {s} is the page selection: {n}, {n}-{m}, -{m}, or {n}-. The setting -BURST:P is equivalent to -BURST:P0,1- when using -PRINTER:HT or -BURST:P1 when using -PRINTER:EH. Note. -BURST:P and -BURST:S require -PRINTER:EH or -PRINTER:HT. The Page range selection feature of -BURST:P requires -PRINTER:HT. -BURST:T requires -PRINTER:HT

Flag	Description
-CB	(Windows) Forces the communication box.
-Dnn	(Non-Microsoft Windows) Causes SQR to display the report output on the terminal at the same time it is being written to the output file. nn is the maximum number of lines to display before pausing. If no number is entered after -D, the display scrolls continuously. The printer type must be LP; otherwise, SQR does not display any output. If the program is producing more than one report, the display is for the first report only.
-DBdatabase	(SYBASE) Causes the SQR program to use the specified database, overriding any USE command in the SQR program.
-E[file]	Directs error messages to the named file, or to the default file program.err. If no errors occur, no file is created.
-EH_APPLETS:dir	Specifies the directory location of the Enhanced HTML applets. The default directory for these applets is IMAGES. Note. The following 13 flags are applicable only when either the -PRINTER:EH or -PRINTER:EP flag is specified.
-EH_BQD	Generates a {report}.bqd file from the report data. Also associates a BQD icon with {report}.bqd in the navigation bar.
-EH_BQD:file	Associates the BQD icon with the specified file.
-EH_BROWSER:xx	Specifies the target browser. When set to ALL, SQR auto-determines which browser is being used, invokes a browser-specific file, and generates HTML designed for that browser. When set to BASIC, SQR generates HTML suitable for all browsers. When set to IE, SQR generates HTML designed for Internet Explorer. When set to NETSCAPE, SQR generates HTML designed for Netscape.
-EH_CSV	Generates a {report}.csv file from the report data.
-EH_CSV:file	Associates the CSV icon with the specified file.
-EH_CSVONLY	Creates a CSV file, but does not create an HTML file.
-EH_FULLHTML:xx	Switches between HTML 3.0 and HTML 3.2. When set to TRUE, SQR generates HTML 3.2. When set to FALSE, SQR generates HTML 3.0
-EH_Icons:dir	Specifies the directory in which the HTML should find the referenced icons.
-EH_LANGUAGE:xx	Sets the language used for the HTML navigation bar. You can specify English, French, German, Portuguese, or Spanish.
-EH_PDF	Associates a PDF icon with {report}.pdf in the navigation bar.
-EH_Scale:{nn}	Sets the scaling factor from 50 to 200.

Flag	Description
-F[library file(member)]	(AS400) Overrides the default output file name library/lis(program). The default action places the LIS in a source file called LIS in the same library as the member {program}. To use the current library, specify -F without an argument. To change the name of the output file, specify -F with the new name. If the new name does not specify a library, the file is created in the current library. The output file is not created until data is actually printed on the page. If no data is printed, no output file is created. To specify an alternate filename: -FSQRDIR/LIS(file)
-F[file directory]	Overrides the default output file name, program.lis. The default action places the program.lis in the same directory as the program.sqr file. To use the current directory, specify -F without an argument. To change the name of the output file, specify -F with the new name. If the new name does not specify a directory, the file is created in the current directory. The output file is not created until data is actually printed on the page. If no data is printed, no output file is created. Specify the file name and directory for different operating systems as follows:
	Operating System
	VMS
	UNIX
	MVS
-Gfile_mode	(VM) Specifies the file mode to use when the report output file is created. See the VM C Library manual, "afopen" function, for a complete description of all the valid keywords and values.
-Gfile_attributes	(VMS, OpenVMS) Specifies the file attributes to use when the report output file is created. You can specify up to 10 sets of attributes, separated by commas (.). See the VAX C Library manual, "creat" function, for a complete description of all the valid keywords and values.
-GPRINT=YES NO	(MVS) -GPRINT=YES causes SQR's report output file to have ANSI control characters written to the first column of each record of the file.
-ID	Displays the copyright banner on the console.
-KEEP	In addition to LIS files, creates an SPF file for each report that your program generates.
-NOLIS	Prevents the creation of LIS files. Instead, SPF files are created.
-O[file]	Directs log messages to the specified file or to programt.log if no file is specified. By default, the sqr.log file is used in the current working directory.
-P	(MVS, AS400) Suppresses printer control characters from column 1.
-PB	(Informix) Causes column data to retain trailing blanks.
-PRINTER:xx	Causes printer type xx to be used when creating output files.
	xx
	EH

Flag	Description
	EP
	HP
	HT
	LP
	PD
	PS
	WP
	Types LP, HP, and PS produce files with the .lis extension. Types EH and HT produce .htm file output. Type HT produces version 2.0 HTML files with the report content inside of <PRE></PRE> tags. Type EH produces reports in which content is fully formatted with version 3.0 or 3.2 HTML tags. On Microsoft Windows systems, the WP extension sends the output to the default Microsoft Windows printer. To specify a non-default Microsoft Windows printer, enter -PRINTER:WP:{Printer Name}. The {Printer Name} is the name assigned to your printer. For example, to send output to a Microsoft Windows printer named NewPrinter, use -PRINTER:WP:NewPrinter. If your printer name has spaces, enclose the entire argument in quotes. If you also want to create an .SPF file, use -KEEP.
-S	Requests that the status of all cursors be displayed at the end of the report run. Status includes the text of each SQL statement, number of times each was compiled and executed, and the total number of rows selected.
-Tnn	Specifies that you want to test your report for nn pages. SQR ignores All ORDER BY clauses in SELECT statements to save time during testing. If the program is producing more than one report, SQR stops after the specified number of pages defined for the first report have been printed.
-T{B}	(SYBASE CT-Lib, ODBC) Trims trailing blanks from database character columns. (MVS/DB2, DB2/400) Prevents SQR from removing trailing blanks from database character columns.
-T{B Z BZ ZB}	(MVS/DB2) -TB prevents SQR from removing trailing blanks from database character columns. -TZ prevents SQR from removing trailing zeros from the decimal portion of numeric columns. -TBZ or -TZB prevents both. (AS400/DB2) -TB trims trailing blanks from database character columns. -TZ trims trailing zeros from the decimal portion of numeric columns. -TBZ or -TZB is legal.
-Vserver	(SYBASE) Uses the named server.
-XB	(Non-Microsoft Windows) Suppresses the SQR banner and the "SQR... End of Run" message.
-XCB	(Microsoft Windows) Do not use the communication box.

Flag	Description
-XL	Prevents SQR from logging on to the database. Programs run in this mode cannot contain any SQL statements. -XL enables you to run SQR without accessing the database. You still must supply at least an empty slash (/) on the command line as a placeholder for the connectivity information. For example: <code>sqr myprog / -xl</code> Some database files must be available for SQR to run whether SQR logs onto the database. See information about your particular operating system and database to determine which files you need.
-XMB	(Microsoft Windows) Disables the error message display so that a program can be run without interruption by error message boxes. Error messages are only sent to an .err file. See the -E flag for more information.
-XNAV	Prevents SQR from creating the navigation bar in .HTM files generated with -PRINTER:HT. This occurs when only a single .HTM file is produced. Multiple .HTM files generated from a single report always contain the navigation bar.
-XP	(SYBASE DBLib) Prevents SQR from creating temporary stored procedures. See “BEGIN-SELECT” for more information.
-XTB	Preserves the trailing blanks in an .lis file at the end of a line.
-XTOC	Prevents SQR from generating the Table of Contents for the report. SQR ignores this flag when either -PRINTER:EH or -PRINTER:HT is also specified.
-ZIF {file}	Sets the full path and name of the SQR initialization file, SQR.INI.
-ZIV	Invokes the SPF Viewer after generating program.spf file. This flag implicitly invokes the -KEEP flag to create program.spf. In case of multiple output files, only the first report file is passed to the Viewer.
-ZMF {file}	Specifies the full path and name of the SQR error message file, sqrrr.dat.
-ZRF {file}	Sets the full path and name of an alternate registry.properties file. The following is a common default path to the registry.properties file on an NT system: <code>c:\peoplesoft\properties\registry.properties</code> The registry.properties file lists Data Sources that SQR Server can access.

See Also

“Understanding SQR for PeopleSoft,” Understanding the SQR Command Line.

CHAPTER 5

Using SQR Print

This chapter describes SQR Print and discusses how to:

- Generate output from the command line.
- Use SQR print command-line flags.
- Generate output in Microsoft Windows.

Understanding SQR Print

SQR Print enables you to create printer-specific reports for any of the file types supported by SQR. SQR Print converts portable printer-independent files (SPF) into printer-specific files. SQR and SQR* create SPF files when you use the **-KEEP** and **-NOLIS** command line flags.

Generating Output from the Command Line

To begin running SQR Print, enter the following command. (If you are in Microsoft Windows, invoke SQRWP rather than SQRP.)

```
SQRP [spf-file] [flags...]
```

The following table describes *spf-file* (SPF file) and *flags*.

SQR Print writes an LIS file with the same name as the SPF file, but with the LIS extension. You can override this name with the **-F** command line flag.

The **-PRINTER** command line flag specifies printer type. SQR offers several options:

- Line Printer
- HP LaserJet
- PostScript
- HTML
- Enhanced HTML
- ADOBE PDF

- Enhanced HTML & ADOBE PDF

If your report contains graphics and you select Line Printer, SQR Print ignores graphic elements such as lines, boxes, and charts and prints only text.

SQR Print Command-Line Flags

The following table describes the command-line flags of SQR Print.

Flag	Description
-A	Appends the output to an existing output file of the same name. If the file does not exist, it creates a new one. This is useful when you want to run the same report more than once, but want to create only a single output file.
-BURST:{xx}	Specifies the type of bursting to be performed. -BURST:T generates the Table of Contents file only. -BURST:S generates the report output according to the symbolic Table of Contents entries set in the program with the TOC-ENTRY command's "level" argument. In -BURST:S[{l}], {l} is the level at which to burst upon. The setting -BURST:S is equivalent to -BURST:S1. -BURST:P generates the report output by report page numbers. In -BURST:P[{l} , {s} [, {s}] ...] , {l} is the number of logical report pages that each .HTM file contains and {s} is the page selection: {n}, {n}-{m}, -{m}, or {n}-. The setting -BURST:P is equivalent to -BURST:P0,1- when using -PRINTER:HT or -BURST:P1 when using -PRINTER:EH. Note. -BURST:P and -BURST:S require -PRINTER:EH or -PRINTER:HT. The Page range selection feature of -BURST:P requires -PRINTER:HT. -BURST:T requires -PRINTER:HT.
-Dnn	(Non-Microsoft Windows) Displays the report output on the terminal at the same time it is being written to the output file. nn is the maximum number of lines to display before pausing. If no number is entered after -D, the display scrolls continuously. The printer type must be LP or the display is ignored. If the program is producing more than one report, the display is for the first report only.
-E[file]	Directs error messages to the named file, or to the default file program.err. If no errors occur, no file is created. Note. This following 12 flags are applicable only when either the -PRINTER:EH or -PRINTER:EP flag is specified.
-EH_APPLETS:dir	Specifies the directory location of the Enhanced HTML applets. The default directory for these applets is IMAGES.
-EH_BQD	Generates a {report}.bqd file from the report data. Also associates a BQD icon with {report}.bqd in the navigation bar.
-EH_BQD:file	Associates the BQD icon with the specified file.

Flag	Description
-EH_BROWSER :xx	Specifies the target browser. When set to ALL, SQR auto-determines which browser is being used, invokes a browser-specific file, and generates HTML designed for that browser. When set to BASIC, SQR generates HTML suitable for all browsers. When set to IE, SQR generates HTML designed for Internet Explorer. When set to NETSCAPE, SQR generates HTML designed for Netscape.
-EH_CSV	Generates a {report}.csv file from the report data.
-EH_CSV:file	Associates the CSV icon with the specified file.
-EH_CSVONLY	Creates a CSV file but does not create an HTML file.
-EH_FULLHTML L:xx	Switches between HTML 3.0 and HTML 3.2. When set to TRUE, SQR generates HTML 3.2. When set to FALSE, SQR generates HTML 3.0
-EH_Icons:dir	Specifies the directory in which the HTML should find the referenced icons.
-EH_LANGU GE:xx	Sets the language used for the HTML navigation bar. You can specify English, French, German, Portuguese, or Spanish.
-EH_PDF	Associates a PDF icon with {report}.pdf in the navigation bar.
-EH_Scale:{nn}	Sets the scaling factor from 50 to 200.
-F[library file(member)]	(AS400) Overrides the default output file name library/lis(program). The default action places the lis in a source file called lis in the same library as the member {program}. To use the current library, specify -F without an argument. To change the name of the output file, specify -F with the new name. If the new name does not specify a library, the file is created in the current library. The output file is not created until data is actually printed on the page. If no data is printed, no output file is created. To specify an alternate filename: -FSQRDIR/LIS(file)
-F[file directory]	Overrides the default output file name, program.lis. The default action places the program.lis in the same directory as the program. sqr file. To use the current directory, specify -F without an argument. To change the name of the output file, specify -F with the new name. If the new name does not specify a directory, the file is created in the current directory. The output file is not created until data is actually printed on the page. If no data is printed, no output file is created. Specify the file name and directory for different operating systems according to the following:
	Operating System
	VMS
	UNIX
	MVS
-Gfile_mode	(VM) Specifies the file mode to use when the report output file is created. See the VM C Library manual, "afopen" function, for a complete description of all the valid keywords and values.

Flag	Description
-Gfile_attributes	(VMS, OpenVMS) Specifies the file attributes to use when the report output file is created. You can specify up to 10 sets of attributes, separated by commas (.). See the VAX C Library manual, "creat" function, for a complete description of all the valid keywords and values.
-ID	Displays the copyright banner on the console.
-O[file]	Directs log messages to the specified file or to programt.log if no file is specified. By default, the sqr.log file is used in the current working directory.
-P	(MVS, AS400) Suppresses printer control characters from column 1.
-PRINTER:xx	Causes SQR to use printer type xx when creating output files.
	xx
	EH
	EP
	HP
	HT
	LP
	PD
	Types LP, HP, and PS produce files with the .LIS extension. Types EH and HT produce .htm file output. On Microsoft Windows systems, the WP extension sends the output to the default Microsoft Windows printer. To specify a non-default Microsoft Windows printer, enter -PRINTER:WP:{Printer Name}. The {Printer Name} is the name assigned to your printer. For example, to send output to a Microsoft Windows printer named NewPrinter, you would use -PRINTER:WP:NewPrinter. If your printer name has spaces, enclose the entire argument in quotes. If you want to create an .SPF file, use -KEEP.
-XB	(Non-Microsoft Windows) Suppresses the SQR banner and the "SQR... End of Run" message.
-XNAV	Prevents SQR from creating the 'Navigation Bar' in HTM files generated with -PRINTER:HT. This occurs when only a single HTM file is produced. Multiple HTM files generated from a single report will always contain the navigation bar.
-XTB	Preserves the trailing blanks in a LIS file at the end of a line.
-XTOC	Prevents SQR from generating the Table of Contents for the report. This flag is ignored when either -PRINTER:EH or -PRINTER:HT is also specified.
-ZIF {file}	Sets the full path and name of the SQR initialization file, SQR.INI.
-ZMF {file}	Specifies the full path and name of the SQR error message file, sqrrr.dat.
-ZRF {file}	Sets the full path and name of an alternate registry.properties file. The following is a common default path to the registry.properties file on an NT system: c:\peoplesoft\properties\registry.properties The registry.properties file lists Data Sources that SQR Server can access.

Generating Output in Microsoft Windows

If you are a Microsoft Windows user, the graphical user interface of SQR Print enables you to generate output from the Print dialog box. In addition to the previously mentioned SQR Print output options, you can also select a Windows printer. This spools your SQR output to your default Microsoft Windows printer or print server.

To generate output in Microsoft Windows:

1. From the File menu, select Print.

The Print dialog box appears.

2. Under **Generate output for**, select the option next to the type of output you want.
3. Specify a file path.
4. Select the **Print to file** check box.
5. Click **OK**.

CHAPTER 6

Avoiding Older SQR Commands

This chapter provides a list of the current set of older SQR commands and provides detailed information concerning each.

Understanding Older SQR Commands

PeopleSoft recommends that you avoid incorporating these commands in your SQR code. Even though these commands are technically supported in this release, they do not interact well with the current SQR lexicon. Incorporating these commands into your SQR code can cause unpredictable results. PeopleSoft recommends that you phase these commands out of your code as soon as it is feasible to do so. SQR may not support these commands in future releases.

If you still have older SQR commands in your program code, refer to this chapter as you replace them with their updated alternatives.

<i>Old Commands</i>	<i>Use Instead</i>
BEGIN-REPORT (END-REPORT)	BEGIN-PROGRAM (END-PROGRAM)
DATE-TIME	datenow function
DECLARE PRINTER	DECLARE-PRINTER
DECLARE PROCEDURE	DECLARE-PROCEDURE
DOLLAR-SYMBOL	ALTER-LOCALE
GRAPHIC FONT	ALTER-PRINTER
MONEY-SYMBOL	ALTER-LOCALE
NO-FORMFEED	DECLARE-LAYOUT
PAGE-SIZE	DECLARE-LAYOUT
PRINTER-DEINIT	DECLARE-PRINTER
PRINTER-INIT	DECLARE-PRINTER
PRINT...CODE	PRINT...CODE-PRINTER

Note. Two older commands, DECLARE PRINTER and DECLARE PROCEDURE, do not contain hyphens. The new commands, DECLARE-PRINTER and DECLARE-PROCEDURE, contain hyphens.

Older SQR Command Details

The following sections discuss each of the older SQR commands.

BEGIN-REPORT

Syntax

```
BEGIN-REPORT
```

Note. This command may be discontinued in a future release. PeopleSoft recommends that you no longer use this command. To use the newer SQR functionality, use BEGIN-PROGRAM.

Description

Begins a report.

After processing the commands in the SETUP section, SQR starts program execution at the BEGIN-REPORT section. The PROGRAM section typically contains a list of DO commands, though you can also use other commands. This is the only required section in an SQR program.

Example

```
begin-report
  do startup
  do main
  do finish
end-report
```

DATE-TIME

Syntax

```
DATE-TIME position [date_format[col_var]]
```

Note. This command may be discontinued in a future release. PeopleSoft recommends that you no longer use this command. To use the newer SQR functionality, use the datenow function in the LET command.

Description

Retrieves the current date and/or time from the local machine (or from the database for Oracle and some DB2 platforms) and places it in the output file at the specified position or into a column variable.

If *col_var* is specified, a *date_format* must be supplied and the current date and time is retrieved each time this command is executed. Otherwise, the date is retrieved only at program start and the same date and/or time is printed each time.

If a *date_format* is not specified, then the date is returned in the default format for that database. The following table provides the default date-time formats for SQR-supported databases.

Database	Default Date-Time Format
DB2	YYYY-MM-DD-HH:MI YYYY-MM-DD- HH:MI:SS.NNNNNN
Informix	YYYY-MM-DD HH:MI YYYY-MM-DD HH:MI:SS.NNN
Oracle	DD-Mon-YYYY HH:MI PM
SYBASE	DD-MON-YYYY HH:MI

For some databases, there are two default formats. The first format prints the date-time, as in the following example:

```
date-time (+1,1)
```

The second format retrieves the date-time into a column variable, as follows:

```
date-time () ' ' &date1
```

For those databases with only one default format, that format is always used in either of these cases.

See the table showing miscellaneous functions under the “LET” command for information about the valid edit mask format codes.

Parameters

Parameter	Description
<i>position</i>	Specifies the position for printing the date.
<i>date_format</i>	A string literal containing the date format mask.

Parameter	Description
<i>col_var</i>	Places the retrieved date-time into a column variable rather than in the output file.

Example

```

date-time (1,50) MM/DD/YY
date-time (1,1) 'Day Mon DD, YYYY'
date-time () HH:MI &time
date-time (+1,70) 'MON DD YYYY HH24:MI' &datetime
date-time (#i, #j) 'YYYY-MM-DD' &date1

```

See the \$current-date reserved and datenow function described in the table showing miscellaneous functions under the “LET” command.

See Also

ALTER-LOCALE command.

DECLARE PRINTER

Syntax

```

DECLARE PRINTER
[TYPE=printer_type_lit]
[ORIENTATION=orientation_lit]
[LEFT-MARGIN=left_margin_num_lit]
[TOP-MARGIN=top_margin_num_lit]
[LINE-SIZE=line_size_num_lit]
[CHAR-SIZE=char_size_num_lit]
[LINES-INCH=lines_inch_int_lit]
[CHARS-INCH=chars_inch_num_lit]
[POINT-SIZE=point_size_num_lit]
[FONT-TYPE=font_type_txt_lit]
[SYMBOL-SET=symbol_set_id_lit]
[STARTUP-FILE=file_name_txt_lit]
[FONT=font_int_lit]
[BEFORE-BOLD=before_bold_string_txt_lit]
[AFTER-BOLD=after_bold_string_txt_lit]

```

Note. This command may be discontinued in a future release. PeopleSoft recommends that you no longer use this command. To use the newer SQR functionality, use DECLARE-LAYOUT and DECLARE-PRINTER.

Description

Specifies the printer type and sets printer characteristics.

You can use the DECLARE PRINTER command in either the SETUP section or in the body of the report. Generally, you should use it in the SETUP section. However, if you do not

know what type of printer you will be using until the report is run, or if you need to change some of the arguments depending on user selection, you could put several DECLARE PRINTER commands in the body of the report and execute the one you need.

The following arguments take effect only once, upon execution of the first PRINT command, and thereafter have no effect even if changed:

LINE-SIZE
CHAR-SIZE
LINES-INCH
CHARS-INCH
ORIENTATION

SQR maps its line and column positions on the page by using a grid determined by the LINE-SIZE and CHAR-SIZE (or LINES-INCH and CHARS-INCH) arguments. Each printed piece of text is placed on the page using this grid. Because the characters in proportional fonts vary in width, it is possible that a word or string is wider than the horizontal space you have allotted, especially in words containing uppercase letters. To account for this behavior, you can either move the column position in the PRINT statement or indicate a larger CHAR-SIZE in the DECLARE PRINTER command.

Parameters

The following table describes the arguments for the DECLARE PRINTER command.

Argument	Choice or Measure	Default Value	Description
TYPE	LINEPRINTER, POSTSCRIPT, HPLASERJET	LINEPRINTER	SQR creates output specific to each printer. LINEPRINTER files generally consist of ASCII characters and can be viewed by a text editor. POSTSCRIPT files consist of ASCII characters, but you need to know PostScript to understand what will be shown on the printer. HP LaserJet files are binary files and cannot be edited or viewed.

<i>Argument</i>	<i>Choice or Measure</i>	<i>Default Value</i>	<i>Description</i>
ORIENTATION	PORTRAIT, LANDSCAPE	PORTRAIT	Portrait pages are printed vertically. Landscape pages are printed horizontally. Printing in landscape on HP LaserJet printers requires landscape fonts.
LEFT- MARGIN	inches	0.5	This argument does not apply to LINE-PRINTER printers. This is the amount of blank space to leave at the left side of the page.
TOP- MARGIN	inches	0.5	This argument does not apply to LINE-PRINTER printers. This is the amount of blank space to leave at the top of the page.
LINE-SIZE	points	12	This argument does not apply to LINE-PRINTER printers. This is the size of each SQR line on the page. There are 72 points per inch. If LINE-SIZE is not specified, it follows the value for POINT-SIZE, if specified. The default value of 12 points yields 6 lines per inch.

Argument	Choice or Measure	Default Value	Description
CHAR- SIZE	points	7.2	This argument does not apply to LINE-PRINTER printers. This is the size of each SQR horizontal character column on the page (for example, the distance between the locations (1,12) and (1,13)). If CHAR-SIZE is not specified and the POINT-SIZE is less than 8.6, CHAR-SIZE is set to 4.32, which yields 16.6 characters per inch. The default value of 7.2 yields 10 characters per inch.
LINES- INCH	lines	6	This argument does not apply to LINE-PRINTER printers. This is an alternate way of indicating the line size, in lines per inch, rather than in points for the LINE-SIZE.
CHARS- INCH	characters	10	This argument does not apply to LINE-PRINTER printers. This is an alternate way of indicating the width of each SQR character column, in characters per inch, rather than points for CHAR-SIZE.
POINT- SIZE	points	12	This argument does not apply to LINE-PRINTER printers. This is the beginning size of the selected font.

<i>Argument</i>	<i>Choice or Measure</i>	<i>Default Value</i>	<i>Description</i>
FONT- TYPE	PROPORTIONAL, FIXED	Depends on the font	This argument applies only to HP LaserJet printers and must be specified only for font types not defined in the table showing the fonts available for HP LaserJet printers in SQR in the previous “DECLARE-PRINTER” section.
SYMBOL- SET	HP defined sets	0U	This argument applies only to HP LaserJet printers. The default value of "0U" is for the ASCII symbol set. For a complete list of the symbol sets, see the HP LaserJet Technical Reference Manual.
STARTUP- FILE	filename	POSTSCRI. STR	This argument applies only to PostScript printers. Use this to specify an alternate startup file. Unless otherwise specified, the default startup file is located in the directory specified by the environment variable SQDIR.

<i>Argument</i>	<i>Choice or Measure</i>	<i>Default Value</i>	<i>Description</i>
FONT	font_number	3	This is the font number of the typeface to use. For HP LaserJet printers, this is the typeface value as defined by Hewlett-Packard. For a complete list of the typeface numbers, see the HP LaserJet Technical Reference Manual. For PostScript printers, SQR supplies a list of fonts and arbitrary font number assignments in the file POSTSCRI.STR. The font numbers are the same as those for HP LaserJet printers, wherever possible, so that you can use the same font number for reports to be printed on both types of printers. You can modify the font list in POSTSCRI.STR to add or delete fonts. Read the POSTSCRI.STR file for instructions. See the table that lists the fonts available for HP LaserJet printers that are available in SQR internally in the previous “DECLARE-PRINTER” section. See also the table that lists the fonts available in the SQR POSTSCRI.STR file also in the previous “DECLARE-PRINTER” section.

Argument	Choice or Measure	Default Value	Description
BEFORE- BOLD	any string	(none)	The BEFORE-BOLD and AFTER- BOLD arguments are for LINEPRINTER printers only. They specify the character string to turn bolding on and off. If the string contains blank characters, enclose it in single quote marks ('). To specify non-printable characters, such as ESC, enclose the decimal value inside angle brackets as follows: BEFORE-BOLD=<27>[r ! Turn on bold AFTER-BOLD=<27>[u ! Turn it off These arguments work in conjunction with the BOLD argument of the PRINT command.
AFTER- BOLD	any string	(none)	See BEFORE-BOLD.

The font you choose—in orientation, typeface, and point size—must be an internal font, available in a font cartridge, or downloaded to the printer.

For fonts not listed in the table that lists the fonts available for HP LaserJet printers that are available in SQR internally in the previous “DECLARE-PRINTER” section, you must indicate the font style using the FONT-TYPE argument, or the correct typeface cannot be selected by the printer.

DECLARE PROCEDURE

Syntax

```
DECLARE PROCEDURE
[BEFORE-REPORT=procedure_name]
[AFTER-REPORT=procedure_name]
[BEFORE-PAGE=procedure_name]
[AFTER-PAGE=procedure_name]
```

Note. This command may be discontinued in a future release. PeopleSoft recommends that you no longer use this command. To use the newer SQR functionality, use `DECLARE-PROCEDURE`.

Description

Defines specific event procedures.

`DECLARE PROCEDURE` can be issued either in the `SETUP` section or in the body of the report. You can use the command as often as you like.

If you issue multiple `DECLARE PROCEDURE` commands, the last one takes precedence. In this way, you can turn procedures on and off while the report is executing. The referenced procedures do not take any arguments; however, they may be **LOCAL**. In addition, they can only `PRINT` into the body of the report, that is, they cannot `PRINT` into the header and/or footer areas.

Parameters

<code>BEFORE-REPORT</code>	Specifies a procedure to execute at the time of the first <code>PRINT</code> command. Use this to create a report heading, for example.
<code>AFTER-REPORT</code>	Specifies a procedure to execute just before the report file is closed at the end of the report. Use this to print totals or other closing summary information. If no report was generated, the procedure does not execute.
<code>BEFORE-PAGE</code>	Specifies a procedure to execute at the beginning of every page, just before the first <code>PRINT</code> command for the page. Use this to set up page totals, for example.
<code>AFTER-PAGE</code>	Specifies a procedure to execute just before each page is written to the file. Use this to display page totals, for example.

Example

```
declare procedure
  before-page=page_setup
  after-page=page_totals
```

DOLLAR-SYMBOL

Syntax

```
DOLLAR-SYMBOL  new_symbol
```

Note. This command may be discontinued in a future release. PeopleSoft recommends that you no longer use this command. To use the newer SQR functionality, use `ALTER-LOCALE`.

Description

Redefines the currency symbol within numeric edit masks.

The dollar sign (\$) is the default currency symbol for coding edit masks in the program that prints on report listings. The DOLLAR-SYMBOL provides a way to change that symbol for both the edit mask and for printing.

If you want to change the symbol that prints on the report, use MONEY-SYMBOL in the PROCEDURE section. Use DOLLAR-SYMBOL and MONEY-SYMBOL together to configure your SQR programs and the reports they produce.

This command is used only in the SETUP section.

Note. The MONEY-SYMBOL command has the same effect as these options of the ALTER-LOCALE command: MONEY-SIGN and MONEY-SIGN-LOCATION=LEFT.

The following table lists the characters that DOLLAR-SYMBOL cannot take.

<i>Type</i>	<i>Character</i>
Numbers	0, 8, 9
Alphabetical	b
	e
	n
	r
	v
	B
	E
	N
	R
	V
Symbols	.
	,
	-
	+
	!
	*
	—
	`
	<

Type	Character
	>
	(
	)

Parameters

Parameter	Description
<i>new_symbol</i>	Specifies a new, single character to be used in edit masks instead of the dollar sign (\$).

Example

The following example indicates how to use the DOLLAR-SYMBOL command:

```
begin-setup
  dollar-symbol £                               ! Define £ as the currency symbol
end-setup
begin-procedure
  ...
  print #amount () edit £££,999.99
  ...
end-procedure
```

In the previous example, if you used the dollar sign in the edit mask after defining the dollar symbol as £, the following error message appears:

```
Bad numeric 'edit' format: $$$,999.99
```

See Also

See the ALTER-LOCALE command for a description of other locale-specific parameters.

GRAPHIC FONT

Syntax

```
GRAPHIC  ()
FONT { font_number_int_lit|_var } [point_size_int_lit|_var[{1|0}
[pitch_int_lit|_var]]]
```

Note. This command may be discontinued in a future release. PeopleSoft recommends that you no longer use this command. To use the newer SQR functionality, use ALTER-PRINTER and DECLARE-PRINTER to set the FONT, FONT-TYPE, POINT-SIZE, and PITCH.

Description

Changes a font.

Parameters

Parameter	Description
<i>font_number</i>	For HP LaserJet printers, the specified font must be installed in the printer. For PostScript printers, the font must be defined in the POSTSCRI.STR file.
<i>point_size</i>	If the <i>point_size</i> is omitted, the size from the most recent DECLARE-PRINTER or GRAPHIC FONT command is used.
{1 0}	This argument is for HP LaserJet printers only. It is needed only if you are using a font that SQR does recognize. (See the table that lists the fonts available for HP LaserJet printers that are available in SQR internally under the DECLARE-PRINTER command.) 1 indicates a proportional font, and 0 indicates a fixed pitch font. The default is proportional.
<i>pitch</i>	If the specified font is fixed pitch, you should also indicate the pitch in characters per inch.

Example

The following example indicates the GRAPHIC FONT command:

```
graphic () font 23 8.5! Century Schoolbook, 8.5 points
graphic () font 6 12 0 10! Letter Gothic, 12 points,
                        ! fixed, 10 characters per inch
graphic () font :#font_number :#point_size
```

See Also

ALTER-PRINTER and DECLARE-PRINTER for information about setting and changing the FONT, FONT-TYPE, POINT-SIZE, and PITCH.

MONEY-SYMBOL

Syntax

```
MONEY-SYMBOL new_symbol
```

Note. This command may be discontinued in a future release. PeopleSoft recommends that you no longer use this command. To use the newer SQR functionality, use the ALTER-LOCALE command.

Description

Redefines the currency symbol to be printed.

If you want to change the symbol that prints on the report, use the MONEY-SYMBOL in the programs PROCEDURE sections. When the MONEY-SYMBOL is set, that value is used until the next MONEY-SYMBOL command executes.

Use DOLLAR-SYMBOL and MONEY-SYMBOL together to configure your SQR application programs and the reports they produce.

To indicate a non-edit character, surround its decimal value with angle brackets (<>). See the table under the DOLLAR-SYMBOL command for characters that cannot be used with MONEY-SYMBOL.

Note. The MONEY-SYMBOL command has the same effect as these options of the ALTER-LOCALE command: MONEY-SIGN and MONEY-SIGN-LOCATION=LEFT.

Parameters

<i>Parameter</i>	<i>Description</i>
<i>new_symbol</i>	Specifies a new, single character to replace the dollar sign (\$) or DOLLAR-SYMBOL character on the printed report.

Example

The following example indicates how to use the DOLLAR-SYMBOL and MONEY-SYMBOL commands:

```
begin-setup
  dollar-symbol £! Define £ as the
    ! currency symbol
end-setup
begin-procedure! If #Amount=1234.56
...
money-symbol £
print #Amount () Edit £££,999.99 ! Prints as: £1,234.56
...
money-symbol $
print #Amount () Edit £££,999.99 ! Prints as: $1,234.56
...
money-symbol
print #Amount () Edit £££,999.99 ! Prints as: 1,234.56
...
end-procedure
```

See Also

DOLLAR-SYMBOL and ALTER-LOCALE commands.

NO-FORMFEED**Syntax**

```
NO-FORMFEED
```

Note. This command may be discontinued in a future release. PeopleSoft recommends that you no longer use this command. To use the newer SQR functionality, use the FORMFEED parameter of the DECLARE-LAYOUT command.

Description

Prevents form feed characters from being written to the output file.

NO-FORMFEED is useful for certain types of reports; for example, flat file output. It is used only in the SETUP section.

Do not write form feed control characters directly into the output file between pages.

Example

```
begin-setup  
  no-formfeed  
end-setup
```

See Also

See the FORMFEED qualifier in DECLARE-LAYOUT.

PAGE-SIZE**Syntax**

```
PAGE-SIZE  page_depth_num_lit  page_width_num_lit
```

Note. This command may be discontinued in a future release. PeopleSoft recommends that you no longer use this command. To use the newer SQR functionality, use the MAX-LINES and MAX-COLUMNS parameters of the DECLARE-LAYOUT command.

Description

Sets the page size.

If you are printing multiple reports, you must use the PAPER-SIZE parameter of the DECLARE-LAYOUT command.

This command is used in the SETUP section only.

Specify the *page_depth* in lines and the *page_width* in columns. An average report printed on 8 1/2 by 11 inch paper might have a page size of 60 lines by 80 columns. A 3 inch by 5 inch sales lead card might have a size of 18 by 50.

If the page size is not specified, the default of 62 lines by 132 columns is used.

For line printers, SQR stores one complete page in a buffer before writing the page to the output file when you issue a NEW-PAGE command or when a page overflow occurs.

You could define a page to be 1 line deep and 4,000 characters wide. This could be used for writing large flat files, perhaps for copying to magnetic tape. Each time a NEW-PAGE occurs, one record would be written. Use the NO-FORMFEED command in the SETUP section to suppress form feed characters between pages.

Use a page width at least one character larger than the right-most position that will be written. This prevents unwanted wrapping when printing. When the last column position on a line is printed, the current position becomes the first position of the next line. This can cause confusion when using relative line positioning with the NEXT-LISTING command. Having a wider page than necessary does not waste any file space because SQR trims trailing blanks on each line before writing the report file.

The size of the internal page buffer used to store a complete page in memory can be determined by multiplying the page depth by the width in the PAGE-SIZE command. For PCs, the page buffer is limited to 64K bytes. On other computers, the page buffer is limited only by the amount of memory available.

Example

```
begin-setup
  page-size 57 132! 57 lines long by 132 columns wide
end-setup
```

See Also

See the PAPER-SIZE parameter of the DECLARE-LAYOUT command.

PRINT ...CODE

Syntax

```
PRINT . . . CODE
```

Description

If you use CODE, the sequence is assumed to be for the printer type specified in the DECLARE-REPORT or default printer, if none is specified.

Parameters

<i>Parameter</i>	<i>Description</i>
CODE	CODE is a qualifier that may be discontinued in a future release. Use CODE-PRINTER instead.

PRINTER-DEINIT**Syntax**

```
PRINTER-DEINIT  initialization_string
```

Note. This command may be discontinued in a future release. PeopleSoft recommends that you no longer use this command. To take advantage of newer SQR functionality, use the RESET-STRING parameter of the DECLARE-PRINTER command.

Description

Sends control or other characters to the printer at the end of a report.

Specify nondisplay characters by placing their decimal values inside angled brackets. For example, <27> is the ESC or escape character.

The PRINTER-DEINIT command is used only in the SETUP section and is designed for use with Line-Printer style output. It has limited functionality with HP LaserJet and PostScript printers.

Example

```
begin-setup
  printer-deinit<27>[7J! Reset the printer
end-setup
```

See Also

See the ENCODE command for another method of printing nondisplay characters. See the *chr* function in the table listing miscellaneous functions under the LET command section.

PRINTER-INIT**Syntax**

```
PRINTER-INIT  initialization_string
```

Note. This command may be discontinued in a future release. PeopleSoft recommends that you no longer use this command. To use the newer SQR functionality, use the INIT-STRING parameter of the DECLARE-PRINTER command.

Description

Sends control or other characters to the printer at the beginning of a report.

Specify non-display characters by placing their decimal values inside angled brackets. For example, <27> is the ESC or escape character.

The PRINTER-INIT command is used only in the SETUP section and is designed for use with Line-Printer output. It has limited functionality with HP LaserJet and PostScript printers.

Example

```
begin-setup
  printer-init<27>[7J! Set the printer
end-setup
```

See Also

See the ENCODE command for another method of printing non-display characters. See the *chr* function in the table listing miscellaneous functions under the LET command section.

CHAPTER 7

Using SQR Sample Programs

This chapter describes the SQR Samples library of programs.

Understanding SQR Samples

SQR samples is a library of SQR programs you can use to customize and experiment with. These programs are stored in the SQR Server directory called SAMPLE on your installation media. You can modify these programs to create configured SQRs.

SQR Sample Programs

The following table lists the sample SQR programs and provides a brief description of each. Each program comprises a report specification and a sample of the output.

<i>Name</i>	<i>Description</i>
_____.DAT	Data files used by the LOADALL.SQR programs
_____.MEM	SQR startup files to run small, medium, and large SQR programs
APPEND.SQR	Demonstrates the append and fixed-nolf commands
APTDIARY.SQR	Demonstrates columns, text wrapping
AREA100.SQR	Demonstrates a 100% area chart
BAR100.SQR	Demonstrates a 100% bar chart
BARCODE.SQR	Demonstrates printing a bar code
CALENDAR.SQR	Demonstrates nondatabase formatting
COMP_FOR.SQR	Prints a graph of the forecasted and actual sales for a given employee
COMP_F_G.SQR	Prints a graph of the forecasted and actual sales for month or quarter
COMP_PLN.SQR	Prints a graph of the planned and actual sales for a given employee
COMP_P_G.SQR	Prints a graph of the planned and actual sales for month or quarter
COVLET02.SQR	Uses SQR to input data from user, enter data in the database, and write a form letter using a DOCUMENT paragraph
CRUPSAL.SQR	(Oracle) Creates stored functions and procedures for Oracle Version 7

Name	Description
CUST.SQR	Prints a list of all of the customers bursted by page
CUSTLBL.SQR	Demonstrates printing mailing labels within columns
CUSTOMER.SQR	Demonstrates multiple detail lines, NEXT-LISTING command
CUSTOMR2.SQR	Demonstrates the use of the ON-BREAK argument to the PRINT command
CUSTOMR3.SQR	Demonstrates the use of the INPUT command to change report output
CUSTOMR4.SQR	Demonstrates the use of arrays
CUSTOMR5.SQR	Demonstrates dynamic queries to enable user to qualify a report as it runs
CUST_SUM.SQR	Prints a group of information about each customer in the customer table
CUSTTAPE.SQR	Demonstrates the flat file output for magnetic tape or other post-processing
DATAA.DAT	Needed for append.sqr
DATAB.DAT	Needed for append.sqr
DROPALL.SQR	Drops all the SQR sample tables created by the LOADALL program
DROPPROC.SQR	(SYBASE) Deletes leftover temporary stored procedures belonging to the user
DYNAMCOL.SQR	Demonstrates use of dynamic columns, dynamic tables and variables passed to ON-ERROR procedure
EMP.SQR	Prints a list of all of the employees bursted by page
EMP_COMM.SQR	Calculates each employee's commission based on sales
EMP_P_Q.SQR	List all employee quotas for a given month or quarter
ENVELOPE.SQR	Demonstrates use of printing envelope with proper bar code
EXPORT.SQR	Creates two SQR reports: one to export a database table, the second to import that table. Data from the table is stored in an external operating system file in compressed format, with trailing blanks removed.
FLATFILE.SQR	Creates an SQR report to extract a database table and place it into a flat file
FLOATBAR.SQR	Demonstrates a floating bar chart
FOR_CUST.SQR	Sales forecast for given customer grouped by month or quarter
FOR_EMP.SQR	Sales forecast for given employee grouped by month or quarter
FOR_PROD.SQR	Sales forecast for given product grouped by month or quarter
FOR_REG.SQR	Sales forecast for given region grouped by month or quarter

Name	Description
FOR_SUM.SQR	Creates a table of projected product sales with links to more information
FORMLETR.SQR	Demonstrates form letters using a DOCUMENT paragraph
HILO.SQR	Demonstrates a high-low-close chart
HISTGRAM.SQR	Demonstrates a histogram chart
INQUIRY.SQR	Creates an SQR program to display rows at your terminal selected from a database table you specify. The resulting SQR program prompts you to qualify rows to be selected, display those rows, then repeat.
INVOICE.SQR	Demonstrates multiple reports, printing invoices, and printing envelopes
LOADALL.SQR	Creates and loads sample tables used in the above SQR programs
MAKEDATA.SQR	Creates a data file with fixed length and NOLF attributes
MAKEREPT.SQR	Helps you create SQR reports more quickly
MITI1.EPS	Needed for sqrlogo.sqr
MULTIPLE.SQR	Demonstrates creating multiple reports
NESTREPT.SQR	Demonstrates nesting of procedures
ORDERS.SQR	Lists all the orders and the orderlines associated with them
ORD_MONG.SQ R	List all orders for a given month and group them by employee number
ORD_M_Q.SQR	List all orders for a given month or quarter
ORD_PROD.SQR	List all orders for a given product
ORD_REGG.SQ R	Creates a report of all orders from a given region grouped by month or grouped by quarter
ORD_SUM.SQR	Displays an order's summary by month
ORD_S_Q.SQR	Prints a graph of the percent of orders for each region (in a year) and four graphs of the percent of orders for each region (one for each quarter of that year)
OVERBAR.SQR	Demonstrates an overlapped bar chart
PHONELST.SQR	Demonstrates printing within columns, page headings, and page footings
PLN_EMP.SQR	Sales plan for given employee grouped by month or quarter
PLN_GEN.SQR	Sales plan grouped by month or quarter
PLN_REG.SQR	Sales plan for given region grouped by month or quarter
PRODUCT.SQR	List of products and their prices and a graph of orders of products
SALELEAD.SQR	Demonstrates DOCUMENT paragraphs

Name	Description
SALES.SQR	Demonstrates charting from stored data and printing several charts on one page
SCATTER.SQR	Demonstrates a scatter chart
SHOWPROC.SQR	(SYBASE) Shows any leftover temporary stored procedures belonging to the user
STCKAREA.SQR	Demonstrates a stacked area chart
SQR3DBAR.SQR	Demonstrates a 3D bar chart
SQRLASER.SQR	Demonstrates graphic and file I/O commands
SQRLINE.SQR	Demonstrates a line chart
SQRLOGO.SQR	Demonstrates printing images
SQRPIE.SQR	Demonstrates a pie chart
TABREP.SQR	Creates a tabular SQR report for a table you select
UPDATE.SQR	Generates an SQR program that enables you to query and update database tables. The created program uses the SHOW command to simulate a menu interface.
UPDSAL.SQR	A sample report that demonstrates use of stored functions and procedures in Oracle 7

CHAPTER 8

Using the PSSQR.INI File

The PSSQR.INI file is the initialization file for SQR for PeopleSoft. This file contains settings and parameters that SQR uses during the compile and execution phases. This chapter provides an overview of the PSSQR.INI file and discusses the following topics:

- PSSQR.INI installation process.
- Default settings.
- Processing-limits section.
- Environment section.
- Locale section.
- Fonts section.
- HTML images section.
- PDF fonts section.
- Colors section.

Installation of PSSQR.INI

The installation process installs a default initialization file called PSSQR.INI. This file is located in the SQR directory under the PS_Home directory.

For Microsoft Windows Platforms Only:

SQR looks for the initialization file in the following locations in order:

1. The file name specified by the -ZIF {file} command-line flag.
2. The directory in which the executable image resides.
3. The Windows system directory.

For All Other Platforms:

SQR looks for the initialization file in the following locations in order:

1. The file name specified by the `-ZIF{file}` command-line flag.
2. The current working directory.
3. The directory specified using the SQRDIR environment variable.

SQR automatically sets up SQRDIR. SQRDIR is no longer kept in PSSQR.INI.

You can make changes or additions to the PSSQR.INI file.

The format of the file is as follows:

```
; Comments are lines which start with a semicolon.  The semicolon
; must be the first character of the line and therefore cannot be
; part of another line.
;
; Leading and trailing space characters are ignored.  To preserve
; the space characters you must surround the value with either
; single (') or double (") quote characters.  SQR will remove
; them when the entry is processed.
;
[Section_Name]
Entry = Value
.
.

[Another_Section_Name]
Entry = Value
.
.
```

[Default-Settings] Section

This section defines the various SQR default actions.

Entry	Value	Description
<code>AllowDateAsChar={TRUE FALSE}</code>	TRUE FALSE	The default setting is FALSE. By default, SQR produces an error when a dynamic column specification does not match the database definition of the column. That is, character equals character, date equals date, and numeric equals numeric. When this value is set to TRUE, SQR allows character to equal either character or date columns. When a date column is “type cast” to be a character, SQR creates the string according to the following rules: For DATETIME columns, SQR uses the format specified by the <code>SQR_DB_DATE_FORMAT</code> setting. If this has not been set, the first database-dependent format as listed in the table showing default database formats in the “Edit” section under the PRINT command. For DATE columns, SQR uses the format specified by the <code>SQR_DB_DATE_ONLY_FORMAT</code> setting. If this has not been set, the format listed in the table showing default database formats in the “Edit” section under the PRINT command. For TIME columns, SQR uses the format specified by the <code>SQR_DB_TIME_ONLY_FORMAT</code> setting. If this has not been set, the format as listed the table showing the TIME column formats in the “Edit” section under the PRINT command. In the following example, <code>AllowDateAsChar=True</code>. This allows <code>\$Col1</code> to be either date or text. <pre> Begin-Select [\$Col1] &col1=Text [\$Col2] &col2=Date [\$Col3] &col3=Number from MyTable End-Select </pre>
<code>OUTPUT-FILE-MODE</code>	LONG SHORT	Specifies the filename convention used for HTML output. SHORT specifies DOS style (8.3) and LONG specifies UNIX style (non 8.3). The default is LONG. (Ignored on 16-bit platforms) The <code>Declare-TOC</code> command and <code>-Burst</code> force <code>Output-File-Mode = LONG</code>.

Entry	Value	Description
		The following represent the file formats for UNIX, DOS, Windows NT, and VMS. SQR and SQR*: {Program} is the name of the SQR/SQT file without the extension For Output-File-Mode = SHORT, SQR-generated filenames are limited to a DOS 8.3 format Output file = {Program}.LIS for first, and {Program}.Lnn for multi-reports SFP file = {Program}.SFP for first, and {Program}.Snn for multi-reports PDF file = {Program}.PDF for first; and {Program}.Pnn for multi-reports HTM file = {Program}.HTM for “frame, and {Program}.Hbb for report bodies GIF file={Program}.Gxx for all reports bb ranges from 00 to 99 and represents the report number. nn ranges from 01 to 99 and represents the report number. xx ranges from 00 to ZZ and represents the graphic number. For Output-File-Mode = LONG, SQR-generated filenames are not constrained to a DOS 8.3 format. {Output}={Program} of first report and {Program}_nn for multi-reports. Output file = {Output}.LIS SPF file = {Output}.SPF PDF file = {Output}.PDF GIF file = {Output}_zz.SPF HTM files = {Output}.HTM, {Output}_bb.HTM, {Output}_frm.HTM, {Output}_toc.HTM, {Output}_nav.htm bb ranges from 01 to ZHJOZI and represents the burst page group number. nn ranges from 01 to 99 and represents the report number. zz ranges from 00 to ZHJOZI and represent the graphic number.

Entry	Value	Description
		SQRP: {Filename} is the name of the SPF file without the extension For Output-File-Mode = SHORT, SQR-generated filenames are limited to a DOS 8.3 format. Output file = {Filename}.LIS GIF file = {Filename}.Gxx PDF file = {Filename}.PDF HTM file = .HTM and {Filename}.H00 xx ranges from 00 to ZZ and represents the graphic number. For Output-File-Mode = LONG, SQR-generated filenames are not limited to a DOS 8.3 format. Output file = {Filename}.LIS PDF file = {Filename}.PDF GIF file = {Filename}_zz.SPF HTM files = {Filename}.HTM, {Filename}_bb.HTM, {Filename}_frm.HTM, {Filename}_toc.HTM, {Filename}_nav.htm bb ranges from 01 to ZHJOZI and represents the bursted page group number. zz ranges from 00 to ZHJOZI and represents the graphic number.
LOCALE	Name of a locale defined in the SQR.INI file or the name SYSTEM.	Specifies the initial locale that SQR loads when the program starts to execute. The value of SYSTEM is used to reference the default locale. See the “ALTER-LOCALE” command section for a complete description.
DEFAULT-NUMERIC	INTEGER FLOAT DECIMAL[(p)] V30	Specifies the default numeric type for variables. The command line flag -DNT overrides this setting. Similarly, the DECLARE-VARIABLE command overrides this setting. See the DECLARE-VARIABLE command for complete details on the meaning of the values.
OutputFormFeedwithDashD={TRUE FALSE}	TRUE FALSE	The default value is FALSE. When set to TRUE, the -Dnn command line flag outputs the Form-Feed character that denotes a page break.

Entry	Value	Description
OutputTwoDigitYearWarningMsg={TRUE FALSE}	TRUE FALSE	The default value is TRUE. When set to TRUE, SQR generates a warning message (sent to the warning file) when a YY or RR date edit mask is encountered during a program run. This affects only SQR code that is processed.
UseY2kCenturyAlgorithm={TRUE FALSE}	TRUE FALSE	The default value is FALSE. When set to TRUE, SQR treats the YY date edit mask as though it is an RR edit mask. See the RR edit mask.

Note. Use the setting V30 to process numbers in the same manner as in prior releases (before V4.0). Specifically, all numeric variables and literals are declared as FLOAT, including integer literals.

Processing-Limits Section

The Processing-Limits section is used to define the sizes and limitations of some of the internal structures used by SQR, which has a direct impact on the memory requirements. The entries are the same as those used in the file specified with the **-Mfile** command line flag. If the **-Mfile** command line flag is used, then the Processing-Limits section of the file is not processed.

Entry	Default Value	Maximum Value	Entry Size	Description
BREAKS	100	64K-1	4	Number of BREAK arguments allowed per EVALUATE or IF command.
DYNAMICARGS	70	32K-1	14	Maximum number of dynamic SQL arguments.
EXPRESSIONSPACE	8192	64K-1	1	Maximum length, in bytes, of temporary string storage used during LET operations.
FORWARDREFS	200	32K-1	8	Maximum number of column forward references.

Entry	Default Value	Maximum Value	Entry Size	Description
LONGSPACE	32K-2	32K-2	1	Maximum buffer size to transfer text and image data in bytes.
ONBREAKS	30	64K-1	8	Maximum number of ON-BREAK LEVEL=values per SET.
POSITIONS	1800	64K-1	14	Maximum number of placement parameters, "(10,5,30)".
PROGLINEPARS	18000	64K-1	2	Maximum number of arguments for all program lines. This value is generally 3 or 4 times the value set for PROGLINES.
PROGLINES	5000	32K-1	8	Maximum number of program lines (SQR commands).
QUERIES	60	32K-1	60	Maximum number of BEGIN-SQL and BEGIN-SELECT paragraphs. This is database dependent and can vary. This size is used as a close approximation.
QUERYARGS	240	64K-1	6	Maximum number of arguments (bind variables) for all SQL or SELECT statements. The number of arguments required is one more than the number used in your report file.

Entry	Default Value	Maximum Value	Entry Size	Description
SQLSIZE	4000	64K-1	1	Maximum length of an SQL statement in characters.
STRINGSPACE	15000	64K-1	1	Maximum size of string space for program line arguments, in bytes.
SUBVARS	100	32K-1	8	Maximum number of run-time substitution variables.
VARIABLES	1500	64K-1	18	Maximum number of variables (string, float, integer, decimal), literal values, and database columns. Add 4 to the entry size for Informix and Ingres.
WHENS	70	64K-1	4	Maximum number of WHEN arguments allowed per EVALUATE command.

The maximum value refers to the number of entries allowed as shown in the previous table; however, limits are lower for PCs. In any case, SQR indicates the limit if you exceed it.

In addition to increasing the sizes you may also lower them to decrease the amount of memory used. This might be advantageous, for example, for certain applications running in the PC environment, where memory is limited.

[Environment: *environment*] Sections

The [Environment: { Common | DB2 | Informix | Ingres | ODBC | Oracle | RDB | Redbrick | SQLBase | Sybase | DDO }] sections define environment variables to be used by SQR. An environment variable can be defined in multiple environment sections; however, a definition in a database-specific environment section takes precedence over an assignment in the [Environment:Common] section.

The environment variables that can be set are **SQRDIR**, **SQRFLAGS**, and **DSQUERY** (Sybase only). On Microsoft Windows systems, **SQRDIR** is required and is automatically defined in the appropriate database-specific environment section during the SQR installation. The other

environment variables are optional. **SQRFLAGS** specifies the default command-line flags for all invocations of SQR. **DSQUERY** identifies the default SYBASE server to use.

On Microsoft Windows systems only, the [SQR Extension] section defines DLLs containing new user functions (ufunc) and user calls (ucall). 'Ufunc' and 'ucall' now reside inside SQREXT.DLL and/or other DLLs.

When SQRW.DLL and SQRWT.DLL are being loaded, they look for SQREXT.DLL in the same directory, and for any DLLs specified in the [SQR Extension] section in SQR.INI such as:

```
[SQR Extension]
c:\sqrexts\sqrext1.dll=
c:\sqrexts\sqrext2.dll=
c:\sqrexts\sqrext3.dll=
```

Any new extension DLLs containing new user functions must be listed in the [SQR Extension] section in SQR.INI.

See the *SQR for PeopleSoft Developer's Guide*, “Interoperability”.

For Windows/Oracle, SQR uses dynamic binding of Oracle routines. When SQR attempts to access an Oracle database, it searches for the Oracle DLL as follows:

- The file described by the value of ORACLE_DLL entry in the [Environment:Oracle] section of the SQR.INI file.
- OCIW32.DLL (Oracle supplied)
- ORANT71.DLL (Oracle supplied)

Additional DDO variables are:

- SQR_DDO_JRE_CLASS=<classpath for DDO drivers, support files>
- SQR_DDO_JRE_CLASSn=<optional; additional entries to the classpath>
- SQR_DDO_JRE_PATH=<classpath information for the local JRE>

Each of these entries is automatically entered upon product installation (NT only). You can specify additional classpath entries using up to nine SQR_DDO_JRE_CLASSn variables, where n is a number from 1-9. These additional variables are available to augment the normal 512-character line limit for entries in the sqr.ini file.

[Locale:locale-name] Section

The [LOCALE:locale-name] section specifies the default settings for the locale identified by *locale-name* (which can consist of A-Z, 0-9, hyphen, or underscore). A number of locales are predefined in the SQR.INI file. Depending on your application, the settings for these locales may have to be altered or new locales may have to be added. A locale can be referenced or

altered at runtime using the ALTER-LOCALE command. The entries for a locale section are described in the following table.

Note. The SYSTEM locale is provided for your reference, but is commented out. The settings for the SYSTEM locale, if set, are ignored. Use the ALTER-LOCALE command to change the SYSTEM locale settings at runtime.

Entry	Description
NUMBER-EDIT-MASK	The default numeric edit mask format when the keyword NUMBER accompanies the DISPLAY, MOVE, PRINT, or SHOW command.
MONEY-EDIT-MASK	The default numeric edit mask format when the keyword MONEY accompanies the DISPLAY, MOVE, PRINT, or SHOW command.
DATE-EDIT-MASK	The default date edit mask format when the keyword DATE accompanies the DISPLAY, MOVE, PRINT, or SHOW command, or the LET dateto-str() or strtodate() functions.
INPUT-DATE-EDIT-MASK	The default date format to use with the INPUT command when TYPE=DATE is specified with the command or the input variable is a DATE variable.
MONEY-SIGN	Specifies the characters to replace the '\$' edit character.
MONEY-SIGN-LOCATION	Specifies the MONEY-SIGN characters location. Valid values are LEFT and RIGHT.
THOUSAND-SEPARATOR	Specifies the character to replace the ',' edit character.
DECIMAL-SEPARATOR	Specifies the character to replace the '.' edit character.
DATE-SEPARATOR	Specifies the character to replace the '/' character.
TIME-SEPARATOR	Specifies the character to replace the ':' character.
EDIT-OPTION-NA	Specifies the characters to replace the 'na' option.
EDIT-OPTION-AM	Specifies the characters to replace 'AM'.
EDIT-OPTION-PM	Specifies the characters to replace 'PM'.
EDIT-OPTION-AD	Specifies the characters to replace 'AD'.
EDIT-OPTION-BC	Specifies the characters to replace 'BC'.
DAY-OF-WEEK-CASE	Specifies how the case for the DAY-OF-WEEK-FULL or DAY-OF-WEEK-SHORT entries are affected when used with the format codes 'DAY' or 'DY'. Valid values are UPPER, LOWER, EDIT, and NO-CHANGE. UPPER and LOWER forces the output to either all uppercase or lowercase, ignoring the case of the format code in the edit mask. Use EDIT to follow the case as specified with the format code in the edit mask. Use NO-CHANGE to ignore the case of the format code and output the day of week as explicitly listed in the DAY-OF-WEEK-FULL or DAY-OF-WEEK-SHORT entries.

Entry	Description
DAY-OF-WEEK-FULL	Specifies the full names for the days of the week. SQR considers the first day of the week to be Sunday. All seven days must be specified.
DAY-OF-WEEK-SHORT	Specifies the abbreviated names for the days of the week. SQR considers the first day of the week to be Sunday. All seven abbreviations must be specified.
MONTHS-CASE	Specifies how the case for the MONTHS-FULL or MONTHS-SHORT entries is affected when used with the format codes 'MONTH' or 'MON'. Valid values are UPPER, LOWER, EDIT, and NO-CHANGE. UPPER and LOWER force the output to either all uppercase or lowercase, ignoring the case of the format code in the edit mask. Use EDIT to follow the case as specified with the format code in the edit mask. Use NO-CHANGE to ignore the case of the format code and output the month as explicitly listed in the MONTHS-FULL or MONTHS-SHORT entries.
MONTHS-FULL	Specifies the full names for the months of the year. SQR considers the first month of the year to be January. All 12 months must be specified.
MONTHS-SHORT	Specifies the abbreviated names for the months of the year. SQR considers the first month of the year to be January. All 12 abbreviations must be specified.

Database	DATE Column Formats
DB2	YYYY-MM-DD
INFORMIX	MM/DD/YYYY
ODBC	DD-MON-YYYY

Database	TIME Column Formats
DB2	HH24.MI.SS
ODBC	HH24:MI:SS

[Fonts] Section

The [Fonts] section lists the fonts available to SQR when printing on Microsoft Windows printer devices (using the **-PRINTER:WP** command-line flag). This section does not apply to PostScript or HP LaserJet printer types. See the **DECLARE- PRINTER** command for a listing of available fonts for these alternate printer types.

Adding [Fonts] entries

Within the [Fonts] section, there are a number of predefined font entries. You can add entries by using the font numbers 900 through 999. Each entry consists of a font name, a font style (fixed or proportional), and a bold indicator, all of which are associated with a font number.

For example:

```
4=Arial,proportional
or
300=Courier New,fixed,bold
```

Note. Proportional is assumed if the second parameter starts with a "P". Also, bold is assumed if a third parameter is supplied.

Using the font number, commands such as **ALTER-PRINTER** and **DECLARE-PRINTER**, can reference a particular font style.

Specifying Character Sets in Windows

If you are a Microsoft Windows user, you can use the CharacterSet entry either to determine the Microsoft Windows default character set or to specify a character set. This enables you to print any standard character set to a Windows printer (- **PRINTER:WP**) or to view an SPF file displaying the appropriate character set.

Syntax:	CharacterSet=DEFAULT AUTO character_set
Arguments:	DEFAULT reflects current SQR functionality. AUTO automatically determines the default character set of the Microsoft Windows installation and uses the default set when generating reports. character_set specifies one of these keywords: ANSI, ARABIC, BALTIC, CHINESEBIG5, EASTEUROPE, GB2312, GREEK, HANGUL, HEBREW, JOHAB, MAC, OEM, RUSSIAN, SHIFTJIS, SYMBOL, THAI, TURKISH, VIETNAMESE.

[HTML-Images] Section

The [HTML-Images] section defines the parameters that SQR uses when generating HTML report output files.

<i>Entry</i>	<i>Value</i>	<i>DefaultValue</i>	<i>Description</i>
FIRST-PAGE	HEIGHT, WIDTH, NAME	60,60,firstpg.gif	Specifies the NAME of the graphic image file that accesses the first page of the report. The HEIGHT and WIDTH are values specified in pixels.
PREV-PAGE	HEIGHT, WIDTH, NAME	60,60,prevpg.gif	Specifies the NAME of the graphic image file that accesses the previous page of the report. The HEIGHT and WIDTH are values specified in pixels.
NEXT-PAGE	HEIGHT, WIDTH, NAME	60,60,nextpg.gif	Specifies the NAME of the graphic image file that accesses the next page of the report. The HEIGHT and WIDTH are values specified in pixels.
LAST-PAGE	HEIGHT, WIDTH, NAME	60,60,lastpg.gif	Specifies the NAME of the graphic image file that accesses the last page of the report. The HEIGHT and WIDTH are values specified in pixels.
WALLPAPER	NAME		Specifies the NAME of the graphic image file used as the background image for the report.
Navbar Background	NAME		Background image of navigation bar.

Note. SQR does not perform any validation of the graphic image filenames provided. The user is responsible for ensuring that the graphic image files are in a location that the browser can access.

[PDF Fonts] Section

The [PDF Fonts] section lists the available fonts for SQR when printing using the -PRINTER:PD command-line flag. Fonts specified are case sensitive.

[Enhanced-HTML] Section

The [Enhanced-HTML] section is used to define various default actions that SQR will take when generating HTML output utilizing the **-EH** command line flag.

Entry	Value	Description
Browser={ALL IE NETSCAPE CAPE}	ALL IE NETSCAPE	When set to ALL, the generated HTML auto-determines which browser is being used and invoke the proper browser-specific file. When set to IE, the generated HTML is designed for Internet Explorer. When set to NETSCAPE, the generated HTML is designed for Netscape. The default action is to generate HTML suitable for ALL browsers.
Language={English French German Portuguese Spanish}	English French German Portuguese Spanish	The default setting is English. Sets the language used for the HTML navigation bar.
FullHTML={TRUE FALSE}	TRUE FALSE	When set to TRUE, HTML 3.2 is generated. When set to FALSE, HTML 3.0 is generated. The default setting is FALSE.

[Colors] Section

The [Colors] section defines the default colors that you can use in your SQRs. Enter the default colors in the format of:

[Colors]

color_name = ({rgb})

color_name = ({rgb})

.

.

.

color_name = ({rgb})

The default colors implicitly installed are:

- black = (0,0,0)
- white=(255,255,255)
- gray=(128,128,128)
- silver=(192,192,192)
- red=(255,0,0)
- green=(0,255,0)
- blue=(0,0,255)
- yellow=(255,255,0)
- purple=(128,0,128)
- olive=(128,128,0)
- navy=(0,0,128)
- aqua=(0,255,255)
- lime=(0,128,0)
- maroon=(128,0,0)
- teal=(0,128,128)
- fuchsia=(255,0,255)

CHAPTER 9

Understanding SQR Messages

This chapter provides all the messages produced by SQR for PeopleSoft. It contains two tables listing all:

- Unnumbered messages
- Numbered messages

Two digits (*nn*) appear as replacement markers in the messages. Descriptions of these replacement markers are listed with the message. The messages contain the proper value when they appear on the screen.

Unnumbered Messages

This table lists all unnumbered SQR for PeopleSoft messages.

Error	Description
Out of memory.	This occurs when a call to the C routine 'malloc()' fails. (PC) Use the -Mfile to reduce some of the different memory requirements. Remove unneeded TSRs. (Unix) Increase the size of the system swap file. (VAX) Increase the amount of memory allowed for that user.
No cursors defined.	From the -S command line flag. The SQR program did not contain any commands that required a database cursor.
Not processed due to report errors.	From the -S command line flag. SQR cannot provide information about the cursor due to errors in the program.
Enter `01`02	Type the value to be assigned to the specified variable. `01` = First character of the variable name `02` = Rest of the variable name
NOPROMPT used - Enter value below	(Windows) This message appears when an INPUT command is defined with the NOPROMPT argument.
Enter `01	Enter the value to be assigned to the specified substitution variable. `01` = Name of the substitution variable
Enter this run's parameters:	Enter the values for the parameters defined in the program.

Error	Description
Error on line `01: `02	SQR detected an error while processing the report file. Correct the error and rerun. `01 = Source line number `02 = Source line
Error in include file "`01" on line `02: `03	SQR detected an error while processing the report file. Correct the error and rerun. `01 = Name of the include file `02 = Source line number `03 = Source line
Warning on line `01: `02	SQR detected a nonfatal error while processing the report file. `01 = Source line number `02 = Source line
Warning in include file "`01" on line `02: `03	SQR detected a nonfatal error while processing the report file. `01 = Name of the include file `02 = Source line number `03 = Source line
Type RETURN for more, C to continue w/o display, X to exit run:	Informational message that is used in conjunction with the -D command line flag.
Error at: `01 Loading Oracle DLL Failed!!!	(Oracle) Title for the dialog box that informs the user that SQR could not load the Oracle DLL.
Errors were found in the program file.	Correct the errors and rerun.
Errors were found during the program run.	Correct the errors and rerun.
`01: End of Run.	Informational message. `01 = Image name (for example, SQR)
Enter report name:	Enter the name of the report (.sqr or .sqt) to run.
Enter database name:	(All but Ingres) Enter the name of the database.
Enter database[/username]: Enter Username:	Enter the user name to log onto the database.
Enter Password:	Enter the password. For security reasons, the password is not be echoed.
Customer ID:	Text message
Press Enter to close...	Text message

Error	Description
`01: Program Aborting.	Informational message. `01 = Image name (for example, SQR)
*** Internal Coding Error ***	Informational message.
SQL DataServer Message	(Windows) A title for the error message dialog box.
Operating-System error	(Windows) A title for the error message dialog box.
DB-Library error	(Windows) A title for the error message dialog box.
`01 is running. Click the Cancel button to interrupt it.	(Windows) This is the body of the -C cancel dialog box. This lets the user quit the program run by clicking the Cancel button.
Table of Contents	Text for HTML driver
Previous	Text for HTML driver
Next	Text for HTML driver
First Page	Text for HTML driver
Last Page	Text for HTML driver
PAGE	Text for HTML driver

Numbered Messages

This table lists numbered messages.

Error Number	Error Message	Suggestion/Interpretation
000001	Error while opening the message file: `01' (`02): `03	Try reloading the sqrrr.dat file from the release media. If the error persists, contact technical support. `01 = Name of the error message file `02 = System error code `03 = System error message
000002	Error while reading the message file. (`01): `02	Try reloading the sqrrr.dat file from the release media. If the error persists, contact technical support. `01 = Name of the error message file `02 = System error code `03 = System error message
000003	Error while closing the message file. (`01): `02	Try reloading the sqrrr.dat file from the release media. If the error persists, contact technical support. `01 = Name of the error message file `02 = System error code `03 = System error message

Error Number	Error Message	Suggestion/Interpretation
000004	Error while seeking the message file. ('01): '02	Try reloading the sqrrrr.dat file from the release media. If the error persists, contact technical support. '01 = Name of the error message file '02 = System error code '03 = System error message
000005	Corrupt message file: Invalid header information.	Try reloading the sqrrrr.dat file from the release media. If the error persists, contact technical support.
000006	Corrupt message file: Invalid count (Got '01, Should be '02).	The header contains an invalid entry count. (1) Make sure SQRDIR points to the correct directory. (2) Try reloading the sqrrrr.dat file from the release media. If the error persists, then contact technical support. '01 = The value read from the header '02 = What the value should be
000007	Cannot handle message file version '01.	This release of SQR does not support the header version. (1) Make sure SQRDIR points to the correct directory. (2) Try reloading the sqrrrr.dat file from the release media. If the error persists, then contact technical support. '01 = Unsupported version read from the header
000010	Invalid SEMCode encountered: '01.	An invalid code was passed to the error message handler. Try reloading the files from the release media. If the error persists, then contact technical support. '01 = Invalid code
000011	Unknown conversion type ('01) for code '02.	Try reloading the sqrrrr.dat file from the release media. If the error persists, then contact technical support. '01 = Invalid type '02 = Internal error code
000012	Message '01 must be either Preload or BuiltIn.	The type error code is not correct. Try reloading the sqrrrr.dat file from the release media. If the error persists, then contact technical support. '01 = Error code
000013	Cannot point to message '01.	The error handler cannot position to the desired error code. Try reloading the sqrrrr.dat file from the release media. If the error persists, then contact technical support. '01 = Error code

Error Number	Error Message	Suggestion/Interpretation
000014	The required environment variable `01 has not been defined.	Define the named environment variable and restart SQR. `01 = Environment variable name
000015	The Meta ESC characters do not match (Got `01', Should be `02').	The meta escape character defined in the header does not match what the error message handler expects. Try reloading the sqrrr.dat file from the release media. If the error persists, then contact technical support. `01 = What was found in the header `02 = What was expected to be found
000016	`01() called to process (`02) and the message file is not open.	The specified error routine was called but the error message file was not open. Try reloading the files from the release media. If the error persists, then contact technical support. `01 = Name of the routine `02 = Error code
000017	Message `01 must be ReportParameters or CopyrightNotice.	Try reloading the sqrrr.dat file from the release media. If the error persists, then contact technical support. `01 = Error code
000018	Allocation header does not point to a valid heap.	(Windows) This is the result of a memory overwrite. Record the steps leading up to the error and contact technical support.
000019	Allocation header has an invalid size.	(Windows) This is the result of a memory overwrite. Record the steps leading up to the error and contact technical support.
000020	GLOBAL header has an invalid size.	(Windows) This is the result of a memory overwrite. Record the steps leading up to the error, and contact technical support.
000021	Cannot free GLOBAL allocation.	(Windows) This is the result of a memory overwrite. Record the steps leading up to the error and contact technical support.
000028	Cannot access the initialization file: `01 (`02): `03	The initialization file specified by the -ZIF command line flag cannot be accessed. `01 = Name of the file `02 = System error code `03 = System error message
000202	DPUT: Bad field number.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.

Error Number	Error Message	Suggestion/Interpretation
000203	DARRAY: Unknown command number.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000204	`01: Cannot find `02 command.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Name of the routine `02 = Name of the command
000205	DDO: DO arguments do not match procedure's.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000206	SDO: Bad params for DO command.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000207	SDO: Bad params for BEGIN-PROCEDURE command.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000208	SGOTO: Bad command numbers.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000209	SGOTO: Bad goto function parameters.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000210	SGOTO: Could not find beginning of section or paragraph.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000211	SGOTO: Bad label: from parameters.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000212	COMPAR: Unknown relational (numeric) operator.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.

Error Number	Error Message	Suggestion/Interpretation
000213	COMPAR: Unknown relational (string) operator.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000214	DONBRK: Unknown case for putlin.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000215	`01: Bad length case for numeric `02.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Name of the routine `02 = name of the variable
000216	GARRAY: Unknown command number.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000217	GCMDS: No Gfunc found.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000218	GDOC: Unknown document type.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000219	GLET: Bad operator.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000220	GLET: Stack incorrect for expression - arg `01.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Number of the argument
000221	GLET: Unknown operator type.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000222	GLET: Unknown operator in expression.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.

Error Number	Error Message	Suggestion/Interpretation
000223	GPARS: Column not SCOL, TCOL or NCOL type.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000224	GPARS: Bad parameter format: `01 =`02=	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Internal command format string `02 = Bad format field found
000225	GPARS: No end of required word in parfint: `01	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Internal command format string
000226	GPARS: Bad parfint entry: `01	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Internal command format string
000227	GPARS: Bad parameter string.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000228	GPARS: Repeat count bad: `01	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Internal command format string
000229	GPARS: Only a,b,8,9 allowed for repeats: `01	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Internal command format string
000230	GPARS: Missing required x: `01	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Internal command format string
000231	GPARS: Bad type in 'ckvrpr()'.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.

Error Number	Error Message	Suggestion/Interpretation
000232	GPROC: No Gfunc found.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000233	GRDWRT: Unknown command number.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000234	GSHOW: Unknown SHOW option.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000235	PGMPARS: 'addvar()' passed maxlen but not column.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000238	PGMPARS: '01' passed invalid parameter number: '02.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Routine name '02 = Invalid parameter number
000239	PGMPARS: 'fxclrf()' encountered bad column reference type: '01.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Internal variable type code
000240	PLCMNT: 'getplc()' passed invalid element number: '01.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Invalid element number
000241	RDPGM: Command array size exceeded (change COMDMAX to at least '01).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Maximum internal command number supported.
000242	RDPGM: Bad match adding internal variable: '01	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Internal variable name

Error Number	Error Message	Suggestion/Interpretation
000243	RDPGM: No cmdget function found for BEGIN_S.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000244	Function `01 not included in run-time package.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Name of the SQR routine
000245	SETSQL: Could not find variable `01', in Run Time.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Variable name
000248	SIFWHL: Command number incorrect.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000249	SPINIT: Bad parameters.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000251	DBFFIX: DBDATLEN returned out of range status.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000252	DPRPST: Error converting Sybase type for EXECUTE.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000254	SETSQL: Could not find variable entry in list.	(Oracle) This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000255	DBDESC: SQLD not = number of select columns.	(DB2, Informix, Ingres) This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
000256	DBFETCH: Unknown variable dbtype encountered: `01 (`02)	(DB2, Informix, Ingres) This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Variable name `02 = Unknown database type

Error Number	Error Message	Suggestion/Interpretation
000257	WRITE_SPF: Unknown code encountered: `01	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Unknown SPF code
000258	`01: Cannot find LOAD-LOOKUP table: `02	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Name of the routine `02 = Name of the table
000259	PGMPARS: "`01' called with wrong variable "`02'	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Name of the routine `02 = Name of the variable
000260	SQTMGT: Could not find 'vars' entry with 'nvars' index of "`01'.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Index into nvars table
000261	MODIFYVAR: Attempt to change variable which is not xVAR (`01).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = name of the variable
000262	MODIFYVAR: Incompatible variable types (`01) and (`02).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Variable type (from) `02 = Variable type (to)
001100	Out of query arguments; use -Mfile to increase QUERYARGS.	This is the total number of variable references (\$Var, #Var, &Col) allowed in the context of a BEGIN-SQL or BEGIN- SELECT command. Use the -Mfile flag on the command line to specify a file containing an entry that increases the currently defined value.
001201	Cannot open the argument file: "`01'. (`02): `03	Depends on the system error message. `01 = Name of the file `02 = System error code `03 = System error message

Error Number	Error Message	Suggestion/Interpretation
001202	Cannot close the argument file. (`01): `02	Depends on the system error message. `01 = System error code `02 = System error message
001203	Cannot open the - MFile: "`01". (`02): `03	Depends on the system error message. `01 = Name of the file `02 = System error code `03 = System error message
001204	Minimum value for "`01" in the -MFile is `02.	Correct the -Mfile entry. `01 = Keyword in question `02 = Minimum value allowed
001205	Maximum value for "`01" in the -Mfile is `02.	Correct the -Mfile entry. `01 = Keyword in question `02 = Maximum value allowed
001206	Invalid -MFile entry: "`01".	Correct the -Mfile entry. `01 = The line from the -Mfile
001207	Cannot close the - MFile. (`01): `02	Depends on the system error message. `01 = System error code `02 = System error message
001209	The minimum value for "`01" (`02) is `03.	Value out of range. `01 = Entry name `02 = Specified value `03 = Minimum value
001210	The maximum value for "`01" (`02) is `03.	Value out of range. `01 = Entry name `02 = Specified value `03 = Maximum value
001211	The value for "`01" (`02) is not an integer number.	Value must be a integer value. `01 = Entry name `02 = Specified value
001300	Bind list does not match query (do not use '@__p' string).	(All but Ingres) SQR reserves the variable names that start with "@__p" for internal use. Edit the source code and use different variable names.
	Bind list does not match query (do not use '?' character).	(Ingres) SQR reserves the variable name "?" for internal use. Edit the source code and use different variable names.
001301	Forward references not permitted in select list bind variables.	Within the body of BEGIN-SQL paragraphs, forward references to &column names are not permitted. Move the BEGIN-SQL paragraph after the &column definition.

Error Number	Error Message	Suggestion/Interpretation
001302	SQL buffer too small; use -Mfile to increase SQLSIZE.	The SQL statement exceeds the size of the internal SQL buffer. Use the -Mfile flag on the command line to specify a file containing an entry that increases the currently defined value.
001303	Error in SQL (perhaps missing &name after expression):	The database server has determined that the SQL statement is in error. The actual error text from the server follows this message. Correct the SQL statement.
001304	Check SELECT columns, expressions and 'where' clause for syntax.	The database server has determined that the SQL statement is in error. The actual error text from the server follows this message. Correct the SQL statement.
001305	CMPSQL: Unknown data type in database: `01.	Contact technical support with the version of the database you are connected to. `01 = Datatype in question
001306	Bind value too large (IMAGE, TEXT not allowed).	IMAGE and TEXT data types cannot be used as bind variables. Modify your SQL statement to use other columns to perform the same selection logic.
001307	CMPSQL: DBDEFN failed.	(ODBC, Oracle, Informix, Ingres, SQLBase) This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
001308	`01: Could not bind column `02.	(ODBC, Oracle, Informix, Ingres, SQLBase) This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = Name of the SQR routine `02 = Name of the column
001309	The type for '&`01' (`02) does not match the type from the database (`03).	Correct the source code. `01 = Name of the column/expression pseudonym `02 = User specified type `03 = Database type
001400	Only numerics allowed for arithmetic.	Only #numeric variables, &columns, and literals are permitted in the arithmetic commands. Correct the source code.
001401	Optional qualifier is ROUND=n (0-`01).	Correct the syntax. `01 = Maximum value for ROUND=
001402	Optional qualifiers for DIVIDE are ON- ERROR={HIGH ZERO} and ROUND=n.	Correct the syntax.

Error Number	Error Message	Suggestion/Interpretation
001403	Attempting division by zero.	Use the ON-ERROR = HIGH ZERO option to prevent this error from halting the program.
001404	Bad number of digits to ROUND or TRUNC (0-15).	Correct the syntax.
001405	WARNING: The ROUND or TRUNC qualifier is greater than the number's precision.	Correct the syntax.
001500	Array element out of range ('01) for array '02' on line '03.	Correct the source logic. '01 = Element number passed '02 = Name of the array '03 = Program line number
001501	Field element out of range ('01) for array '02', field '03', on line '04.	Correct the source logic. '01 = Element number passed '02 = Name of the array '03 = Name of the field '04 = Program line number
001502	WARNING: Attempting division by zero on line '01. Array field '02' unchanged. Run continuing...	The ARRAY-DIVIDE command has attempted division by zero. The division has been ignored; the result field is unchanged. Add logic to account for this possibility. '01 = Program line number '02 = Name of field
001601	'FILL' not appropriate for numeric data.	The FILL argument to the PRINT command may be used only for text fields. Move the #numeric variable to a \$string variable, and then print the string variable.
001700	Report '01': Columns must be between 1 and the page width ('02).	The specified value is wider than the width of the page. Correct the source line. '01 = Name of the current report '02 = Page width
001702	Report '01': GOTO- TOP='02 must be between 0 and the page depth ('03).	The value specified on the GOTO-TOP argument of the NEXT-COLUMN command was either less than 1 or greater than the page depth. Correct the source line. '01 = Name of the current report '02 = Goto-Top value '03 = Page width
001703	Report '01': ERASE- PAGE='02 must be between 0 and the page depth ('03).	The line number specified on the ERASE-PAGE argument of the NEXT-COLUMN command is greater than the page depth. Correct the source line. '01 = Name of the current report '02 = Erase-Page value '03 = Page width

Error Number	Error Message	Suggestion/Interpretation
001704	Report '01': The NEXT-COLUMN command is not legal in the '02 section with the qualifier AT-END=NEWPAGE.	Correct the source line. '01 = Name of the current report '02 = Name of the section
001705	Report '01': Column number '02 is not defined.	The column number specified with the USE-COLUMN command is greater than the highest column defined in the COLUMNS command. Correct the source line. '01 = Name of the current report '02 = Column number
001800	Format for CONNECT: username/password [ON-ERROR=procedure[(arg1[,argi]...)]]	Correct the syntax.
001801	Cannot use CONNECT while SQL statements are active.	Correct the program logic to ensure that all BEGIN-SELECT paragraphs have completed before executing the CONNECT command.
001802	Logoff failed prior to CONNECT.	The database server returned an error while trying to log off from the database. SQR ends the program run since it cannot continue.
001803	CONNECT failed. Perhaps username/password incorrect.	The specified connectivity information is incorrect or there might have been a network failure. Use the ON-ERROR flag to trap any errors during the program run; otherwise SQR ends the program run.
001804	Sybase extensions SET and SETUSER not permitted in SQR.	Remove SET and SETUSER from the source.
001805	USE allowed once in SETUP section only, not in BEGIN-SQL. Elsewhere, specify db.[user].table...	Correct the source.
001806	Out of query space. Use -Mfile to increase QUERIES.	The number of SQL statements has been exceeded. Use the -Mfile flag on the command line specify a file that contains an entry that increases a greater value than is currently defined.
001807	The requested database connection ('01) is already active.	The -Cnn value specified is being used by another BEGIN-SELECT paragraph that is currently selecting data. Use another connection number. '01 = Connection number
001808	Cannot find inactive database cursor. Program too large.	Too many BEGIN-SELECT and BEGIN- SQL paragraphs are active at the same time. Reduce the complexity of the program.

Error Number	Error Message	Suggestion/Interpretation
001809	Database commit failed.	(DB2, Ingres, ODBC, Oracle, SQLBase) This is an error that should never occur during normal operations. Record the steps leading up to the error and contact your system administrator.
001810	Database rollback failed.	(DB2, Ingres, ODBC, Oracle, SQLBase) This is an error that should never occur during normal operations. Record the steps leading up to the error and contact your system administrator.
001811	Cannot open database cursor.	(Ingres, ODBC, Oracle, HPIW) This is an error that should never occur during normal operations. Record the steps leading up to the error and contact your system administrator.
	Cannot open database cursor.	(SQLBase) This is an error that should never occur during normal operations. Record the steps leading up to the error and contact your system administrator. '01 = SQL statement
001901	Variable for date-time must begin with '&'.	Correct the syntax.
001913	Format code must be SYYYYY when specifying signed year.	Correct the edit mask.
001914	Bad input data ('01) for edit mask: '02'.	Correct the input. '01 = Data being converted '02 = Edit mask
001915	Year cannot be zero.	Correct the date.
001916	Year must be between -4713 and 9999 inclusive.	Correct the date.
001917	Ambiguous date-time.	Correct the date.
001918	'01' is not a valid date part.	Correct the date part. '01 = Date part.
001919	Invalid day of week.	Correct the date.
001920	Format code cannot appear in date input format: '01'.	Correct the edit mask. '01 = Improper format characters.
001921	Bad date mask starting at: '01'.	Correct the edit mask. '01 = Improper format characters.
001922	Seconds past midnight must be between 0 and 86399.	Correct the date.
001923	Seconds must be between 0 and 59.	Correct the date.
001924	Minutes must be between 0 and 59.	Correct the date.
001925	Month must be between 1 and 12.	Correct the date.

Error Number	Error Message	Suggestion/Interpretation
001926	Day must be between 1 and `01.	Correct the date.
001927	Hour must be between 1 and 12.	Correct the date.
001928	Hour must be between 0 to 23.	Correct the date.
001929	HH24 precludes the use of meridian indicator.	Correct the edit mask.
001930	HH12 requires meridian indicator.	Correct the edit mask.
001931	Day of year must be between 1 and 365 (366 for leap year).	Correct the date.
001932	Date string too long.	Correct the date.
001933	The month (`01) is not valid for the current locale or database.	Correct the date. `01 = Name of the month.
001934	The format mask must be a literal when the date-time is not loaded into a variable.	Correct the format mask. The format mask must be a literal when the date-time is not loaded into a variable.
001935	Date-time format too long.	Correct the format mask.
001936	Bad date-time format.	Correct the format mask.
001937	Bad SQL for default date-time. (Table sysssql.systables required for syntax.)	(SQLBase) Possibly the format mask needs to be corrected; otherwise, there is a problem with the database server.
001937	Bad SQL for default date-time. (Table DUAL required for syntax.)	(Oracle) Possibly the format mask needs to be corrected; otherwise, there is a problem with the database server.
	Bad SQL for default date-time. (Table DUAL required for syntax.)	(DB2) Possibly the format mask needs to be corrected; otherwise, there is a problem with the database server.
001938	Cannot recompile sql.	A fatal error relating to the SQL statement used to retrieve the date-time was encountered. Record the steps leading up to the error and contact your system administrator.
001939	Problem executing cursor.	A fatal error relating to the SQL statement used to retrieve the date-time was encountered. Record the steps leading up to the error and contact your system administrator.
001940	Error fetching row.	A fatal error relating to the SQL statement used to retrieve the date-time was encountered. Record the steps leading up to the error and contact your system administrator.

Error Number	Error Message	Suggestion/Interpretation
001941	Cannot redefine variable addresses.	A fatal error relating to the SQL statement used to retrieve the date-time was encountered. Record the steps leading up to the error and contact your system administrator.
001942	The date '01' is not in the format SYYYYMMDD[HH24[MI[SS[N NNNN]]]].	When specifying an SQR date, at a minimum, the date must be specified; the time is optional. '01 = The invalid date.
001943	The date '01' is not in one of the accepted formats listed below: MM/DD/YYYY [BC AD] [HH:MI[:SS[.NNNN N]] [AM PM]] MM-DD-YYYY [BC AD] [HH:MI[:SS[.NNNN N]] [AM PM]] MM.DD.YYYY [BC AD] [HH:MI[:SS[.NNNN N]] [AM PM]] SYYYYMMDD[HH24[MI[SS[N NNNN]]]]	The date specified with the INPUT command was not in one of the default formats. Please reenter the date in a valid format. '01 = The invalid date.
001944	The date '01' is not in the format specified by SQR_DB_DATE_FORMAT or in one of the accepted formats listed below: DD-MON-YY SYYYYMMDD[HH24[MI[SS[N NNNN]]]]	(Oracle) The date was not in one of the expected formats for this database. '01 = The invalid date.
001944	The date '01' is not in the format specified by SQR_DB_DATE_FORMAT or in one of the accepted formats listed below: Mon DD YYYY [HH:MI[:SS[.NNN]] [AM PM]] Mon DD YYYY [HH:MI[:SS[:.NNN]] [AM PM]] YYYYMMDD [HH:MI[:SS[.NNN]] [AM PM]] YYYYMMDD [HH:MI[:SS[:.NNN]] [AM PM]] SYYYYMMDD[HH24[MI[SS[N NNNN]]]]	(Sybase) The date was not in one of the expected formats for this database. '01 = The invalid date.

Error Number	Error Message	Suggestion/Interpretation
001944	The date '01' is not in the format specified by SQR_DB_DATE_FORMAT or in one of the accepted formats listed below: Mon DD YYYY [HH:MI[:SS[.NNN]] [AM PM]] Mon DD YYYY [HH:MI[:SS[.NNN]] [AM PM]] YYYYMMDD [HH:MI[:SS[.NNN]] [AM PM]] YYYYMMDD [HH:MI[:SS[.NNN]] [AM PM]] YYYYMMDD[HH24[MI[SS[N NNNN]]]]	(ODBC) The date was not in one of the expected formats for this database. '01 = The invalid date.
001944	The date '01' is not in the format specified by SQR_DB_DATE_FORMAT or in one of the accepted formats listed below: YYYY-MM-DD [HH24:MI[:SS[.NNN NNN]] YYYYMMDD[HH24[MI[SS[N NNNN]]]]	(Red Brick) The date was not in one of the expected formats for this database. '01 = The invalid date.
	The date '01' is not in the format specified by SQR_DB_DATE_FORMAT or in one of the accepted formats listed below: YYYY-MM-DD HH:MI:SS.NNN YYYYMMDD[HH24[MI[SS[N NNNN]]]]	(Informix) The date was not in one of the expected formats for this database. '01 = The invalid date.
	The date '01' is not in the format specified by SQR_DB_DATE_FORMAT or in one of the accepted formats listed below: YYYY-MM-DD[- HH.MI.SS[.NNNNNN]] MM/DD/YYYY DD.MM.YYYY YYYYMMDD[HH24[MI[SS[N NNNN]]]]	(DB2) The date was not in one of the expected formats for this database. '01 = The invalid date.

Error Number	Error Message	Suggestion/Interpretation
	The date '01' is not in the format specified by SQR_DB_DATE_FORMAT or in one of the accepted formats listed below (Times can also use "."): DD-MON-YY[YY][HH[:MI[:SS[:NNNNN N]]]] [AM PM] MM-DD-YY[YY][HH[:MI[:SS[:NNNNN N]]]] [AM PM] DD.MM.YYYY[HH[:MI[:SS[:NNNNN N]]]] [AM PM] YYYY-MM-DD[HH[:MI[:SS[:NNNNN N]]]] [AM PM] SYYYYMMDD[HH24[MI[SS[N NNNNN]]]]	(SQLBase) The date was not in one of the expected formats for this database. '01 = The invalid date.
	The date '01' is not in the format specified by SQR_DB_DATE_FORMAT or in one of the accepted formats listed below: DD-MON-YYYY[HH:MI[:SS]] MM/DD/YYYY[HH:MI[:SS]] MM-DD-YYYY[HH:MI[:SS]] YYYY.MM.DD[HH:MI[:SS]] YYYY_MM_DD[HH:MI[:SS]] SYYYYMMDD[HH24[MI[SS[N NNNNN]]]]	(Ingres) The date was not in one of the expected formats for this database. '01 = The invalid date.
001945	SQR does not support dates before '01'.	SQR does not support dates before the one specified in the message. '01 = Smallest date
001946	The date variables are incompatible with each other.	The SQR function references two date variables which cannot logically be used together (for example, DateDiff of 'date- only' and 'time-only' dates).
002000	Procedure name used more than once: '01'.	Give the procedure a unique name. '01 = Procedure name
002001	Could not find procedure: '01'.	Check for a misspelled procedure name. '01 = Procedure name
002002	DO arguments do not match procedure's.	The argument lists for the DO and BEGIN-PROCEDURE commands must match in both type and count. Correct the source line.
002003	DO argument must be \$string or #number to accept returned value.	Correct the syntax.
002100	Edit string too long.	The edit mask must be less than 255 characters. Reduce the length of the edit mask.

Error Number	Error Message	Suggestion/Interpretation
002101	Bad numeric 'edit' format: `01	The numeric edit mask contains an invalid character. See the PRINT command for the valid numeric edit mask characters. `01 = Invalid character
002103	DOLLAR-SYMBOL must be a single alphanumeric character or its decimal value enclosed in brackets: <nnn>.	Correct the syntax.
002104	DOLLAR-SYMBOL cannot be any of the following characters: `01	Correct the syntax. `01 = List of invalid characters
002106	MONEY-SYMBOL must be a single alphanumeric character or its decimal value enclosed in brackets: <nnn>.	Correct the syntax.
002107	MONEY-SYMBOL cannot be any of the following characters: `01	Correct the syntax.
002200	ENCODE string too large; maximum is `01.	Break up the ENCODE command. `01 = Maximum length of an ENCODE string supported by this version of SQR
002300	EXIT-SELECT failed.	The database command to cancel the query returned an error. Try running the SQR program again. The error could be related to a network or server problem. If the error persists, contact your system administrator.
002301	EXIT-SELECT valid only within SELECT paragraph.	Remove the EXIT-SELECT command.
002400	Duplicate label's - do not know which one to GOTO.	Labels must be unique within the section or paragraph where they are defined. Give each label a unique name.
002401	(Labels must be in same section or paragraph as GOTO.) Cannot find a matching label for GOTO command.	Check the source code.
002500	Error getting INPUT.	The C routine "fgets()" returned an error and SQR ends the program run.
002501	Unknown INPUT datatype: type={char number integer date}	Correct the syntax.
002502	INPUT STATUS= must reference #variable.	Correct the syntax.
002503	Unknown qualifier for INPUT.	Correct the syntax.
002506	Too long. Maximum `01 characters.	The response to the INPUT statement was too long. Re-enter the data. `01 = Maximum characters allowed

Error Number	Error Message	Suggestion/Interpretation
002507	Incorrect. Format for floating point number: [+ -]99.99[E99]	Invalid number was entered for an INPUT request. Re-enter the data.
002508	Incorrect. Format for integer: [+ -]999999	Invalid integer was entered for an INPUT request. Re-enter the data.
002510	A format mask can only be specified when TYPE=DATE is used.	Correct the syntax.
002511	The format mask cannot be stored in a date variable.	Correct the syntax.
002512	The input variable type does not match the TYPE qualifier.	Correct the syntax.
002513	Number too large for INTEGER. Valid range is -2147483648 to 2147483647.	The number was too large to be stored as an integer. Values are from -2147483648 to 2147483647. Re-enter the data.
002514	Enter a date in one of the following formats: MM/DD/YYYY [HH:MI[:SS[.NNNNN N]] [AM PM]] MM-DD-YYYY [HH:MI[:SS[.NNNNN N]] [AM PM]] MM.DD.YYYY [HH:MI[:SS[.NNNNN N]] [AM PM]] SYYYMMDD[HH24[MI[SS[N NNNNN]]]]	The date cannot be blank. Enter a date in one of the specified formats.
002515	`01 required user interaction but user interaction was disabled by the -XI command line flag.	The specified command required user interaction, but user interaction was disabled by the -XI command line flag. `01 = Name of the command
002600	LOAD-LOOKUP table `01' has not been defined.	Add a LOAD-LOOKUP command. `01 = Load lookup table name
002601	Missing value for `01= in LOAD-LOOKUP.	Correct the syntax. `01 = Name of missing required parameter
002602	Bad value for `01= in LOAD-LOOKUP.	Correct the syntax. `01 = Name of the parameter
002603	LOAD-LOOKUP `01= cannot reference a variable in the Setup section.	Either move the LOAD-LOOKUP command from the Setup section or remove the variable reference. `01 = Name of the parameter
002604	LOAD-LOOKUP names must be unique.	Give each LOAD-LOOKUP array a unique name.

Error Number	Error Message	Suggestion/Interpretation
002605	Cannot compile SQL for LOAD-LOOKUP table "01'.	The database server returned an error while trying to compile the SQL statement needed to process the LOAD-LOOKUP command. Check the column and table names. Also check the WHERE= clause for errors. `01 = Load lookup table name
002606	Could not set up cursor for LOAD-LOOKUP table "01'.	The database server returned an error while trying to compile the SQL statement needed to set up the LOAD-LOOKUP command. Check the column and table names. Also check the WHERE= clause for errors. `01 = Load lookup table name
002607	Problem executing the cursor for LOAD- LOOKUP table "01'.	The database server returned an error while trying to execute the SQL statement needed to process the LOAD-LOOKUP command. `01 = Load lookup table name
002609	Integers only allowed in numeric lookup keys.	Correct the source line.
002610	Numeric lookup keys must be <= `01 digits.	Correct the source line. `01 = maximum length supported
002611	Bad return fetching row from database in LOAD-LOOKUP table "01'.	The database server returned an error while fetching the data. `01 = Load lookup table name
002612	Field return code for `01 = `02 Bad return code after fetch in LOAD-LOOKUP table "03'.	(SQLBase) The database server returned an error while fetching the data. `01 = Columns name `02 = Return code `03 = Load lookup table name
002613	Loading "01' lookup table ...	This message can be inhibited by using the QUIET argument on the LOAD-LOOKUP command. `01 = Name of the load lookup table `02 = Number of rows loaded
002615	Warning: `01 duplicate keys found in "02' lookup table.	This message can be inhibited by using the QUIET argument on the LOAD-LOOKUP command. `01 = Number of duplicate keys `02 = Name of the load lookup table
002616	LOAD-LOOKUP `01= must reference a numeric variable or literal.	Correct the source line. `01 = Name of the parameter
002617	LOAD-LOOKUP `01= must reference a string variable or literal.	Correct the source line. `01 = Name of the parameter

Error Number	Error Message	Suggestion/Interpretation
002618	LOAD-LOOKUP `01= variable `02' has not been defined.	Correct the source line. `01 = Name of the parameter `02 = Name of the undefined variable
002619	LOAD-LOOKUP cannot support `01 rows; maximum is `02.	Reduce the ROWS= value. `01 = ROWS= value `02 = Maximum value allowed
002620	`01 command not allowed with -XL option in effect.	Either use the #IF command to conditionally compile the program when -XL is being used or do not execute this SQR report with the -XL option. `01 = SQR command
002700	Line to stop erasing for 'NEW-PAGE' is larger than the page depth.	Correct the source line.
002800	'ON-BREAK' not appropriate for numeric data.	The ON-BREAK argument to the PRINT command may be used only for text fields. Move the #numeric variable to a \$string variable, and then print the \$string variable.
002801	SET= and LEVEL= must be >= zero when indicated.	Correct the source line.
002802	Cannot use old style PROCEDURE= with BEFORE= or AFTER=.	Correct the syntax.
002803	Out of ON-BREAKS; use -Mfile to increase ONBREAKS.	Use the -Mfile flag on the command line to specify a file containing an entry that increases the currently defined value.
002804	SET= must be same for all ON-BREAKs in Select.	All the ON-BREAKS in a query must belong to the same SET. Use SET= to differentiate between ON-BREAKs in different queries. Correct the source line.
002805	ON-BREAK with BEFORE or AFTER must be inside Select.	Correct the source line.
002806	SAVE= must be a \$string variable.	Correct the syntax.
002900	Record :types are FIXED, VARY or FIXED_NOLF (default is VARY).	Correct the syntax.
002901	STATUS variable for `01 must be #Numeric.	Correct the syntax. `01 = SQR command affected
002902	OPEN missing required qualifiers: RECORD={rec_len} FOR-READING FOR- WRITING FOR-APPEND	Correct the syntax.

Error Number	Error Message	Suggestion/Interpretation
002903	Too many external files opened; maximum is `01.	Reduce the number of open external files needed by the program. `01 = Maximum number of open external files supported by this version of SQR
002904	File number already opened.	Check your program logic.
002905	Cannot open file `01' AS `02.(`03): `04	SQR quits.. `01 = Filename `02 = File number `03 = System error code `04 = System error message
002906	Cannot close file `01. (`02): `03	SQR quits. `01 = File number `02 = System error code `03 = System error message
002907	Problem closing user file(s) at the end of run.	This message may indicate system problems.
002908	Warning: Cannot CLOSE file `01 -- file not opened.	While not an error, this message indicates a problem with your SQR code. `01 = File number
003000	PAGE-NUMBER strings too long.	The pre-and post-PAGE-NUMBER strings must be less than 74 characters. Correct the source line.
003100	Cannot find document marker referenced in POSITION command.	Defines the specified @ marker in a BEGIN-DOCUMENT paragraph. Check for a misspelled @ marker name.
003101	Only 'COLUMNS nn...' allowed after document marker in POSITION command.	Correct the syntax.
003200	Specified file number not opened for reading.	Files must be opened for reading in order to use the READ command with them. Correct the program logic.
003201	Line `01: Error reading the file. (`02): `03	`01 = Program line number `02 = System error code `03 = System error message
003202	Specified file number not opened for writing.	Files must be opened for writing in order to use the WRITE command with them. Correct the program logic.
003203	Line `01: Error writing the file. (`02): `03	`01 = Program line number `02 = System error code `03 = System error message

Error Number	Error Message	Suggestion/Interpretation
003204	Length of variables exceeds record length.	The total of the lengths indicated in the command must be less than the RECORD= argument used on the OPEN command. Check for a typographical error or recalculate the RECORD= value.
003205	Numeric binary transfer allowed with FIXED or FIXED_NOLF records only.	By default, all files are opened in VARY (variable length) mode, thus prohibiting the transfer of numeric binary data. Add the:FIXED or FIXED_NOLF option to the RECORD= argument on the appropriate OPEN command.
003206	Command not complete.	Correct the syntax.
003207	File number must be a numeric literal, variable, or column.	Correct the syntax.
003208	Missing required :length in READ command.	Correct the syntax.
003209	Bad :length for READ or WRITE command.	Correct the syntax.
003210	\$String or #numeric variables required for READ.	Correct the syntax.
003211	#Numeric variables and literals must have :length of 1, 2 or 4 bytes.	Correct the syntax.
003212	#Numeric variables and literals on CDC may only have :length of 1 or 3 bytes.	Correct the syntax.
003213	:Length not allowed for \$date variables, length of 18 is assumed.	Correct the syntax.
003300	Unknown qualifier for STOP.	Correct the syntax.
003301	Program stopped by user request.	Informational message.
003400	Wrap not appropriate for numeric data.	The WRAP argument to the PRINT command may be used only for text fields. Move the #numeric variable to a \$string variable first, and then print the \$string variable.
003401	Max `01 chars/line for reverse WRAP.	Reduce the number of characters specified. `01 = Maximum number of characters supported by this version of SQR.
003402	Max `01 chars/line for WRAP with ON= or STRIP=	Reduce the number of characters specified. `01 = Maximum number of characters supported by this version of SQR
003403	Bad <number> in WRAP qualifier.	The number inside the angled brackets must be a valid ASCII number (1 - 255). Correct the source line.

Error Number	Error Message	Suggestion/Interpretation
003404	Missing '>' in WRAP qualifier.	A leading "<" in the ON= or STRIP= qualifier indicates that a numeric value is following, which must be ended by a closing ">". Correct the source line.
003405	The value for '`01' ('02) must be `03 0.	The value specified for the specified qualifier is invalid. Correct the program logic. `01 = Qualifier name `02 = Value encountered `03 = Relation to zero (<,<=,>,>=)
003500	PUT, GET or ARRAY- xxxx command incomplete. Required word missing.	Correct the syntax.
003501	Did not find end of literal.	The ending quote character (') was not found at the end of the literal. Add the ending quote character.
003502	Literal too long.	Literal strings can be up to 256 characters long. Break up the literal into smaller pieces and combine using the LET command.
003503	Unknown variable type.	Variable names must begin with \$, #, or &. Correct the source line.
003504	Cannot find 'array_name (#element)'.	The element number was not specified. Correct the source line.
003505	'(#Element)' variable not found for array.	Each GET or PUT command must indicate the element or row number to access in the array. Correct the source line.
003506	Array specified not defined with CREATE-ARRAY.	Use the CREATE-ARRAY command to define each array before referencing that array in other commands. Check for a misspelled array name.
003507	Bad element reference for array (#variable 123).	The element number is larger than the number of rows defined in the CREATE- ARRAY command. Check program logic to make sure that the element number was not inadvertently changed.
003508	Did not find ending ')' for field.	The "occurs" number for an array field is missing a right parenthesis. Correct the source line.
003509	Field not defined in array: `01	Check for a misspelled field name against the CREATE-ARRAY command. `01 = Undefined field name
003510	More variables than fields specified in array command.	The ARRAY command must not have more variables listed to the left of the array name than there are matching fields defined for the array. Check against the CREATE-ARRAY command.

Error Number	Error Message	Suggestion/Interpretation
003511	More variables in command than fields in array.	The ARRAY command must not have more variables listed to the left of the array name than there are matching fields defined for the array. Check against the CREATE-ARRAY command.
003512	Only numeric variables and fields allowed with array arithmetic commands.	The ARRAY-ADD, ARRAY-SUBTRACT, ARRAY-MULTIPLY, and ARRAY-DIVIDE commands may have only numeric variables or literals as the source fields. Move the string data into a #numeric variable and then reference the #numeric variable.
003513	GET can only be used with \$string or #numeric variables.	You can move array fields only into \$string variables or #numeric variables. Correct the source line.
003514	PUT and GET variables must match array field types.	When moving data into or out of arrays, the source or destination variables must match the array fields in type. CHAR fields can be stored into/from strings, NUMBER fields into/from numeric variables. Check the CREATE-ARRAY command.
003515	More fields than variables found in array command.	The ARRAY command must not have more variables listed to the left of the array name than there are matching fields defined for the array. Check against the CREATE-ARRAY command.
003516	Too many arrays defined; maximum is '01.	Reduce the number of arrays needed by the program. '01 = Maximum number of arrays supported by this version of SQR
003517	Missing '=specifier' in qualifier: '01	Correct the syntax. 01 = Name of missing required parameter
003518	Duplicate array name: '01	Change the name of the array. '01 = Array name in question
003519	Too many fields defined; maximum is '01.	Reduce the number of fields. '01 = Maximum number of fields allowed per array
003520	Missing ':type' in CREATE-ARRAY FIELD= '01	Correct the syntax. '01 = The name of the field
003521	Duplicate FIELD name: '01	Change the name of one of the fields. '01 = The name of the field
003522	Optional :nn for FIELD must be between 1 and 64K.	Correct the source line.
003523	CREATE-ARRAY FIELDS :type must be one of the following: '01	Correct the syntax.

Error Number	Error Message	Suggestion/Interpretation
003525	Missing NAME= in CREATE-ARRAY.	Correct the syntax.
003526	Missing or incorrect SIZE= in CREATE- ARRAY.	Correct the syntax.
003527	Missing FIELD= statements in CREATE- ARRAY.	Correct the syntax.
003528	Array dimensioned too large for PC in CREATE-ARRAY.	On PC-based systems, the maximum allocation that can be made is 65520 characters. The array as specified would exceed this limit. Reduce the number of entries.
003529	Missing or invalid initialization value for field `01.	Correct the syntax. 01 = Name of the field
003600	Missing 'ask' variable name.	Correct the syntax.
003601	Out of substitution or #DEFINE variables; use -Mfile to increase SUBVARS.	Use the -Mfile flag on the command line to specify a file containing an entry that increases the currently defined value.
003603	WARNING: Substitution variables do not vary when saved with run-time.	Informational message.
003605	No substitution variable entered.	The C routine "fgets()" returned an error and SQR ends the program run.
003700	Did not find end of paragraph: `01	Missing the END-paragraph command to match the specified paragraph. Correct the source file. `01 = BEGIN-paragraph in question
003701	Invalid command.	Check for a misspelled command.
003702	Command not allowed in this section: `01	Correct the syntax. `01 = Offending command name
003703	Paragraph not allowed inside procedure.	The BEGIN-paragraph command is not allowed here. Check your SQR code for a misplaced paragraph.
003704	Missing procedure name.	Correct the syntax.
003705	Extra argument found.	Correct the syntax.
003706	Missing Comma.	Correct the syntax.
003707	Bad Argument List.	The DO or BEGIN-PROCEDURE command has an error in its argument list, possibly extra characters after the final right parentheses. Correct the source line.

Error Number	Error Message	Suggestion/Interpretation
003708	Empty Argument.	The DO or BEGIN-PROCEDURE command has an error in its argument list, possibly two commas in a row inside the parentheses. Correct the source line.
003709	Only \$string and #number variables allowed for BEGIN- PROCEDURE parameters.	Correct the syntax.
003710	Unknown argument type.	An argument in a DO or BEGIN-PROCEDURE command is incorrect. Check for a misspelled variable type.
003711	Indicate :\$string or #number returned values in BEGIN- PROCEDURE only.	Correct the syntax.
003712	Missing).	Correct the syntax.
003713	`01 paragraph not allowed with -XL option in effect.	Either use the #IF command to conditionally compile the program when -XL is being used or do not execute this SQR report with the -XL option. `01 = Name of the BEGIN-paragraph
003714	Bad database connection number.	The -Cnn value must be a non-zero value. Correct the source line.
003715	Did not find end of paragraph: `01 (No 'from...' clause found.)	Correct the source code. `01 = BEGIN-command in question
003716	Error in SQL statement.	The database server has determined that the SQL statement is in error. The actual error text from the server follows this message. Correct the SQL statement.
003717	Extra characters after expression continuation.	Remove the extra characters after the dash.
003718	Did not find end of expression.	An expression in a SELECT list must end with either a &column variable or a position parameter "(Row,Col,Len)". Correct the source line.
003719	Columns names and expressions must be unique or be given unique pseudonyms (&name).	Columns retrieved from the database are assigned names by prepending an "&" to the beginning of the name. You are trying to select the same &column name more than once. Change the assigned &column name by using an alias after the name.
003720	Bad number specified for 'LOOPS=' on 'BEGIN-SELECT; Maximum is 32767'.	If your program logic requires that you stop processing after more than 32767 rows have been retrieved, you could count the rows manually and use the EXIT-SELECT command to break out of the SELECT loop.

Error Number	Error Message	Suggestion/Interpretation
003721	Bad param found on 'BEGIN-SELECT' line; Format is: BEGIN-SELECT [DISTINCT] [-Cnn] [LOOPS=nn] [ON-ERROR=procedure[(arg1[,argi,...])]] [-DB=database]	(HPIW) Correct the syntax.
003721	Bad param found on 'BEGIN-SELECT' line; Format is: BEGIN-SELECT [DISTINCT] [-Cnn] [- Bnn] [LOOPS=nn] [ON-ERROR=procedure[(arg1[,argi,...])]]	(DB2) Correct the syntax.
003721	Bad param found on 'BEGIN-SELECT' line; Format is: BEGIN-SELECT [DISTINCT] [-Cnn] [LOOPS=nn] [ON-ERROR=procedure[(arg1[,argi,...])]]	(Informix) Correct the syntax.
003721	Bad param found on 'BEGIN-SELECT' line; Format is: BEGIN-SELECT [DISTINCT] [-Cnn] [LOOPS=nn] [ON-ERROR=procedure[(arg1[,argi,...])]]	(Ingres) Correct the syntax.
003721	Bad param found on 'BEGIN-SELECT' line; Format is: BEGIN-SELECT [DISTINCT] [-Cnn] [LOOPS=nn] [ON-ERROR=procedure[(arg1[,argi,...])]] [-DB=database]	(ODBC) Correct the syntax.
	Bad param found on 'BEGIN-SELECT' line; Format is: BEGIN-SELECT [DISTINCT] [-Cnn] [- Bnn] [LOOPS=nn] [ON-ERROR=procedure[(arg1[,argi,...])]]	(Oracle) Correct the syntax.
	Bad param found on 'BEGIN-SELECT' line; Format is: BEGIN-SELECT [DISTINCT] [-Cnn] [LOOPS=nn] [ON-ERROR=procedure[(arg1[,argi,...])]] [-NR] [-DB=database] [-SORT=nn] [-LOCK=ll] [-INMSG=nn] [-OUTMSG=nn]	(SQLBase) Correct the syntax.

Error Number	Error Message	Suggestion/Interpretation
	Bad param found on 'BEGIN-SELECT' line; Format is: BEGIN-SELECT [DISTINCT] [-Cnn] [- XP] [LOOPS=nnn] [ON-ERROR=procedure[(arg1[,argi]...)]]	(Sybase) Correct the syntax.
	Bad param found on 'BEGIN-SELECT' line; Format is: BEGIN-SELECT [DISTINCT] [-Cnn] [LOOPS=nnn] [ON-ERROR=procedure[(arg1[,argi]...)]] [-DB=database]	(Red Brick) Correct the syntax.
003722	Could not set up cursor.	An error occurred while trying to compile the SQL statement. Look closely at any \$string variable references. Correct the SQL statement or use the ON-ERROR= option to trap the error during the program run.
003723	Problem executing cursor.	An error occurred while trying to execute the SQL statement. Look closely at any \$string variable references. Correct the SQL statement or use the ON-ERROR= option to trap the error during the program run.
003724	Could not exit query loop.	The database command to cancel the query returned an error. Try running the SQR program again. The error could be related to a network or server problem. If the error persists, contact your system administrator.
003725	Bad return fetching row from database.	The database returned an error status for the last row that was fetched, commonly due to the buffer not being large enough. If selecting expressions, make sure that the length of the first expression will be adequate for all rows selected.
003726	Literal in SQL expression missing closing quote.	Literals must be surrounded by single quotes ('). To embed a quote within a literal use two single quotes in sequence ("). Correct the source line.
003727	SQL expression not ended, perhaps parentheses not balanced.	An expression in a SELECT list must end with either a &column variable or a position parameter "(Row,Col,Len)". Correct the source line.
003728	SQL expression not ended, perhaps missing &name.	An expression in a SELECT list must end with either a &column variable or a position parameter "(Row,Col,Len)". Correct the source line.

Error Number	Error Message	Suggestion/Interpretation
003729	SQL expression is missing &name or has unbalanced parentheses.	An expression in a SELECT list must end with either a &column variable or a position parameter "(Row,Col,Len)". Correct the source line.
003730	Incorrect arguments for BEGIN-SQL: [-Cnn] [ON- ERROR=procedure[(arg1[,argi]...)] [-DB=]	(HPIW) Correct the syntax.
003730	Incorrect arguments for BEGIN-SQL: [-Cnn] [ON- ERROR=procedure[(arg1[,argi]...)]	(DB2) Correct the syntax.
003730	Incorrect arguments for BEGIN-SQL: [-Cnn] [ON- ERROR=procedure[(arg1[,argi]...)]	(Informix) Correct the syntax.
003730	Incorrect arguments for BEGIN-SQL: [-Cnn] [ON- ERROR=procedure[(arg1[,argi]...)]	(Ingres) Correct the syntax.
003730	Incorrect arguments for BEGIN-SQL: [-Cnn] [-NR] [ON- ERROR=procedure(arg1[,argi]...)] [-DB=]	(ODBC) Correct the syntax.
	Incorrect arguments for BEGIN-SQL: [-Cnn] [ON- ERROR=procedure[(arg1[,argi]...)]	(Oracle) Correct the syntax.
	Incorrect arguments for BEGIN-SQL: [-Cnn] [-NR] [ON- ERROR=procedure(arg1[,argi]...)] [-DB=] [-SORT=nn] [-LOCK=ll] [-INMSG=nn] [-OUTMSG=nn]	(SQLBase) Correct the syntax.
	Incorrect arguments for BEGIN-SQL: [-Cnn] [-XP] [ON- ERROR=procedure[(arg1[,argi]...)]	(Sybase) Correct the syntax.
	Incorrect arguments for BEGIN-SQL: [-Cnn] [ON- ERROR=procedure[(arg1[,argi]...)] [-DB=]	(Red Brick) Correct the syntax.
003731	Did not find 'END- SQL' after 'BEGIN-SQL'.	Correct the source file.
003732	ON-ERROR= for 'BEGIN-SQL' in SETUP section must be STOP, WARN or SKIP.	03732 Correct the syntax.

Error Number	Error Message	Suggestion/Interpretation
003733	Could not create procedure for SQL.	(Sybase) SQR could not create a stored procedure for the SQL statement. The most likely cause for failure is that the user name you are using to run the report under does not have the proper privileges. Either grant the user CREATE PROCEDURE privilege or use the -XP command line option to inhibit SQR from creating temporary stored procedures for SQL statements.
003734	Could not compile SQL.	Correct the SQL statement or use the ON-ERROR= option to trap the error during the program run.
003735	Could not execute SQL.	An error occurred while trying to compile the SQL statement. Correct the SQL statement or use the ON-ERROR= option to trap the error during the program run.
003736	Please use BEGIN- SELECT - END- SELECT section for SELECT statements.	(Informix, Ingres, ODBC, Oracle, SQLBase, HPIW) Correct the source code.
003737	Bad fetch buffer count.	(Oracle, Sybase) The -B flag specifies an illegal value. Correct the source code.
003738	Report interrupted by request.	Informational message.
003741	Dynamic column must be \$string variable.	Correct the syntax.
003742	Dynamic column missing '01'.	Correct the syntax. '01 = Missing character
003743	Dynamic columns must have a &pseudonym.	Correct the syntax.
003744	&pseudonym =type must be 'char', 'number', or 'date'.	Correct the syntax.
003745	Only a variable name may be between the '01' and '02' characters.	Correct the syntax. '01 = Leading character '02 = Trailing character
003746	When dynamic columns are used all non-dynamic columns and expressions must be defined with &name=type.	Add &name=type to all expressions and non-dynamic columns.
003747	When the table name is dynamic each column and expression must be defined with &name=type.	Add &name=type to all expressions and non-dynamic columns.

Error Number	Error Message	Suggestion/Interpretation
003800	Too many document paragraphs; maximum is `01.	There are too many BEGIN-DOCUMENT paragraphs. Reduce the number of DOCUMENT paragraphs needed by the program. `01 = Maximum number supported by this version of SQR
003801	Too many document markers; maximum is `01.	There are too many BEGIN-DOCUMENT paragraphs. Reduce the number of DOCUMENT paragraphs needed by the program. `01 = Maximum number supported by this version of SQR
003802	Duplicate document marker.	Give the document marker a unique name.
003803	Did not find 'END- DOCUMENT' after 'BEGIN-DOCUMENT'.	The BEGIN-DOCUMENT paragraph must end with END-DOCUMENT. Correct the source code.
003900	EXECUTE command is incomplete.	Correct the syntax.
003901	Bad -Cnn connection number for EXECUTE.	The -Cnn value must be a nonzero value. Correct the source line.
003902	@#Return_status must be #numeric (missing #).	Correct the source line.
003903	Missing '=' after `01.	Correct the source line. `01 = The parameter in question
003904	Unknown variable type.	Variable names must begin with \$, #, or &. Correct the source line.
003905	OUT[PUT] variables for EXECUTE may only be \$variable or #variable.	Correct the syntax.
003906	The only EXECUTE option is WITH RECOMPILE.	Correct the syntax.
003907	You must EXECUTE ... INTO &columns.	Correct the syntax.
003908	Unknown datatype for EXECUTE...INTO &columns.	Check for a misspelled data type. If the data type is correct, then contact customer technical support so SQR can be updated.
003909	EXECUTE...INTO &columns must be unique.	The &column name assigned to the column must be unique throughout the report. Give the column a unique name.
003910	Missing (length) for datatype in EXECUTE.	Correct the source line.
003911	Datatype should not have (length) in EXECUTE.	Correct the source line.
003912	DO= in EXECUTE requires INTO... variables.	Correct the syntax.

Error Number	Error Message	Suggestion/Interpretation
003913	Could not EXECUTE stored procedure.	Record the database error message displayed with this message. If needed, contact your system administrator.
003914	Bad return fetching row from database.	Record the database error message displayed with this message. If needed, contact your system administrator.
003915	Could not set up EXECUTE cursor.	The database server returned an error while trying to compile the SQL statement needed to set up the EXECUTE command.
003916	Missing '(' before EXECUTE params.	(Ingres) Correct the syntax.
003917	Missing EXECUTE params.	(Ingres) Correct the syntax.
004000	Result #variable or \$variable or '=' missing in expression.	The LET command is not properly formatted. Correct the source line.
004001	Expression too complex.	The expression is either too long or is too deeply nested. Break the expression into smaller expressions.
004002	Parentheses unbalanced in expression.	A left or right parenthesis is missing. Correct the source line.
004003	Too many variables; maximum is '01.	Break the expression into smaller expressions. '01 = Maximum number supported by this version of SQR
004004	Empty expression.	The expression is invalid. Correct the source line.
004005	Extra comma in expression.	An argument is missing after a comma in the expression. Correct the source line.
004006	Unknown operator '01'. Do you mean '02 ?	The concatenation operator is . Correct the source line.
004007	Too many &column forward references in expression; maximum is '01.	The expression contains too many forward references. Break the expression into smaller expressions. '01 = Maximum number supported by this version of SQR
004008	Unknown function or variable in expression: '01	The specified function is not an SQR built-in function nor does it exist in the user-modifiable file UFUNC.C. Check for a misspelled function name. '01 = Function name
004009	Function '01' missing parentheses.	All functions in an expression must be followed by their arguments enclosed in parentheses. Correct the source line.

Error Number	Error Message	Suggestion/Interpretation
004010	Empty parentheses or expression.	A pair of parentheses were found with nothing inside them. Remove the () in question from the source line.
004011	User function '01' has incorrect number of arguments.	Look at the file UFUNC.C to determine the correct number and type of arguments required for the specified function. '01 = User function name
004012	Function '01' has incorrect number of arguments.	Correct the syntax of the function. Functions are described under the LET command. '01 = SQR function name
004013	Missing operator in expression.	Correct the source line.
004014	Operator '01' missing argument.	Correct the syntax of the function. Functions are described under the LET command. '01 = Operator
004015	Function '01' missing argument.	Correct the syntax of the function. Functions are described under the LET command. '01 = SQR function name
004016	Function or operator '01' missing arguments.	Correct the syntax of the function. Functions are described under the LET command. '01 = SQR function name
004017	User function '01' requires character argument.	Look at the file UFUNC.C to determine the correct number and type of arguments required for the specified function. '01 = User function name
004018	User function '01' requires numeric argument.	Look at the file UFUNC.C to determine the correct number and type of arguments required for the specified function. '01 = User function name
004019	User function '01' requires \$string variable.	Look at the file UFUNC.C to determine the correct number and type of arguments required for the specified function. '01 = User function name
004020	User function '01' requires #numeric variable.	Look at the file UFUNC.C to determine the correct number and type of arguments required for the specified function. '01 = User function name
004021	User function '01' has incorrect argument type list. Must be of: c,n,C,N	The UFUNC.C file has a bad definition for the specified function. Correct the UFUNC.C program file; recompile UFUNC.C; and recreate the SQR executable. '01 = User function name

Error Number	Error Message	Suggestion/Interpretation
004022	User function '01' missing arguments.	Look at the file UFUNC.C to determine the correct number and type of arguments required for the specified function. '01 = User function name
004023	User function '01' has incorrect return type. Must be c or n.	The UFUNC.C file has a bad definition for the specified function. Correct the UFUNC.C program file; recompile UFUNC.C; and recreate the SQR executable. '01 = User function name
004024	'isnull' requires a &column, \$string or \$date argument.	#numeric variables cannot be NULL. Correct the source line.
004025	'nvl' requires a &column, \$string or \$date as its first argument.	#numeric variables cannot be NULL. Correct the source line.
004026	Function or operator '01' requires character argument.	Correct the source line. '01 = Function or operator
004027	Function or operator '01' requires numeric argument.	Correct the source line. '01 = Function or operator
004028	IF or WHILE expression must return logical result.	The expression used must evaluate a statement that will be TRUE or FALSE. Correct the source line.
004029	Attempting division by zero in expression.	The expression tried to divide a number by zero. Use the COND() function to check if the divisor is zero; then divide by something else (for example, 1).
004030	Attempting division by zero with '%'.	An attempt was made to divide a number using the " %" operator. Use the COND() function to check if the divisor is zero; then divide by something else (for example, 1).
004031	The number used with '%' ('01) is out of range.	The "%" operator works with integers only. Correct the program logic. '01 = Maximum value allowed
004032	User function has unknown return type -- expecting n or c -- need to recompile Run-Time file?	SQR detected an error while processing a user defined function. If you are running an .sqt file, it probably needs to be recompiled because the user function has changed its definition. If you are running an .sqr file, then you need to correct the UFUNC.C program file; recompile UFUNC.C, and recreate the SQR executable.
004033	In user function use C type with allocated string to change \$variable.	SQR detected an error while processing a user defined function. Correct the UFUNC.C program file recompile UFUNC.C and recreate the SQR executable.

Error Number	Error Message	Suggestion/Interpretation
004034	Could not find array '01' in ARRAY function.	Check for a misspelled array name. '01 = Array name
004035	Could not find array field '01' in ARRAY function.	Check for a misspelled array field name. '01 = Array field name
004036	Math error in expression (usually over- or under-flow).	Most of the SQR mathematical built-in functions have a corresponding C library routine. One returned an error. Break the expression into discrete expressions in order to identify the function that caused the error.
004037	Error executing expression.	Record the steps leading up to the error and contact technical support.
004038	Out of space while processing expression; Use -Mfile to increase EXPRESSIONSPACE.	The expression requires more temporary string storage than is currently allocated. Use the -Mfile flag on the command line to specify a file that contains an entry that increases by a greater value than is currently defined.
004039	'01' assumed to be a variable name, not an expression.	Warning message. '01 = Expression in question
004040	The array '01' has not been defined.	Define the array using the CREATE- ARRAY command. '01 = Array name
004041	The field '01' is not valid for array '02'.	Correct the source code. '01 = Field name '02 = Array name
004042	The array reference '01' has an incorrect number of parameters specified.	Correct the source code. '01 = Array name
004043	The array reference '01' requires numeric parameters for the element and occurs arguments.	Correct the source code. '01 = Array name.
004045	Function or operator '01' requires date argument.	Correct the source code. '01 = Array name
004046	Incompatible types between expression and variable.	Correct the source code.
004047	The field '01' is must be 'char' or 'float'.	Correct the source code. '01 = Field name
004048	Function or operator '01' must be a string or date argument.	Correct the source line. '01 = Function or operator
004100	Use 'print' command to format data outside SELECT query.	You must precede PRINT command arguments (WRAP, ON-BREAK.) with an explicit PRINT command when outside of a BEGIN-SELECT paragraph. Correct the source line.

Error Number	Error Message	Suggestion/Interpretation
004101	Cannot find required parameter.	Correct the syntax.
004102	Bad number found.	A command expecting a numeric literal or :#numeric variable reference found an illegal number definition or a reference to a string variable or column. Correct the source line.
004103	Cannot find required numeric parameter.	Correct the syntax.
004104	Cannot find placement parameters.	The position qualifier "(Row,Col,Len)" was not found. Check for a missing parentheses.
004105	Placement parameter incorrect.	The "Row", "Column" or "Length" fields are invalid or ill-formed. Correct the source line.
004106	Invalid second function on line.	An SQR command used as a qualifier for a primary command (for example, PRINT) is incorrect. Correct the source line.
004107	Second function must be FORMAT type.	The PRINT command may have format command qualifiers such as WRAP, CENTER, or FILL. Other qualifier commands are not permitted.
004108	Missing operator =, <, >, ...	Correct the source line.
004109	Invalid operator.	Correct the source line.
004110	Missing variable.	Correct the syntax.
004111	Please give this expression a &pseudonym.	Expressions selected in a BEGIN-SELECT paragraph should be given an &Name or be followed by a print position "(Row,Col,Len)". Correct the source line.
004112	Wrong variable type.	Correct the syntax.
004113	Command incomplete, expected '01'.	Correct the syntax. '01 = What was expected
004114	Expecting '01', found '02'.	Correct the syntax. '01 = What was expected '02 = What was encountered
004115	Unknown command or extra parameters found (missing quotes?).	Correct the syntax.
004116	Duplicate references to parameter '01'.	Correct the syntax. '01 = Duplicated parameter
004117	Unexpected equal sign found with '01'.	Correct the syntax. '01 = Parameter name
004118	Qualifier '01' cannot be used with the following qualifiers:	Correct the syntax. '01 = Qualifier name

Error Number	Error Message	Suggestion/Interpretation
004119	Expecting numeric column, found string column.	Correct the syntax.
004120	Date variables (`01) cannot be used with this command.	Correct the syntax. `01 = Parameter name
004200	Page width and depth must be > 0 and < 32767.	The values specified with the PAGE-SIZE command are out of specified range. Specify legal values.
004201	Page buffer must be < 65536 on PC SQR.	The maximum allocation on a PC is 65536. The Page-Depth * Page-Width cannot exceed this value. Reduce the Page-Depth or Page-Width.
004202	Cannot generate line printer output for this report because position qualifier(s) may be out of range. If you are running this report, specify PRINTER: {HP,EH,HT,PS,WP} for graphical printer output.	The report output cannot be generated for a Line Printer. If your report was designed for a graphics printer, specify - PRINTER: {HP,EH,HT,PS,WP} for graphical printer output.
004300	Missing end of placement (...) in SHOW.	The placement parameter is ill-formed. Correct the source line.
004301	Bad (...) location in SHOW.	Screen positions must be valid numbers. Correct the source line.
004302	Missing literal or variable name to EDIT in SHOW.	The literal or variable name must immediately precede the EDIT, NUMBER, MONEY, or DATE keywords.
004303	Missing edit mask in SHOW.	The word EDIT must be followed by a valid edit mask. Correct the source line.
004304	Only string variable allowed for dynamic edit mask.	Dynamic edit masks may only be stored in \$Variables. Correct the line.
004305	Unknown option for SHOW.	Correct the syntax.
004400	Program too large; use -Mfile to increase PROGLINES.	The SQR program contains too many SQR command lines. Use the -Mfile flag on the command line to specify a file containing an entry that increases the currently defined value.
004401	Out of param storage; use -Mfile to increase PROGLINEPARS.	The SQR program contains too many SQR command line parameters. Use the -Mfile flag on the command line to specify a file containing an entry that increases the currently defined value.
004402	Out of string storage; use -Mfile to increase STRINGSPACE.	The space allocated to hold the static string variables ('...') has been used. Use the - Mfile flag on the command line to specify a file containing an entry that increases the currently defined value.

Error Number	Error Message	Suggestion/Interpretation
004403	Out of variables; use - Mfile to increase VARIABLES.	There are too many variables (string, numeric), literals and database columns. Use the -Mfile flag on the command line to specify a file containing an entry that increases the currently defined value.
004405	Out of forward &column or \$variable references; use -Mfile to increase FORWARDREFS.	A forward referenced variable is a variable that is referenced before it is defined. Use the -Mfile flag on the command line to specify a file containing an entry that increases the currently defined value.
004406	Number `01 not allowed.	Use a different value. `01 = Internal number
004407	Referenced variables not defined:	References were made to column variables (&var) that were not defined in the program. The list of variable names follows this message.
004500	Out of Print positions; use -Mfile to increase POSITIONS.	A print position is the "(Row,Col,Len)" parameter. Use the -Mfile flag on the command line to specify a file containing an entry that increases the currently defined value.
004501	Use '+' and negate variable for reverse relative placement.	The use of "-#Variable" is not legal here. Negate the #Variable value and use "+#Variable".
004503	Fixed line placement #variable must be > 0. Use relative positioning, (+#line,10,0).	Correct the source line as indicated.
004504	Fixed column placement #variable must be > 0. Use relative positioning, (5,+#col,0).	Correct the source line as indicated.
004505	Length placement #variable must be >= 0.	The length field cannot be a negative value. Correct the source line.
004600	CODE not appropriate for numeric data.	The CODE qualifier to the PRINT command may only be used for text fields. Move the "#Variable" to a "\$Variable" first and then print the "\$Variable".
004601	Unknown option for GRAPHIC command: BOX, HORZ-LINE, VERT-LINE or FONT	Correct the syntax.
004602	GRAPHIC BOX out of bounds. Row: `01, Column: `02, Width: `03, Depth: `04	SQR ends the program run. `01 = Row `02 = Column `03 = Width `04 = Depth

Error Number	Error Message	Suggestion/Interpretation
004603	GRAPHIC VERT- LINE out of bounds. Row: `01, Column: `02, Length: `03	SQR ends the program run. `01 = Row `02 = Column `03 = Length
004604	GRAPHIC HORZ- LINE out of bounds. Row: `01, Column: `02, Length: `03	SQR ends the program run. `01 = Row `02 = Column `03 = Length
004700	Cannot open the program file: "`01" (<`02): `03	Depends on the system error message. `01 = Name of the program file `02 = System error code `03 = System error message
004701	Cannot logon to the database.	The connectivity information is either incorrect or the database server is unavailable. Check the connectivity information and the server availability.
004702	Line found outside paragraph.	All commands must be within BEGIN-... END-statements. Correct the source code.
004703	Cannot close the program file. (<`01): `02	Depends on the system error message. `01 = System error code `02 = System error message
004704	#ENDIF not found for #IF.	Missing an #ENDIF to complete conditional compilation. Correct the source code.
004705	Program line too long; maximum is `01.	Break the program line into smaller lines. `01 = Maximum line length supported by this version of SQR
004706	Substitution variable `{`01} would cause this line to exceed the maximum line length of `02 characters.	The substitution variable value would cause this line to exceed the maximum line size. Break the program line into smaller lines. `01 = Name of the substitution variable `02 = Maximum line length supported by this version of SQR
004707	No value found for substitution variable: `{`01}	An empty value was found for the substitution variable. Check for a misspelled name. `01 = Name of the substitution variable
004708	#ELSE without preceding #IF.	Missing an #IF or #IFDEF or #IFNDEF to begin conditional compilation. Correct the source code.
004709	#ENDIF without preceding #IF.	Missing an #IF or #IFDEF or #IFNDEF to begin conditional compilation. Correct the source code.

Error Number	Error Message	Suggestion/Interpretation
004710	#IF's nested too deeply; maximum is '01.	Reduce the number of nested #IF directives. '01 = The maximum depth supported by this version of SQR
004711	#INCLUDE files nested too deeply; maximum is '01.	Reduce the number of nested #INCLUDE directives. '01 = The maximum depth supported by this version of SQR
004712	Include file name too long; Modify -I flag.	The combined -I directory name with the #INCLUDE file name exceeds the maximum length permitted for a complete pathname. Check the spelling of both the -I command flag and the #INCLUDE filename.
004713	Cannot open the #INCLUDE file: '01' ('02): '03	'01 = Include file name '02 = System error code '03 = System error message
004714	Cannot close the #INCLUDE file: '01' ('02): '03	'01 = Include file name '02 = System error code '03 = System error message
004716	'BEGIN-REPORT' command not found in program.	This section is required for all reports. Correct the source code.
004717	Cannot open the report output file: '01' ('02): '03	'01 = Output file name '02 = System error code '03 = System error message
004719	Cannot logoff the database.	The database server returned an error while trying to log off from the database. SQR ends the program run.
004720	Cannot open the run- time file: '01'. ('02): '03	SQR ends the program run. '01 = Run-Time file name '02 = System error code '03 = System error message
004721	Cannot close the run- time file. ('01): '02	SQR ends the program run. '01 = System error code '02 = System error message
004722	Error reading the run- time file. ('01): '02	SQR ends the program run. '01 = System error code '02 = System error message
004723	Run time file must be recreated for this version of SQR.	The runtime file was created by a earlier version of SQR and is incompatible with the current version. Recreate the .sqt (runtime) file.
004724	The -XL option cannot be specified with this run-time file because access to the database is required.	Do not use the -XL option.

Error Number	Error Message	Suggestion/Interpretation
004725	Cannot open cursor.	The database server returned an error indicating that a new database cursor or logon could not be completed. See the error message from the database server.
004726	Cannot create procedure for SQL statement.	(Sybase) SQR could not create a stored procedure for the SQL statement. The most likely cause for failure is that the user name you are running the report under does not have the proper privileges. Either grant the user CREATE PROCEDURE privilege or use the -XP command line option to inhibit SQR from creating temporary stored procedures for SQL statements.
004727	Error writing the run- time file. (`01): `02	`01 = System error code `02 = System error message
004728	You must specify a Partitioned Data Set name and member to build a .sqt(member) run-time file. Could not create the run-time file.	(MVS) Use the proper format to specify the name of the .sqt file.
004729	Cannot find inactive database cursor. Program too large.	(DB2, Oracle) The program has too many concurrent database cursors. Reduce the complexity of the program.
004730	Run-time saved in file: `01	Informational message. `01 = Name of the .sqt file created
004734	Cannot select user.	(Ingres) SQR could not select the default user name from the database. Contact your system administrator.
004735	Unknown variable type encountered in run-time file: `01	SQR ends loading the runtime file. `01 = Variable type
004736	Unexpected End-Of- File while processing the run-time file.	SQR ends loading the runtime file.
004737	Cannot load the run- time file because it was built for the `01 database and `02 is built for the `03 database.	SQR ends loading the runtime file. `01 = Database name from runtime file `02 = SQR image name `03 = Database that SQR is built for
004738	'END-REPORT' not paired with 'BEGIN- REPORT'.	Correct the source code.
004739	'END-PROGRAM' not paired with 'BEGIN- PROGRAM'.	Correct the source code.
004743	#INCLUDE filename must be enclosed in quotation marks.	Correct the syntax.
004744	#INCLUDE command format is: #Include 'filename'.	Correct the syntax.

Error Number	Error Message	Suggestion/Interpretation
004745	Array field ('01.'02) specification exceeds the PC 64K limit.	Reduce the size of the field requirements. '01 = Array name '02 = Field name
004746	Layout '01' specifications exceeds the PC 64K limit.	The layout is too large for the PC version of SQR to handle. '01 = Layout name
004747	The SQT file is corrupted and cannot be processed.	SQR ends loading the runtime file.
004748	The user function '01' needs to be defined as entry '02' in the user function table. It requires a definition of: Return Type = '03' Arg Count = '04 Arg Types = "'05"	The SQT file requires that the specified user function be defined. '01 = User function name '02 = Entry in the user function table '03 = Return type '04 = Argument count '05 = Argument types
004749	An attempt was made to move '01 characters into '02'. The maximum allowed is '03 characters.	An attempt was made to move too much data into an SQR string variable. '01 = Number of characters to be moved '02 = Variable name '03 = Maximum characters allowed
004802	PRINTER TYPE must be HTML, HPLASERJET, POSTSCRIPT, or LINEPRINTER.	Correct the syntax.
004805	Both BEFORE-BOLD and AFTER-BOLD must be specified.	Correct the syntax.
004807	Unknown DECLARE qualifier.	Correct the syntax.
004900	Out of dynamic SQL arguments [\$...]; use -Mfile to increase DYNAMICARGS.	Use the -Mfile flag on the command line to specify a file containing an entry that increases the currently defined value.
004901	Date variables ('01) cannot be used in BEGIN-SQL or BEGIN-SELECT paragraphs.	Correct the source code. '01 = Variable name
005000	Report '01' heading section size exceeds the page depth.	Reduce the size of the heading or increase the page depth.
005001	Report '01' footing location must be less than the page depth.	Reduce the size of the footing or increase the page depth.
005002	Check 'BEGIN- HEADING' commands: Discovered 2nd page- initialization while heading in progress.	The BEGIN-HEADING procedure either caused an overflow of the current page or it issued a command that caused a page eject to occur. Check any procedure invoked by the BEGIN-HEADING section to ensure that the commands do not overflow the page or cause a page eject.

Error Number	Error Message	Suggestion/Interpretation
005003	Check 'BEGIN-FOOTING' commands; perhaps number of footing lines is too small. Discovered 2nd page- write while footing in progress.	The BEGIN-FOOTING procedure either caused an overflow of the current page or it issued a command that caused a page eject to occur. Check any procedure invoked by the BEGIN-FOOTING section to ensure that the commands do not overflow the page or cause a page eject.
005004	Attempt to execute the `01 command while processing the `02 section.	Change the SQR program logic to prevent the command from executing while the specified section is active. `01 = Command name `02 = Section name
005005	Report `01' already has been assigned a `02 section.	Correct the source code. `01 = Report name `02 = Duplicated section name
005006	You cannot define more than one default `01' section.	Correct the source code. `01 = Duplicated section name
005007	Report `01' has overlapping heading and footing sections.	Correct the source code. `01 = Report name
005008	TOC `01' already has been assigned a `02 section.	Correct the source code. `01 = Table of Contents name `02 = Duplicated section name
005100	'IF', 'WHILE', 'EVALUATE' commands nested too deeply; maximum is `01.	Reduce the nested commands. `01 = Maximum depth allowed by this version of SQR
005101	'BREAK' found outside 'WHILE' or 'EVALUATE' statement.	The BREAK command is valid only in the context of a WHILE or EVALUATE statement. Correct the source code.
005102	Out of Break commands; Use -Mfile to increase BREAKS.	This is the number of BREAK commands allowed per EVALUATE command. Use the -Mfile flag on the command line to specify a file containing an entry that increases the currently defined value.
005103	END-WHILE found without matching 'WHILE'.	Correct the source code.
005104	'IF' or 'EVALUATE' command not completed before 'END- WHILE'.	Correct the syntax.
005105	'ELSE' found without matching 'IF'.	ELSE can be used only within the context of an IF command. Correct the source code.
005106	Single 'ELSE' found inside 'WHILE' or 'EVALUATE' statement.	ELSE can be used only within the context of an IF command. Correct the source code.
005107	Only one 'ELSE' allowed per 'IF'.	Rewrite the source code to use nested IF statements.

Error Number	Error Message	Suggestion/Interpretation
005108	Found 'END-IF' without matching 'IF'.	Each IF command must have a matching END-IF command. Correct the source code.
005109	'WHILE' or 'EVALUATE' command not completed before 'END-IF'.	You are missing a closing END-WHILE or END-EVALUATE command before END- IF. IF, WHILE, and EVALUATE statements can be nested, but they cannot cross each other's boundaries. Each inner statement must be complete before a closing statement is ended. Correct the source code.
005110	EVALUATE statements nested too deep; maximum is `01.	Reduce the number of nested statements. `01 = Maximum depth supported by this version of SQR
005111	'WHEN' found outside 'EVALUATE' clause.	WHEN may be used only in the context of an EVALUATE clause. Correct the source code.
005112	'IF' or 'WHILE' not completed before 'WHEN' statement.	Correct the syntax.
005113	Out of When commands; Use -Mfile to increase WHENS.	Use the -Mfile flag on the command line to specify a file containing an entry that increases the currently defined value.
005114	Incorrect types for comparison. Both must be of the same type (string, numeric or date).	Correct the source line.
005115	'When-other' found outside 'Evaluate' statement.	WHEN can be used only in the context of an EVALUATE statement. Correct the source code.
005116	'IF' or 'WHILE' not ended before 'WHEN- OTHER' command.	Correct the syntax.
005117	Only one 'WHEN- OTHER' allowed per 'EVALUATE'.	Correct the syntax.
005118	Found 'END-EVALUATE' without matching 'EVALUATE'.	Each EVALUATE command must have a matching END-EVALUATE command. Correct the source code.
005119	'IF' or 'WHILE' command not completed before 'END- EVALUATE'.	Correct the syntax.
005120	'WHEN-OTHER' must be after all 'WHEN's.	Correct the syntax.
005121	No 'WHEN's found inside 'EVALUATE' statement.	Correct the syntax.
005122	'IF', 'EVALUATE' and 'WHILE' statements cannot cross sections or paragraphs.	These commands must be contained within a single section or paragraph. Correct the source code.

Error Number	Error Message	Suggestion/Interpretation
005200	Did not find '>' after <....	A leading left angled bracket "<" indicates that you are beginning an ASCII value, which must be ended by a right angled bracket ">". Correct the source line.
005201	Bad ascii character in <...>.	Numbers in angled brackets <> must be between 1 and 255. Correct the source line.
005202	Bad ascii number in <...>.	Numbers in angled brackets <> must be between 1 and 255. Correct the source line.
005203	<...> string is too long; maximum is `01 characters.	Reduce the length of the string. If this is not possible, use a PRINT-DIRECT command in a BEGIN-REPORT or END-REPORT procedure. `01 = Maximum number of characters supported by this version of SQR
005300	Did not find '=' after qualifier: `01	Correct the syntax. `01 = Qualifier name
005301	Qualifier `01' requires a numeric value.	Correct the syntax. `01 = Qualifier name
005302	Incorrect value for qualifier `01'. Valid values are:	Correct the source line. `01 = Qualifier name
005303	Invalid qualifier `01'. Valid qualifiers are:	Correct the source line. `01 = Qualifier name
005304	Qualifier `01' requires a numeric literal, variable, or column.	Correct the source line. `01 = Qualifier name
005305	Qualifier `01' references a numeric variable that has not been defined.	Correct the source line. `01 = Qualifier name
005306	Qualifier `01' requires a string literal, variable, or column.	Correct the source line. `01 = Qualifier name
005307	List not terminated.	Correct the syntax.
005308	Missing comma in list.	Correct the syntax.
005309	Required argument `01' was not specified.	Correct the source line. `01 = Qualifier name
005310	Qualifier `01' has already been specified.	Correct the source line. `01 = Qualifier name
005311	Qualifier `01' requires a string literal.	Correct the source line. `01 = Qualifier name
005312	Qualifier `01' requires a list of values: (val [,val]...).	Correct the source line. `01 = Qualifier name
005313	Qualifier `01' requires a integer value.	Correct the source line. `01 = Qualifier name

Error Number	Error Message	Suggestion/Interpretation
005314	Invalid character in variable name '01'.	Correct the source line. '01 = Invalid character
005315	Qualifier '01' references a string variable that has not been defined.	Correct the source line. '01 = Qualifier name
005316	Qualifier '01' uses an invalid Unit-Of-Measure suffix. Valid suffixes are: dp pt mm cm in	Correct the source line. '01 = Qualifier name
005400	Second page write attempted while writing current page. Check BEFORE- PAGE, AFTER- PAGE procedures.	Check any procedure invoked by the BEFORE- PAGE or AFTER- PAGE procedures to ensure that the commands do not overflow the page or cause a page eject.
005402	String cannot be placed on page: '01 -- placement specified is out of range. ('02, '03, '04)	Ensure the values are within the page limits. '01 = Text value '02 = Row '03 = Column '04 = Length
005403	Error writing the output file. ('01): '02	'01 = System error code '02 = System error message
005404	Cannot open the Postscript startup file: '01 ('02): '03	'01 = Name of the file '02 = System error code '03 = System error message
005405	SQR trial copy exiting after '01 pages.	'01 = Number of pages.
005406	Exiting after requested number of test pages ('01).	'01 = Number of pages
005408	Program stopped by user request.	Informational message.
005500	Cannot set parse_only option.	(Sybase) The DB-Library routine dbsetopt() returned an error. This should never happen. Contact technical support.
005501	Cannot reset parse_only option.	(Sybase) The DB-Library routine dbclropt() returned an error. This should never happen. Contact technical support.
005502	Cannot drop SQR generated stored procedure: '01.	(Sybase) See the database server error message that was also output. This should never happen. Contact technical support. '01 = Stored procedure name
005503	Cannot use '01 datatype as bind variable.	(Sybase) Use another database column. '01 = The database datatype.
005504	Unknown datatype for bind variable: '01 Cannot create stored procedure.	(Sybase) Please contact technical support. '01 = Unknown database datatype

Error Number	Error Message	Suggestion/Interpretation
005505	SQL too large to create stored procedure.	(Sybase) The size of the SQL text needed to create the stored procedure is too large for SQR to process. Add the -XP option to the BEGIN-SQL or BEGIN-SELECT command.
005506	SQR's EXECUTE command not available for this version of Sybase.	(Sybase) Some early versions of Sybase SQL Server or Microsoft SQL Server do not support Remote Procedure Calls (RPCs). Update your database server.
005507	Could not add param to remote procedure call.	(Sybase) A DB-Library routine returned an unexpected error. See the error message from the database.
005508	The number of EXECUTE...INTO &columns does not match the procedure.	(Sybase) Check the definition for the stored procedure you are referencing.
005509	Incorrect number of INTO &columns defined in EXECUTE.	(Sybase) Check the definition for the stored procedure you are referencing.
005510	Error converting OUTPUT Sybase type for EXECUTE.	(Sybase) The DB-Library routine dbconvert() failed to convert the data from the stored procedure. Contact technical support.
005511	Number of OUTPUT parameters from EXECUTE is incorrect.	(Sybase) Check the definition for the stored procedure you are referencing.
005512	Missing default database name for USE.	(Sybase) Correct the syntax.
005512	Missing default database name for USE.	(ODBC) Could not connect to the specified datasource.
	Missing default database name for USE.	(Red Brick) Could not connect to the specified datasource.
005513	You may only specify 'USE db' once, before any SQL statements are executed.	(Sybase) Only one USE command is allowed in a report. Place the SETUP section at the beginning of the SQR report.
005514	DBLON: 'sqlini() failed.	(SQLBase) SQR could not initialize the database interface. Contact technical support.
005515	Undefined variable referenced in -DB flag: `01	(ODBC, SQLBase) Check for a misspelling. `01 = Variable name
005516	SQLEPO: Cannot get error position in SQL.	(SQLBase) SQR could not get the position of the error in the SQL statement from the SQLBase database. Contact technical support.
005517	SQLNSI: Cannot get number of select items.	(SQLBase) SQR could not determine the number of columns in the Select list from the SQLBase database. Contact technical support.

Error Number	Error Message	Suggestion/Interpretation
005518	Could not connect to database specified in - DB variable: '01'.	(SQLBase) Correct database name. '01 = Database name
005519	SQLROW: Cannot get number of rows.	(SQLBase) SQR could not determine the number of rows from the SQLBase database. Contact technical support.
005520	SQLRLO: Error reading Long datatype.	(SQLBase) SQR could not read the long datatype column from the SQLBase database. Contact your system administrator.
005521	Warning: LONG datatype truncated; use -Mfile to increase LONGSPACE.	(SQLBase) Use the -Mfile flag on the command line to specify a file containing an entry that increases the currently defined value.
005522	SQLELO: Error reading Long datatype.	(SQLBase) SQR could not read the long datatype column from the SQLBase database. Contact your system administrator.
005523	Database commit failed.	The database command to perform a commit returned an error. Try running the SQR program again. The error could be related to a network or server problem. If the error persists, contact your system administrator.
005524	Cannot close database cursor.	The database command to close the database cursor returned an error. Try running the SQR program again. The error could be related to a network or server problem. If the error persists, contact your system administrator.
005525	'01 Could not get database error message.	(SQLBase) SQR could not get the error message from the SQLBase database. '01 = Reason for failure
005528	HPIW SQL '01 error '02 in cursor '03: '04	(HPIW) '01 = Routine name '02 = Error code '03 = SQR cursor number '04 = Error message from database
005528	DB2 SQL '01 error '02 in cursor '03:	(DB2) '01 = Routine name '02 = Error code '03 = SQR cursor number
005528	INFORMIX SQL '01 error '02 (ISAM: '03) in cursor '04: '05	(Informix) '01 = Routine name '02 = Error code '03 = ISAM code '04 = SQR cursor number '05 = Error message from database

Error Number	Error Message	Suggestion/Interpretation
005528	INGRES SQL `01 error `02 in cursor `03: `04	(Ingres) `01 = Routine name `02 = Error code `03 = SQR cursor number `04 = Error message from database
005528	ODBC SQL `01 error `02 in cursor `03: `04	(ODBC) `01 = Routine name `02 = Error code `03 = SQR cursor number `04 = Error message from database
	ORACLE `01 error `02 in cursor `03: `04	(Oracle) `01 = Routine name `02 = Error code `03 = SQR cursor number `04 = Error message from database
	SQLBase `01 error `02 in cursor `03: `04	(SQLBase) `01 = Routine name `02 = Error code `03 = SQR cursor number `04 = Error message from database
	Sybase `01 error in cursor `02: `03	(Sybase) `01 = Routine name `02 = SQR cursor number `03 = Error message from database
	Red Brick `01 error `02 in cursor `03: `04	(Red Brick) `01 = Routine name `02 = Error code `03 = SQR cursor number `04 = Error message from database
005532	System 10 files are missing.	(Sybase) Contact your system administrator.
005533	Not a System 10 SQL Server.	(Sybase) The CT-Library version of SQR can only connect to a System 10 server. Use the DB-Library version of SQR to connect to a pre-System 10 server.
005534	SQL too long for PRE-PARE/DECLARE; maximum `01 characters.	(DB2) The SQL statement is too large. `01 = Maximum number of characters supported by this version of SQR
005536	Unknown error message number: `01.	(DB2) `01 = Error message number
005537	Empty error message returned from system for error number: `01.	(DB2) `01 = Error message number
005538	Invalid SELECT statement; COMPUTE clauses are not supported.	(Sybase) The select statement contains a COMPUTE clause that is not supported.

Error Number	Error Message	Suggestion/Interpretation
005539	Could not connect to datasource specified in -db variable: '01'.	(ODBC) Could not connect to the specified datasource.
	Could not connect to datasource specified in -db variable: '01'.	(Red Brick) Could not connect to the specified datasource.
005540	Not connected to a database, database access is not allowed.	The SQR program is no longer connected to a database. Commands that access the database can no longer be used. This situation can occur if the CONNECT fails and the ON-ERROR option was used.
005541	Bind variables are not supported.	(HPIW) Bind variables are not supported.
005543	Specify the Oracle DLL name in the SQR.INI file in [Environment:Oracle] section for ORACLE_DLL entry, such as ORACLE_DLL=orant 71.dll	(Oracle) SQR was unable to load the Oracle DLL. By default, SQR looks first for "ociw32.dll" or the DLL specified by the ORACLE_DLL entry in the [Environment:Oracle] section of the SQR.INI file. If that DLL could not be loaded, then SQR attempts to load 'orant71.dll'.
005600	GETWRD: Word too long; maximum is '01.	Reduce the length of the "word". '01 = Maximum size of a "word" supported by this version of SQR
005700	Cannot call SQR recursively.	SQR cannot be called recursively. This error can only occur if a User Function from either UFUNC.C or UCALL.C calls the sqr() routine. Do not call sqr() from a UFUNC.C or UCALL.C routine.
005701	Too many SQR command line arguments; maximum is '01	To pass more than this number of arguments, use a @file argument file containing one argument per line. '01 = Maximum number supported by this version of SQR.
005702	Log file name specified is too long.	Reduce the length of the log file name.
005703	Error opening the SQR log file: '01' ('02): '03	'01 = Name of the file '02 = System error code '03 = System error message
005704	Missing program name.	The name of the program file was not found on the command line. The program name must be the first parameter on the command line.
005705	Program file name specified is too long.	Reduce the length of the program file name.
005707	Error opening the -E error file: '01' ('02): '03	'01 = Name of the file '02 = System error code '03 = System error message

Error Number	Error Message	Suggestion/Interpretation
005708	Cannot find `01 in SQRDIR, PATH or \SQR.	The specified file cannot be located in any of the directories pointed to by the mentioned environment variables or default directories. Make sure the "file" is present in one of the locations searched. `01 = File name
005709	`01 environment variable is not defined.	As of version 2.5, the environment variable SQRDIR must be defined. `01 = Name of the environment variable
005710	`01 path too long.	The length of the directory path plus the length of the file name to be opened is too long for SQR to handle. Reduce the length of the directory path. `01 = Environment variable name
005711	Bad number in -T test flag.	The number specified must be > zero. Correct the value.
005712	-G option requires arguments.	(VAX) The command line option is ill- formed. Correct the syntax.
005713	Too many arguments to -G option; maximum is `01.	(VAX) The command line option is ill- formed. Correct the syntax. `01 = Maximum number of arguments supported by this version of SQR
005714	-G attribute too long; maximum is `01.	(VAX) The command line option is ill- formed. Correct the syntax. `01 = Maximum number of each attribute supported by this version of SQR
005716	Unknown flag on command line: `01	Correct the syntax. `01 = Unknown command line flag
005717	Cannot open channel to TT; status = `01	(VAX) This is an error that should never occur during normal operations. Record the steps leading up to the error and contact your system administrator. `01 = System status
005718	Cannot read from TT; status = `01	(VAX) This is an error that should never occur during normal operations. Record the steps leading up to the error and contact your system administrator. `01 = System status
005719	Cannot close channel to TT; status = `01	(VAX) This is an error that should never occur during normal operations. Record the steps leading up to the error and contact your system administrator. `01 = System status

Error Number	Error Message	Suggestion/Interpretation
005720	Error opening tty. (`01): `02	(DG, UNIX) This is an error that should never occur during normal operations. Record the steps leading up to the error and contact your system administrator. `01 = System error code `02 = System error message
005721	Error with 'ioctl()'. (`01): `02	(DG, UNIX) This is an error that should never occur during normal operations. Record the steps leading up to the error and contact your system administrator. `01 = System error code `02 = System error message
005722	Error reading tty. (`01): `02	(DG, UNIX) This is an error that should never occur during normal operations. Record the steps leading up to the error and contact your system administrator. `01 = System error code `02 = System error message
005723	Error closing tty. (`01): `02	(DG, UNIX) This is an error that should never occur during normal operations. Record the steps leading up to the error and contact your system administrator. `01 = System error code `02 = System error message
005724	Bad number in -B flag.	(Oracle, Sybase) The number specified must be > zero. Correct the value.
005734	No program name given.	The report name must be the first command line argument.
005735	Cannot use both -U and db/user.	(Ingres) Correct the command line.
005736	-P requires an argument: role_id/role_password.	(Ingres) Supply the role information.
005737	Unknown printer type specified with - PRINTER: switch.	The printer type can be EH, HT, LP, HP, PS, or WP. WP is valid only with PC/Windows.
005738	Database name needs to be included with - DB switch.	(ODBC) Could not connect to the specified datasource.
005738	Database name needs to be included with - DB switch.	(Red Brick) Could not connect to the specified datasource.
	Database name needs to be included with - DB switch.	(Sybase) Supply the database name.
005739	Too many -F switches; maximum is `01.	Reduce the number of -F switches. `01 = Maximum number allowed

Error Number	Error Message	Suggestion/Interpretation
005740	-F and outfile name are required with DDN or DD style SQR {program} name.	(MVS) Correct the JCL stream.
005741	Attempting to use SQR {program} file for outfile.	(MVS) Correct the JCL stream.
005742	Attempt to invoke viewer (using Win-Exec) failed; error code = `01.	(Windows) `01 = System error code
005743	Unknown numeric type specified with -DNT: switch.	Correct the command line.
005744	-DNT:Decimal precision (`01) is out of range (`02 - `03).	Correct the command line. `01 = Specified precision `02 = Minimum allowed `03 = Maximum allowed
005745	The specified default numeric type `01 = `02' is invalid.	Correct the SQR.INI file entry. `01 = Entry `02 = Value
005746	The decimal precision `01 = `02' is out of range (`03 - `04).	Correct the SQR.INI file entry. `01 = Entry `02 = Value `03 = Minimum allowed `04 = Maximum allowed
005747	The following error(s) occurred while processing the [`01] section from the SQR.INI file.	See the error message(s) that follow. `01 = Name of the section
005750	The -Burst switch is not properly formatted.	The "Burst" command line flag is not properly formatted.
005751	The -Burst switch cannot be used with the -NOLIS switch.	The "Burst" command line flag cannot be specified when the -NOLIS command line flag is also specified.
005752	The -Burst switch requires either the -Printer:HT or -Printer:EH switch to be specified.	The "Burst" command line flag is applicable only when HTML code is produced. You must specify either the -PRINTER:HT or -PRINTER:EH switch.
005753	The -Burst:S and -Burst:T switches can only be used against an SPF file which was generated with SQR v4.1 and above.	The "Burst" command line flag can only be specified when processing a SPF file that was generated by SQR v4.1 and above. Older SPF files do not contain the proper information that permits bursting.
005754	The -Burst switch caused no output to be generated.	The "Burst" command line flag was specified with a set of page ranges that prevented any output to be created. Change the page ranges.
005900	Bad number in -`01	(Windows) Specify a valid number. `01 = Command line option

Error Number	Error Message	Suggestion/Interpretation
005901	Bad filename in -`01	(Windows) Specify a valid file name. `01 = Command line option
005902	Bad directory in -`01	(Windows) Specify a valid directory path. `01 = Command line option
005903	Cannot access the @ parameter file (`01): `02	(Windows) Depends on the system error message. `01 = System error code `02 = System error message
005904	The argument list is too long; maximum is `01.	(Windows) To pass more than this number of arguments, use a @file argument file containing one argument per line. `01 = Maximum number supported by this version of SQR.
005905	Cannot open the report file (`01): `02	(Windows) Depends on the system error message. `01 = System error code `02 = System error message
005906	Invalid filename entered.	(Windows) Re-enter with a valid file name.
006000	Error writing the printer file. (`01): `02	This is an error that can occur during normal operations due to the system environment (for example, file locking, permissions). Record the steps leading up to the error and contact your system administrator. `01 = System error code `02 = System error message
006001	Error reading the printer file. (`01): `02	This is an error that can occur during normal operations due to the system environment (for example, file locking, permissions). Record the steps leading up to the error and contact your system administrator. `01 = System error code `02 = System error message
006002	Cannot open the printer file: `01 (`02): `03	This is an error that can occur during normal operations due to the system environment (for example, file locking, permissions). Record the steps leading up to the error and contact your system administrator. `01 = Name of the file `02 = System error code `03 = System error message
006003	Unexpected End-Of- File while processing the printer file.	Possibly the file is corrupted. Try to recreate the .spf file. If the error persists, contact technical support.

Error Number	Error Message	Suggestion/Interpretation
006004	Encountered unknown SPF code ('01) while reading the printer file.	Possibly the file is corrupted. Try to recreate the .spf file. If the error persists, contact technical support. '01 = Unknown SPF code
006100	Duplicate chart ('01).	Each chart must be given a unique name. '01 = Chart name
006101	Unknown chart ('01).	Chart could not be found. '01 = Chart name
006102	Number of chart data- array columns specified ('01) exceeds the number of array columns ('02).	Correct the source code. '01 = Number of data-array columns '02 = Number of array columns
006103	Number of chart data- array rows specified ('01) exceeds the number of array rows ('02).	Correct the source code. '01 = Number of data-array rows '02 = Number of array rows
006104	Too many pie segments ('01). Max is '02.	Correct the source code. '01 = Number of segments '02 = Maximum allowed segments
006105	Chart module is not initialized.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006106	XY charts may have only numeric columns.	Correct the syntax.
006107	The 3rd column in the data array must be a character column to specify USE-3RD- DATA-COLUMN.	Correct the syntax.
006108	Invalid chart size or placement.	Correct the source code.
006120	INTERNAL: Bad chart index from stack ('01).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Chart index
006121	INTERNAL: Unknown SQRB-GInterface message ('01).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Message code
006122	INTERNAL: Unsupported Graftsman chart type ('01).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Chart type

Error Number	Error Message	Suggestion/Interpretation
006123	INTERNAL: Unsupported pie-explode setting ('01).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Setting value
006124	INTERNAL: Unsupported tick-mark placement ('01).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Placement value
006125	Grafsman interface message ('01) not supported.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Message code
006126	Unrecognized return code ('01) from Grafsman command message ('02).	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Return code '02 = Message code
006127	Cannot fit Chart/Image into the current page. Position: ('01, '02) Size: ('03, '04)	Correct the source code. SQR ends the program run. '01 = Row '02 = Column '03 = Width '04 = Depth
006128	Check coordinate values.	Correct the syntax.
006140	Duplicate image ('01).	Images must be given unique names. '01 = Image name
006141	Unknown image ('01).	Image name could not be found. '01 = Image name
006142	Cannot open image file ('01). ('02); '03	'01 = Name of the file '02 = System error code '03 = System error message
006150	INTERNAL: Bad image index from stack ('01).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Image name
006200	This report has already been defined.	Each report must be given a unique name.
006201	This layout has already been defined.	Each layout must be given a unique name.
006202	This printer has already been defined.	Each printer must be given a unique name.
006203	The values for '01' must be > 0.	Correct the syntax. '01 = Qualifier name

Error Number	Error Message	Suggestion/Interpretation
006204	Qualifiers '01' and '02' are mutually exclusive.	Correct the syntax. '01 = Qualifier name '02 = Qualifier name
006205	Qualifier '01' is not applicable with a 'default' printer.	Correct the syntax. '01 = Qualifier name
006206	The list must contain report names or ALL.	Correct the syntax.
006207	'ALL' must be specified by itself.	Correct the syntax.
006208	No report name was specified.	Correct the syntax.
006209	No layout name was specified.	Correct the syntax.
006210	No printer name was specified.	Correct the syntax.
006211	The name cannot be 'ALL'.	Correct the syntax.
006212	The name can only contain characters [0-9 A-Z _ -].	Correct the syntax.
006213	Report '01' is referenced by multiple '02' printers.	Correct the syntax. '01 = Report name '02 = Printer type
006214	Qualifier '01' is not allowed with a '02' printer.	Correct the syntax. '01 = Qualifier name '02 = Printer type
006215	The value for '01' must be '02 0.	Correct the syntax. '01 = Qualifier name '02 = Relation to zero (<,<=,=,>=,>)
006216	Report '01' does not exist.	Correct the syntax. '01 = Report name
006217	The report name can be a string literal, variable, or column.	Correct the syntax. '01 = Report name
006218	Referenced layouts not defined:	A list of undefined layouts follows this message.
006219	Referenced reports not defined:	A list of undefined reports follows this message.
006220	Referenced printers not defined:	A list of undefined printers follows this message.
006221	The following SQR commands (listed below) cannot be used when any of the following NEW SQR commands are also used in the same report:	Correct the syntax.

Error Number	Error Message	Suggestion/Interpretation
006224	No printer type was specified.	Correct the syntax.
006225	Incorrect value for printer type. Valid values are:	Correct the syntax. A list of valid printer types follows this message.
006226	Attempt to execute the '01 command while processing the '02 procedure.	SQR ends the program run. '01 = SQR command '02 = Procedure name
006227	Incorrect value for 'paper-size'. Specify the actual dimensions or one of the following names:	Correct the syntax. A list of valid predefined paper-size names follows this message.
006228	Referenced TOC (Table Of Contents) not defined:	A list of undefined Table of Contents follows this message.
006229	This TOC (Table Of Contents) has already been defined.	Each Table of Contents must be given a unique name.
006230	The list must contain TOC (Table of Contents) names or ALL.	Correct the syntax.
006231	The TOC (Table Of Contents) entry cannot be positioned given the LEVEL ('01) and INDENTATION ('02) values.	The Table of Contents entry will not fit given the specified level and current indentation values. '01 = Specified LEVEL= value '02 = Current INDENTATION= value
006232	'01 command not allowed while generating the Table of Contents.	The specified command cannot be used while the Table of Contents is being generated. '01 = SQR command
006233	The TOC (Table of Contents) entry "A" cannot be processed because the existing entry "B" is positioned below it. A: Line = '01, Level = '02, Text = '03' B: Line = '04, Level = '05, Text = '06'	Correct the program logic to eliminate the conflict between the two TOC (Table of Contents) entries. '01 = A: Line number '02 = A: Level value '03 = A: Text value '04 = B: Line number '05 = B: Level value '06 = B: Text value
006300	Unknown parameter ('01).	Correct the syntax. '01 = Parameter name
006301	Value not valid for parameter ('01).	Correct the syntax. '01 = Parameter name
006302	Invalid option ('02) for parameter ('01).	Correct the syntax. '01 = Parameter name '02 = Option
006303	Parameter ('01) is required, but has not been specified.	Correct the syntax. '01 = Parameter name

Error Number	Error Message	Suggestion/Interpretation
006304	Parameter ('01) already specified.	Correct the syntax. '01 = Parameter name
006305	Parameter ('01) does not support &columns.	Correct the syntax. '01 = Parameter name
006306	Parameter ('01) requires equal sign.	Correct the syntax. '01 = Parameter name
006307	Parameter ('01) has an unquoted string.	Correct the syntax. '01 = Parameter name
006308	Missing part of specification for parameter ('01).	Correct the syntax. '01 = Parameter name
006309	Parameter ('01) requires literal.	Correct the syntax. '01 = Parameter name
006310	Parameter ('01) requires valid numeric value.	Correct the syntax. '01 = Parameter name
006311	Parameter ('01) requires integer value.	Correct the syntax. '01 = Parameter name
006312	Parameter ('01) does not support type supplied.	Correct the syntax. '01 = Parameter name
006313	Parameter ('01) requires valid string. Perhaps quote or \$ is missing.	Correct the syntax. '01 = Parameter name
006314	Parameter ('01) does not accept 'NONE' in this context.	Correct the syntax. '01 = Parameter name
006315	Parameter ('01) requires proper object name.	Correct the syntax. '01 = Parameter name
006316	Parameter ('01) requires array name.	Correct the syntax. '01 = Parameter name
006317	Parameter ('01) does not accept 'AUTOSCALE' in this context.	Correct the syntax. '01 = Parameter name
006318	Parameter ('01) has improper value list.	Correct the syntax. '01 = Parameter name
006320	Parameter ('01) does not support relative values.	Correct the syntax. '01 = Parameter name
006350	Conversion [('01) to ('02)] is not supported.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = From type '02 = To type

Error Number	Error Message	Suggestion/Interpretation
006352	INTERNAL: Unsupported option/request ('01) in ('02).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Option/request code '02 = Function name
006354	INTERNAL: Unknown data type, ('01).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Data type
006355	INTERNAL: Unable to retrieve parameter value, ('01).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Parameter name
006356	INTERNAL: Data type ('02) not valid for parameter ('01).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Parameter name '02 = Data type
006357	INTERNAL: Data location ('02) not valid for data type ('01).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Data location '02 = Data type
006358	INTERNAL: Cannot decode string ('01) to index.	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = String to decode
006359	INTERNAL: Cannot set bit value ('02) for parameter ('01).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = Parameter name '02 = Value
006360	INTERNAL: Unknown program state ('01).	This is an internal error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. '01 = State

Error Number	Error Message	Suggestion/Interpretation
006400	Unsupported background color.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006401	Unsupported border color.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006402	Border width out of range.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006403	X position out of range.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006404	Y position out of range.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006405	X size out of range.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006406	Y size out of range.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006407	Unsupported font.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006408	Unsupported font style.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006409	Unsupported font color.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006410	Unsupported horizontal text justification value.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006411	Unsupported vertical text justification value.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006412	Unsupported font path.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006413	Unsupported font rotation.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.

Error Number	Error Message	Suggestion/Interpretation
006414	Font size out of range.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006415	Text line id# out of range.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006416	Unsupported chart type.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006417	Unsupported chart sub-type.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006418	Unsupported chart orientation (not H or V).	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006419	Unsupported perspective (not 2D or 3D).	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006420	Unsupported axis (not X or Y).	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006421	Unsupported axis label data type.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006422	Dataset id# out of range.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006423	Unsupported dataset type.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006424	Unsupported dataset color.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006425	Unsupported dataset line style.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006426	Unsupported dataset fill pattern.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006427	Unsupported dataset marker.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.

Error Number	Error Message	Suggestion/Interpretation
006428	Chart type does not support Y-axis datasets.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006429	Pie-chart segment id# is out of range.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006430	Unsupported pie- segment color.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006431	Unsupported pie- segment border color.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006432	Unsupported pie- segment pattern.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006433	Unsupported pie- segment explode setting.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006434	Command only valid for charts of type 'pie'.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006435	Pie-chart radius out of range.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006436	Pie-chart starting angle out of range.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006437	Unsupported pie-chart fill direction. Must be clockwise or counter-clockwise.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006438	Unsupported pie- segment label position.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006439	Unsupported pie- segment quantity display position.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006440	Unsupported pie- segment per-cent display position.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006441	Unsupported legend style.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.

Error Number	Error Message	Suggestion/Interpretation
006442	Unsupported legend horizontal position.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006443	Unsupported legend vertical position.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006444	Text charts do not support legend.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006445	Number of datasets specified does not match data.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006446	Unsupported axis label position.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006447	Unsupported axis type (not LINEAR or LOG).	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006448	Pie and text charts do not support axis control.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006449	Unsupported axis min scaling.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006450	Unsupported axis max scaling.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006451	Unsupported axis max scaling.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006452	Beginning of tickmarks is after end.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006453	Unsupported tickmark type.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006454	Unsupported grid type.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006455	Unsupported grid color.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.

Error Number	Error Message	Suggestion/Interpretation
006456	Grid line width out of range.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006457	Unable to open grafcap file.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006458	Unsupported grafcap device.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006459	Error in grafcap entry specification.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006460	Unable to open chart output destination.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006461	Internal error during ggDraw.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006462	Improper parameters passed to gscale.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006463	The shared library specified in the grafcap file could not be found.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006464	A function called from the shared library specified in the grafcap file could not be found.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006500	The bar code could not be positioned on the page. Row: `01, Column: `02, Height: `03	Correct the source code. `01 = Row `02 = Column `03 = Height
006501	Unknown BCL error (`01) encountered.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support. `01 = BCL error code
006502	Invalid bar code type (`01): Valid values are from 1 to 15.	Correct the source code. `01 = Bar code type.
006503	The length of the bar code text `01' must be between 1 and 30 characters.	Correct the source code. `01 = Bar code text
006504	The length of the caption text '01' must be between 1 and 30 characters.	Correct the source code. `01 = Caption text

Error Number	Error Message	Suggestion/Interpretation
006505	Invalid printer type (`01): Valid values are from 0 to 13.	Correct the source code. `01 = Printer type
006506	Invalid offset: Valid values are from 0 to 250.	Correct the source code.
006507	Invalid height (`01): Valid values are from 0.1 to 2.0 inches.	Correct the source code. `01 = Height
006508	Invalid checksum: Valid values are from 0 to 2.	Correct the source code.
006509	Invalid pass: Valid values are from 1 to 6.	Correct the source code.
006510	The bar code text '01' is not valid for the type of bar code (`02) selected.	Correct the source code. `01 = Bar code text `02 = Bar code type
006511	Internal error: Could not generate the bar code.	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006512	Internal error: Bar code buffer required too large (>32K).	This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006601	Cannot allocate the device context for the default printer.	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator.
006602	Failed to start printing the document.	(Windows) This is an error that can occur due to lack of system resources or a problem with the printer. Record the steps leading up to the error and contact your system administrator.
006603	New-page (start) failed on page `01.	(Windows) This is an error that can occur due to lack of system resources or a problem with the printer. Record the steps leading up to the error and contact your system administrator. `01 = Page number
006604	New-page (end) failed on page `01.	(Windows) This is an error that can occur due to lack of system resources or a problem with the printer. Record the steps leading up to the error and contact your system administrator. `01 = Page number
006605	End document failed.	(Windows) This is an error that can occur due to lack of system resources or a problem with the printer. Record the steps leading up to the error and contact your system administrator.

Error Number	Error Message	Suggestion/Interpretation
006606	Error reading font information from the [Fonts] section in SQR.INI. Using the default font.	(Windows) Correct the [Fonts] section in the SQR.INI file.
006607	Failed to create a brush for shading.	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator.
006608	Failed to select font `01.	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator. `01 = Font name
006609	Failed to modify font `01.	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator. `01 = Font name
006610	Failed to create a pen that was required to draw a box.	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator.
006611	Failed to create a pen that was required to draw a horizontal line.	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator.
006612	Failed to create a pen that was required to draw a vertical line.	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator.
006613	Failed to open the image bitmap file (`01). (`02): `03	(Windows) This is an error that can occur during normal operations due to the system environment (file locking, permissions). Record the steps leading up to the error and contact your system administrator. `01 = Name of the file `02 = System error code `03 = System error message
006614	The file (`01) does not contain a valid bitmap.	(Windows) Specify a valid bitmap file. `01 = Name of the file
006615	Failed to create the palette for image (`01).	(Windows) This is an error that can occur due to lack of system resources or an invalid bitmap. Record the steps leading up to the error and contact your system administrator. `01 = Name of the file

Error Number	Error Message	Suggestion/Interpretation
006616	Failed to load RLE into memory for image ('01).	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator. '01 = Name of the file
006617	Failed to convert DIB to DDB for image ('01).	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator. '01 = Name of the file
006618	Failed to draw the bitmap image ('01).	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator. '01 = Name of the file
006619	Cannot access the default printer's driver.	(Windows) This is an error that can occur due to lack of system resources or a problem with the printer. Record the steps leading up to the error and contact your system administrator.
006620	Cannot select the charting clip area onto the printers DC.	(Windows) This is an error that can occur due to lack of system resources or a problem with the printer. Record the steps leading up to the error and contact your system administrator.
006621	Cannot select create a metafile required for business graphics.	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator.
006622	Cannot create a region required for business graphics.	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator.
006623	Cannot create a DC required for business graphics.	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator.
006624	Cannot create a bitmap required for business graphics.	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator.
006625	Business graphics failed while setting up the device (ggWinDevice).	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator.

Error Number	Error Message	Suggestion/Interpretation
006626	Cannot draw business graphics.	(Windows) This is an error that can occur due to lack of system resources or it can be due to a damaged LIBSTI.INI file. The LIBSTI.INI file resides in the Windows main directory. Make sure that the GPATH= and IPT= entries point to a valid SQR BINW directory. Record the steps leading up to the error and contact your system administrator.
006700	SQRDIR is not defined.	(Windows) The variable SQRDIR must be defined in the SQR.INI file.
006701	Could not allocate memory while attempting to register the .spf filename extension.	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator.
006702	Could not allocate memory for the page cache.	(Windows) This is an error that should never occur during normal operations. Record the steps leading up to the error and contact technical support.
006704	Cannot open or read file (`01) (`02): `03	(Windows) This is an error that can occur during normal operations due to the system environment (for example, file locking, permissions, and so on.). Record the steps leading up to the error and contact your system administrator. `01 = Name of the file `02 = System error code `03 = System error message
006705	File (`01) is not in SPF packet format.	(Windows) The file was not produced by SQR or it has been corrupted. `01 = Name of the file
006706	Failed to identify the start of the report (`01).	(Windows) The file was not produced by SQR or it has been corrupted. `01 = Name of the file
006707	An invalid seek was made for page `01.	(Windows) This is an internal error which should not occur under normal operations. Please contact technical support. `01 = Page number
006708	Too many errors were encountered while processing the file. Processing has been stopped.	(Windows) This is an error that can occur due to lack of system resources. Record the steps leading up to the error and contact your system administrator.

Error Number	Error Message	Suggestion/Interpretation
006709	Failed to open the image bitmap file ('01). (`02): `03 This message is displayed only once per SPF file.	(Windows) This is an error that can occur during normal operations due to the system environment (for example, file locking, permissions, and so on.). Record the steps leading up to the error and contact your system administrator. `01 = Name of the file `02 = System error code `03 = System error message
006800	`01: Detected internal program error.	This is an internal error that should never occur during normal operation. Record the steps leading up to the error and contact technical support. `01 = Name of the routine
006801	`01: Null Operand Passed as input.	This is an internal error that should never occur during normal operation. Record the steps leading up to the error and contact technical support. `01 = Name of the routine
006802	`01: Decimal Exponent Under/Overflow.	Exponent Under/Overflow: Exponent of decimal number has exceeded the valid boundaries established for the decimal type. Check the documentation for the current upper and lower bounds of a decimal object. `01 = Name of the routine
006803	`01: Decimal to Integer Conversion Under/Overflow.	Integer Under/Overflow: Cannot convert input decimal object into a valid integer number. Decimal object exceeds the established integer boundaries for this machine architecture. Check the magnitude and sign of the decimal object to ensure that it falls within the upper and lower bounds of an integer number. `01 = Name of the routine
006804	`01: Decimal to Float Conversion Under/Overflow.	Floating Point Under/Overflow: Cannot convert input decimal object into a valid floating point number. Decimal object exceeds the established floating point boundaries for this machine architecture. Check the magnitude and sign of the decimal object to ensure that it falls within the upper and lower bounds of a floating point number. `01 = Name of the routine

Error Number	Error Message	Suggestion/Interpretation
006805	`01: Decimal Precision Under/Overflow.	Decimal Precision Under/Overflow: Attempt made to initialize decimal object with an invalid precision. Check the input precision value against the documented upper and lower boundaries for a decimal object. `01 = Name of the routine
006806	`01: String to Decimal Object Conversion Error.	String To Decimal Conversion Error: Length of input string is greater than precision of underlying decimal object. Either increase the precision of the decimal object or reduce the size of the input mantissa to match the decimal object precision. `01 = Name of the routine
006807	`01: Truncation/Rounding Error - Outside Valid Range for Decimal Object.	Truncation/Rounding Error: Input truncation or round value is outside the valid range for this decimal object. Please ensure that the truncation/round value is greater than or equal to zero and less than the precision of the underlying decimal object. `01 = Name of the routine
006808	`01: Decimal Error: Cannot Divide by Zero.	Decimal Math Divide by Zero Error: Attempt made to divide a decimal object by zero. Please check divisor to ensure that it does not equal zero before attempting to divide. `01 = Name of the routine
006900	There is no default printer set up on your system. Use the Control Panel "Printers" applet to define it.	(Windows) SQR Print requires that a default printer be defined. Use the "Printers" applet in the Control Panel to define one.
007000	The locale '`01' is not defined in the SQR.INI file.	Check for a misspelled locale name and/or the SQR.INI file. `01 = Locale name
007001	At least one qualifier must be specified.	Correct the source code.
007002	The value for '01' must be a list of 02 string literals, variables or columns.	Correct the source code. `01 = Qualifier `02 = Number of entities in list
007003	The values for '01' and '02' cannot be the same.	Correct the source code. `01 = Qualifier `02 = Qualifier
007004	The value for '01' (`02) must be a single character which is not in the list: "03".	Correct the source code. `01 = Qualifier `02 = Value `03 = List of invalid characters

Error Number	Error Message	Suggestion/Interpretation
007005	The value for '01' (02) is invalid. Valid values are:	Correct the source code. '01 = Qualifier '02 = Value
007006	The last character of the '01' value (02) cannot be a digit or the minus sign or the same as either of the separators.	Correct the source code. '01 = Qualifier '02 = Invalid character
007007	The first character of the '01' value (02) cannot be a digit or the minus sign or the same as either of the separators.	Correct the source code. '01 = Qualifier '02 = Invalid character
00700	The following errors occurred while processing the (01) locale from the SQR.INI file.	This message precedes error messages encountered while processing the SQR.INI file. '01 = Locale name
007009	The value for '01' cannot be 'DEFAULT' or 'SYSTEM'.	Correct the syntax. '01 = Qualifier
007010	The value for '01' (02) is not properly formatted: Did not find the '>' for the '<nnn>' construct.	Correct the syntax. '01 = Qualifier '02 = Value
007011	The value for '01' (02) is not properly formatted: The value of an '<nnn>' construct must be from 1 to 255.	Correct the syntax. '01 = Qualifier '02 = Value
007012	The default locale (01) specified in the [02] section of the SQR.INI file has not been defined.	Correct the syntax. '01 = Locale name '02 = Section name
007013	The value for '01' (02) must be a list of 03 quoted string literals.	Correct the syntax. '01 = Qualifier '02 = Value '03 = Number of entities in list
007014	The entry (01 = 02) is not valid.	Correct the SQR.INI entry. '01 = Qualifier from the SQR.INI file '02 = Qualifier's value
007100	The use of an edit mask or the keywords NUMBER, MONEY or DATE is not legal when storing numeric variables.	Correct the source code.
007101	The last keyword is not '01'.	Correct the source code. '01 = Keyword
007102	Incompatible source and destination variable types.	Correct the source code.
007103	The keyword (01) is not compatible with the variable (02).	Correct the source code. '01 = Keyword '02 = Variable name

Error Number	Error Message	Suggestion/Interpretation
007104	The use of an edit mask or the keyword DATE is not legal if both variables are date variables.	Correct the source code.
007200	The specified precision ('01) is out of range ('02 - '03).	Correct the source code. '01 = Specified precision '02 = Minimum precision '03 = Maximum precision
007201	The precision is specified by a value from '01 to '02 surrounded by parentheses.	Correct the source code. '01 = Minimum precision '02 = Maximum precision
007202	Variable ('01) is not a decimal variable and cannot have a precision associated with it.	Correct the source code. '01 = Variable name
007203	A string variable name is required here.	Correct the source code.
007204	A numeric variable name is required here.	Correct the source code.
007205	The variable ('01) has already been defined as '02' and may not be redefined.	Correct the source code. '01 = Variable name '02 = Variable type
007206	The variable type has not been specified.	Correct the source code.
007207	This command is only allowed within local procedures.	Correct the source code.
007208	This command must be before all other commands in the procedure.	Correct the source code.
007209	Only string (\$) and numeric (#) variables may be declared.	Correct the source code.
007210	Invalid variable name specified.	Correct the source code.
007211	You cannot declare a global variable from within a procedure.	Correct the source code.
007400	The specified character is invalid in the current character set.	Correct the program logic.
007401	'01' is not a valid value for the ENCODING environment variable.	The specified encoding scheme is not known by SQR. '01 = ENCODING environment variable setting.
007402	The Double-Byte LET function '01' is not supported in this version of SQR.	The SQT file contains a reference to a LET function, which is not supported by this version of SQR. '01 = LET function name

Error Number	Error Message	Suggestion/Interpretation
007403	The Double-Byte SQR command '01' is not supported in this version of SQR.	The SQT file contains a reference to an SQR command, which is not supported by this version of SQR. '01 = SQR command name
007404	Double-Byte .sqt files are not supported by this version of SQR.	The runtime file was created by the Double-Byte version of SQR and is incompatible with the current version.
007405	The barcode text '01' cannot contain double- byte characters.	Correct the source code. '01 = Bar code text
007501	Using '01 edit mask from ('02) against ('03)	A date edit mask element was detected which could cause date data to be incorrectly interpreted. This warning message can be turned off by setting the "OutputTwoDigit-YearWarningMsg" entry int the [Default-Settings] section of the SQR.INI file to FALSE. '01 = Edit mask element '02 = Edit mask being used '03 = Value being applied to the edit mask
007601	Cannot access the Java file ('01) ('02): '03	SQR cannot access the required file. '01 = Name of the file '02 = System error code '03 = System error message
007602	-EH_Scale: value ('01) is out of range ('02 - '03).	Correct the command line. '01 = Specified scale '02 = Minimum allowed '03 = Maximum allowed
007603	-Printer:EH functionality is not available on this platform.	Enhanced HTML functionality is not available on this platform.
007604	-Printer:PD functionality is not available on this platform.	PDF functionality is not available on this platform.
007701	Did not find end of paragraph: '01 (No 'end-execute' clause found.)	Correct the source code. '01 = BEGIN-command in question.
007702	Invalid entry for keyword, '01='02'	Correct the source code.
007703	May only specify either PROCEDURE=, or COMMAND=, or GETDATA=, exclusive.	Correct the source code.
007704	Must specify a SCHEMA.	Correct the source code.
007705	Must specify either a PROCEDURE, COMMAND, or GETDATA.	Correct the source code.
007706	CONNECTION '01' not found. No such connection.	Correct the source code.

Error Number	Error Message	Suggestion/Interpretation
007707	The returned set of Procedure parameters (INOUT and OUT) (length = '01 items) did not include one or more of the specified items.	Stored procedure error.
007708	Encountered a parameter of type ''01'. Valid types are either IN, OUT, or INOUT. If no type is entered, the type defaults to IN.	Stored procedure error.
007709	The datasource failed to provide the expected return status value. Verify the query metadata.	Datasource error.
007710	The datasource failed to provide the expected number of elements in the return status list.	Datasource error.
007711	Failed to login to the requested datasource (Connection='01', username='02'). DETAILS: '03	Logon failed.
007712	The requested rowset ('01) was not available. Verify the query metadata.	Not enough rowsets.
007713	Missing or invalid Registry.properties file. Verify that the CLASSPATH includes SQDIR, that SQDIR contains the folder with the Registry.properties file, and that the Registry.properties file is valid.	Incorrect environment setup.
007714	The datasource ('01) does not support the requested capability ('02). Check the capabilities list for the datasource, located in the Properties folder.	Invalid query for datasource.
007715	Failed to start the Java Virtual Machine (JVM). Possible causes are: missing or invalid jdk files, incorrect CLASSPATH, or insufficient resources.	Incorrect environment setup.
007716	The current rowset ('01) contained no rows. Check the return status and/or metadata for the requested service to determine the cause.	No data.
007717	The query failed. DETAILS: '01	Query failed.
007718	Failure setting property ''01'. DETAILS: '02	Property-set failed.
007719	The value for keyword ''01' exceeds the maximum length of '02 characters.	Keyword value too long.

Error Number	Error Message	Suggestion/Interpretation
007720	A fatal error occurred while fetching against the current rowset: DETAILS: `01	A failure occurred during row fetch.
007721	Parameter `01 (`02) was passed to the PROCEDURE as data type `03; expected (`04) type `05. Verify the query metadata.	A failure occurred during row fetch.
007722	Invalid query parameter: Reason: `01	Bad procedure parameter.
007723	Too many parameters (= `01) were supplied to the query. Verify the query metadata.	Bad procedure parameter.
007724	Parameter `01 (`02) was passed to the PROCEDURE as type `03; expected type `04. Verify the query metadata.	Bad procedure parameter.
007725	Parameter `01 (`02', JDO-type `03), specified 'NULL', is a required-parameter. Specify a value or variable name.	Bad procedure parameter.
007726	The list-variable parameter to the query is too long. Maximum number of elements is 30.	List too long.
007727	Unable to retrieve metadata for Procedure=`01, Schema=`02. DETAILS: `03	Metadata check failed.
007728	Parameter list type mismatch (#`01, SQR type = `02). The datasource expected a parameter of type `03. Verify the query metadata.	Parameter list mismatch.
007729	List size mismatch detected while fetching data of type ROW, `01 items, into SQR list- variable, `02 items. Fetching will proceed to the smaller size.	List size mismatch.
007730	Incorrect syntax for BEGIN-SELECT ... FROM. Options are: FROM ROWSETS=... FROM PARAMETER= \$strvar strlit	Bad begin-select syntax.
007731	Attempted to pass as INOUT or OUT a parameter which was of type ROWSET (`01). Use of such parameters is supported as IN only, after which they may be used in a BEGIN- SELECT construct.	Bad parameter keyword.

Error Number	Error Message	Suggestion/Interpretation
007732	Attempt to use a scalar SQR variable ('01') to reference a ROWSET procedure parameter ('02'). Use either the keyword 'NULL', or an SQR LIST variable (%var). Verify the query metadata.	Bad proc parameter.
007733	The list of keywords entered to the PARAMETERS keyword must be terminated with a semicolon.	Bad proc parameter. Correct the source code.
007734	Datasource '01' not found. The Connection being used by this query specifies a datasource which is not listed in the DDO Registry ('02'). DETAILS: '03.	Bad proc parameter. Correct the source code.
007735	Missing one or more DDO {fname}.jar files. Verify the location of the original- installation files, and that they are accessible. Error code: '01. Classpath: '02.	Bad environment.
007736	Unable to open Connection ('01') to datasource ('02'). Possible causes: (a) the Declare- or Alter-connection specification is invalid, or (b) the data-source is no longer available. DETAILS: '03.	Bad environment.
007737	Unable to locate one or more entry points in an SQR {fname}.jar file. Verify that the original-installation files have not become corrupted.	Bad environment.
007738	At least one JNI method pointer was lost. This should never occur: record the steps leading up to this failure, and contact Technical Support. DETAILS: Schema='01', Proc='02'.	Bad environment.
007739	Unable to locate query object '01' in the specified schema ('02). DETAILS: '03.	Bad environment.
007740	Invalid &pseudonym or 'TYPE=' data-type specified for a begin- select column-variable. Valid types are: CHAR, TEXT, DATE, NUMBER.	Correct the syntax.
007741	Illegal attempt to fetch a non-scalar field into a column variable. Correct the query.	Correct the syntax.

Error Number	Error Message	Suggestion/Interpretation
007742	The output parameter specified in 'Begin- Select ... From Parameter = `01' is not available. Available parameters: `02.	Bad command.
007743	The output parameter specified in 'Begin- Select ... From Parameter = `01' is not of type ROWSET. Verify the query metadata.	Bad command.
007744	Illegal attempt to assign an SQR variable ('01') of type '02' the value from a DDO object ('03') of type '04'. Verify the query metadata.	Bad var assignment.
007745	Illegal attempt to assign an SQR column variable ('01') of type '02' the value from a DDO object of type '03'. Verify the query metadata.	Bad var assignment.
007746	Failed to locate the requested Rowset (`01) while processing the query. The last available Rowset number is `02. Verify the query metadata.	Not enough RowSets.
007747	The query raised a DDO exception. DETAILS: `01.	Bad query.
007748	A BEGIN-SELECT paragraph was coded, but the query returned no Rows.	No data warning.
007749	Invalid syntax for PARAMETERS=(...) statement. Use: PARAMETERS=(%v \$v #v &v NULL SKIP numlit datelit textlit [IN INOUT], ...) All parameters must be specified. Optional parameters which are to be ignored may be specified by the keyword 'NULL' or 'SKIP'. Correct the syntax.	Incorrect syntax.
007750	FATAL: Failure creating Java object.	General failure.
007751	Attempt to create a List variable of size greater than the maximum size of `01 items.	General failure.
007752	Parameter-list item "`01' is not a member of the parameter list for this Query. Verify the query metadata.	No such input/inout parameter.
007753	Attempt to access List- row (`01) beyond the List size (`02 rows).	Bad list assignment/setup.

Error Number	Error Message	Suggestion/Interpretation
007754	Attempt to assign/modify a List row is not compatible with the List definition.	Bad list assignment/setup.
007755	Attempt to assign a row to a non-existent List variable. Define the List first, using the syntax: let %lname[size] = list(NUMBER DATE TEXT #var \$var [, ...])	Bad list assignment/setup.
007756	Incorrect syntax for List-variable reference. Use: let [\$ #]var = %listname[nlit #var].colname	Bad list assignment/setup.
007757	Alter-connection statement missing 'DSN=...!.	Improper alter-conn.
007758	List-definition size specifier must be literal.	Improper alter-conn.
007759	Attempt to access a non-existent List-column ('01').	No such list column name.
007760	Must specify one of the keywords, FROM- ROWSETS or FROM_PARAMETER .	Incorrect syntax for Load-lookup.
007761	Incorrect syntax to Load-lookup 'PARAMETERS=' keyword. Use: PARAMETERS=(slit nlit \$var #var %var &var, ...) No line wrapping is allowed for this usage.	Incorrect syntax for Load-lookup.
007762	Too many parameters ('02) entered to Load- Lookup command. Max parameters is '01.	Incorrect syntax for Load-lookup.
007763	Problem executing the cursor for LOAD- LOOKUP table "'01'. DETAILS: '02.	The database server returned an error while trying to execute the SQL statement needed to process the LOAD-LOOKUP command. '01 = Load lookup table name
007764	Bad return fetching row from database in LOAD-LOOKUP table "'01'. DETAILS: '02	The database server returned an error while fetching the data. '01 = Load lookup table name
007765	DC, DI sort options not supported with this SQR version. To sort, use SORT=SC or SORT=SI.	Database sort not supported for Load- Lookup with DDO.
007766	Must specify a query keyword; PROCEDURE=, COMMAND= or GETDATA=.	Incorrect syntax for Load-lookup. Specify a keyword representing the query.
007767	Unknown column variable type.	Unknown data type returned by the server.

Error Number	Error Message	Suggestion/Interpretation
007768	The property '01' was not found in the property sheet for the specified datasource ('02). Available property names are: '03. The datasource property sheet does not include the named property.	Verify the metadata and correct the syntax.
007769	The specified CONNECTION ('01') references a datasource whose property sheet does not show support for the Get-Data query method. The datasource property sheet does not show support for Get-Data.	Verify the metadata and property sheet and correct the syntax.
007770	Attempt to create a Selector (or MDSelector) object failed. This event should not occur. Contact your system administrator.	Failed to create the requested object. Contact your system administrator.

Glossary

The terms in this glossary are used among multiple Financials and Supply Chain Management applications.

Numbers

401(a)(17) Limits

The limitations on the earnings that may be included in the calculation of benefits under qualified U.S. pension plans.

1st Year Amount

In PeopleSoft Workforce Analytics, 1st Year Amount is an employee-level compensation amount, totaling the calculations for the first calendar year's worth of accounting periods, in a compensation scenario.

A

Abend

Abnormal End (to a process).

ABM (Activity-Based Management)

See PeopleSoft Activity-Based Management.

ABPS (Activity-Based Planning and Simulation)

See Activity-Based Planning and Simulation.

Absence

An absence occurs when an employee is not at work (absent) during a normally scheduled work period. Absences may be scheduled or non-scheduled, compensated or uncompensated, excused or unexcused. An absence may occur for a variety of reasons like illness, family emergency, civic obligations (e.g. Military duty or jury duty), or vacation.

Absence Entitlement

Element which defines the rules for granting paid time off for valid absences, such as sick time, vacation, and maternity leave. An absence entitlement element defines the entitlement amount, frequency, and entitlement period.

Absence Take

Element which defines the conditions that must be met before a payee is entitled to take paid time off.

Accepted Exception

An exception that has been reviewed and validated (see Time Management).

Accommodations

Accommodations are efforts your organization is able to make for employees or applicants with disabilities, such as purchasing special equipment or making structural changes to a work environment.

Account Management

In PeopleSoft Demand Planning, a feature that enables you to divide a centrally held corporate forecast into multiple subsections for easier maintenance and management. These subsections are separate databases that can be distributed to account managers for use and updates, then rejoined with the main database at a later date.

Account

A code for recording and summarizing financial transactions as expenditures, revenues, assets, or liabilities balances. This is a delivered PeopleSoft ChartField, specific use of which is typically defined by the organization during implementation of PeopleSoft General Ledger.

Account Type

A name for one of the different kinds of accounts used in a PeopleSoft General Ledger, such as Asset, Liability, Equity, Revenue, and Expense.

Accounting Class

In PeopleSoft Enterprise Performance Management, an attribute that defines how the particular resource would be treated for generally accepted accounting practices. Inventory denotes whether a Resource will become part of a balance sheet account such as inventory or fixed assets, while Non-inventory denotes that the Resource will be treated as an expense of the period during which it occurs.

Accounting Date

The date that a transaction is recognized as opposed to the date the transaction actually occurred—the **Transaction Date** (although the two dates can be the same). The accounting date determines the period in the general ledger to which the transaction is to be posted. You

can only select an accounting date that falls within an open period in the ledger to which you are posting. The accounting date for an item is normally the invoice date. In PeopleSoft Asset Management, the difference between accounting date and transaction date determines whether prior period depreciation must be calculated, and how much. Accounting Date must be later than or equal to Transaction Date.

Accounting Entry

A set of related debits and credits. An Accounting Entry is made up of multiple ***Accounting Lines***. In most PeopleSoft applications, accounting entries are always balanced (debits = credits). Accounting entries are created to record accruals, payments, payment cancellations, manual closures, project activities in general ledger, and so forth (depending on the application).

Accounting Entry Template

A user-defined table that controls the use of system-generated accounting lines in the posting processes.

Accounting Split

Method indicating how expenses are allocated or divided among one or more sets of accounting ChartFields.

Accredited Education

Education above the high school level completed in a U.S. college, university, or other educational institution that has been credited by one of the accrediting agencies or associations recognized by the Secretary, U.S. Department of Education.

Accrual

Any hours that employees accumulate for use at another time in the form of earned vacation time or sick leave, for example.

Accrual Basis Accounting

Accounting that records the impact of a business event as it occurs, regardless of whether the transaction affected cash.

Accrual Class Codes

Classes or categories of accruals.

Accrual Type

Defines an accrual such as annual leave or sick leave.

Accumulate Demand

In PeopleSoft Demand Planning, a transfer process function that adds demand quantities for an item to any quantities that already exist for the period.

Accumulator

Element which allows you to combine several elements. For example, an accumulator could consist of all voluntary deductions, or all company deductions, enabling you to accumulate amounts. It allows total flexibility for time periods and values accumulated. See also Time Administration.

Accumulator [Global Payroll]

Element which provides a means for storing the cumulative values of defined items as they are processed. As you make payments, take deductions, and perform calculations, you'll use accumulators to track accumulated amounts, or balances. You can accumulate a single value over time or multiple values over time, as your requirements specify. For example, an accumulator could consist of all voluntary deductions, or all company deductions, enabling you to accumulate amounts. It allows total flexibility for time periods and values accumulated.

Action

In PeopleSoft Deduction Management, a task that you perform to obtain information required to resolve a deduction.

Action and Conditions

A process that defines actions and conditions independently of one another and then combines them to create a complete rule (see Rule Creation).

Action Code

In PeopleSoft Engineering, a user-defined code associated with an event/action triggered by the implementation of an engineering change order (ECO). Actions could include analyzing an item's existing quantity on hand, scrapping existing inventory, or modifying current documentation.

In PeopleSoft Product Configurator, a 2-character code that identifies rule types. For example, **FP** is the action code for the Finalize Price rule, and **CN** is the action code for the Condition rule. The rules control the processing path for configured items.

Action List

An online list of customers who meet predefined credit management criteria. The list also includes appropriate procedures for each action and contact information for the customer.

Action Owner

In PeopleSoft Deduction Management, the individual assigned a task to obtain information to resolve a deduction.

Action Reason

The reason an employee's job or employment information is updated. The action reason is entered in two parts: a *personnel action*, such as a promotion, termination, or change from one paygroup to another—and a *reason* for that action. Action Reason is used by PeopleSoft Human Resources, PeopleSoft Benefits Administration, PeopleSoft Stock Administration, and the COBRA Administration feature of the Base Benefits business process.

Active Control

A target control requiring that the user validate the budget against the planning targets before submitting it. If the budget totals are not within the tolerance levels, the system indicates that the status is invalid and the user cannot submit their budget until the budget is modified and the amount is within the tolerance range of the planning target.

Activity

In PeopleSoft Receivables and Deduction Management, an action taken on an item, such as creating an item, unposting an item, or writing off an item.

In PeopleSoft Projects, the unit of work that provides a further breakdown of projects—usually into specific tasks. Resources are assigned directly to activities within a project, not directly to projects.

A self-contained task that is part of one or more business processes. Business process maps display the activities that make up the process. An activity consists of steps representing the pages the user needs to complete and events representing the workflow routings triggered by the user's actions.

In PeopleSoft Enterprise Warehouse, the work of an organization and the aggregation of actions used for Activity-Based Costing.

Activity Attributes

Activity Attributes provide pieces of activity information. For example: capacity and performance, cost drivers, cycle time and performance measures.

Activity-Based Costing (ABC)

A methodology that measures the cost and performance of activities, resources and cost objects, assigns resources to activities and activities to cost objects based on their use and recognizes the causal relationships of cost drivers to activities.

Activity-Based Management (ABM)

See PeopleSoft Activity-Based Management (ABM).

Activity-Based Planning and Simulation (ABPS)

ABPS, a feature of PeopleSoft Activity-Based Management, calculates resource demands, new rates, costs, and activity volumes based on demand forecasts. It converts the new

resource demands into new cost requirements at the General Ledger item level to feed as input for budgeting.

Activity Driver

An Activity Driver indicates the amount of demand there is for a particular activity and it is used to assign cost to cost objects. In some instances, an activity driver may represent the yield of an activity.

Activity Fragmentation

The part of the Employee Profile feature that provides information about the number of employees that is involved in completing a particular activity on a full or part-time basis.

Activity ID

A unique 15-character alphanumeric identifier given to each activity within a project. Activity IDs need only be unique within a single project.

Activity List

In PeopleSoft Pension Administration, a checklist used to monitor pension-related activities.

Activity Type

A user-definable identifier for grouping activities.

Activity Type

Also known as Activity Code. A categorization of work effort. Typically work effort is categorized as productive or non-productive; Repair, Maintenance, Enhancement, or Improvement; or Development or Construction. Activity type is usually required to support cost accounting or financial accounting (recording) functions. It may also be required to support some organizational administration requirements such as organizational productivity goals, or employee performance measurement. In some companies, activity type is inferred from job function, work group affiliation, or organization.

Activity Use

An attribute used to describe the behavior of an Activity as defined within PeopleSoft Enterprise Performance Management. A Primary Activity is an activity that is performed for the purpose of directly generating revenue within the course of business. A Secondary Activity is generally performed in direct support of a Primary Activity such as activities related to human resources or MIS.

Actual Base Hours

This defines the number of hours that an employee is expected to work within a given period under analysis within PeopleSoft Enterprise Performance Management. Hours worked in excess of Actual Base Hours are generally considered overtime, while hours worked less than Actual Base Hours would illustrate that the employee is working part-time.

Actual Contribution Percentage (ACP)

The amount of an employee's after-tax or employer matching contributions made in a Section 401(m) plan on behalf of highly compensated plan participants, divided by the employee's annual compensation, or an amount determined in the same manner with respect to non-highly compensated employees. The Base Benefits business process is set up to perform ACP nondiscrimination tests for Section 401(m) plans. See Nondiscrimination Tests and Highly Compensated Employee.

Actual Date

Calendar date in which a punch occurred (see Time Reporting).

Actual Deferral Percentage (ADP)

The amount of salary reduction contributions made by an employee to a Section 401(k) plan for a year, divided by the employee's total compensation for that year. The Base Benefits business process is set up to perform ADP nondiscrimination tests for Section 401(k) plans. See Nondiscrimination Tests and Highly Compensated Employee.

Actual Demand

In PeopleSoft Demand Planning, an **Array** of demand by historical period imported from an external system. The demand figures are determined by imported values and typically include shipments, orders booked, orders booked by requested ship date, or shipments.

Actual Rates

An Actual Rate is the rate that your business currently uses for its business practice.

Actuarial Assumptions

Any assumptions used to calculate an equivalent benefit for an optional form of payment or an alternative retirement date.

Actuarial Valuation

A comparison of a pension plan's assets and liabilities.

Actuarial Valuation Extract

A PeopleSoft Pension Administration data extract containing data that a plan actuary needs in order to determine the plan's assets and liabilities.

Address Type

A high-level address classification that identifies addresses associated with a **Material Issue**. Examples include Ship To Address, Bill To Address, and Ship Notification Address.

Adjusted

In the Enterprise Planning and Simulation forecasting process, in addition to versions of the statistical forecast, there is an adjusted version of the forecast. Managers create this version by reviewing the forecasts and entering adjustments that cannot be inferred statistically. For example, there may be a promotional campaign next quarter that is expected to boost volume for certain products over several weeks.

Adjusted Demand

In PeopleSoft Demand Planning, an **Array** of demand after adjustments have been made to the actual demand values. The adjusted figures may include both manual and system-generated changes, such as demand filtering and depromotion. The system uses adjusted demand rather than actual demand in the Forecasting Reset process and in the recalculation of model components during period-end processing.

Adjusted Forecast

In PeopleSoft Demand Planning, a **Statistical Forecast** that has been adjusted using management overrides, proration, or summarization.

Adjustment

See **Bill Adjustment** or **Inventory Adjustment**.

Adjustment Voucher

A PeopleSoft Payables voucher that enables you to apply an adjustment to an existing voucher or to relate one voucher to another.

Advice

The Form that employees who choose direct payroll deposit receive in lieu of a check.

Affiliate

A control person of a corporation. Generally, an officer, director, or major shareholder that has the ability to influence the corporate management decisions.

After-tax Deductions

Deductions that reduce net pay. These deductions are subtracted from gross pay after taxes have been taken out. Also called “post-tax” deductions.

Agency

Any Department or independent establishment of the Federal Government, including a government-owned or -controlled corporation, that has the authority to hire employees in the competitive, excepted, and senior executive services.

Aggregated

In Enterprise Planning and Simulation, each period the statistical forecast is calculated automatically by the system. A forecast for each individual product can be computed using history for that product. Then these forecasts can be aggregated (that is, summarized) into forecasts for the product family.

Aggregate Reporting

The ability to report time as a collection or mass. In Time and Labor aggregate time reporting features include the ability to report time in a lump sum, as a pattern, in a range of dates, or for an entire crew.

Aging Data

Updating data from separate sources, and separate dates, to a common date using an annualized factor.

Aging ID

A code representing rules for aging open items.

Alias

Any of several PeopleSoft Pension Administration utilities that look up or calculate employee information.

Allocated

In Enterprise Planning and Simulation, the computed forecast and the summarized forecast are two different versions of the statistical forecast. In addition, the forecast at the product family level can be allocated down to the individual products. Usually this allocation is done in proportion to the calculated product forecasts at that level. This version of the (statistical) forecast is called the allocated or prorated statistical forecast.

Allocated Inventory

The inventory assigned to a specific stock request.

Allocation Manager

Perform allocations using the Allocation Manager. Allocations enable you to distribute revenue, expense, and statistical quantities across business units, departments, and so on. You can allocate budget planning to detail levels so that you may perform detailed budgeting. The type of allocation you select determines the output.

Allocation Manager Rules

In the PeopleSoft Enterprise Warehouse, Allocation Manager rules allow you to specify the basis as well as the target tables for moving, aggregating, or multidimensionalizing your output. Rules use Allocation Manager methods to enrich the PeopleSoft Enterprise Warehouse data. *See* Allocation Manager Methods.

Allocation Manager Methods

There are several methods: Arithmetic Operation, Prorata, and Spread Even. Each method enables you to move and/or enrich output.

Allocations

A process of distributing budget amounts to and from other Budget Centers. Budget amounts are allocated to cover, or offset, the costs in one Budget Center by charging them to another Budget Center. An allocation is also the budget amount that is distributed to or from a Budget Center. A budget amount that is charged to another Budget Center appears as a negative amount. This same budget amount appears as a positive amount in the other Budget Center receiving the allocation. PeopleSoft Budgeting-specific.

Allotment

This is a voluntary deduction from pay. Employees may elect up to two allotments from pay, transmitted to a financial institution to the employee's checking or savings account.

ALM (Asset Liability Management)

See PeopleSoft Asset Liability Management.

Allowances

The amount owed to an employee in addition to base salary and which is not defined as part of gross salary. For example, vacation can be considered an allowance. PeopleSoft Budgeting-specific.

Alternate Account

A feature in PeopleSoft General Ledger that enables you to create a statutory chart of accounts and enter statutory account transactions at the detail transaction level as required for recording and reporting by some national governments.

Alternate BOM

Identifies the multiple ways in which an item can be produced. The primary production BOM is designated as BOM code 1. By using BOM codes, you can associate up to 98 other alternate BOMs with the item.

Alternate Routing

A routing, usually less preferred than the primary routing, but resulting in an identical item. You can specify up to 98 alternate routings for production routing types by entering additional Routing Codes (greater than 1) for the same routing type.

Alternative Minimum Tax (AMT)

AMT is calculated by adjusting the taxpayer's regular taxable income with a number of tax preference items and adjustments. Tax preference items are positive items increasing

Alternative Minimum Taxable Income (AMTI) and are excluded from regular taxable income. Tax preference items include gain from the exercise of incentive stock options.

Amount Type

In PeopleSoft Workforce Analytics, the Amount Type specifies whether a benefits compensation amount is a value or expense, to the employee or the employer.

Analysis Base

Defined static, historical data used both to seed and compare against proposed budgets.

Analysis Group

A grouping of analysis types. Analysis groups can be used for project analysis and grouping or for mapping analysis types.

Analysis Template

A set of pre-defined reports that you can view and publish online. These templates access data in the Enterprise Warehouse tables, and organize it by function, role and industry. The templates allow you to pivot, sort, rank, drill and chart the data, for your analysis needs.

Analysis Type

A 3-character, user-definable identifier that enables you to label the different types of costs. For example, you might want to track budgeted costs (BUD), committed costs (COM), and actual costs (ACT).

Analytical Applications

See PeopleSoft Analytic Applications.

Analytic Forecasting

Analytic Forecasting is the part of the Planning and Simulation feature that creates forecasts for your business requirements.

Annual Amount

In PeopleSoft Workforce Analytics, Annual Amount is an employee-level compensation amount, totaling the calculations for a full fiscal year's worth of accounting periods, in a compensation scenario.

Annual Declaration Report

The French Annual Declaration report is a payroll report which checks establishment profiles to see whether an establishment has to produce the report, and then calculates the amount of all the social security contributions for this establishment.

Annual Leave

Annual leave is absence from work with pay and must be approved by the employee's supervisor in advance. This type of leave (Plan Type 51) is accrued based on years of service: Full-time Permanent/Full-time Seasonal employees ...0-3 years - 4 hours per biweekly pay period; 3-15 years - 6 hours per biweekly pay period (plus an additional 4 hours in the final pay period of the leave year); and 15+ years - 8 hours per biweekly pay period. Part-time Permanent/Part-time Seasonal employees...0-3 years - 1 hour for every 20 hours worked; 3-15 years - 1 hour for every 13 hours worked; 15+ years - 1 hour for every 10 hours worked. Generally, there is a leave year ceiling of 240 hours on accrual; amounts accrued in excess of the ceiling and not used prior to leave year-end are forfeited.

Annual Shareholders Meeting

A meeting of corporation's directors, officers, and shareholders held for the purpose of communicating the operating and financial results for the prior year, the prospects for the future and major decisions of management.

Annual Workforce Survey by Nationality and Professional Category (Enquête sur l'activité et les conditions d'emploi de la main d'oeuvre)

In France, companies are required to submit the Annual Workforce Survey by Nationality and Professional Category to the Ministry of Labor. This report provides an analysis of the company's foreign workforce, which includes any employee who does not have French citizenship.

Annualized Tax Method

A payroll tax calculation method that divides the tax on an annualized amount by the number of pay periods in the year to find withholding for a given pay period, based on the number of withholding allowances. Annualized is the most common tax method.

Annuitant Amount

The gross monthly annuity a federally retired employee receives.

Annuitant CSA Number

A unique number assigned by OPM for a retired employee.

Annuitant Indicator

A code used to indicate the status of an annuitant appointed to a position in the Federal civilian service. Text for the codes is as follows:

1. Reemployed annuitant - Civil Service/FERS
2. Retired military officer receiving pay
3. Retired military non-officer (enlisted) receiving pay
4. Retired military officer receiving pay and a reemployed annuitant - Civil Service

5. Retired military non-officer (enlisted) receiving pay and a reemployed annuitant - Civil Service
6. Not applicable (none of the above)

Annuitant Indicator (cont)

- A. Reemployed Annuitant – FERS
- B. Former Annuitant - FERS
- C. Retired Officer/Reemployed Annuitant - FERS
- D. Retired Officer/Former Annuitant - FERS
- E. Retired Enlisted/Reemployed Annuitant - FERS
- F. Retired Enlisted/Former Annuitant - FERS

Annuity

A series of periodic payments made to an individual. Under a pension plan, these payments are generally made monthly.

Anti-Dilutive

Typically, options or shares where the price is greater than the current fair market value of the security.

APE (Activité Principale Exercée) Codes

APE codes classify the type of industry or activity your French company is in, such as software, banking or insurance. The APE codes are a normalized set of codes that are required by law and are used in regulatory reporting.

API

An Application Programming Interface (API) is the technology that a software product supplies so you can control it or communicate with it from another application. PeopleSoft APIs enable the user to perform desired actions upon PeopleSoft data without having to know the internal logic or rules of the program.

Applicant Hire Process

The procedure of hiring an applicant who has been tracked and administered in the Recruitment pages. Once you assign an Employee ID, the system uses recruitment data to populate the fields in the Personal Data pages.

Application agent

An application agent is an online agent that is loaded into memory with a PeopleSoft page. It detects when a business rule has been triggered and determines the appropriate action.

Application Designer

The integrated development environment used to develop PeopleSoft applications.

Application Engine

PeopleTools batch processes consisting of a set of defined SQL statements. Application Engine processes is more efficient than COBOL or SQR, since they operate within the database system, and don't rely on external processing.

Application Journal Template

A set of rules and default values to control the creation of journals from accounting entries.

Application Processor

The Application Processor is the PeopleTools runtime engine that controls processing of the application from the time the user requests a panel group from an application menu through the time that the database is updated and processing of the panel group is complete.

Application Server

The application server is the centerpiece of PeopleSoft's three-tier architecture. It utilizes Tuxedo, BEA Systems' transaction monitor, to manage client transactions and provide the business rules and workflow capabilities of PeopleSoft's enterprise applications.

Application Server Domain

The collection of server processes and associated resource managers defined by a single PSTUXCFG configuration file. Each application server domain is configured to connect to a single database. Multiple application server domains can exist on the same server machine.

Appointing Authority

The basis that authorized the appointing officer to effect personnel actions on an employee.

Appointing Officer

Denotes if the employee has appointment authority based on laws and regulations.

Approve Time

The Time and Labor feature that approves all employee daily time before it can be sent to payroll for processing. You can approve time by group or by individual employee. You can also unapprove previously approved time.

Approving Official

Individual with the delegated authority responsible for signing the action(s) taken on an employee.

Array

An ordered grouping of data by period and year. PeopleSoft Demand Planning uses arrays in forecasting demand.

Array

Element which enables you to extract information based on a column value. One way of thinking of an array is that it is a SQL statement that retrieves data from an existing table.

Array Dimension

Determines which inventory-stocking possibilities are included in a **Cube View**. This standard one-level dimension consists of the key fields that include, for example, order quantity, safety stock, and turn rate.

Arrears Balance

An amount owed to either the employer or employee, usually the result of a deduction not fully taken.

Ask Price

The price at which someone who owns a security offers to sell it; also known as the asked price.

As-of-Dated

Refers to a snapshot of the data at a given point in time.

Asset Assignment

A streamlined means of associating project costs to assets or asset profiles within PeopleSoft Projects.

Asset Budgeting

Budget for planned asset acquisitions and the associated depreciation expense that can be associated with a Capital Acquisition Plan (CAP).

Asset Catalog

A list of asset profiles which includes information about that asset type, including Cost, Life, Salvage Value, Depreciation Method, Currency Code, and Asset and Depreciation Account.

Asset Category

A standard group of assets. Typical asset categories include Furniture and Fixtures, Machinery and Equipment, Land, Buildings, Leasehold Improvements, and the like. These generally correspond to General Ledger asset accounts. Assets in one category usually share some depreciation characteristics, such as estimated service life and depreciation limits.

Asset Class

An asset group used for reporting purposes. It can be used in conjunction with Category to refine asset classification.

Asset Liability Management

See PeopleSoft Asset Liability Management.

Asset Life

The number of years an asset will depreciate, after which time it might be kept or sold for its Salvage Value. Also *see* Useful Life.

Asset Profile

A template that contains standard depreciation criteria for an asset type and its corresponding asset books. You can use the information in asset profiles as default values when adding assets.

Assignment of Life Insurance

Effective 10/3/94, Federal employees can assign their Basic, Option A and Option B insurance to another person(s), firm(s), or trust(s); Option C is excluded. The assignment of benefits transfers ownership of the FEGLI coverage to the assignee(s). The insured no longer has control over his/her insurance coverage and can no longer designate beneficiaries.

Assignment is irrevocable. Either all or none of the insurance can be assigned. Assignment does not have to be to the same person or firm. Assignments must be made in percentages of total insurance versus an assignment of Basic Insurance to one person and Option A to another. Additionally, terminally ill employees can assign their insurance to a Viatical Settlement Firm in exchange for cash (approx. 60% - 85% of the face value of the coverage). Life Expectancy is usually 24 months or less for a Viatical Settlement Agreement.

Assignment Type

This defines the behavior of the object, (resource, activity, or cost object) within PeopleSoft Activity-Based Management. If the object is identified as a source then costs may be allocated from that object to another object, which must be identified as a target. If an object ID is identified as a target it may be allocated costs from another object ID but may not allocate costs. An object ID can be both a source and a target, thereby having the functionality of each.

Associated Primary BOM

With multiple outputs, it's possible that a given co-product can be created in more than one way – in other words, an item is a co-product on more than one items' primary BOM. By assigning an associated primary BOM to a co-product, you are telling the system which BOM to use in exploding the co-product to the next level.

AT Section

In France, this stands for Section Accident du Travail, or Work Accident Section. It is information needed to identify the establishment risk code for insurance purposes.

ATP Reserved Order

An order that has been promised against future supply. The user has an obligation to the customer to fulfill the order quantity by a certain date. ATP-reserved orders are also referred to as *promised orders*.

Attendance

A component of time reporting application whose purpose is to apply business rules related to Benefit Entitlement and Administration and Organizational Administration to time reported as worked or not worked, and to satisfy a variety of reporting needs.

Attendance Reporting

A Time and Labor report that indicates an employee's attendance record. It includes sick leave, vacation time, and other leaves taken.

Attribute

An attribute is an element within a dimension. For example, the element "Store" is an attribute of the dimension "Geography" for the retail industry. An attribute is also a column heading on an analysis and reporting template.

Audit Trail

See Drill-Back Calculation.

Auditor

Person designated to review expense sheets and cash advances before payment.

Automatic Revision Incrementing (Auto Rev)

The ability to automatically set up revision control and generate revisions for revision-controlled items at the business unit level. This includes setting up a revision scheme or a predetermined, ordered list of revision names.

Automatic Spouse Benefit

A joint and survivor pension benefit provided without any actuarial reduction to a pension benefit. The automatic benefit is a n% joint and survivor; the employee is still entitled to choose any optional form of payment and any beneficiary for the remainder of the benefit.

Availability Date

The date a lot becomes acceptable for fulfillment in PeopleSoft Inventory or for consumption in PeopleSoft Production Management. (Availability Date = Creation Date + Availability Lead Time)

Available to Promise (ATP)

The projected supply of a product less the actual demand, which informs the sales and marketing department of the products that can still be sold without modifying the master schedule. ATP isn't cumulative – it's calculated for each period.

Average Daily Balancing

A feature in PeopleSoft General Ledger that enables you to target the ChartFields on which you base average balance calculations, summarize amounts for selected ChartField values according to your reporting requirements, and define the periods for these calculations.

Used by the financial analytic applications in Enterprise Performance Management. For a reporting period (usually monthly) this refers to the average daily balance of an account as opposed to the month-end-balance, which is the balance as of the last day of the month.

Average Daily Balance Ledger (ADB_Ledger)

In the PeopleSoft Enterprise Warehouse, the Average Daily Balance Ledger table (PF_ADB_LEDGER_F00) is similar to the functionality of the PF Ledger table (PF_LEDGER_F00), in that it too supports reporting. However, the Average Daily Balance Ledger is used for average daily balances. It is a table that is used mostly for processes associated with the financial services industry.

Average Inventory

In PeopleSoft Inventory Planning, one half of the average lot size plus the safety stock when demand and lot sizes are expected to be relatively uniform over time. When demand and lot sizes are not uniform, the stock level versus time can be charted to determine the average.

Average Price

The average price derived from either the bid and ask prices (for bid/ask/average) or from the high and low prices (for high/low/average).

Average Static Calc Flag

In PeopleSoft Inventory Planning, a method used with static policies. The average method sets the static policy equal to the weighted-average, time-phased policy over the next argument periods.

Award

A special payment to an employee for certain prescribed kinds of activities or accomplishments.

B

Back Pay Interest

Under certain circumstances, an employee can be eligible to receive additional pay relative to a delayed receipt in salary caused by administrative error in processing a personnel action. The U.S. Office of Personnel Management has established guidelines for Federal agencies on when and how to make these calculations.

Background Process

Any task or process that is grouped with another and runs in the background. Background processes are usually scheduled to run on a regular basis. All background processes are executed through process-specific COBOL programs run outside the Windows environment.

Backlog Reason Code

An identifier indicating the reason an item could not be shipped. Example codes might include out of stock, discontinued, or seasonal.

BAD Forecast Ratio

In PeopleSoft Demand Planning, the maximum acceptable value of the ratio of the and the base component (Standard Deviation/Base Component). When this value is exceeded, the system automatically resets forecast model parameters. The higher the value, the less likely it is that the system will reset the parameters. In most organizations, a BAD ratio of 1.00 or lower is appropriate for most items.

Balance Segmentation

Balance Segmentation is used in Funds Transfer Pricing to divide balances in deposit accounts between core (stable) and non-core (volatile) segments. Core funds represent the minimum balances that are retained on a long-term basis, building a relatively reliable source of funding to the bank. Non-core funds are temporary in nature due to their volatility caused by customer preferences for liquidity, and cannot be utilized on a long-term basis.

Balance Type

Balance Type is a lookup code used to define the type of instrument balances that will be stored in the PeopleSoft Enterprise Warehouse and processed by the analytic applications. Examples of different Balance are Current Balance, Average Daily Balance, Period Ending Balance, or Commitment Balance.

Balanced Scorecard

See PeopleSoft Balanced Scorecard.

BAM

Business Analysis Model. XXX I think this term is incorrect because we use BAM to refer to the application. If we were referring to the business analysis model, we would say BAM model (that is, Business Analysis Modeler model.)

BAM Model

The BAM database published from the template. The model contains both the data and analytic structure used in the application. The BAM database is physically separate from the Enterprise Warehouse database. Data is sent to the model through migration processes.

BAM Template

A file created using BAM design tools, representing the model prior to its creation as a database. This file has an extension of .MDL. This file is published to a BAM database once the model design process is complete. Each application using BAM will deliver templates which the customer will review and publish to a database in their environment.

Bank Identification Number (BIN)

In PeopleSoft Payables, a part of the bank information that identifies business unit banks.

Base Budget

The initial budget defined by the Budget Coordinator. The base budget is distributed as a starting point for Budget to review and edit. The base budget can be zero-based or incremental.

Base Compensation

In PeopleSoft Workforce Analytics, Cash Compensation that is typically categorized as fixed. It includes base pay and shift differentials as well as associated merit, equity, and step increases.

Base Currency

Base Currency is used to consolidate and report financial results of a multinational company. When a company transacts its business operations in different transaction currencies, those currencies are translated to the base currency for reporting purposes.

Base Currency Equivalent (BCE) Amount

If the monetary amount is in a currency other than the base currency, either the Extract-Transform-Load (ETL) process or the Multi Currency Engine can be used to convert the monetary amount to the Base Currency Equivalent (BCE) Amount.

Base Factor

In PeopleSoft Demand Planning, an element of a smoothing constant simulation set that controls base component smoothing in the Model Reset Simulation process.

Base Metric

Metric found on a fact table. A base metric usually contains an aggregate operator, for example “sum” or “count”.

Base Pay

A pay component included in the job comp (job compensation rate) calculation. It is pay for a regularly assigned workweek. For example, you can set up a regular hourly rate plus a shift rate, a union-negotiated rate for hazardous work, and so on.

Base Pay Structure

A PeopleSoft Workforce Rewards module you use to create or revise pay structures, and to assess the cost and impact of implementing new structures.

Base Time Zone

Customer defined time zone used for converting reported time to a common time zone for ease of applying rules (see Time Administration).

Batch

Batch systems are used when realtime updates are not needed. Batch-oriented data collection applications, developed in-house or by a third-party vendor, produce transactions that are collected in an ASCII text file. The text file is fed to a PeopleSoft SQR program that loads the transactions into the database.

Batch Processes

Any of the background programs in the client/server environment of PeopleSoft applications. Batch processes perform operations—such as pay confirmation, deduction calculation, and so forth—on groups of records, and are usually scheduled to run on a regular basis. You run these processes from the Process Scheduler, and they are executed through process-specific COBOL programs.

Before-Tax Deduction

Deduction that reduces net pay and FWT taxable gross, applied prior to the calculation of federal and state/provincial withholding taxes. Also called “pre-tax” deductions.

Begin Calc Date

The date on which PeopleSoft Asset Management begins to deduct from an asset's life.

Begin Depr Date

The date on which PeopleSoft Asset Management begins to calculate depreciation for an asset. Begin Depr Date is calculated using In-Service Date and Prorate convention.

Benchmark Job

In PeopleSoft Workforce Analytics, this refers to a Job Code for which there is corresponding salary survey data from published, third party sources. Jobs for which there is no corresponding salary survey data are referred to as non-benchmark jobs.

Benefit Commencement Date (BCD)

The date on which a pension payee elects to begin receiving payments.

Benefit Deduction

Any amount taken from an employee's pay check to offset all or part of the cost of the employee's benefits.

Benefit Eligibility

The PeopleSoft Pension Administration function that determines if an employee is eligible for retirement or ancillary benefits. A plan may have several retirement types—normal, early, late, death, and disability—each with its own eligibility criteria.

Benefit Entitlement

Any rules governing the circumstances under which employees are entitled to receive certain benefits. Typically, entitlement to benefits is based on type of employee (for example, full time, part time, occasional), length of employment, and specific rules which apply thereto, i. e., work group affiliation, and compensation base. Other criteria may also apply, such as reasons-for-claiming or job performance.

Benefit Formula

The formula that determines a participant's pension benefit in a defined benefit plan, as well as the PeopleSoft Pension Administration function that calculates the benefit.

Benefit Group

Part of a group of defaults assigned to job codes. Benefit group may include medical, dental, and health benefits dependent on individual company parameters.

Benefit Plan

A specific benefit within a plan type. For example, your company's life plan type might include benefit plans of one times salary, two times salary, and three times salary.

Benefit Plan Type

Any category of benefit, such as health, life, or savings.

Benefit Program

A set of benefits and deductions valid for an employee or group of employees. A single company may have any number of programs. An individual employee may belong to only

one program; the deductions and benefits contained in that program are the only valid deductions and benefits for that employee.

Benefit Tables

Any of the tables that contain employee benefits information. These are often relevant to payroll processing.

Benefits Base

The salary used for benefit calculations. The benefits base will be either the employee Annual Rate or Annual Benefits Base Rate.

Benefits Compensation

In PeopleSoft Workforce Analytics, Benefits Compensation is value associated with employment benefits. It can include benefits types for Health and Welfare (Medical, Life Insurance), Retirement (annuities, savings plans, pensions), and Paid Time Off (Vacation Leave, Sick Leave). Benefits compensation is sometimes fixed, and sometimes variable, depending upon the benefit type.

Betriebszählung (Company Statistics Report)

Also called the OFIAMT report. This report provides statistics required by the Swiss Federal Department of Statistics (BFS).

Bias Signal Limit

In PeopleSoft Demand Planning, a number between one and six that indicates how many **Forecast Period** to test for bias. If the bias test is violated, the system records a **Tracking Signals** error in the period up to the number of periods determined by the bias signal limit.

Bias Test

In PeopleSoft Demand Planning, a forecasting test that sets the limit for tripping a **Tracking Signals**. The lower the value, the more likely it is that a tracking signal is set.

Bid Price

The price a prospective buyer is prepared to pay at a particular time for trading a unit of a given security.

BIF file

This is the bulk insert file (input.bif) used with the Verity search engine to specify the documents to be submitted to a collection (search index). It contains a unique key, document size (in bytes), field names and values, and document location in the file system.

Bilan Social Report

See Employee Survey Report.

Bill

In PeopleSoft Billing, any group of bill lines.

Bill Adjustment

The process of making credit or credit and rebill adjustments to an invoiced billing activity.

Bill By Identifier

The Bill By Identifier is used to define how billing activity is grouped when added to a bill through the billing interface or the Populate Billing process.

Bill Header

The record containing information that pertains to the bill as a whole. Each bill has a unique bill header that identifies it within the system.

Bill Inquiry Phone

Bill Inquiry Phone is the number printed on your invoices for your customers to call if they have any questions about their bill.

Bill Line

The basic unit of billing activity representing a billable charge, including the charge identifier, quantity, price, and any other information regarding an individual transaction. Every bill line is related to a bill header that may have one or more bill lines related to it.

Bill Search

A method of finding a bill or bill line when you don't have enough information to call up the bill directly. **Customer Bill Search** enables you to locate a bill by Customer Name. You can also choose other parameters to limit your search. With **Bill Line Search** you first search for a particular bill and then a line on that bill. Parameters for bill line search include Reference, Date, and Amount.

Bill Source

The point where billing activity originates. Bill sources may be external to the system (imported through the billing interface) or entered directly online. Examples of bill sources include order management, project costing, and contract administration.

Bill To Customer

A customer who receives an invoice.

Bill Type

A category of billing activity variety. Examples of Bill Types include standard and custom order activities.

Bill Update

The process that adjusts bills that have either been entered manually or generated within the system.

Billable Indicator

A status flag that identifies an item as eligible for billing to a customer.

Billback Discount (BB)

A per unit discount which typically requires a customer to perform one or more merchandising activities to receive the discount. A BB discount is not deducted from the customer invoice, but once the customer performs the merchandising activity, a sales representative or broker can approve payment for the discount amount. Billback discounts can originate from a National Allowance or Customer Promotion, and are passed to PeopleSoft Order Management for informational purposes only. Billback discounts are recognized as a liability when the product is shipped.

Billing Location

A number identifying a customer address. Each customer may have multiple locations, but must have one *Primary Location* at which you contact them.

Blackout Period

The period of time, determined by the company, which prohibits certain activity in the company stock. Blackout Periods can affect the trading of some key individuals or can be placed on the entire company.

Bonus Tax Method

Annualizes your year-to-date earnings by multiplying them by the number of pay periods in the year. This method is used for Canadian tax processing.

Book

In PeopleSoft Asset Management, a data location storing financial information—like cost, depreciation attributes, and retirement information—on assets.

Borrow/Loan

The temporary reassignment of an employee to other task reporting or compensation requirements to allow the business to meet unexpected, short-term, fluctuations in staffing or work load. Typically, this kind of reassignment is done informally at a local level, where HR isn't involved and a new job record isn't created. Companies may have specific rules about how long an employee may be borrowed/loaned, how and where productive, non-productive, and compensated absence time will be charged, and what business rules to apply to the borrowed employee's time for the purpose of compensation and benefit entitlement and administration. See also Casual work Assignment.

Bracket

Brackets are a way to look up and retrieve database table values. After you've defined a table, the system finds a corresponding row on that table and returns the value of the bracket. The result is then available for use in other items such as formulas.

Branch

A tree node that rolls up to nodes above it in the hierarchy, as defined in the Tree Manager.

Branch Of Military Service

Identifies, if any, military service in which the employee served.

Breadcrumbs

Breadcrumbs show the navigation path to the current web page location. As you drill down through the different levels of the registry, a “breadcrumb trail” appears that shows the path you’ve selected. Each registry level is separated by an angled brace (>), and you can select any level to navigate directly back to that level.

A typical Breadcrumb would look like this:

Home > HR > Administer Workforce > Benefits

Break Funding

Charges assessed for mortgages that are paid off before maturity. In the Funds Transfer Pricing (FTP) application, Break Funding charges are factored into the transfer price for a loan that may be prepaid.

Break in Service

A period of time for which an employee does not meet stated service requirements.

Break Price

The price used to determine which options are eligible for repricing. For example, if the break price is \$36, then all outstanding option with a grant price of \$36 and greater are eligible for repricing.

Break Punch

An in/out punch of when a time reporter takes a break.

Brokers

Individuals or organizations who buy and sell securities. Often they are account executives who work for firms registered with the Stock Exchanges and the SEC. Unlike Transfer Agents, (who are not responsible for sales) Brokers do not maintain records on all your company’s certificates. They maintain only sales records and stocks for their clients.

BSC (Balanced Scorecard)

See PeopleSoft Balanced Scorecard.

Budget Activity

A type of activity performed using PeopleSoft Budget Planning. Budget activities include Line Item Budgeting, Line Item Mass Adjustments, Budget Allocations, and Position Budgeting. PeopleSoft Budget Planning-specific.

Budget Amount Ledger

Stores budget amounts and is updated by posting budget entries, transfers, and adjustments.

Budget Analyst

A role within PeopleSoft Budgeting. Budget Analysts are typically people within an organization responsible for reviewing and analyzing a prepared budget before submitting it to the Budget Coordinator. PeopleSoft Budgeting-specific.

Budgetary Account Only

An account used by the system only and not by users; this type of account will not accept transactions. You can only budget with this account. Formerly called “system-maintained account.”

Budget Category

A set of related expenses that are accumulated for proposal budgets and reporting to a sponsor. The estimated cost for a set or class of accounts.

Budget Category

Numeric/alpha identification given to each category of positions.

Budget Center

In PeopleSoft Budgets, any entity responsible for producing or reviewing budget data. For example, a Budget Center might be the individual departments responsible for producing budgets.

Budget Center Dimension

In PeopleSoft Budgets, the dimension by which you distribute budget data. If you budget by department, your department dimension will be your Budget Center Dimension. You'll assign Budgets Users to the nodes and detail values on the tree you use to build your Budgets Center Dimension.

Budget Check

In commitment control, the processing of source transactions against control budget ledgers, to see if they pass, fail, or pass with a warning.

Budget Check Override

Selective suspension of Budget Processing. With this feature you can override the controlled budget for a transaction that failed budget checking due to insufficient funds; or override the tolerance limits for a transaction rejected due to exceeded tolerance limits. When you push the Override button, the system flags the transaction to allow the Budget Processor to process successfully regardless of available funding. You can cancel the override any time before the Budget Processor is run by clicking the Cancel Override button.

Budget Control

In commitment control, it ensures that commitments and expenditures don't exceed budgets. It enables you to track transactions against corresponding budgets and abort a document's cycle if the defined budget conditions are not met. For example, you can prevent a purchase order from being dispatched to a vendor if there are insufficient funds in the related budget to support it.

Budget Coordinator

A role within PeopleSoft Budgeting. Budget coordinators are responsible for monitoring the budget process. The Budget Coordinator is typically located within an organization's central budget office and builds the budgeting model. PeopleSoft Budgeting-specific.

Budget Detail

A level of itemization that when combined makes up a major budget category.

Budgeted Rates

In PeopleSoft Activity-Based Management, the rate your organization uses based on the budget.

Budget Error Exception

A transaction that fails budget checking, causing an Error or Warning to be issued. See **Error Exception** and **Warning Exception**.

Budgeting Functions

PeopleSoft Budgeting's six main action categories, including: system administration, budgeting setup, budgeting preparation, budgeting analysis, data integration and my profile. Your user role determines how many of these functions display and are available.

Budgeting Model

The framework for an organization's budget development process. Business unit defines a Budgeting Model. The Budget Coordinator typically defines the model and includes the time period of a budget cycle, time period for phases within a budget cycle, the sources of data that will be available to budget users, the methods that will apply to line-item budgets, and other budget options and control parameters. PeopleSoft Budgeting-specific.

Budgeting Type

Associated with the budget ledger type set definition, a budget type is an indication of whether the organization uses a standard budget ledger, project budget ledger, or controlled budget ledger for budgeting.

Budget Justification

Written explanation further defining the what and why of a budget category.

Budget Period

The period in which you define plans to meet your organizations training requirements.

The interval of time (such as 12 months or 4 quarters) into which a period is divided for budgetary, and reporting purposes. The ChartField allows maximum flexibility to define operational accounting time periods without restriction to only one calendar.

Budget Phase

In PeopleSoft Budgets, a span of time during which a budget or portion of a budget is to be completed. You'll filter dimensions, assign alternate Budgets Users, enable Position and Asset budgeting, and specify Budgets User notification options at the Phase level.

Budget Plan

In PeopleSoft Workforce Rewards, when working with a Compensation Planning BAM model. A budget plan is a rollup of like compensation rules. For example, for base pay rules budget plans are a rollup of values for like Action Reasons. For variable pay rules budget plans are a rollup of the values for like Variable Compensation Plan IDs.

Budget Preparer

A role within PeopleSoft Budgeting. Budget preparers are typically people within an organization responsible for developing the detailed budget for a Budget Center and submitting it to a Budget Reviewer or Analyst for review and approval. PeopleSoft Budgeting-specific.

Budget Reviewer

A role within PeopleSoft Budgeting. Budget reviewers are typically people within an organization responsible for reviewing and approving a prepared budget submitted by a Budget Preparer. PeopleSoft Budgeting-specific.

Budget Seeding

Represents a new budget or forecast, such as historical data that is manipulated to develop a more current representation for a proposed budget. Uses detail data as the budget seed or basis to create the base budget that represent the level of detail in which budget numbers are prepared.

Budget Translation Trees

Trees translate (summarize) source transactions into the appropriate levels for processing against control budgets. This is because you usually budget above the level of your source transaction ChartFields on a tree.

Budget Type

Indicates whether a budget is for expenditures or revenues.

Budget Warning

See **Warning Exception**.

Budgets User

In PeopleSoft Budgets, any user who needs to gain access to the Budgets. You'll designate Budgets Users on the Budgets Users page through the Coordinate Budgets window. You'll also assign these users to the tree representing your Budget Center Dimension.

Budget View

A user-defined view where selected dimensions, columns and rows of data determine the layout of line-item budgets affecting the view or entry of data.

Budget Year

The institutionally defined, consecutive, 12-month period to which a financial transaction or summary applies.

Build Option

A detailed PeopleSoft Planning model that specifies a method of building an assembly item. This model specifies the routing, resources, and materials that are necessary to produce the item.

Built-in function

Prior to PeopleTools 8.0, there were only built-in functions, like FetchValue, ScrollSelect, etc. A built-in function, in your code, is on a line by itself, and doesn't (generally) have any dependencies. You don't have to instantiate anything before you can use a built-in.

Business Interlink Definition

A definition encapsulating an external Transaction or Query and providing a set of generically typed input/outputs that can be assigned to PeopleCode variable or Record Fields at runtime. A Business Interlink Definition is added to the Application Designer's objects at the same level as Fields, Records, Panels, etc.

Business Interlink Design-Time Plug-in

An XML file that, when coded for an external system, encapsulate that external system and provide a catalog of Transactions, Classes and Criteria specific and meaningful to that external system.

Business Interlink Framework

The framework for integrating any external system with PeopleTools application objects. It is composed of the following components:

1) An External System, 2) Generic definitions for a Transaction/Query command interfaces, 4) Business Interlink Definitions, 4) Business Interlink Plug-in.

Business Interlink Object

An instantiation based on a Business Interlink Definition. Actual data can be added to the inputs of the Business Interlink Objects once the appropriate bindings are provided. The Business Interlink Object can be executed to perform the external service. Once a Business Interlink Object is executed, the user of that object can retrieve the outputs of the external service. The Business Interlink Objects use buffers to receive input and send output. When a Business Interlink Object is executed, the transaction/query/class associated to the Business Interlink Object will be executed once per each row of the input buffers corresponding to the input Records. If there is only one row, after appropriate substitution by the driver, it is executed only once.

Business Interlink Runtime Plug-in

A set of C++, Visual Basic, or other high-level language methods that, when coded for an external system, encapsulate that external system and provide the execution methods to match the Business Interlink Design-Time Plug-in. (The catalog of Transactions, Classes and Criteria provided by the Design-Time Plug-in can also be provided by the Runtime Plug-in.)

Business Objects

A way of identifying those mass changes that have been designed to be referenced by a flexible formula and provide them with a shorter name to simplify the creation of flexible formulas.

Business Planning

The type of planning that focuses on elimination activities that are not needed by changing the drivers.

Business Rules

Rules that can process information differently depending on the values of data in the PeopleSoft Enterprise Warehouse.

Business Unit

A corporation or a subset of a corporation that is independent with regard to one or more operational or accounting functions. PeopleSoft General Ledger business units typically comprise individual entities for accounting purposes.

Business units in PeopleSoft Projects represent operational structures but not necessarily independent financial units.

PeopleSoft Payables business units are either *Vouching* (have payables accrued to them) or *Charge to* (have voucher expense distributions charged to them), and pass journals to general ledger units.

PeopleSoft Purchasing business units share vendor, purchase order, and receiving information with PeopleSoft Payables units in the same SetID.

A PeopleSoft Inventory business unit is a storage facility that maintains its own replenishment and costing methods, as well as its own definitions and guidelines.

The Manufacturing business unit must be identical to the Inventory business unit in order to link the manufacturing and inventory processes.

The Order Management business unit controls certain order processing parameters (tax and freight calculation methods, base currency, credit card hold options, and so on) for its associated PeopleSoft eStore and Mobile Order Management merchant variants.

Business Unit Audit List

One or more business units specifically targeted for expense report and cash advance audits.

Buying Agreement

You can structure flexible and easy-to-use buying agreements for customers or groups of customers. You can set up maximum amounts and specify the minimum dollar value per order placed against it. You can automatically generate sales orders or create sales orders online from buying agreements. Rebate and penalty calculations can be implemented for buying agreements.

C**Cafeteria-Style Benefits**

Any programs offering several benefit plans from which participants make elections. Cafeteria-style benefits may or may not include flexible credits.

Calculation

In PeopleSoft Pension Administration, the determination of a participant's pension benefit.

Calculation Rule

Criteria for calculating benefits, including as-of dates for age, service, premium, and coverage calculations; rounding rules; and minimum and maximum coverage amounts. Any number of program and plan combinations can use a single set of calculation rules.

Calculation Rule [Global Payroll]

Any rule you develop using combinations of elements to command the system to perform a type of calculation.

Calendar

In PeopleSoft Manufacturing, a list defining the days your enterprise is available and the hours of operation for each day. The system first looks to see whether you are using a work center specific calendar. If none is defined, it looks at the production calendar. If no production calendar is defined, planning and scheduling functions base start and due dates on a five-day workweek.

In PeopleSoft Demand Planning and Inventory Planning, a list defining the start and end dates for each time-phased period. It also contains daily weights for distributing raw data into different period buckets.

In PeopleSoft General Ledger, your accounting calendar defines the time periods to which you post transactions for different ledger group and business unit combinations. You can have multiple calendars, so you can keep a calendar for actuals, another for budget and forecast activity, and still others for special reporting or transitional needs.

Calendar Group ID

Allows you to group together multiple Calendars that you want to run together at the same time. It also controls the order in which the Calendars are processed. You can only group calendars together that are for the same country (based on pay entity country).

Calendar Scope

A time period type (Day-Factored, Month-Factored, or Week-Factored) for use in building your time period calendar.

Canada Academic Teaching Surveys

Statistics Canada requires that all Canadian universities (all degree granting institutions) produce full-time and part-time *Canada Academic Teaching Surveys*. These reports are a legislative requirement. PeopleSoft HRMS 8 provides you with the functionality to code HRMS information using Statistics Canada codes and create both the full-time and part-time Academic Teaching Surveys.

Canadian Industrial Sector

The Canadian industrial classification code with which employees are associated for Canadian employment equity reporting purposes.

Canadian National Occupational Classification (NOC) Codes

NOC codes are occupational classification codes for Canadian companies provided by the government.

Canadian Standard Occupational Classification (SOC) Codes

SOC codes are occupational classification codes for Canadian companies provided by the government.

Cancellation

A process that terminates stock fulfillment requests, allowing reserved and allocated items to be returned to inventory.

Cancellation

In the context of an employee stock plan, a transaction (usually triggered by a specific event, such as a termination of employment) in which outstanding securities are declared void and inactive and returned to the pool of securities reserved for issuance under the plan or retired.

Candidate Keys

In PeopleSoft Demand Planning, elements of data that can be used to construct the **Forecast Item** key field at different levels of the forecast.

Capacity Rate

A rate you assign to a capacity cost object. This enables you to track and report on excess capacity.

Capacity Fence

A time fence that indicates that date and time after which PeopleSoft Enterprise Planning or Production Planning solvers ignore capacity violations. The solvers do not use this date in processing capacity violations.

Capacity Multipliers

A multiple used in PeopleSoft Enterprise Planning and Production Planning to determine the available capacity on a resource. Since a capacity multiplier is effective-dated, you can use it to vary the resource's available capacity over time.

Capital Acquisition Plan (CAP)

A method of projecting and tracking capital expenditures for a project. Budgeted assets and actual expenditures can be associated with a CAP Plan so the owner can track planned against actual costs.

Capital Gain

The difference between an asset's purchase price and selling price, when the difference is positive. Capital gains can be either short-term (where the capital asset was held for 12 months or less) or long-term (where the capital asset was held for 12 months or more).

Capital Gains Tax

A tax on profits from appreciation in owned real property, recognized at the time the property is sold; real property includes owned company shares.

Capitalization

The total types and amount of the outstanding securities that have been issued by a corporation. Generally includes both equity and debt securities.

Capital Markets Instrument

In the financial services industry, Capital Market Instruments are assorted financial instruments issued by organizations to raise capital for funding operations. Participants are made up of interested parties that choose to supply or acquire the capital funding through such vehicles. Derivatives, debt instruments, equities and foreign exchange instruments that are traded in highly liquid markets represent the instruments. In the PeopleSoft financial analytic applications, Capital Market securities refer to instruments that are bought/sold by the institution for its own investment account. The capital markets set the product prices and interest rates.

CAP Sequence Number

The number that distinguishes a small project belonging to a CAP plan. Budgeted assets can be associated with an overall CAP Plan and a CAP Sequence, if that level of detailed tracking is desired.

Carry-Forward

Residual contributions that remain in a stock purchase participant's account after the purchase of shares that are used toward future purchases.

Carrying Cost

In PeopleSoft Inventory Planning, a value that shows the cost associated with holding a dollar of inventory for one year. The value is presented as a percentage.

Case Officer

In Germany employees in your company are designated as Case Officers, and have responsibilities for handling health and safety incidents.

Cash Balance Accounts

The PeopleSoft Pension Administration function that tracks the activity in an employee's hypothetical account under a cash balance plan.

Cash Balance Plan

A defined benefit plan designed to look like a defined contributory plan. The plan periodically credits a percentage of pay to each employee's hypothetical account.

Cash Compensation

In PeopleSoft Workforce Analytics, Cash Compensation is a component of direct compensation. Cash Compensation consists of direct cash payments made to an employee for base compensation and short-term variable compensation.

Cash Exercise

At the time of exercise, the optionee is required to pay in cash the total option price plus any withholding taxes due to the company.

Cash Flow Generator

This is a support module for the PeopleSoft financial services analytic applications. It generates actual and projected cash flows for financial instruments by using output from the other support modules, such as loan prepayment rates, deposit runoff rates, product pricing indices, discount rates, and product definitions (such as start and end dates, balance amount, interest rate, term, payment dates, repricing and compounding frequency, and accrual basis) to generate the cash flows. The Financial Performance Measures module accesses the cash flow results to calculate the required financial measures.

Casual Preparer

An additional user role at the lowest level of budget preparation for a budget center. This user performs the same activities as the Budget Preparer role when access is granted. The system does not, however, enable the Casual Preparer role to define their own private views for line-item budgeting.

Casual Work Assignment

The temporary assignment of an employee to a work position or location to meet the needs of the business. Typically, there is no Human Resource activity to support the work assignment (that is, a new Job record is NOT created). Often compensation rules that accrue to the temporary assignment override the compensation rules that apply to the employee's normal work assignment. See also Borrow/Loan.

Catalog

The list of transactions, classes, and queries used to interface to the external system. Integration users are presented with this list when they pick the type of Business Interlink Plug-in they are going to use. There are four types of catalogs: transaction, class, operator, and configuration parameter.

Catalog

A way of organizing your training courses into classifications for increased flexibility. Catalogs consist of categories and subcategories.

Category

Categories are the primary level of a two-tier structure of training courses. Categories can consist of subcategories that provide further course definition.

Category Tree

A hierarchical structure that groups products by category to control how they are displayed in PeopleSoft eStore web pages. Used also by Mobile Order Management to enable product information to be accessed by a wireless device.

CBM

See PeopleSoft Customer Behavior Modeling.

Census Metropolitan Area (CMA) Code

In Canada this code is prescribed by the government and refers to the area of an urbanized core with a population of at least 100,000.

Central Personnel Data File (CPDF)

Two types of reporting made by agencies to the OPM include the Dynamic and Status files (quarterly and monthly, respectively) covering a range of employee personnel/payroll data.

Certain and Continuous Payment Option

A form of pension payment where the benefit is paid out for the lifetime of the participant with a specified number of payments guaranteed so that a beneficiary will receive payments until the end of the guarantee period if the employee dies before the guaranteed payments are complete. For example, under a ten-year certain and continuous payment option, a retiree who lives less than ten years receives payments until death, then the retiree's beneficiary continues to receive payments for the remainder of the ten year period. A retiree who lives longer than ten years continues receiving payments after the ten year period until death. Also known as a "Term Certain and Continuous" payment option.

Certain Only Payment Option

A form of pension payment where the benefit is paid out entirely over a specified period of time—usually five, ten, or fifteen years—with no ongoing payments after the specified period. If the retiree dies before payment period is over, the remaining payments are made to a beneficiary. Also known as a "Term Certain" payment option.

Change To Lower Grade

- For positions under the General Schedule or under the same wage grade schedule, a change-to-lower grade changes the employee to a lower grade; and
- When both the old and new positions are under the same type ungraded wage schedule, or in different pay-method categories, a change-to-lower grade changes the employee to a position with a lower rate of basic pay.

Charge Out

A **Material Issue** used when the item is scheduled for future return.

ChartField

A field storing a chart of accounts, resources, and so on, depending on the PeopleSoft application. ChartField values represent individual account numbers, department codes, and so forth.

ChartField Balancing

PeopleSoft enables you to set up ChartFields and indicate that you want specific ChartFields to match (balance) on the debit and the credit side of a transaction. When you work with Controlled Budgets, the Fund and Budget Period are already set up in the system to balance (match). For example, suppose you want to balance by Class and Program. You indicate that these on a panel that these ChartFields are required, along with Fund and Budget Period which should already be selected. When you enter a transaction, you must enter the same Class, Program, Fund, and Budget Period ChartFields on both sides of the accounting entry. but you can modify any ChartFields, other than these four, on the user-defined line. The system always requires that total debits equal credits.

ChartField Combination Edit

Also called *Combo Edit*. The process of editing journal lines for valid ChartField combinations based on user-defined rules.

ChartKey

One or more fields that uniquely identify each row in a table. Some tables contain only one field as the key, while others require a combination.

ChartViews

Charts of data in the model, presented through the Worksheet which retains the ability to drag dimensions on the chart as desired.

Check In/ Check Out

The process of retrieving planning activities from the BAM database (check out) and posting changes and results back into the database (check in).

Child

A node or detail on a tree linked to another, higher-level node (referred to as the parent). Child nodes—projects, customers, and so on—can be rolled up into the parent. A node can be a child and a parent at the same time depending on its location within the tree.

Child

A node or detail of a tree linked to another, higher-level node referred to as the parent. Child nodes can be rolled up into their parent. A node can be a child and a parent at the same time depending on its location within the tree.

Chunking

Chunking is a PeopleSoft Enterprise Warehouse mechanism that makes voluminous processing easier through the use of multiple small parallel processes. By enabling chunking, multiple jobs are spawned from one Jobstream. These jobs run in parallel (behind the scenes) to process data efficiently.

Citizenship Code

Numeric indicator as to whether the employee is a U.S. citizen or a foreign national serving in the U.S. The codes are:

- citizen
- other

Civil Service Retirement System (CSRS)

A retirement plan available to employees of the federal government. CSRS covers all employees appointed to a position in the federal government before January 1, 1984. Coverage includes a basic annuity plan with employee contributions and the Medicare Hospital Insurance component (1.45%) of the Social Security tax.

Class catalog

Lists classes used to interface to an external system. A class contains data members of basic types and/or objects that are typed after another class. A Class can also contain lists of basic types or objects.

Class ChartField

A ChartField value that identifies a unique appropriation budget key when you combine it with a Fund, DeptID, and Program Code as well as a Budget Period. Formerly called “sub-classification.”

Classification Code

Need App A code that categorizes an engineering change. Example classification codes include the following: Mandatory, Optional, Upgrade, Quality, and Safety.

Clock Hour Reporting

Method of reporting time by recording actual times in and out (start and stop) (see Time Reporting).

Clone

To create a unique copy of an object. When used in PeopleCode, clone will always mean to make a unique copy. Copy, on the other hand, may or may not mean making a unique copy. Copy may mean making a new reference to an object, so if the underlying object is changed, both the copy and the original change.

Cloning

The process that enables you to copy run controls to create employee schedules from existing Run Control ID's that have already been executed and saved.

Close Date

The date in which time entry is no longer allowed for a given pay period. Defined as an offset number of days to the pay period end date.

Close Price

The price of the final trade for a security at the end of the trading day.

Closure Calendar

A calendar that establishes closure dates for shipping, receiving, and materials management operations for a specific **Business Unit**. Typically, application processes account for these closure dates when determining Lead Time and dates for anticipated fulfillment processing dates (scheduled shipment dates, scheduled arrival dates, and lot retest dates, for example).

CMA (Census Metropolitan Area) Code

In PeopleSoft Workforce Analytics, the CMA code is prescribed by Statistics Canada, and refers to the main labor market area of an urbanized core with a population of at least 100,000.

COBRA (Consolidated Omnibus Budget Reconciliation Act)

In PeopleSoft Workforce Analytics, this refers to legislation that requires employers to offer continued health care coverage to employees, and their dependents, who lose benefits coverage under certain defined conditions such as voluntary termination, divorce, becoming an overage dependent, or retirement. Any individual, whether employee or dependent, that is covered under a health plan at the time of a qualifying event, has the option to elect COBRA coverage.

Codepage

One character set.

Collection

To make a set of documents available for searching in Verity, you must first create one or more collections. A collection is set of directories and files that allow search application users to use the Verity search engine to quickly find and display source documents matching various search criteria. A collection is a set of statistics and pointers to the source documents, stored

in a proprietary format on a file server. Since a collection can only store information for a single locale, PeopleSoft maintains a set of collocations (one per language code) for each search index object.

Combined Federal Campaign (CFC)

A vehicle used by federal employees to contribute to a charity or charities of their choice.

Commercial-Off-The-Shelf (COTS)

Equipment or software that is currently sold commercially to at least one customer.

Commission Tax Method

A payroll tax calculation method that adds year-to-date earnings to earnings for this pay period and finds the annualized gross by multiplying by the number of pay periods in the year; the gross is then divided by the number of tax periods specified on the paysheet. This method is used for Canadian processing only.

Commitment Control

Commitment control includes budget control and commitment accounting functionality.

Common Shares Issued and Outstanding

Represents the residual ownership interests in the corporation. This is the composite number of shares available and tradable on the open market.

Community Background

In the United Kingdom Community Background refers to the religious category, such as Catholic or Protestant, of employees, job applicants or appointees. See the Northern Ireland Report for more information.

Compa-Ratio

In PeopleSoft Workforce Analytics, Compa-Ratio is most commonly defined as the relationship between current pay and the midpoint calculated as: $(\text{Incumbent Pay} / \text{Midpoint}) * 100$. Usually expressed in whole numbers, or in percentage form by dropping the multiplication operation. Much less common is the use of a compa-ratio calculation as: $\text{range midpoint} / \text{market rate}$.

Compensation Frequency

In PeopleSoft Workforce Analytics, this is the frequency at which a job is paid. This is the value you use for reporting or quoting pay. Examples include Annually, Monthly and Weekly.

Compensation Planning

In PeopleSoft Workforce Analytics, this is the process through which employee compensation plans are defined, and compensation budgets are allocated throughout an organization. Major components of compensation planning include designing pay structures, setting individual pay levels, and budgeting and forecasting compensation spending.

Compensation Rate

In PeopleSoft Workforce Analytics, this is the compensation rate for a job. This is the rate the company uses for quoting and reporting pay.

Comp time (compensatory time)

A PeopleSoft Time and Labor-managed employee benefit where time off is granted in exchange for time worked based on customer-defined criteria; is associated with an expiration and is used as reported time (see Attendance).

Compensation

The process by which a worker is remunerated for services rendered to, or work performed on behalf of a business entity.

Compensation Package

All of the base and non-base components on a job row.

Compensation Rules

Business methodology or logical process that is applied to reported time in order to determine payable time (see Time Administration).Competency

In PeopleSoft Workforce Analytics, Competency is a knowledge, ability, skill, accomplishment, or National Vocational Qualification (NVQ).

Competency Inventory

All of the roles, tasks, competencies and accomplishments possessed by the workers in the current workforce. This data is migrated from internal source systems into the data warehouse tables of the PeopleSoft Enterprise Warehouse.

Competency Strategy

The type and number of roles, tasks, competencies and accomplishments essential to accomplishing a business scenario based on your strategic business goals.

Competitive Appointment

An appointment to a position in the competitive service following open competitive examination or under direct-hire authority. The competitive examination, that is open to all applicants, may consist of a written test, an evaluation of an applicant's education and

experience, and/or an evaluation of other attributes necessary for successful performance in the position to be filled.

Competitive Service

All positions as defined by 5 USC 2102 in the executive branch of the Federal Government are in the competitive service unless they are specifically excluded from it. Positions in the legislative and judicial branches are outside of the competitive service unless they are specifically included.

Compress

The act of placing a Planning task as early as possible in the schedule without violating any constraints.

Compressed Split

In PeopleSoft Demand Planning, an optional function that allows a split database to be compressed so it can be transferred to an account manager's computer.

Concurrent Offerings

Multiple stock purchase offerings that are active and outstanding at the same time. The end date is measured from the employee's grant date.

Concurrent Processing

The situation in which you run multiple batch processes at a time. In PeopleSoft Benefits Administration, for example, simultaneous open enrollment and event maintenance qualifies as concurrent processing.

Configuration Code

A unique 50-character identification code that accurately tracks and costs inventory with the PeopleSoft Product Configurator. It corresponds to a lot number for a non-configured item.

Configuration Costing

The overall process of reviewing and evaluating anticipated cost data for a configured item.

Configuration parameter catalog

Used to configure an external system with PeopleSoft. For example, it might set up configuration and communication parameters for an external server.

Consolidate Assets

In PeopleSoft Asset Management, the process of consolidating multiple load lines, usually coming from a separate application, into one asset.

Consolidate Depreciation

In PeopleSoft Asset Management, the process of summing all open Add and Adj transactions by transaction type, **Transaction Date**, and accounting date for all composite members reporting to one composite asset.

Consolidated Bill

A grouping of bills gathered together for invoice presentation. The bills belonging to a consolidated bill are invoiced and printed together, with a page summarizing the bills as a group.

Consolidations

The PeopleSoft Pension Administration functions that accumulate hours, earnings, and pension contributions based on payroll data.

Consolidations-Elimination Set

A related group of intercompany accounts that is processed during consolidations. Once eliminated, this group of accounts should normally net to zero.

Constraint

A limit to a schedule, that, when violated, must be repaired to produce a valid schedule. User-configurable Planning constraints include Missed Request Dates, Missed Promise Dates, BI Shortages, RM Shortages, Capacity Overloads, Missed Inventory Targets, Changeovers, and Excess Inventory. See also **Temporal Constraint**.

Constraints

In the PeopleSoft Enterprise Warehouse, a constraint can consist of one or more filters and is used to define complex business logic. Constraints are based on DataMaps.

Consumption Pattern

In PeopleSoft Activity-Based Management, an attribute used to describe how an activity interacts with the cost objects to which it has been assigned. A unit type activity can expect to be performed on a regular basis so that each time a product is produced. A batch type activity may only be performed periodically for a given range of transactions. For example, each time a machine is setup to produce another product type. Sustaining type activities generally occur to support the overall operation of a company unrelated to products produced or customers served.

Contact

A person associated with a Customer ID. Contacts can be internal contacts or external contacts. Internal contacts are your employees who manage the relationship with your customers, from handling billing inquiries to product/warranty questions, to basic product/service questions. Interactions with customers can be recorded via PeopleSoft Conversations. Self service interactions can be recorded through PeopleSoft Contact Us. External contacts are your customer's representatives who can access self-service transactions

and receive documents such as sales order acknowledgements. Contacts must have a User ID to access self-service transactions.

Contact Us

A method by which customers and unregistered guest users send email messages to specific addresses or members of the merchant's organization. Merchants can also define automatic response messages.

Container

An Inventory stock unit for receiving, putaway, bin to bin transfers, picking, shipping, adjustments, and physical accounting. Each container is associated with a unique container ID.

Content Reference

Content references are pointers to some kind of content registered in the portal registry. These are typically either URLs or iScripts. Content references fall into three broad categories: target content, templates, and template pagelets.

Contextual reference

PeopleCode refers to a row or buffer field determined by the current context; that is, the context in which the PeopleCode program is currently executing.

Contingent Beneficiary

In PeopleSoft Pension Administration, any non-spouse pension beneficiary, including a child, other relative, or a trust. Spousal consent is required in order for an employee to name a contingent beneficiary.

Contracting Officer (CO)

Individual who has the authority and the official responsibility to produce a sound acquisition document.

Contracting Officer's Technical Representative (COTR)

Individual responsible for monitoring a contract and its associated tasks and deliverables.

Contractor

Any individual or non-employee reporting time that will not be paid through the payroll system.

Contribution

Represents money a stock purchase participant elects to contribute to the plan. Contributions are deducted from the participant's paycheck and used to purchase stock pursuant to the offering and purchase period they are enrolled.

Contributory Plan

A Pension plan to which employees contribute. Contributions are typically a percentage of pay deducted from the employees' paychecks.

Control Budget

Commitment control enables you to establish budgets that provide extensive, active budgetary controls over transactions, rather than just passively recording transactions.

Control ChartField

A control ChartField is a key ChartField that you designate to be the control field. Designating a ChartField as the control allows you to set attributes for a specific value of the ChartField that are different from the attributes specified for the budget type in general. For example, if the tolerance for a Projects budget type is set to 10% in general, you can override this value, making it higher or lower for specific projects.

Control Group

A mechanism to relate vouchers together for the purpose of controlling voucher input into PeopleSoft Payables. Generally used for assigning vouchers to data entry personnel and for reviewing input.

A set of parameters that determines the major forecast process options. The Control Group code is assigned to a group of **Forecast Item** and controls the forecast development and tracking for each item in the group.

Control groups are used by the Analytic Forecast Component to govern particular properties of the forecast rule, such as what accuracy to expect and what statistical method to apply. Forecast elements are assigned to exactly one control group. They manage differences among forecasts within a set.

Control Hierarchy

The relationship between business units, origins, vendors, and control groups in PeopleSoft Payables that defines which processing data will be automatically entered on each voucher.

Control Number

A sequential identifying number used to identify an exercise.

Control Plan

In PeopleSoft Quality, a plan that brings together application, measurement, and control and response criteria for a specific product and process.

Conversation

Any notes, transcript, or detail of a telephone call between an employee and a customer. Conversations may be tied to items, payments, purchase orders, document references, or bills of lading.

Conversion data profile

A conversion data profile takes the values from a particular PeopleSoft database table (such as the table holding bank transaction codes) and specifies how that value appears in PeopleSoft Business Documents.

Conversion data profile

A conversion data profile takes the values from a particular PeopleSoft database table (such as the table holding bank transaction codes) and specifies how that value appears in PeopleSoft Business Documents.

Conversion Loader

A sample SQR delivered with PeopleSoft Asset Management that transfers data from multiple fixed-length ASCII files into sample, relational conversion tables.

Copy Bill

In PeopleSoft Billing, the online environment providing for the replication of a single bill, generating a new bill with its own unique invoice number.

Core Functionality

Core functionality is the set of information in PeopleSoft HRMS that is common to your entire global workforce tracking needs—and is always displayed on the primary page.

Core hours

The hours a workday, workweek or pay period in which a time reporter must be present for work in a flexible work schedule (see Scheduling).

Corporate Account

In PeopleSoft applications, this is equivalent to the Account (ACCOUNT) ChartField. The term is used to make a distinction between the chart of accounts typically used to record and report financial information for management, stockholders, and the general public, as opposed to a chart of statutory (Alternate) accounts required by a regulatory authority for recording and reporting financial information.

Corporate Reporting

Companies with more than \$10 million in assets whose securities are held by more than 500 owners must file annual and other periodic reports. Publicly held companies are required to file documents with the SEC which include:

- Registration statements for newly-offered securities
- Annual and quarterly filings (Forms 10-K and 10-Q)
- Proxy materials sent to shareholders before an annual meeting

- Annual reports to shareholders
- Documents concerning tender offers (a tender offer is an offer to buy a large number of shares of a corporation, usually at a premium above the current market price)
- Filings related to mergers and acquisitions

Corporate Repurchase

When a corporation elects to repurchase some of its own securities. This reduces the Common Shares Issued and Outstanding. Typically, used to improve the valuation of the company's common securities outstanding as well as the Earnings Per Share (EPS).

Correction to IRR

An IRR type used when corrections need to be made to an original IRR that has already been submitted to the Office of Personnel Management (OPM). Federal employees covered by the CSRS retirement plan require SF-2806-1. Federal employees covered by the FERS retirement plan require SF-3101. A Correction IRR is also used if original retirement deductions were over-reported. See also Individual Retirement Record (IRR).

Correspondence Customer

A customer to whom all correspondence (statements) is addressed, often a corporate customer receiving correspondence for associated child customers.

Cost Accounting

A method where business costs are accumulated and distributed to products, processes, or discrete undertakings on an equitable basis. There are a variety of cost accounting methods, but they all share the same basic functions: classifying costs, recording costs, allocating costs to products or activities, summarizing and reporting costs to management. Cost accounting requirements and financial accounting requirements are not necessarily synonymous.

Cost Assignment

Resources assigned to cost objects or activities.

Cost Basis

Typically, this refers to the original price of an asset used in determining capital gains. However, in the case of death of an optionee, the appraised value of the asset at the time of death is the cost basis.

Cost Center

A Time and Labor Business Unit, in which all related costs attributable to some center within a business (such as an activity, an organization, or a program), are segregated for accounting or reimbursement purposes.

Cost Element

See **Inventory Cost Element** and **Manufacturing Cost Element**.

Cost Flow

Determines how depletions will occur for purposes of costing a transaction. Cost flows available include Specific Lot ID, Specific Serial ID, FIFO, and LIFO.

Cost Objects

Cost objects represent cost information about products, customers, and channels. They are the final results of the activities performed by your business, representing the focal point of costing and profitability analysis. Examples are products, customers and channels. They are the final results of the activities performed by your business. Your model's resources and activities are linked to the cost objects. They are often the focal point of profitability analysis.

Cost of Capital

An attribute used to describe the behavior of a particular cost object. A primary cost object is typically the main focus of the activity-based management analysis. This may be a product, customer or channel that you wish to calculate cost for. A support cost object may be used in a similar manner but may be further allocated to other support cost objects or primary cost objects.

Cost Of Living Allowance (COLA), Non-Foreign

A cost-of-living allowance payable to an employee at a location in a non-foreign area where living costs are substantially higher than those in the Washington, DC area.

Cost Profile

A combination of a receipt cost method, a cost flow, and a deplete cost method. A profile is associated with a cost book and determines how items in that book are valued, as well as how the material movement of the item is valued for the book.

Cost Profile Group

A grouping of items for the purpose of costing transactions and valuing inventory for a given book. Assigning an item to a cost profile group determines the books used by the item when accounting for that item.

Cost Roll-up

A process for calculating item costs. Cost roll-up provides a summation of all of the costs associated with the bill of material structure and the routing used in producing the item.

Cost Row

A cost transaction and amount for a set of ChartFields.

Cost Template

A collection of cost components that you can apply to a group of purchased items.

Cost Type

A user-defined method of categorizing item costs in Manufacturing for simulations and what-if analysis. Examples of cost types include current costs (which reflect the item's current bill of material or routing), proposed costs (which could be used in preparation for the next standard cost period), or activity-based costs (which include costs for items that consume a given activity).

Cost Version Type

A combination of cost types and cost versions used in cost rollups. Valid values include production (rolls up only manufacturing data and uses only the primary BOM and routing, each with a code of 1), engineering (can roll up with either manufacturing or engineering data, with any combination of BOM/routing codes), or simulation (only rolls up with manufacturing data, but can use any combination of BOM/routing codes).

Count Grade

A user-defined evaluation of a counting event.

Count Point

A predefined step on a routing or operation list where you can gather operation completion information. You define the appropriate points on the routing, record completions at these count points, and the system automatically backflushes the prior operations. This is only used on production IDs.

Counts

Count elements allow you to count the number of days or hours from a specific period of time. Counts are used primarily during proration calculations, but can potentially be utilized in other situations as well.

Court-Ordered Benefits Coverage

As prescribed in Title 5, United States Code and Title 5, Code of Federal Regulations, court orders that stipulate that an employee must continue or begin the coverage features for all employee benefits must be enforced. Federal employees are mandated by court orders to continue covering or begin covering their former spouses and/or children under their federal employee benefit programs (health, life, and thrift savings).

Court-Ordered Garnishments

As prescribed in Title 5, United States Code and Title 5, Code of Federal Regulations, court orders enforcing child support, alimony, or collection of commercial indebtedness are served on the appropriate entity within the Federal agency and implemented as offsets against the employee's salary.

Coverage

An employee's chosen benefit plan and coverage level; that is, what sort of benefit is provided as well as the value.

CPAM (Caisse Primaire d'Assurance Maladie)

In France, CPAMs are the local social security offices that manage health coverage for French workers. CPAMs are regulated and established by the French government. If you're managing a French workforce you'll need to identify and track the CPAM offices that impact your enterprise.

CRAM (Caisse Régionale d'Assurance Maladie)

In France, CRAM is the regional social security body which oversees the running of CPAMs. CRAM offices work with companies to both prevent and compensate workers for industrial injury.

Create Date

The date that you extracted a deduction or offset to PeopleSoft Deduction Management or created a split deduction.

Create Missing Items

In PeopleSoft Demand Planning and Inventory Planning, a feature that enables automatic system generation of master records that don't exist in the system.

Created Time

Time collecting device time or elapsed time generated by the system based on the time reporter's schedule (see Time Administration)

Creating Time

The preliminary generation of time segments as close as possible to their likely values when you officially report time—so that the information on the time records is as fresh and current as possible. The system shows you time that has already been created, rather than you having to create it “on the fly” when you come in to report. The process fills in reporting day gaps as defined by work schedules.

Credit Analyst

A required field used in PeopleSoft Receivables, Billing, Order Management, and Deduction Management when working with items. Each item must be assigned to a credit analyst. If no credit analyst is assigned to an item, the credit analyst assigned to the customer is used as the default.

Credit Risk Spreads

In the financial services industry, the additional charge to a risk-free interest rate, based on a riskier credit rating.

Credits

See Flexible Credits.

CREF

Acronym for Content Reference.

Crew Reporting

A Time and Labor process that enables you to report the earnings which consist of one or several time reporting codes and associated quantities of hours, amounts, or units, and task information for one date under report for a user-defined crew. The system transforms the information into instances of daily time for each crew member for the entered date.

Critical Success Factors (CSFs)

In PeopleSoft Balanced Scorecard, things that an organization must do well or excel at to achieve its goals. One or more key factors or objectives that must be accomplished for a particular strategic thrust. Key Performance Indicators are attached to CSFs.

CRM Warehouse

See Warehouses.

Cross Border Walker

This term is used in Europe for an employee who lives near a border in one country and works in another country. Such employees are subject to different tax and social security rules.

Cross-Plan Validation

The process by which the PeopleSoft Benefits Administration determines enrollment prerequisites for benefit plans. You can define four types of cross-plan validation prerequisites: prerequisites based on plan types, benefit plans, dependent enrollments, and coverage percentage limits for Life and AD/D plans.

Cross-View Reconciliation

In PeopleSoft Demand Planning, a process that enables the balancing of forecasts between selected levels of related views with the same **Forecast Item** key. The process is used when adjustments have been made to a working view and are then required in a related view.

Cube

See **Multidimensional Database (MDDB)**.

Cube View

In PeopleSoft Demand Planning, defines the user's own view of a forecast. The parent working view and dimensions determine what forecast data is included and how aggregates are formed.

Cumulative Tax Method

A payroll tax calculation method that adds together year-to-date earnings and earnings for the current pay period, then annualizes the result before calculating tax. This method is useful when Payrolls vary greatly in amounts from pay period to pay period, such as in the case of sales commissions.

Currency Calendar

In the financial services industry, business calendars for markets outside the organization's domestic operations that reflect the foreign markets' holiday schedules.

Currency Conversion Engine

A PeopleSoft Enterprise Warehouse Engine that processes financial information in multiple currencies.

Current Period

The earliest pay period for which the close date has not passed (see Time Reporting).

Current Period (Time and Labor)

In Time and Labor, the employee's current pay period which will be determined via the employee's Pay Group affiliation. Although there can be only one definition of Current Period per installation, the user can change it manually.

Current View

A reporting screen in Time and Labor whose effective date is within the date boundaries of an employee's current pay period, and for which pay has not yet been confirmed. A *Future Time Reporting Transaction* is one that has an effective date after the last day of the employee's current pay period. An *Historical Time Reporting Transaction* is one that has an effective date before the first day of the employee's current pay period.

Current Year

A period for event maintenance processing.

Curve Generator

A supporting module (common to financial services industry applications) that enables you to construct curves used to determine appropriate interest rates for given maturities and / or time periods. You can import market data from outside sources such as Bloomberg, upload the data from a spreadsheet, or manually enter the data. You can then build configured curves from segments or combinations of other curves.

CUSIP Number

A nine digit alphanumeric number associated with issuers' securities. CUSIP (Committee on Uniform Securities Identification Procedures). A uniform numbering system widely used to identify specific securities and their issuers.

Custom Statement

A user-created logical or mathematical expression that determines information about an employee in PeopleSoft Pension Administration. Custom Statements commonly define employee groups and benefit formulas.

Customer Inquiry

A window containing options to review customer balances, aging, history, items, actions, and conversations.

Customer Scorecard

See PeopleSoft Customer Scorecard.

Customer Tree

A user-defined graphical representation of your current sales organization. A customer tree is used to establish and distribute funds and to determine authority levels for promotional activities.

Cut Session

Cut sessions are a means of dividing a course session. You use cut sessions where a course session does not run on consecutive days from start to finish, or if there are multiple instructors or locations. Each cut session has its own start/end date, location, and instructor. For example, if you have a course that runs for two days a week for a month, you would divide the course session into four cut sessions, each of which is two days long.

Cycle Count

A manual counting event that does not cover an entire inventory business unit. Usually includes every item (and lot, if applicable) in a location or family.

Cycle Interval

The number of days between cycle counts.

Cycle Procedures

Inventory planning tasks that need to be performed on a regular basis to ensure an up-to-date **Inventory Policy**. The tasks can be performed either at the end of a processing period or within the period, and should always be performed if the forecast or **Control Group** or **Policy Item** parameters change. Tasks include generating a policy and reviewing **Work Queue** messages.

D

DAT file

A text file (input.dat) used with the Verity search engine that contains all of the information from documents that will be searchable but not returned in the results list.

Data Elements

Data elements, at their most simple level, define a subset of data and the rules by which to group it.

For PeopleSoft Balanced Scorecard, data elements are used as the basis for key performance indicators, and as target values for Key Performance Indicator (KPI) objects.

For Workforce Analytics, data elements are rules that tell the system what measures to retrieve about your workforce groups.

Data Entry Access List

Used to present a concise list of often-performed data entry tasks to a user. You can assign multiple control plans to a single data entry access list.

Data Extract

A report that creates a file used to transmit data to a third party on magnetic media. There is no meaningful printed output for this type of report.

Data Loader

Data Loader is a PeopleSoft Enterprise Warehouse utility that moves data from the Operational Data Store staging area to either the ODS reporting area or the Data Warehouse. The Data Loader utility is made up of several pages that allow you to enter Metadata to define your source and target records and your transformation rules and then perform the load by running an Application Engine.

Data Loader Map

Defines how to extract data from the Operational Data Store (ODS), transform it, and load to a Target Table. The target table can reside in the warehouse or the ODS layer.

Data Manager

A PeopleSoft Enterprise Warehouse engine that distributes revenue, expense, analytical application engine results, statistical quantities and other measures across business units, departments, products, customers and channels—any field or logical group in the chart of accounts. You can define a number of types and options within this engine. It is also used as a means of posting to the Performance Ledger.

Data Manager Rules

In the PeopleSoft Enterprise Warehouse, Data Manager rules use Constraints to specify the source as well as the target tables for moving, aggregating, or multidimensionalizing your engine output. Rules use Data Manager methods to enrich the PeopleSoft Enterprise Warehouse data.

See Data Manager Methods.

Data Manager Methods

There are several methods: Copy, GL Mapper, Prorata, Spread Even, and Tree Aggregation. Each method enables you to move and/or enrich engine output.

Data Mart

A Data Mart is a data structure that uses a central fact table and related dimension tables to generate a “relational cube” or directly generate an Insight report.

Data Mart Builder

The Data Mart Builder is a multiple Application Engine (AE) process, that is, a framework of procedural programs, that creates a Data Mart.

DataMaps

Information that builds upon the data captured in the TableMap records. DataMaps enable you to define a logical view of the physical PeopleSoft Enterprise Warehouse tables. DataMaps bring together information from many different tables and fields and define it all as one entity or table.

Data Row

Contains the entries for each field in a table. To identify each data row uniquely, the system uses a key consisting of one or more fields in the table.

DataSet

DataSets are used as input for various engines and processes, for instance, the Analytic Forecasting component, the Data Manager, user defined functions, drivers in Activity-Based Management, and data elements in the Key Performance Indicator Manager. DataSets provide a user defined set of information to the engines. DataSets use Constraints to restrict used columns and restrict returned rows. Each DataSet is created by a process specific setup. However, the underlying logic is the same, enabling you to more easily understand the functional aspects of the process.

Data Warehouse

A large database containing data summarized from one or more transactional systems, optimized to support the analysis needs of the enterprise. An ideal data warehouse contains all the data necessary to make business decisions. Users analyze the data in the warehouse using Online Analytical Processing (OLAP) tools and ad hoc query/reporting tools. An increasing

number of organizations have "virtual" data warehouses, where the data warehouse is not one physical database, but rather a collection of specialized (and distributed) data marts.

See also PeopleSoft Enterprise Warehouse.

Data Warehouse Tables

Data Warehouse tables act as the portal for getting data into the PeopleSoft Enterprise Warehouse from PeopleSoft, OLTP applications or other "outside" sources. These tables are used:

- As targets for loading operational data.
- For error detection and handling
- For data validation.
- For aggregation.

Database Alias

The PeopleSoft Pension Administration utility that looks up employee data.

Dataset

A file containing data to be analyzed by the Quality Server program. The dataset is similar in content to a spreadsheet.

In PeopleSoft Planning, a file that stores schedule information such as tasks, resources, calendars, and so on.

Date

See **Accounting Date** **Transaction Date** or **Effective Date**.

Date

If you want to either include a date in a calculation, or determine a new date by taking a starting date and either adding or subtracting a period of time to come up with another date, you use a date element.

Date Classified

Date the Position Description is approved by Management/Position Management.

Date Eligible To Retire

Date an employee is eligible to optionally retire based on the combination of age and service that meets legal requirements.

Date Under Report

The date (day) in PeopleSoft Time and Labor for which time is being reported. The Date Under Report does not have to equal today's date.

Day Breaker

Customer defined time that is used to determine when one day becomes the next. It's used to determine the "logical" date of a punch. (See Understanding Workgroups.)

Days Supply

In PeopleSoft Inventory Planning, a method that can be used with several types of **Inventory Policy**. Using this method, a specific number of days of supply for an item should be used to calculate the item's inventory policy.

Deal Type

PeopleSoft Treasury has categorized deals into several basic deal types from which you can choose when defining an instrument.

Death Coverage

The PeopleSoft Pension Administration function that determines the factor used to reduce an employee's benefit when the plan charges for PRSA coverage.

Decompressed Split

In PeopleSoft Demand Planning, a function for returning a compressed split database to its original form. See also Compressed Split.

Deduction

Any amount taken from an employee's pay check each pay period. Deductions may include health or medical benefits, union dues, and so on. See also Benefit Deduction and General Deduction.

Deduction Date

The as of date for the deduction item in PeopleSoft Receivables.

Deduction Item

An individual item that you created in receivables and is an open receivable on the customers account due to a deduction that they took in a payment for a receivable item.

Deduction Reason

Code that describes the type of deduction. When assigned to a write-off resolution, it determines what accounting entries to create.

Deduction Specialist

The individual responsible for tracking and resolving deductions in PeopleSoft Deduction Management.

Deduction Subset

A group of deductions selected from a company's standard set of deductions. Deduction subsets minimize data entry time in special processing situations such as bonus check runs.

Default Mode (DM) model

In the financial services industry, an approach used by financial institutions to predict a decline in portfolio value. Only two outcomes are considered – default or non-default. If the debt does not default, there is no change in the value. If the debt does default, then the loss is calculated as the difference between what was contractually owed and the value of any collateral recovered.

Defection Analysis

In PeopleSoft Workforce Analytics, the identification of employees who are likely to leave the organization based on predefined assessment criteria.

Deferred Compensation

Compensation payments that are payable to an individual in the future such as pension plan payments, annuities, stock awards and profit sharing. Note: Profit sharing can be considered direct pay if paid out in cash on a periodic basis or deferred pay if cumulative with the intention of payment in the long-term future.

Deferred Vesting

The adjustment made to the original option's vesting schedule that pushes the vesting into the future.

Defined Benefit Plan (DB Plan)

A retirement income plan (usually called a pension plan) where the employee's benefit is definitely determinable based on a plan-specified benefit formula.

Definition or Function Definition

The parameters for any of PeopleSoft Pension Administration's nineteen core functions. A definition has to be explicitly associated with an employee Group Definition before it can be applied.

Dekit

The ability to return material issued in kits to inventory. This is used when entire kits need to be returned; individual components are handled through kit issues/returns.

Delete Non-Matching Items

In PeopleSoft Inventory Planning, an option used in the Generation process to delete Inventory Planning items that don't have corresponding items in Demand Planning. The item deletion occurs when the system generates the policy.

Delta

When retroactive processing occurs for a given payee, the system recalculates each element generated for the payee. The system compares the recalculated results to the original results. The difference between these results is typically referred to as the retro "delta." A retro delta can represent either an underpayment or an overpayment that results in an adjustment to the payee's earnings.

Demand

Collection of training requests. This could be an employee demand, a departmental one or a company-wide demand.

Demand Filter Width

In PeopleSoft Demand Planning, specifies the confidence interval within which demand is considered to be reasonable. Actual demand that is outside the confidence interval is automatically filtered and replaced by the value at the edge of the interval. The value is expressed as a percentage.

Demand Filtering

In PeopleSoft Demand Planning, provides a way to detect and highlight unusual demands and forecast errors. If the demand falls outside of a band that is considered reasonable, the system automatically adjusts it to the level of the boundary and logs a message to the **Work Queue**.

Demand Number

The configured product sub-component sequence number.

Demand Planning

In PeopleSoft Activity-Based Management, this type of planning focuses on studying the impact of cost objects and activity volumes.

Demand Priority

The placing of importance on independent demand. The Planning engine uses the demand priority value to determine the order in which you fulfill the demand. You can set a demand priority from 1 to 998 with 1 being the most important level. The priority value of 999 is reserved for the system.

Demand Priority Rules

In PeopleSoft Inventory, a set of rules that will sort demand so the most important demand will have the first opportunity to reserve available inventory. If demand priority rules have

been defined, the Material Reservations process (INPLDMND) sequences orders by priority rank, processing those with the lowest rank value first.

Deplete Cost Method

Determines how you cost a depletions transaction within a book. The deplete cost methods available include Actual, Non-Cost, Perpetual Weighted Average, Periodic Weighted Average, and Value at Current Standard.

Depreciate When in Service

A switch that indicates whether PeopleSoft Asset Management should allocate depreciation as of the date an asset was placed in service. This is valid only in the year the asset was acquired.

Depreciation - Declining Balance

Budgeting calculates this as: Cost minus Accumulated Depreciation divided by Life divided by number of periods per year. It results in a higher depreciation expense in the early years of an asset, which decreases as you near the end of its useful life.

Depreciation - Double Declining Balance

Budgeting calculates this as: Cost minus Accumulated Depreciation multiplied by 2 divided by Life divided by number of periods per year. It results in a higher depreciation expense in the early years of an asset, which decreases as you near the end of its useful life.

Depreciation Methods

The various methods of spreading the acquisition cost across the life of an asset rather than expense the full value of an asset at the time you acquire it. The value of the asset consequently decreases (or depreciates) through time. The four depreciation formulas delivered with PeopleSoft Budgeting include: declining balance, double declining balance, straight line, and sum of the years.

Depreciation - Straight Line

A method of depreciating asset value in equal amounts across the life of the asset. Per-Period Straight-Line depreciation is calculated as the cost of an item minus the salvage value divided by the number of periods to depreciate.

Depreciation - Sum of the Years

A depreciation method equal to the value of the remaining years of life divided by the sum of the years remaining is multiplied by the Net Book Value. This figure is then multiplied by the percent of years to depreciate. This results in a higher depreciation expense in the early years of an asset, which decreases as you near the end of its useful life.

Depromote

In PeopleSoft Demand Planning, the process of making an adjustment to actual demand data that removes the effect of a promotion during a defined period. As a promoted period moves into history, the system creates an adjusted demand entry that is equal to the **Prorated Forecast**.

DeptID

A ChartField that defines departments or administrative offices that have operational, fiscal and/or budgetary responsibility for specific sets of activities.

Derived Metric

The result of a calculation on a report of base metrics.

Detail

A temporary assignment to a different position for a specified period when the employee is expected to return to his/her regular duties at the end of the assignment. This employee is considered for pay and strength count purposes to be permanently occupying his/her regular position. Unless the agency chooses to use an SF50, a detail is documented with an SF52.

Detail Tree

A tree that employs ranges of detail values under each node; you must manually specify the detail values.

DFI ID (Depository Financial Institution ID)

A PeopleSoft Payables bank identifier, consisting of Transit Number, Swift ID, or CHIPS ID.

Dimension

A single element of a business model, such as product, department, or location. Cube Manager uses the term Conforming Dimension.

In terms of data analysis, dimensions can be thought of as criteria, such as time, product, and location, used to pinpoint a particular piece of data. For example, in the retail industry a set of dimensions could be geography, product, time, customer, and vendor. The geography dimension would include company, chain, region, district, and finally store attributes. A dimension is also a column heading on an analysis and reporting template which you can drill through or roll up to the multiple levels.

In PeopleSoft Budgeting, a view option that assists in summarizing the rows of data in line-item budgeting.

A single element of a budgeting model, such as account, product, project, department, or operating unit. In PeopleSoft Budgeting, these dimensions typically represent the ChartFields used by your organization during your budgeting process.

Dimension Table

In the PeopleSoft Enterprise Warehouse, Dimension Tables store additional attributes or data about Facts. Some example dimensions include Customer, Channel, Geography and Product.

Direct Compensation

In PeopleSoft Workforce Analytics, Direct Compensation is payment made to workers in exchange for their contributions to the organization. Direct Compensation is typically categorized as including Cash Compensation and Long-term Variable Compensation.

Cash payments made to workers in exchange for their contributions to the organization. Direct pay is typically categorized as fixed pay (for example, base pay, shift differentials) and variable pay (for example, profit sharing, incentive, bonus). Note: Profit sharing can be considered direct pay if paid out in cash on a periodic basis or deferred pay if cumulative with the intention of payment in the long-term future.

Direct Calculation

Calculate actual and directly assigned dollars.

Direct Cost

In PeopleSoft Workforce Analytics, a direct cost of an activity or a cost object. An example is the salary cost of employees working on a project.

Director

An affiliate of the company who holds a seat on the Board of Directors for the corporation. A Director, generally, is not an employee of the corporation.

Disability and Discrimination Act of 1995

In the United Kingdom this act makes it unlawful to discriminate against individuals on the basis of their disability in relation to recruitment, promotion, training, benefits, terms and conditions of employment, and dismissal.

Disability Rate Code

The desired percentage of disabled persons that should be employed by French employers, as mandated by the French government.

Disbursement View

In PeopleSoft Demand Planning, a **Forecast View** that allows the forecast from a working view to be reported on using an alternate key. Disbursement views are built directly from the working views and inherit many working view attributes, including time period and associated user data definitions, from the parent working view.

Discounted Stock Option

Rights to a stock option at a price less than 100 percent of fair market value at the time of grant.

Discretionary Plan

In PeopleSoft Workforce Analytics, this is a plan for distributing compensation awards that provide managers the ultimate discretion over a pool of money which is either funded based on company, group, or employee performance, or it's budgeted. The discretionary award determination is sometimes guided by a pre-determined percent of the participant's salary, expressed as an opportunity. This figure can then be modified based upon management's perception of actual value created by the group or employee.

Disqualifying Disposition (DD)

When an optionee sells or otherwise disposes of the shares of stock acquired through the exercise of an incentive stock option or through an employee stock purchase plan before the holding period for preferential tax treatment has lapsed.

In the case of Incentive Stock Options, the holding period is one year of the date of exercise and two years of the date of grant. At the time of disposition, the individual recognizes compensation income equal to the difference, if any, between the option price and the fair market value of the corporation's stock on the date of exercise. If the sale price is less than the fair market value of the stock on the date of exercise, the compensation income is limited to the total sales price less the total option price, less any fees.

In the case of purchases through an employee stock purchase plan, the holding period is one year from the purchase date and two years from the enrollment date. Compensation income in a disqualifying disposition is equal to the difference between the total fair market value on the purchase date and the total purchase price.

Distribution

Provide a repository of time and associated estimated and actual allocated labor costs to other systems

The process of assigning values to ChartFields. A distribution is a string of ChartField values assigned to items, payments, and budget amounts.

Distribution Network

A distribution network is a prioritized list of Inventory business units (IBUs). When a customer orders a product, the system uses this network to determine which warehouse the stock ships from.

Distribution Profile

A definition of ChartField distributions assigned for compensation costs. A distribution profile can be used to set up defaults for how the system should distribute costs associated with a position's salary, benefits, and earnings. PeopleSoft Budgeting-specific.

Distribution Rule

You use distribution rules to determine the order in which the system searches for matches against the distribution sets matrix when sales orders are entered.

Distribution Set

Distribution Sets assign account distribution information to combinations of defining elements used on sales orders.

Distribution Type

An identifier that defines one of the different transactions that move an item into or out of an inventory business unit. Distribution types are used to create debit and credit transactions to the general ledger via the Journal Generator.

Dividend

Distribution of earnings back to shareholders, prorated by the class of security and paid typically in the form of money or stock. The amount of a dividend is decided by the Board of Directors and is usually paid quarterly.

Document Management

The process through which a user has complete control of document version including the ability to view, query, and edit documents in a secure vault. Document management enables you to seamlessly perform online document queries and view documents directly, launching them from within PeopleSoft applications. You can associate pertinent documents with engineering change requests (ECR), engineering change orders (ECO), item revisions, bills of material, manufacturing and engineering routings, production component lists, and production operation lists.

Document Sequence Number

A value that the PeopleSoft system assigns to a document (such as an invoice, voucher, or journal) when you create a document for a business unit that you have enabled for document sequencing. The system determines the number by the values of the business unit, accounting date, and document type.

Document Sequencing

A flexible method that sequentially numbers the financial transactions (for example, bills, purchase orders, invoices, and payments) in your system for the purpose of statutory reporting and tracking of commercial transaction activity. Document sequencing requires that you classify all financial transactions into three transaction types—journal type, journal code, and document type—and that within each transaction type, all documents you enter are numbered sequentially. When you create a document (such as an invoice, voucher, or journal), the PeopleSoft system assigns a document sequence number to that document.

Document Type

The final level of three categories for defining a financial transaction (or document), necessary when using document sequencing. It represents the business purpose of a financial transaction, such as domestic customer invoice or customer credit memo. Document type is within one and only one journal code; journal code is within one and only one journal type. Document type is the only required category, because the values of the other two categories can be derived from document type.

Dollar Tolerance

In PeopleSoft Inventory, the acceptable cost difference between expected cycle count quantities and actual quantities counted. This value allows a margin of error for an item during cycle count reconciliation based on item cost.

Domestic Relations Order (DRO)

A preliminary version of a court order (usually stemming from a divorce settlement) ordering a division of a participant's pension benefits. The order is not in effect until it is determined to be "qualified" by virtue of meeting certain requirement. At that point it becomes a Qualified Domestic Relations Order, or QDRO.

Double Byte Characters

If you're working with Japanese or other Asian employees, you can enter the employee's name using double-byte characters. The standard double byte character set name format in PeopleSoft applications is: [last name] space [first name].

Draft Worksheet

A work space used in PeopleSoft Receivables to track a draft through its processing life cycle.

Drill-Back Calculation

Assigns indirect dollars and Drill-Back calculations. Also, this picks-up all costs in the Calculations Detail (CALC_DETAIL_F00) that was assigned during direct calculations.

Drill Down

The ability to go down to the next level of detail in a set of data. For instance, if you're looking at an expense figure for a division, you can drill down to the expenses for each department in the division.

Drill Down

The ability to go down to the next level of detail in a set of data. For instance, if you're looking at an expense figure for a division, you can drill down to the expenses for each department in the division.

Driver Lookup Table

Tables associated with a driver that enable different rates and amounts unique to a budget center.

Drivers

In PeopleSoft Activity Based Management, drivers are a means of assigning dollar amounts from resources, activities, and cost objects to other resources, activities, and cost objects throughout the model in PeopleSoft Activity-Based Management. Drivers can also be assigned across business units. There are different types of driver categories (transactional, duration, and intensity), and different ways of specifying how those dollar amounts are calculated (amount, percentage, spread even, and direct), as well as different ways that dollar amounts are assigned (depending on assignment type and object type).

In PeopleSoft Business Planning, a driver can be defined as a set of values that are used as an input to another process. In this context, a driver should be interpreted as a projection of external factors and other indicators. The user can then define a relationship between the driver and a financial result. The driver values and the defined relationship then combine to produce a projection of the financial result. For example, a projection of the number of square feet used (driver) and the price per square foot paid in rent (driver) can combine to produce a projection of rent expense (financial result).

DRO

See Domestic Relations Order.

DSS (Decision Support System)

A DSS is a workstation-based analysis and reporting system, typically aimed at analysts and line managers. OLAP tools provide a powerful DSS.

Duration

In PeopleSoft Pension Administration, the utility that calculates the length of time between two dates.

Duration [Global Payroll]

An element type that calculates a period of time between two dates. For example, if you want to determine a payee's age, you can calculate the duration between his birth date and the calendar period end date.

Dynamic Group

A group in Time and Labor that enables you to establish criteria or attributes for a group of employees. All employees who fit this criteria at processing time belong to the group.

Dynamic Tree

A tree that takes its detail values—*Dynamic Details*—directly from a table in the database, rather than from a range of values entered by the user.

Dynamic Views

In PeopleSoft Demand Planning, a **Forecast View** that allows interaction with the forecast data using an alternate key structure. By using dynamic views, you streamline the working view and can complete the working-view design without having to anticipate all conceivable adjustments.

E***Earliest Change Date***

Determines both the range of dates and the amount of data that will be processed for each time reporter (see Batch Processing)

Early Punch

A punch that is more than the predefined number of hours/minutes before a scheduled punch where a time reporter is warned

Early Retirement Date (ERD)

A retirement date earlier than a plan-specified “normal” retirement date. Employees usually must meet age and/or service requirements to be eligible for early retirement, and early retirement benefits are often reduced to compensate for the longer duration of payments.

Early Retirement Factor

The reduction made to an employee's benefit if the employee elects for early retirement.

Early Warning

In commitment control, warning of possible future budget exceptions. You can specify that you are to receive a warning when commitments and expenditures reach a predetermined percentage of budget. For example, you can instruct the system to let you know when commitments and expenditures reach 50%, 80%, or some other percent of a budget.

Early/Late Adjustments

The PeopleSoft Pension Administration function that calculates early retirement factors or late retirement factors.

Earning Group

Part of a group of defaults assigned to job codes. Earnings group may include non-salaried items such as holidays and bonus pay dependent on individual company parameters.

Earnings

The amount owed to an employee based on salary, hours worked, or other calculation routines, plus other types of compensation and holiday, vacation, and bonus pay.

Earnings [Global Payroll]

An element type that defines the different types of compensation that are added to a person's pay. Examples include salary, commission, bonuses, and retirement pay.

Earnings Accrual Class

Categorizes a set of accruable earnings.

Earnings Code

Codes that represent the various types of earnings such as regular, overtime or leave.

Earnings Per Share (EPS)

The portion of a company's profit allocated to each outstanding share of common stock. Net income (reported or estimated) for a period of time is divided by the total number of shares outstanding during that period.

Earnings Type

An abbreviated and encrypted set of business instructions containing compensation instructions. Earnings Type may also contain Benefit Entitlement and Administration instructions, taxation instructions, Financial Accounting instructions, Organizational Administration instructions, work group and labor affiliation instructions, and other instructions.

Economic Loss

In Funds Transfer Pricing, this refers to the break fund economic loss, calculated by applying the theoretical value of the interest rate differential (IRD) against a cash flow stream, based on the amount of the prepayment or cancelled draw-down.

Economic Value Added

In the financial services industry, Economic Value Added is a financial metric that factors into the measurement of an activity's profitability the cost of economic capital assigned to that activity.

EDGAR (Electronic Data Gathering, Analysis, and Retrieval)

An electronic system implemented by the SEC that enables companies to file documents in conjunction with disclosure requirements mandated by the SEC.

EDI Agent

Used in EDI processing, the inbound EDI Agent loads trading partner data (flat files) into the PeopleSoft database using transaction, map, and trading partner definitions set up using EDI Manager. The outbound EDI Agent extracts information from the PeopleSoft database and generates data files that can then be processed for transmission to a trading partner.

EDI Manager

A suite of online pages used to define transaction sets, trading partner profiles, and translation maps for EDI transactions.

Edit Table

A table on the database that has its own record definition, such as the Department table. As fields are entered into a PeopleSoft application, they can be validated against an edit table to ensure data integrity throughout the system.

EEO Company Code

In the United States companies are assigned this federal code for EEO and VETS100 reporting.

Effective Date

A method of dating information in your system. You can predate information to add historical data to your system, or postdate information in order to enter it before it actually goes into effect.

Effective Date

A method of dating information in your system. You can predate information to add historical data to your system, or postdate information in order to enter it before it actually goes into effect. The Effective Date usually defaults to your system's current date.

Effective Periods

In PeopleSoft Demand Planning, the number of periods of historical demand used in the Model Reset process. The value can be used to exclude older, possibly unrepresentative historical demand data from model **Optimization**.

Effective Sequence

A system-generated number assigned to distinguish between two job entries with the same effective date.

Effective Tax Rate

The ratio of income tax paid over gross income, showing the percentage of income actually paid in taxes.

Effectivity Date

The date on which a component can be added or substituted in production, typically specified on an item's bill of material.

EIS (Executive Information System)

An EIS is a workstation-based analysis and reporting system for executives. An EIS provides a higher-level view of the data than a DSS, and typically requires less knowledge about the underlying transactional systems. OLAP tools provide a powerful EIS.

Elapsed Schedule

A method of scheduling a time reporter's time that is based on TRC and duration. This method can be used for scheduling of elapsed time reporters (see Scheduling.)

Elapsed Time

Reporting non-clock time in increments of hours or partial hours (see Managing Time / Understanding Time Reporting).

Elapsed Time Service

A method of calculating a period of service that uses only the start and end dates of the period to determine the amount of service. Hours worked or other measures of the actual work performed during the period are not taken into account.

Electronic Certification System (ECS)

An automated Payment Voucher authorized by the Certifying Officer for use within the Treasury Department, Financial Management Service's financial system. PeopleSoft provides a method to record and generate data files for on- and off-cycle processed payments.

Element

In PeopleSoft Global Payroll, an element refers to both primary elements and supporting elements. Primary elements are comprised of earnings, deductions, absence entitlements, and absence take elements. Supporting elements are element components that are combined to create primary elements.

In PeopleSoft Enterprise Performance Management, elements are used to create a Profile. An Element can be one or more columns of data in an Enterprise Warehouse table, associated with a single dimension (for example, Customer, Product, Department, or Channel). An Element can also be KPI, Population, subscription data from a third party, preexisting Profiles, and data mining scores.

Element Group

Element Group identifies a group of elements to provide eligibility. You can then use this as a notational shortcut—instead of having to list each element, you can use the element group name. Element Group's expedite the process of manipulating earnings and deductions.

Element Name

Name assigned by the user for data fields, rules, formulas, and tables. For example, the names you give to new rules, elements, or objects.

Element Segment

When an element changes mid-period, requiring the affected element (and perhaps a subset of other elements) to be calculated multiple times on either side of the date on which the change takes place, element segmentation is used. Unlike period segmentation, the system segments only the elements you select, and creates separate result columns only for the specified elements. In element segmentation, there is only one gross-to-net result set.

Eligibility Group

Eligibility groups define the possible earnings, deduction, absence entitlement, and absence take elements that a payee might be eligible to receive. This enables you to group payees so as to assign eligibility for certain pay elements.

Eligibility Rule

PeopleSoft Benefits Administration uses eligibility rules during Benefits Administration processing to determine which benefit programs and options an employee is eligible for. Eligibility rules are closely associated with event rules: they determine what options an employee can *have*, while event rules determine which of those options an employee will actually be able to *choose*.

Elimination Set

See **Consolidations-Elimination Set**.

Email Template

Pre-defined parameters that establish automatic email generation during budget submittal, rejection, publishing, and/or targeting.

Employee

An individual employed by an organization and administered as an employee in the PeopleSoft Human Resources system.

Employee Accounts

The PeopleSoft Pension Administration function that tracks employee contributions to a pension plan.

Employee ID

A unique identification code for an individual associated with your organization.

Employee Paid Benefit

The portion of a pension benefit funded by the employee's own contributions to the pension plan. Also, the PeopleSoft Pension Administration function that determines this amount.

Employee Profile

This PeopleSoft Activity-Based Management feature enables time and labor information to be part of an analysis.

Employee Stock Purchase Plan

A type of statutory stock option plan through which employers grant options to their employees in order to provide them with additional forms of compensation.

Employee Survey

In PeopleSoft Workforce Analytics, a method for capturing information about the activities performed by a given employee as well as the amount of time they spend performing each activity to perform activity-based management.

Employee Survey Report

The Employee Survey Report is an annual regulatory report that the French government requires from employers with more than 200 employees. In French it is called "Le Bilan Social". The report is communicated to both labor unions and the government. It provides a snapshot view of the company over the past 3 years for about 200 indicators.

Employee Training Cost

Amount budgeted to pay for students' salaries while on training courses.

Employer Identification Number (EIN)

In the United States a company is typically defined as a business enterprise that has a unique federal Employer Identification Number (EIN) for payroll tax reporting purposes.

Employer's Liability Insurance Associations (Berufsgenossenschaften)

Social Insurance in Germany is maintained and administered by private organizations that act as employer's liability insurance associations. Employers pay out premiums to these associations, who administer and pay out funds to workers who are injured on the job.

Employment Cost Index (ECI) Adjustment

Annual increase to wages established/permitted by statute.

Employment Equity Computerized Reporting System (EECRS)

Canadian companies are required to report to the Federal Government on employment equity. PeopleSoft Human Resources contains the Canadian Employment Equity report (PER101CN), which creates a data interface file to the federal government's Employment Equity Computerized Reporting System (EECRS).

Employment Record Number (EMPL RCD#)

A field in PeopleSoft Human Resources Management Systems and PeopleSoft Workforce Analytics that indicates an employee has multiple job records in the system. A numeric value (0, 1, 2) is assigned to each job as a way to uniquely identify that job record.

Encumbrance

A claim against funds. It is a projection of future expenses based on the situation, as we know it today. Encumbering funds is not the same as spending them or even guaranteeing that you will spend them. It just means that if the situation as it exists today does not change, you will spend all of those funds by the end of the fiscal year.

Engineering Bill of Material (EBOM)

A listing of all the parts, raw materials, and subassemblies that form the basis of all item and product structures. EBOMs differ from MBOMs (Manufacturing Bills of Material) in that they are not visible within Production Planning or Production Management and are isolated from Manufacturing.

Engineering Change Order (ECO)

A revision to a blueprint or design, released by engineering to modify or correct a part and/or bill of material. PeopleSoft Engineering uses ECOs to manage and document required assembly and component changes.

Engineering Change Request (ECR)

A document that allows you to request manufacturing process improvements and report product defects directly to the engineering department. When workflow is enabled, ECRs can also be routed for review and approval, after which they change into ECOs.

Engineering Cost Version

The process of generating cost versions for new and modified configurations based on engineering bills of material (EBOM) and costing data.

Engineering Workbench

An engineering environment, separate from production, consisting of engineering bills of material (EBOM), engineering change requests, engineering change orders, EBOM cost roll-up capability, online BOM comparisons, and seamless integration to a document management vault.

Engineering Workbench

An engineering environment, separate from production, consisting of engineering bills of material (EBOM), engineering routings, engineering change requests, engineering change orders, EBOM cost roll-up capability, online BOM comparisons, and seamless integration to a document management vault.

Enterprise

In PeopleSoft Time and Labor, all of the business units of the installation site.

Enterprise Performance Management (EPM)

See PeopleSoft Enterprise Performance Management

Enterprise Portal

The PeopleSoft Enterprise Portal is a separate product offering purchased independently of any other PeopleSoft applications. It can be used with or without any PeopleSoft application. It can be used as a standalone corporate portal that does not access PeopleSoft data at all.

Enterprise Resource Planning (ERP)

The encompassing term for all the transaction-oriented database applications an organization deploys across its business enterprise. The term includes financial, manufacturing and supply chain, human resources, and payroll applications, among others.

Enterprise Warehouse (EW)

See PeopleSoft Enterprise Warehouse

Entry Authority

Authorization granted by employees to specific user IDs for entering expense data on their behalf.

Entry Currency

The currency used to enter budget data.

Entry Event

An automated process that generates multiple debits and credits resulting from single transactions, to produce standard supplemental accounting entries.

Entry Event Code

Designation of an Entry Event; an identifier or label.

Entry Event Generator

A mechanism that generates standard, supplemental accounting entries based on Entry Event codes.

Entry Event Process

An accounting transaction. Entry Event processes combine to form Entry Events. For example, requisition posting is a Purchasing process, and cash clearing is a Payables process. Each process can involve one or several Entry Event Steps.

Entry Event Step

Part of an accounting transaction. For example, the BUDG process includes these steps, among others: prepare allotment budgets, prepare organization budgets, and prepare revenue estimates. Entry Event steps combine to form Entry Event processes.

Entry On Duty Date (EOD)

Date that indicates when an employee started to work at his/her current agency.

Entry Type

Any activity that creates or updates an item.

EPM (Enterprise Performance Management)

See PeopleSoft Enterprise Performance Management

Equal Employment Opportunity Commission (EEOC)

In the United States the EEOC requires that most companies file one or more reports from a series named EEO-1 through EEO-9. These reports include counts by federal employment categories of male and female employees in certain ethnic groups.

Equitization

A process that enables parent companies to calculate the net income of subsidiaries on a monthly basis and adjust that amount to increase the investment amount and equity income amount before performing consolidations. For organizations with complicated parent/subsidiary business unit relationships, this automated process reduces the process time and reduces the possibility for errors.

Equity Increase

In PeopleSoft Workforce Analytics, Equity Increases are base pay increases granted to bring an employee's pay up to some internally specified standard for your organization.

Equivalent Standard Deviation

In PeopleSoft Demand Planning, a **Standard Deviation** developed during the Model Reset process that enables you to compare standard deviations from different models. The deviation is calculated by multiplying the **Model Equivalency Factors** defined on the **Control Group** by the standard deviation.

ERISA (Employee Retirement Income Security Act of 1974)

The U.S. Federal legislation enacted to prevent abuses of employee pension rights by employers.

Error Exception

A transaction that is stopped because the budget limits would be exceeded if it continued. For the transaction to proceed, action must be taken, such as canceling or reducing the transaction amount, increasing the budget amount, overriding the budget limits, or transferring available funds from another budget.

Error Ratio

The ratio of the statistical Standard Deviation to the base component that gives an indication of the accuracy of the forecast. The ratio is presented in the PeopleSoft Demand Planning Audit and Accuracy Review and is calculated during the **Forecast Calculation Process**.

Estimated Gross

Estimated labor cost associated with reported time (see Managing Time, Understanding time Reporting Codes).

Estimated Shipments

A group of shipment schedules used to manage sales order requested shipment data and actual shipment data. Used in conjunction with weight and volume pricing and freight charge calculations.

Ethnic Code

The Federal Office of Management and Budget (OMB) racial and ethnic census categories used for classifying individuals in U.S. Government reports.

ETL (Extract-Transform-Load)

See Extract-Transform-Load.

ETL maps

ETL maps provide rules for importing your source data to the data warehouse tables.

Evaluated Receipts Settlement (ERS)

A PeopleSoft Payables feature that matches receipts against purchase orders and generates vouchers without requiring an invoice.

Evaluations Periods

In PeopleSoft Demand Planning, indicates the number of future periods to use for the calculation of forecast errors. For example, if the number of evaluation periods is two, then the forecast error in April 2001 (after posting demand for April) would be determined by comparing the actual demand for April and the April forecast generated in February 2001. Evaluation periods are set separately for each view.

Event

Events are predefined points either in the Application Processor flow or in the program flow. As each point is encountered, the event fires on each component, triggering any PeopleCode program associated with that component and that event. Examples of events are FieldChange, SavePreChange, OnRouteSubscription, and so on.

Event

Occurrence or happening.

Event Class

An event or type of event that results in a change of benefits eligibility for an employee or dependent. Event classes are prominently used in COBRA and Benefits Administration processing.

Event Maintenance

The process that enables you to manage ongoing enrollments during a plan year. Changes involving maintenance include new hires and re-hires, terminations, family status changes, and changes to benefits eligibility.

Event Rule

Used by PeopleSoft Benefits Administration to determine how events are processed by the system. Event rules look at the benefit plan options an employee is eligible for and determine which options the employee can actually *choose*. Event rules are closely associated with eligibility rules but it is important to note that they are not the same. Event rules *should not* be used to determine eligibility.

Event Trigger

You use triggers to tell the system that when a change takes place to certain data (an event), it should perform an action automatically. When the event occurs, the system writes a line to a trigger table. Then when it's time for the action, the system reads the data from the trigger table and performs the appropriate action.

EW (Enterprise Warehouse)

See PeopleSoft Enterprise Warehouse.

See also PeopleSoft Enterprise Performance Management (EPM).

Expected Losses

In the financial services industry, the amount the institution predicts it will lose in portfolio value. Loan loss reserves are set aside to cover the expected losses.

Excepted Service

As defined by 5 USC 2103, the Excepted Service consists of those civil service positions that are not in the competitive service or Senior Executive Service.

Exception

User or system delivered, defined conditions applied to scheduled, reported or payable time that require audit or review (see Time Management)

Exception Rules

A rule (s) that is applied to scheduled, reported time, and payable time in order to determine conditions which require audit or review (see Time Administration).

Exception Severity

The degree of importance associated with an exception. For example, in exception which is a result of an employee clocking in late may have a Medium severity, while an exception which is a result of an employee not clocking in has a High severity (see Time Management).

Exception Time Reporting

A method of time reporting where only differences to the schedule are provided (see Time Reporting).

Excess Plan

A pension plan where the benefit formula provides an increased benefit for Final Average Earnings above a specified integration level. This compensates for the fact that Social Security benefits are based only on earnings up to a specified maximum.

Exchange Rate Variance

In PeopleSoft Cost Management, the change in currency exchange rate between the time the item is received into inventory and vouchered in Accounts Payable.

In PeopleSoft Payables, a matching feature that compares the exchange rate on the purchase order and the invoice and then copies any variance to PeopleSoft Inventory tables for analysis and accounting purposes.

Exclusive Pricing

Supersedes all pricing structures in effect for customers and products, except **Buying Agreement**, and enables you to drive pricing with a promotional structure. Exclusive pricing can be set up for a specific time frame and associated with particular orders.

Executive Schedule (EX)

Compensation and pay plan used by the Executive Branch of the federal government. Statutory pay limits are derived from several of the pay levels within this plan and imposed on the General Schedule and other existing pay plans throughout the Federal government.

Exercisable

The option shares that are available to the optionee to exercise.

Exercise

The transaction in which an individual purchases or “exercises” the right to purchase the option shares. The IRS refers to the purchase of company stock in an employee stock purchase plan as an exercise.

Exercise Date

The date on which an individual purchases underlying shares from and option grant or transacts a simultaneous purchase and sale of underlying option shares through a cashless exercise and collects option profit in cash or shares.

Exercise Price

The price per share required to exercise a stock option.

Exercise Proceeds

Cash, stock or other recognition received by a company as a result of option exercises, including cash or stock paid by individuals to exercise options and cash company tax savings from deducting non-statutory option profits at exercise.

Expense Location

Geographic area defined to enable the recording, tracking, and reporting of expense activity.

Expense Location Amount

Authorized spending defined for an expense type in a particular expense location and currency.

Expense Location Group

Collection of expense locations based on a common classification such as state, country, or continent.

Expense Type

Means of itemizing various kinds of business expenses. Examples are hotel, dinner, or ground transportation.

Expense Type Edit

User-defined requirement that mandates input of additional data—such as an airline ticket number or number of nights in a hotel—when an expense type is selected in an expense report.

Expense Type Group

Expense types that are classified together for reporting and tracking.

Expensed Item

Non-inventory item which may represent software, manuals, documentation, or any item for which no quantity on hand is maintained, but which can be specified on a bill of material (BOM). Expensed items can only exist as components on a BOM and cannot have a BOM, routing, or production area/item definition.

Expiration

The process by which the outstanding shares of an option cease to be exercisable, generally at the end of the option term. The length of the option term and the date of expiration are established in the Grant Agreement.

Expiration Date

In PeopleSoft Inventory, the date a lot exceeds its Shelf Life and is no longer acceptable for fulfillment or consumption. (Expiration Date = Creation Date + Shelf Life)

Expiration Date

The last day of an option term in which the option is canceled and no longer exercisable.

Expiration Grace Period

When you enter a stock action allows the exercise of the already vested shares as of the action date, the system will calculate the date these shares expire based on the grace period defined on the Stock Action Rules page for that stock action. The system will automatically cancel vested shares not exercised at the end of the expiration grace period.

Express Customer

A customer for whom the minimum necessary information is entered.

Expressions

Expressions enable you to create pseudo-columns made up of mathematical calculations based on actual fields on a table. Since expressions are resolved at run-time, duplicate information is not stored on the database.

Express Order

An order entry shortcut in PeopleSoft eStore and Mobile Order Management whereby the customer populates the shopping cart and goes directly to the order summary to checkout, bypassing any billing or shipment modification screens. Billing and shipping information defaults in as previously entered.

External Data

Data from external sources. For instance, in PeopleSoft Workforce Analytics, external data may include third party salary surveys and benchmark metric surveys.

External Scheme

In the United Kingdom an External Scheme is a vocational training, education and job placement program involving an employee, an employer and the government.

External System

Any system that is not directly compiled with the PeopleTools servers.

Extra Time

Any hours worked outside of an employee's normal (scheduled/shift) hours or days. Extra time may be scheduled in advance of when it is worked, and may be subject to special compensation rules. It may be treated differently than standard time for purposes of Benefit Entitlement and Administration.

Extraction

A reusable query that specifies what information should be retrieved from the Quality database.

Extract-Transform-Load (ETL)

The extraction and transport of data from one server to another remote server. In PeopleSoft budgeting ETL specifically refers to the process by which financial and human resource data is extracted from PeopleSoft Financials and HRMS and transferred to the PeopleSoft Enterprise Warehouse which PeopleSoft Budgeting uses to access and record data transactions. Within PeopleSoft Enterprise Warehouse, data migration typically refers to information moved from outside sources into the Operational Data Store tables.

Extrinsic Rewards

Tangible rewards that can be given to the individual. Typically categorized as financial and non-financial rewards. Financial rewards would include direct compensation, indirect compensation and deferred compensation. Non-financial rewards are provided to the individually and viewed as a benefit by the individual based on the culture of the organization such as the size or location of one's office.

In PeopleSoft Workforce Analytics, tangible rewards given to an individual. Typically categorized as financial and non-financial rewards. Financial rewards would include direct compensation, indirect compensation and deferred compensation. Non-financial rewards are provided to the individually and viewed as a benefit by the individual based on the culture of the organization such as the size or location of one's office.

F

Fact

Facts are numeric data values from fields from a source database as well as an analytic application. A fact can be anything you want to measure your business by, for example, revenue, actual, budget data, or sales numbers. A fact is stored on a fact table.

Fact Table

A fact table is where facts are stored in the PeopleSoft Enterprise Warehouse.

Fair Labor Standards Act (FLSA)

A federal regulation governing several time and labor issues. *FLSA Overtime* requires that all nonexempt employees be paid at a rate of time-and-one-half for all hours over 40 physically worked during a workweek. This requirement may be superseded by state or local laws when the lesser law is to the greater benefit of the employee, or by union contract. An *FSLA Workweek* is a permanently established, regular workweek for a group of employees.

Fair Market Value (FMV)

The price of a company stock based on the current market value as determined by supply and demand, or a valuation method. The stock market sets the fair market value for a public company. For a private company the fair market value is more subjective, but typically determined by financial factors or set by an outside valuation company.

Fair Market Value Tracking Methods

Methods used to track and report trading activity on various exchanges (i.e. NYSE, AMEX, NASDAQ, etc...).

Family Medical Leave Act (FMLA)

A federal regulation that protects health benefits and job restoration for employees who must take a leave from work to care for themselves or family members. FMLA regulations contain provisions regarding employer coverage, employee eligibility and entitlement, notice and certification, continuation of health benefits, and job restoration. PeopleSoft Benefits applications offer FMLA Plans that help employers and employees determine FMLA eligibility and schedule and track FMLA leave requests.

Federal Employee Group Life Insurance Program (FEGLI)

Generally, if the employee has Federal retirement coverage or is on a temporary appointment exceeding one year, he/she is eligible to participate in the FEGLI program. Once eligible, he/she is covered automatically for Basic Life Insurance and premiums will be deducted from gross salary unless coverage is waived within the first period of eligibility. The program offers Basic Insurance coverage and three types of optional coverage: Option A (Standard), Option B (Additional), and Option C (Family).

Federal Employee Pay Comparability Act (FEPCA)

This law provides a structure and methodology to determine and authorize locality-based pay adjustments to Federal employees in order to elevate their basic pay to be commensurate with private sector employees working in the same occupations in the same geographic localities. It also includes a feature to authorize agencies to make advance salary payments to attract candidates for open positions which have consistently been hard-to-fill in certain geographic areas.

Federal Employees' Compensation Act (FECA)

This law provides compensation and medical benefits to civilian employees of the United States for disability due to personal injury or disease sustained while in the performance of duty. A feature of this law provides for the continuation of pay (COP) without charge to leave for up to 45 calendar days due to disability and/or medical treatment following a traumatic injury. Employees file claims with the U.S. Department of Labor, Office of Worker's Compensation, which adjudicates the claims and compensates the employing agencies for the employee's pay and benefits during the claim period.

Federal Employees Health Benefits (FEHB)

Generally, the employee is entitled to coverage by the FEHB program if appointed to a position with Federal retirement coverage or has been on the rolls on a temporary appointment for more than one year. The Federal employer shares the cost of the premium (about 75%); actual premiums depend on the plan selected. If under a temporary appointment, the employee pays both the employer and employee shares. If the position is part-time, the employee pays the employee share and a portion of the employer's share.

Federal Employees Retirement System (FERS)

A retirement plan available to employees of the federal government. FERS covers all employees appointed to a position in the federal government after January 1, 1987. Coverage includes Social Security, a basic annuity plan, and a TSP.

Federal Employer Identification Number (EIN)

Used to identify the tax accounts of businesses. Businesses, which have employees or operate business as a partnership or corporation, must obtain an EIN.

Federal Insurance Compensation Act (FICA)

Employee and employer contributions to Social Security.

Federal Reserve Transit Number

A unique identifier for U.S.-based banks, allowing banks to transfer funds within the Federal Reserve system.

Feeder Line

A type of production line replenishment used in PeopleSoft Flow Production. If you are using feeder line replenishment, smaller production lines create subassemblies that feed directly to your production line.

FEGLI Living Benefits Act

Beginning 7/25/95, a Federal employee who is terminally ill may elect to receive a lump-sum payment equal to the full amount of basic life insurance only, or a limited portion designated in multiples of \$1000. An election to receive this benefit is irrevocable; the individual is considered terminally ill if his /her life expectancy is 9 months or less.

FICA (Federal Insurance Contributions Act)

FICA consists of both a Social Security (retirement) payroll tax and a Medicare (hospital insurance) tax. The tax is levied on employers, employees, and certain self-employed individuals.

Fictitious Calculations

Fictitious calculation rules perform temporary calculations. A fictitious calculation is a sub-calculation run during a normal calculation to determine a net that would have been computed if certain parameters were used. This result is then used for further processing in the normal calculation. A fictitious calculation is always started from inside a normal calculation, run for one payee, and run for a specified set of periods.

FIFO (First In First Out)

Method used by companies to record Disqualifying Disposition Income. If a company uses this method they record the optionees disposition of shares by attributing the shares to the earliest exercise, purchase or release dates for which shares remain available for sale.

Fill-In Employment

Employment held by persons during the time period after leaving their regular occupation in anticipation of, but before entering, military service.

Filter

A filter creates a subset of information. Filters are used in templates to limit your information from a pick list of attribute values.

Final Average Earnings (FAE)

The PeopleSoft Pension Administration function that averages earnings from a specified period of an employee's career. The result is used as a component of the pension benefit formula.

Final Forecast

The final forecast is the prorated version of the adjusted forecast, summarized to all levels of the product hierarchy. This is the best-guess version of the forecast that is used to make all decisions dependent on the forecast.

Final Table Merge Engine

Final Table Merge Engine is used by the PeopleSoft Enterprise Warehouse; it moves enriched data from one table into another. When you run a job in a jobstream, the immediate results are stored in temporary tables. At the end of the jobstream, the Merge Engine runs and merges the output temporary tables into the final tables, where processing can continue.

Financial Accounting

The accounting for a business entity's assets, liabilities, revenues, and expenses to determine its net worth and to produce financial statements. Within Generally Accepted Accounting Principles, a business has some latitude as to when and how to record its financial transactions, as long as it continues to meet its legal and regulatory requirements. A business' financial accounting requirements are not necessarily the same as its cost accounting requirements. The one should not be mistaken for the other (i.e. the extent to which a company's financial accounting system meets its cost accounting needs depends on how it has chosen to describe its chart of accounts and the level at which it has chosen to record financial transactions.

Financial Instrument

In the financial services industry, a specific product or service sold by a financial institution to its customers. In terms of the reporting hierarchy, a product falls under a ledger account, while an instrument falls under a product. A product may be treated like a generic description or category, while an instrument is a specific instance of a category.

Financial Performance Measures (FPM)

For the financial services industry, the Financial Performance Measures program performs calculations on financial instruments based on the rules defined in the Financial Calculation Rules module, and using input from the Cash Flow Generator, Stratification engine, Product Pricing, and Curve Generator. Its calculations include: measures of duration, option-adjusted spread and option-adjusted cost for PeopleSoft Funds Transfer Pricing, and Monte Carlo simulation for PeopleSoft Asset Liability Management.

Financial Product

In the financial services industry, a product or service sold by a financial institution to its customers. In terms of the reporting hierarchy, a product falls under a ledger account, while an instrument falls under a product. A product may be treated like a generic description or category, while an instrument is a specific instance of a category.

Financial Services Instrument

In the financial services industry, products created by financial institutions and sold to retail customers. Product prices and interest rates are set by the financial institutions and take into account its customers' behavioral models.

Financial Statement Simulation

A facility within Planning & Simulation which establishes rules for simulating future period, or pro-forma, financial statements. The user defines corporate financial policies, such as corporate tax rates, dividend distribution frequency, and force balancing rules, which are then applied to cash flows for a given future accounting period. The Financial Statement Simulator engine drives costs and revenues to accounts on PF_LEDGER_F00 via a scenario.

Financials Warehouse

See Warehouses.

First Year Amount

See 1st Year Amount.

Fixed Basis

The basis option enables you to create the data for the Basis online, as part of the rule. Fixed Basis is used with the Allocation Manager only. It is available with all methods except when Period-Based Allocation is being used. The Fixed Basis is a predetermined table that can be populated online.

Fixed Offering

The offering type is fixed when the end date of each offering is the same for all employees regardless of the employee's grant dates.

Fixed Percentage

A fixed percentage value. The source pool amount will be split based on this percentage to get the target amount. Used with the Allocation Manager.

Fixed Period Requirements

In PeopleSoft Enterprise Planning and Production Planning, a lot-sizing technique that sets the order quantity to the demand for a given length of time.

Fixed Picking Bin

A dedicated picking location for an inventory item. Fixed picking bins are replenished from bulk locations when the available quantity falls below the optimal quantity.

Fixed Plan

A stock purchase offering period where the ending offering date will be the same as the purchase date. Eligible employees will always purchase stock on the specific purchase dates and by the purchase rules you define.

Fixed Quantity

An **Inventory Policy** method that defines a fixed amount of an item to be ordered to meet replenishment needs. This method can be selected as an inventory policy for order quantity, safety stock, **Reorder Point**, and minimum and maximum parameters.

Fixed Source

The fixed source option enables you to create the data for the Source online, as part of the rule. Fixed Source is used with the Allocation Manager only. It is available with all methods except when Period-Based Allocation is being used. The Fixed Source is a predetermined table that can be populated online.

Flexible Credit

Any credit associated with a given benefits program, plan, or type of coverage. Credits based on an entire program can be applied toward the benefit costs however the employee chooses.

Flexible Hours

Hours during the workday, workweek or pay period during which a time reporter covered by a flexible work schedule may choose to vary his times of arrival and departure from the worksite (see Scheduling)

Flexible Spending Account (FSA)

An account to which an employee and (optionally) an employer pledge an annual amount for a plan year. The employee then submits claims for authorized expenses.

Flexible TimeSpan

A user-defined period into which costs can be collected. Flexible TimeSpans can be as long or as short as you like—covering multiple years or a single day. The main purpose of Flexible TimeSpans is to assist you in analyzing costs.

Flexible Work Schedule

A method of scheduling a time reporter's time that is based on a range of flex hours of start and stop times and core work hours. This method can be used for scheduling clock and elapsed time reporters (see Scheduling)

FLSA Status

A PeopleSoft Human Resources term that is used to indicate whether a job is exempt or nonexempt according to the Fair Labor Standards Act. All employees associated with a

particular job will receive that job's FLSA Status. FLSA Status is an eligibility determination factor for PeopleSoft Benefits Administration.

Forecast Attribution

A FSI (financial services industry) transformation process through which forecasted product originations are pooled and run through the cash flow engine for future periods.

Forecast Calculation Process

In PeopleSoft Demand Planning, the process by which a **Statistical Forecast** is generated for each item at each level of the view. When a **Forecast Item** is set to recalculate, the system tries several forecast calculation methods and picks the one with the least amount of error. This process also makes adjustments for promotions and filters for abnormal demand.

Forecast Definition

Forecast definitions are a set of forecasting rules that generally govern multiple forecasts distinguished by key properties such as products, customers, channels, and so forth.

Forecast Element

Each forecast within a single definition is called a Forecast Element.

Forecast Fulfillment

In PeopleSoft Demand Planning, a process used to manage forecasted demand over a period of time. The process makes it possible to divide the total forecast demand into portions so that certain portions can be met, even if the total forecast cannot be met entirely.

Forecast Item

In PeopleSoft Demand Planning, a logical item used as the basis to forecast demand. The components of a forecast item key are defined for each level in a forecast view.

Forecast Level

See Level.

Forecast Period

A period in time as defined by the calendar for which data is processed through the PeopleSoft Demand Planning model.

Forecast Start Period/Year

Determines the most recent period for which demand data is available for a forecast view. This period can also be described as the last actual demand period to have had an impact on the forecast.

Forecast View

See **View**.

Foreign Education

Education acquired outside of any state of the U.S., the District of Columbia, the Commonwealth of Puerto Rico, a Trust Territory of the Pacific Islands, or any territory or possession of the U.S.

Form 10-K

A form used for annual reports pursuant to Section 13 or 15(d) of the Securities Exchange Act of 1934 for which no other form is prescribed.

Form 10-Q

A form used for quarterly reports under Section 13 or 15(d) of the Securities Exchange Act of 1934, filed pursuant to Rules 13a-13 or Rule 15d-13. This report, which public companies are required to file quarterly with the SEC, provides unaudited financial information and other selected material.

Form 5500 Participant Count Extract

A PeopleSoft Pension Administration data extract containing data that a plan administrator needs in order to complete IRS Form 5500, used to report on the number of plan participants.

Form S-8

A form used to register securities offered by a reporting company under its employee benefits plans, including stock option plans. Also called the Registration Statement under the Securities Act of 1933.

Form W-2

A form used by employers to provide workers with a statement of wages, tips and other compensation from the previous year. This form, distributed employees by January 31 of each year, reflects state and federal taxes, social security, Medicare wages, and tips withheld.

Formula

Element which enables you to define your own formulas for use—gives further flexibility to define complex organizational needs.

Formula Plan

This compensation distribution plan type is based on a pay out rule, as the pay out rule is defined. The pay out rule can be based on a flat amount, a percentage, or a data element. Whereas a Target Plan distributes pay out based on a comparison of a performance measure against a target, in a Formula Plan the pay out is based just on the pay out rule.

French Professional Elections

French companies employing a certain number of employees must hold elections for selecting personnel representatives (Délégués du personnel), and members of the Work Council (Comité d'Enterprise).

Frozen Rate

A rate that is applied to allocate resources to activities in place of the actual, budgeted and capacity rates calculated by the Activity-Based Management (ABM) Engine.

FTE (Full Time Equivalency)

FTE is the percent of full time the employee should normally work in this job. Full time is defined by the Standard Hours specified in either the Salary Plan Table or the Default Standard Hours specified in the Installation Table.

FTP (Funds Transfer Pricing) Adjustments

Adjustments made to the PeopleSoft Funds Transfer Pricing (FTP) base rate, for such factors as geographic premiums, liquidity premiums, embedded options, or incentive programs.

FTP (Funds Transfer Pricing) Base Rate

In PeopleSoft Funds Transfer Pricing (FTP), this refers to the basic charge or credit that is applied to a ledger account, a product, or an off-balance sheet position.

FTP (Funds Transfer Pricing)

See PeopleSoft Funds Transfer Pricing.

Full-Time Equivalent

See FTE.

Function

A category of pension calculation. PeopleSoft Pension Administration divides a pension calculation into nineteen “core functions” such as Service, Final Average Earnings, and Benefit Formula.

Function Result

The calculation rules for any of PeopleSoft Pension Administration’s nineteen core functions. These rules match Definitions—the specific parameters for the function—to the Groups of employees that use that particular definition. Function Result also refers to the value produced by the rules.

Fund ID

In the financial services industry, Fund ID is a lookup code used to track investment funds associated with a financial instrument or account. Provided primarily by the financial analytic applications to track investment funds for insurance policies.

Funds Transfer Pricing (FTP)

See PeopleSoft Funds Transfer Pricing.

Fungible

This term describes a resource used for multiple activities.

Future Period

Any pay period which is not current and whose close date hasn't passed (see Time Reporting).

Future Periods

The number of periods of future forecasts maintained by the PeopleSoft Demand Planning system.

G**Gang Reporting**

See Crew Reporting.

General Deduction

Any non-benefit deduction. Examples include charitable deductions, union dues, parking, garnishments, and bonds. General Deductions are calculated from the General Deduction Table; Benefit Deductions draw on one of the benefits tables.

General Ledger Distribution

The process and guidelines by which accounting information is transferred from your PeopleSoft Receivables or Deduction Management system to a general ledger system.

General Schedule (GS)

Compensation and pay plan used by the Executive Branch of the federal government.

Generation Control

Generation control elements allow you to indicate to the system whether to process an element based upon criteria you define. There are six parameters that control this function and comprise the definition of the generation control element—HR Status, HR Action/Reason, Segment Status, Frequency, Formula, and Run Types.

Generic Conversion Factor

A conversion factor that applies universally between two units of measure. The factor is used in the conversions between levels of PeopleSoft Demand Planning **Forecast Items** and Inventory Planning **Policy Item**.

Generic Process Type

This term applies to Process Scheduler. Process types are identified by a generic process type. For example, the generic process type "SQR" includes all SQR process types, such as "SQR Process," "SQR Report," and so on.

Geo RSZ Code

This code is for Belgian employers to track the geographical location for RSZ codes.

Geographic Location Code

In Canada this code is prescribed by the government and refers to the location a business is in.

Giveaway Adjustment Type

In PeopleSoft Order Management, the price break tables are set up to indicate what product the user receives as a free premium based on defined quantities or prices. The giveaway item does not have to be the same product that the customer is purchasing. For instance, you may set up a price break that indicates that a purchase of 100 widgets entitles the customer to one or more free T-shirts. The system automatically adds an order line for the free item. Giveaways cannot be applied to the total order.

Goals Matrix

In PeopleSoft Workforce Analytics, a matrix used to create calculation rules for group or employee performance goals. You can combine and standardize multiple performance goals into a single, weighted, goal score, against which actual performance is compared. A Goals Matrix can be used to in conjunction with a pay out distribution plan called a Target Plan.

Grace Period

A period that is a number of hours or minutes before or after a scheduled punch where a time reporter's punch is accepted. For Stock Administration, the period of time an optionee has to exercise an option after termination and before the option expires.

Grade

A range of pay in a graduated scale that includes positions of different occupational groups. The work performed should be equivalent as to the level of difficulty and responsibility and the level of qualification requirements of the work. The levels are established and designated within a specific pay plan by law or regulation.

Graduate Education

Successfully completed education in a graduate program for which a bachelor's or higher degree is normally required for admission. To be creditable, such education must show evidence of progress through a set curriculum, i.e., it is part of a program leading to a master's or higher degree, and not education consisting of undergraduate and/or continuing education courses that do not lead to an advanced degree.

Grandfathered Benefit

A benefit that an employee was entitled to prior to a change in the plan and that defines the employee's new minimum level of benefits. The change might be caused by a plan merger, new legislation, or a plan amendment.

Grant

A contractual right giving an individual the option to purchase a specified number of shares of stock through an Equity Compensation Plan. Also known as an option.

Grant Agreement

The legal document issued by a company defining the number of shares granted, grant price, vesting schedule and other terms and conditions of the stock option or stock award.

Grant Date

The date the individual begins participating in a stock purchase offering. The date on which an option or other award is granted. The date the company enters into the grant agreement. The underlying stock's fair market value on this date generally derives the option price.

Grant Price

The price per share at which the stock option was granted. This is the price per share the individual must pay when exercising the option.

Gross Salary

The sum of an employee's salary and earnings defined as part of gross salary. The gross salary is used to calculate budget amounts for benefit plans defined as a percentage of an employee's salary.

Gross-up

The process used to calculate taxes and resultant gross pay from a check for an exact net amount.

Group

In PeopleSoft Billing, a specific term for a posting entity composed of one or more transactions (items, deposits, payments, transfers, matches, or write-offs).

Group

Any set of records associated under a single name or variable in order to run various calculations in PeopleSoft Business Processes. In Time and Labor, for example, employees are placed in groups for time reporting purposes, while in Administer Variable Compensation, groups identify which employees are eligible for what forms of compensation. In PeopleSoft Pension Administration, you'll use Custom Statements to define criteria for grouping employees, then by associating calculation rules (Definitions) with specific Groups, you can vary rules for different classes of employees.

Group Asset

A financial asset with no cost information. It is used to depreciate the sum of the costs of its associated group member assets.

Group Asset Depreciation

The depreciation of a group asset calculated using an average service life set by a local regulatory agency and a calculated group depreciation rate.

Group Coverage (Or Generic) Qualification Standards

Standards prescribed for groups of occupational series that have a common pattern of education, experience, and/or other requirements.

Group Member Asset

A financial asset with cost information. Cost information for all group members of a group asset is summed up to the group asset level, where depreciation is calculated.

Group Security

The ability to grant or deny access to groups. You can set up group security by Group ID or by user ID.

Group Security [Time and Labor]

The ability to grant access to employee time, by providing security through Time and Labor's groups functionality. For example, you might want your employees to only access their own records, or allow your supervisors who handle all of the time input for have access to specific groups. You can restrict the user from accessing everyone, or allow the user to be able to access only their own records, or only a specific group. This feature also provides the ability for employees to report their own time.

Group Type

An indicator of the activity that created the billing group: billing, maintenance, payment, transfers, or unposted.

H

Handicap Code

A code that identifies a type of physical or mental impairment that substantially limits one or more of an employee's major life activities.

Hazard/Disposal Code

An inventory item group sharing a disposal routine.

Headcount

The number of people represented by a given Employee Survey record in the PeopleSoft Enterprise Performance Management product line.

Health and Safety Executive (HSE)

Health and Safety reporting for your UK operations is sent to the local office of the HSE per the requirements of the RIDDOR (Reporting of Injuries, Diseases, and Dangerous Occurrences Regulations).

Health Benefits Code

An alpha/numeric code that identifies each Health Benefit plan.

Health Benefits Effective Date

Date the health benefit plan goes into effect or the effective date of cancellation.

Hierarchy

Hierarchy refers to the relationship between the levels in a dimension.

Highly Compensated Employee (HCE)

An IRS employee category applied to employees who are considered “highly compensated” according to a federally set standard. This distinction is used for the purposes of nondiscrimination tests, to determine that Section 401 and Section 129 plans do not discriminate in favor of highly compensated employees.

HIPAA

The Health Insurance Portability and Accountability Act of 1996. PeopleSoft Benefits applications enable you to comply with this act, which requires that employers provide Certificates of Group Health Plan Coverage to employees who have their health coverage terminated. This certificate lists group health coverage an employee had for the twelve month period prior to the date coverage ended as a result of termination of coverage. The HIPAA certificate will be used by subsequent health coverage carriers to evaluate pre-existing condition clauses, if applicable.

Historical Periods

In PeopleSoft Demand Planning, a component that indicates the maximum number of periods of historical demand maintained for a **Forecast Item** within a **Forecast View**. Historical periods must be a minimum of two years in order to support the development of seasonal models based on an item's demand history.

Historical Rules

An element used to set up rules that retrieve data from prior periods. Historical rules can be used in formulas and fictitious calculations.

Historical Usage Calculation Method

In PeopleSoft Inventory Planning, a method that defines the set safety stock or minimum inventory level. The usage is based on the review of historical demand over the number of effective periods. The historical demand quantity is determined by one of four methods; maximum possible usage, Lead Time, estimated daily or period use, and static values calculations.

Hold Grade/Step

Grade/step the employee was in prior to receiving a temporary promotion.

Hold Last Equivalent Increase (LEI)

Date held by an employee for this event prior to receiving a temporary promotion. Necessary in order to establish the WGI due date if returning to original grade/step.

Hold Position Description

The new position description number that is the result of a reclassification action prior to the NOA being processed.

Hold Purchase

A flag that tells the system to keep this participant in the purchase process. The hold flag is maintained at the contribution page.

Hold Within Grade Increase (WGI) Due Date

WGI due date prior to an employee receiving a temporary promotion.

Holding Period

Typically refers to the holding period required for ISO's and Qualified Section 423 Purchase Plans, to receive preferential tax treatment on a disposition of shares. See Disqualifying Disposition.

Hours Counting Service

A service calculation that uses actual or generated hours to determine the service credited to a pension plan participant.

Hours Equivalence Service

A service calculation that uses hours to determine service, but that uses a set number of hours per day, week, or other period worked rather than counting actual hours.

HR Action/Reason Category

A group of related job actions—for example, hire and rehire—treated similarly for pension purposes in PeopleSoft Pension Administration.

HRMS Warehouse

See Warehouses.

I**Ignore Plan**

Complex event processing feature of PeopleSoft Benefits Administration that enables the user to designate plan types linked to a particular Event Rules/Event Classification combination as being unaffected by Benefits Administration processing.

Ignore Violations

The ability to report over capacity violations but not to score or repair them during the optimization process.

Imputed Income

Theoretical income that a company pays on behalf of an employee but the individual does not actually receive. This “theoretical income” must be added to the employee’s gross wages. In general, imputed income refers to the value of excess Group Term Life or Dependent Life coverage.

In Punch

Indicates start of a shift.

In the Money Option

When the fair market value of the stock is greater than the grant price of an option.

INAIL code

In Italy, the INAIL code is used to classify jobs according to the level of risk associated with the job and the related risk insurance required by the employer. INAIL codes are defined by the employer.

Incentive Pay Plans

In PeopleSoft Workforce Analytics, pay plans that are formula-driven based on the expected results defined at the beginning of a performance cycle. Incentive plans are designed for the individual worker, or for group levels such as teams, business units, divisions, or company-wide. Incentive plans are used for a variety of reasons; including cost control, alignment of employee and shareholder interests, and increased focus on specific performance indicators.

Incentive Plans

Pay plans that are formula-driven based on the expected results defined at the beginning of a performance cycle. Incentive plans can be designed for the individual worker or at group levels such as teams, business units, divisions or company wide.

Incentive Stock Option (ISO)

For an option to be considered an Incentive Stock Option, it must have the following characteristics:

- The option must be granted pursuant to a plan which includes the aggregate number of shares which may be issued under options and the employees (or class of employees) eligible to receive options, and which is approved by the stockholders of the granting corporation within 12 months before or after the date such plan is adopted;
- The option must be granted within 10 years from the date such plan is adopted, or the date such plan is approved by stockholders, whichever is earlier;
- The option is not exercisable after the expiration of 10 years from the date such option is granted;
- The option price is not less than the fair market value of the stock at the time such option is granted;
- The option is not transferable by such individual otherwise than by will or the laws of descent and distribution, and is exercisable, during his lifetime, only by him, and;
- The optionee, at the time the option is granted, does not own stock possessing more than 10% of the total combined voting power of all classes of stock of the employer corporation or of its parent or subsidiary corporation.

Incomplete Punch

A punch that cannot be processed (i.e. missing employee ID, invalid date or time).

Incremental Budgeting

A budgeting option during budget development that uses prior year actual or budget values as a basis and then applies a percentage that increments the base. PeopleSoft Budgeting-specific.

Incumbent

An employee currently assigned to a position.

Indirect Compensation

Typically involves non-cash types of compensation awarded to the individual in exchange for their contribution to the organization. Common types of indirect pay include health and welfare benefits (for example, medical, dental, vision, long-term disability, short-term disability, unemployment insurance), payment for time not worked (for example, holiday, vacation, sick), and employee services and perquisites (for example, club memberships, parking, holiday gifts).

Indirect Cost

A cost that is assigned by management to an activity or a cost object. An example is the cost of office space assigned to an activity.

Individual Occupational Requirements

Requirements, e.g., experience or education, for particular occupational series of positions within a series and are used in conjunction with a group coverage (generic) standard.

Individual Retirement Record (IRR)

Used by the Office of Personnel Management (OPM) as the basic record for determining the retirement benefits payable to separated federal employees and their survivors. Employees covered by the CSRS retirement plan require SF-2806. Employees covered by the FERS retirement plan require SF-3100. In addition, the SF-2806-1 and SF-3101 are used for corrections to the IRR. See also Correction to IRR.

Inherit Control Group Policies

In PeopleSoft Inventory Planning, a feature that controls whether the policy for an item is set explicitly or defaults from the associated **Policy Control Group**. A series of check boxes enable you to define which policies to inherit.

Initial COBRA Events

The event which makes an individual eligible for COBRA coverage. Typical initial COBRA events include loss of benefits eligibility due to termination, reduction in hours, retirement, and military leave, as well as divorce, death of employee, and Medicare entitlement. See COBRA and Secondary COBRA Events.

INSEE (National Institute for Statistical and Economical Studies) Codes

INSEE is an official statistics and economics organization in France. INSEE codes for your French company's organizations are used in regulatory reporting.

INSEE PCS (Classification par Catégorie Socio-Professionnelle) Code

Each PeopleSoft Human Resources French Jobcode is linked to a four-digit INSEE PCS, or social/professional classification code.

In-Service Date

In PeopleSoft Asset Management, the date upon which an asset is placed in service. In-service date is used in conjunction with an asset's prorate convention to determine Begin Depr Date.

Inservice Placement

Includes a noncompetitive action in which a position is filled with a current or former competitive service employee through promotion, reassignment, change to lower grade, transfer, reinstatement, reemployment, or restorations. Inservice placement also includes noncompetitive conversion of appointees whose Federal excepted positions are brought into the competitive service under Title 5 CFR 316.702, and Department of Defense/Nonappropriated Fund (DOD/NAF) and Coast Guard NAF employees whose positions are brought into the competitive service.

Insider

An officer, director or principal shareholder of a publicly owned company and members of his or her immediate family. This category may also include other employees of the company and people who obtain nonpublic information about the company.

Insider Trading

When a person trades a security while in possession of material non-public information in violation of a duty to withhold the information or refrain from trading. The securities law broadly prohibits fraudulent activities of any kind in connection with the offer, purchase, or sale of securities.

Instance

A row of data on the Positive Input table. Instances of positive input can be entered manually, or can be system generated. They can also be received from other applications, such as PeopleSoft Time and Labor.

Integration Level

The salary level in a defined benefit excess plan at which a higher benefit rate becomes applicable. For example, the following formula uses a \$10,000 integration level: 1% of Final Average Earnings up to \$10,000 plus 1.75% of Final Average Earnings over \$10,000.

Integration Template

A high-level template that defines the integration between PeopleSoft Projects and your other financial applications. Each integration template you create defines a specific set of business units from your other financial applications. Each project is then assigned an integration template containing this preset integration information. You can use Integration Templates to set up joint ventures, and new transactions added to that project will reflect the business units defined in the integration template.

Intensity

The cost for each unit of the activity driver.

Interest

Some companies pay interest on the monies that are being withheld from employees' paychecks. The interest plus the employees' stock purchase contributions are used to purchase stock at the end of the purchase period.

Interest Rate Modeling

An FSI feature that allows you to model interactively interest rate scenarios for Asset Liability Management, and to run rate scenarios and analysis in real time.

Interest Rate Sensitivity Model

In the financial services industry, this support module describes in granular terms how a group of customers holding a specific type of instrument with a particular interest rate will respond to changes in interest rates in the market.

Interface Loader

An SQR delivered with PeopleSoft Asset Management that is used to transfer load lines into the PeopleSoft Asset Management loader tables.

Internal Data

Data from PeopleSoft ERP systems, or other legacy ERP systems used by your organization.

Interpolation

To calculate a value of a function, or series, between two known values.

Interunit Account

The account for each business unit to which other business units in the same corporation refer when they need to distribute amounts across business units. These accounts are used to keep the individual ledgers in balance when a single transaction affects multiple business units.

Inter-Unit Drivers

Drivers that provide a means of establishing relationships between the cost objects of one organization with the supporting activities of the organizations that share business units and models.

Interunit Transaction

A transaction that involves moving amounts from an account in one PeopleSoft General Ledger business unit to an account in another General Ledger business unit.

InterUnit Transfer

A transfer that occurs between different business units.

IntraUnit Transfer

A transfer that occurs within one business unit.

Intrinsic Rewards

A reward that is generated by the worker internally such as job satisfaction, as opposed to Extrinsic Rewards which are tangible rewards.

Inventory Adjustment

A process that enables you to change the quantity of an item in the inventory system to match the actual physical quantity found in the **Storage Location**.

Inventory Business Unit

Usually a warehouse. You establish a separate inventory business unit (IBU) for any one of the following reasons: 1) You want on hand visibility to a specific location of your business that manages inventory. 2) You want to define replenishment rules for a specific location of your business that manages inventory. 3) You maintain standard and average costs in a specific location of your business that manages inventory.

Inventory Cost Element

A cost that can be associated with inventory items and inventory transactions. Examples include freight, overhead, and transportation. Each cost element has a unique cost code.

Inventory Item

A tangible commodity that is stored in an Inventory business unit (Ship From warehouse).

Inventory Location

See **Storage Location**.

Inventory Policy

In PeopleSoft Inventory Planning, a set of rules that controls how inventory policy values are calculated for items. Inventory policy is defined at the **Policy Control Group** and stockkeeping-unit levels. The elements that make up inventory policy are order quantity, safety stock, **Reorder Point**, and minimum and maximum policies.

Inventory Transaction

An event that moves inventory into, within, or out of the inventory business unit. Examples include material transfers, inventory adjustments, and standard issues.

Inventory Transaction Group

An identifier that categorizes transactions by type for costing purposes. For example, you can group all types of interunit transfers together.

Invoice Format Identifier

An identifier for the formatting options that determine the sorting and summarization levels of invoice information.

IRC 423 (Internal Revenue Code 423)

The section of the IRC that defines a Qualified Employee Stock Purchase Plan.

IRR Fiscal Data Accumulation

This report accumulates all retirement deductions for employees, as well as any LWOP and any basic pay that was received when an employee was not covered by the CSRS or FERS retirement plans.

IRR Remarks

Special remarks that are documented on an employee's IRR. IRR Remarks can be set up ahead of time and can be system-entered text or employee-specific.

IRR Status

IRRs can be in pending or final status. Those in pending status can be updated and corrected. A final status indicates that the IRR has been processed and can't be updated or corrected except through a Correction IRR or a Supplemental IRR.

IRR Worksheet

A preliminary IRR form that enables an agency to print a pending IRR for a separated employee, review it and make corrections, if necessary. Agencies can also use the IRR Worksheet to view a current IRR for an active employee.

ISO IRS \$100K Limit

The limit the IRS places on the exercisable value of Incentive Stock Options (ISOs) of \$100K per calendar year based upon the fair market value at the time of grant (Section 422 of the Internal Revenue code).

ISO to NQ Grace Period

The period of time after which an Incentive Stock Option is treated as a Non-Qualified Stock Option for tax purposes upon the termination of employment according to Internal Revenue Code Sections 421 and 422. Depending on the termination reason the option is treated:

- If the termination reason is for any reason other than death or disability, and an exercise occurs more than three months from the termination date, the system withholds taxes as if the option is a non-qualified stock option.
- If the termination reason is disability, the system withholds taxes if an exercise occurs more than twelve months from the termination date.
- If the termination reason is death, the system always treats the option as an ISO.

Issue

See **Material Issue**.

Issuer

A legal entity that has the power to issue and distribute a security.

Item

See **Inventory Item Planning Item** or **Receivables Item**.

Item Content Provider

Third-party software consisting of web-based catalogs of item and price information. These systems benefit the design and purchasing of new products by accelerating item location, maximizing design reuse, and reducing acquisition costs. PeopleSoft Purchasing, Engineering, and Inventory integrate to Item Content Providers, and the information is used by many other PeopleSoft applications.

Item Rounding Rules

A set of rules determining how fractional values are rounded so that calculations result in whole numbers. Rounding rules are used in conjunction with **Quantity Precision Rules**.

Item Simulation

In PeopleSoft Demand Planning, a process that enables you to interact with the forecast in a manageable manner and perform "what-if" analysis by comparing the effects of different forecast models.

Item Type

An identifier that defines inventory items at a very high level, and may include sets of Item Families. For example, the families Computer Items and Office Furniture might be categorized by types like Outside Manufacturing, Finished Goods, and Work In Progress.

Item-Specific Conversion Factor

A conversion between the same two units of measure when the measurements have a different value for an item. For example, a conversion between packaging unit and stocking unit.

Iterative Processing

Refers to a concept on only re-calculating those payees who have had changes and need to be recalculated (if you choose to run your payroll multiple times before actually finalizing it). This concept saves you a lot of time as you only have to recalculate those payees who have had a data change or who you indicate you would like to be recalculated.

J***Java Server Handlers (JSH)***

The JSH manages network connectivity, making service requests from the Jolt Repository, and translating Tuxedo buffer data into the Jolt buffer.

Java Station Listeners (JSL)

The JSL handles the work of the client connection, tracking client messages, and session handoff.

Job Code

An ID for a job as defined on the Job Code table.

Job Code Components

The pay components assigned to a job code by associating rate codes with job codes on the Default Compensation page or the Non-Base Compensation page of the Job Code table.

Job Code Cost

Evaluation of salaries for specific job codes.

Job Compensation Rate

The compensation rate of the corresponding job row.

Job Events

Actions relevant to an employee's employment—such as a hire, transfer, or termination—that can affect benefit program or plan eligibility. Used by PeopleSoft Benefits Administration. See Event Class.

Job Order Cost Accounting

A cost accounting method that attempts to develop a discrete cost for each job performed or product produced. Only the material, labor, and overhead required to complete the job are attributed to the job cost.

Joint and Survivor Payment Option

A form of pension payment in which benefits are paid for the life of the participant and a beneficiary. Should the beneficiary outlive the participant, the benefit continues (often in a reduced amount) for the life of the beneficiary.

Joint Staffing Report

In the United Kingdom governmental agencies are required submit the Joint Staffing Report. Although it is mainly designed for government sector organizations, commercial organizations may also use this SQR to provide a summary of their staffing by department, job code, gender and full/part time employment status.

Jolt

A BEA/Tuxedo companion product that runs on an application server domain and is used to listen for Web Client Java requests and transfer them to Tuxedo.

Journal Code

The second highest level of three categories for defining a financial transaction (or document), necessary when using document sequencing. Examples of journal code are domestic sales and export sales. This category is preceded by journal type and followed by document type.

Journal Generator Template

A table containing defaults to be used in journal generation. PeopleSoft Asset Management and Billing require one journal generator template for each transaction type.

Journal Line

A record storing a double-sided, balanced entry for a given journal. A single journal usually includes multiple lines. The sum of the monetary amounts for the journal lines in one journal totals zero (debits = credits).

Journal Template

A list of the characteristics of the general ledger journal entries that will be created from your PeopleSoft Receivables system.

Journal Type

The highest level of three categories for defining a financial transaction (or document), necessary when using document sequencing. Examples of journal types are sales journal and purchase journal. This category is followed by journal code, then document type within the journal code.

Journal Voucher

A PeopleSoft Payables voucher that enables you to make accounting entry modifications while keeping your PeopleSoft General Ledger and Payables systems in sync. Like the adjustment voucher, the journal voucher is linked to an existing voucher.

K***Kanban ID***

A unique identifier used to track Kanban cards and replenishment requests when using PeopleSoft Flow Production.

Keep Ledgers in Sync

An option in PeopleSoft General Ledger that defines how a transaction should be posted—to all ledgers in a ledger group as opposed to only a single specified ledger.

Key

See **ChartKey**.

Key

One or more fields that uniquely identify each row in a table. Some tables contain only one field as the key, while others require a combination.

Key Performance Indicator (KPI)

KPI is used by the PeopleSoft Performance Management analytical applications. KPIs are high-level measurements of how well an organization is doing in achieving critical success factors. A KPI defines the data value or calculation from the Data Warehouse tables upon which an assessment is determined.

KPI (Key Performance Indicator)

See Key Performance Indicator.

Knowledge, Skills, And Abilities (KSA)

Also known as Competencies, these are attributes required to perform a job and are generally demonstrated through qualifying experience, education, or training. *Knowledge* is a body of information applied directly to the performance of a function. *Skill* is an observable

competence to perform a learned psychomotor act. *Ability* is competence to perform an observable behavior or a behavior that results in an observable product.

L

Labor Costs

Actual expenditures associated with *salary* portion of time reporter expense.

Labor Dilution

A process that occurs after the Labor Distribution process in PeopleSoft Time and Labor. The labor dilution process takes the costs that the payroll system has calculated for payable time, determines an average or rate per hour, and applies the average amount evenly across all reported hours for the day.

Labor Distribution

The process of distributing payroll expense to the corresponding payable time entries generated in PeopleSoft Time and Labor.

Labor Distribution Amount

An actual labor cost associated with reported time.

Last Equivalent Increase (LEI)

Reflects the effective date of the last step received in grade or the last promotion, whichever is most current (does not include QSI). Used as the basis to establish an employee's WGI due date.

Last Physical Counting Event

The last date the inventory item was counted. This information is stored with each inventory item.

Last Purchase Date

The item's most recent purchase date in the inventory business unit.

Last Putaway Date

The item's most recent putaway date in the inventory business unit.

Last Putaway Document Number

The item's most recent putaway document identification number in the inventory business unit.

Last Receiving Date

The item's most recent receipt date in the inventory business unit.

Last Shipping Date

The item's most recent ship date in the inventory business unit.

Last Shipping Document Number

The item's most recent shipping document identification number in the inventory business unit.

Law Enforcement Officers (LEOs)

Positions within the Federal government involving law enforcement. Under FEPCA, many of these positions are entitled to additional special pays.

Lead-Time Estimated Usage

An inventory planning method for calculating historical usage of an item. The historical demand is prorated on a daily basis and then multiplied by the number of days lead time for each effective historical period. The maximum period value is then used as the safety stock or minimum stock level. This method should be used for items that have a steady demand pattern throughout each period.

Lead-Time Period Usage

An inventory planning method for calculating historical usage of an item. The purchase lead time is rounded up to a specified number of periods. The historical demand is calculated as the maximum usage during these periods and the safety stock or minimum-stock level is set to this value.

Leave

Time entitled to an employee as a benefit, such as, Sick, Vacation, STD, and LTD. This process is managed by HRMS (see Time Reporting).

Leave Accrual Processing

Processing of leave accruals is used to maintain employee leave balances. All leave benefit plans accrue leave by length of service or number of hours worked. Leave accrual processing is used to determine the employee's leave accrual award and resulting leave balance.

Leave Accruals

Hours that employees earn to use at another time, such as annual leave and sick leave.

Leave Plan

A method for earning and managing leave time.

Leave Without Pay (LWOP) Total (Cumulative)

An employee's cumulative number of hours of leave without pay (LWOP).

Ledger Group

In PeopleSoft General Ledger, a group of ledgers consisting of one primary ledger and secondary ledgers.

Ledger Mapping

Ledger mapping is a process that enables you to relate expense data from your general ledger accounts to resource objects. Multiple ledger line items can be mapped to one or more resource IDs. You can also use ledger mapping to map dollar amounts (referred to as rates) to business units. You can map the amounts in two different ways: an actual amount that represents actual costs of the accounting period, or a budgeted amount that can be used to calculate the capacity rates as well as budgeted model results. In the PeopleSoft Enterprise Warehouse (EW), Ledger Mapping enables you to map general ledger accounts to the EW Ledger table.

Ledger Template

A table containing records and fields common to all ledgers that ensures that all ledgers specified in a ledger group share the same physical layout.

Ledger Type

The unique combination of a single ledger, scenario, and fiscal year. Multiple ledger types make up a ledger type set.

Ledger Type Set

A collection of ledger types, the members of which will represent the members of your ledger type dimension.

Legend ID

A way of recording information that is displayed upon the Issuance Instruction Report. Can be used to record a notice that should appear on the back of a stock certificate indicating that the shares represented are "Restricted Securities." Can also be used to indicate how shares should be processed, as in the case of Swaps, Trades, Repurchases and SAR Exercises.

Level

A section of a tree that organizes groups of nodes.

Defines a set of **Forecast Item** with a common key structure. Each level is related in a hierarchical definition with other levels in the view. A level definition contains descriptive and control data that relates to the operation of the forecast at each level within the view.

Level

The section of a tree that organizes groups of nodes.

Level Income Payment Option

An annuity form of pension payment in which payments are increased in early years (prior to eligibility for Social Security benefits) and decreased in later years when Social Security benefits are also received. The goal is to provide a relatively constant total retirement income both before and after Social Security eligibility.

Life Profile

In PeopleSoft Demand Planning, a feature that enables you to establish product forecasts based on predefined patterns in an item's life cycle.

Lifecycle (of Reported Time)

A representation of time through the various stages of Time and Labor; includes processing of current, future, and previous period time from scheduling and time capture through Time Administration and distribution.

LIFO (Last In First Out)

Method used by companies to record Disqualifying Disposition Income. If a company uses this method they record the optionees disposition of shares by attributing the shares to the most recent exercise, purchase or release dates for which shares remain available for sale.

Line-Item Budgets

The budget amounts associated with ChartField distributions that make up an organization's budget. Line-item budgets include personnel costs as well as operating and maintenance costs. They also include revenue estimates. PeopleSoft Budgeting-specific.

Line Schedule Editor (LSE)

PeopleSoft Production Planning utility or tool that displays production tasks for multiple products on multiple resources across multiple periods of time.

Literal Mapping

In PeopleSoft Demand Planning, a mapping option for formatting data that is common to all records being imported. This enables you to set an available field value for all the loaded rows.

Load

The feature that initiates a process to automatically load information into a PeopleSoft application—for example, populating the PeopleSoft Benefits database with plan-level election information.

Load Activation

Load Activation enables you to specify exactly which part of your Data Mart to build, including security. You set up load activation on the Load Activation page.

Load Planning

The PeopleSoft Inventory feature that picks, packs, and ships orders by Load ID. Load Planning is also used to estimate shipping weight, volume, and charges.

Loader Table

Any table in PeopleSoft Asset Management used to store load lines before they are loaded into the system as open transactions. The loader tables comprise INTFC_FIN, INTFC_PHY_A, and INTFC_PHY_B.

Loan Exercise

A form of cash exercise, typically requiring a loan agreement and a promissory note.

Local Code

In PeopleSoft Demand Planning, a type of validation used for a user-defined field code. If a user-defined field is marked to require local table validation, **User-Field Code** are used to determine the list of valid values for the field.

Local Functionality

Local functionality is the set of information in PeopleSoft HRMS that is available for a specific country. You can access this information when you click on the appropriate country flag push button in the global window, or when you access it by a local country menu.

Location Accounting

An accounting method that captures and records material movement within the warehouse, providing accounting visibility based on where the inventory resides. You can designate certain **Storage Area** as raw material, WIP, or finished goods by assigning the corresponding account ChartField (account, department, product, and project ID) to the storage area. All inventory locations in a storage area use the storage area account.

Location Code

Locations enable you to indicate the different types of addresses a company has—for example, one to receive bills, another for shipping, a third for postal deliveries, and a separate street address. Each of these addresses has a different location number. Every customer role must have a primary location, which will be used throughout the system on all panels that display a customer address. The primary location—indicated by a *1*—is the address you use most often when contacting the customer, and may be different from the customer's main address.

Location Summary

A Picking Plan option that sorts the picking plan according to the highest-level sort options defined and prints the order lines and the total item quantity to pick from each **Storage Location**. Because the layout of the printed report reflects the actual positions of stock to be picked, personnel can follow a serpentine path through the warehouse, fulfilling all orders on the picking plan without revisiting locations.

Lock for Confirm

A flag on the Pay Line record that enables users to access the database 7 days a week, 24 hours a day, without affecting or interrupting payroll processing. Issues a warning message "A payroll is currently in process for this employee. This data will not be processed until the next payroll."

Log file

One way that you can monitor the build process is to review the log files that the build process automatically generates. Keep in mind that the log file is entirely separate from the script file; do not confuse the two. How much information that the log file contains is up to you. You can set up your logging so that all status (both good and bad) appears in the log, or you can specify that just the errors or warnings appear in the log. This section describes the options you can specify in regards to the Build log file.

Long-Term Variable Compensation

In PeopleSoft Workforce Analytics, a component of direct compensation that consists of long-term payments to an employee in the form of stock programs, and deferred compensation.

Lookup Codes

In the financial services industry, these are user-defined codes that enable the system to define and categorize incoming Instrument table information. They also provide a means for you to report on specific data, such as treasury position, balance type, and ledger account.

Lot Status

The status assigned to a lot. In PeopleSoft Inventory, a lot's status can be Hold, Open, Rejected, or Restricted.

Lump Sum

A tax method that determines withholding based on the Canadian Lump-Sum tax table.

Lump Sum Payment Option

A form of pension payment in which some or all of a participant's benefit is paid as a single sum.

Lump Sum Reporting

A Time and Labor process that enables you to report time in a lump sum of hours or units for a single Time Reporting Code, and quantities of time. The system uses a batch process to gather the information you enter, perform edits, and update the daily time tables. The system uses the default assignments you establish for workgroups, taskgroups, shifts and so on.

M***Maintenance Worksheet***

A work space for creating write-offs, matches, or adjustments to clean up posted items.

Manage Base Pay Structure

See Base Pay Structure

Manage Compensation Planning

A PeopleSoft Workforce Rewards module that facilitates modeling and analysis of compensation costs across organization units, specific job classifications, or groups. You can focus on the impact of changes to workforce size, or on changes to fixed and variable compensation elements, and determine their effects on current and future payroll costs.

Manage Market Compensation

A PeopleSoft Workforce Rewards module you use to match your company's jobs to similar jobs found in published market compensation surveys. You then calculate a target market rate based on a weighted average from multiple surveys. This market rate is then used to assess your company's gap to market and to perform cost impact analysis.

Manage Retention Planning

A PeopleSoft Workforce Rewards module that enables organizations to analyze the factors that lead to employee turnover, and how retention of key employees affects business performance and goals.

Manual Checks

Any checks calculated and prepared outside of the PeopleSoft Payroll system that you must enter into the system manually.

Manual Count

A PeopleSoft Inventory procedure in which you enter the actual count data and then create the counting event with its header, item records, and count quantities.

Manual Events

Events that are inserted by the user manually through the BAS Activity table. Events are actions that occur, which potentially change employee benefit coverage eligibility—see Event Class for more information. Used by PeopleSoft Benefits Administration.

Manufacturing Cost Element

A particular category of an item's cost. For example, when you produce a subassembly that has a cost of \$100, the cost can be broken down further into material costs, labor costs, and overhead costs.

Manufacturing Execution Systems (MES)

Third-party system that enables detailed planning and execution of production activities from production order release to completing finished goods. PeopleSoft Manufacturing integrates to MES.

Manufacturing Task

Any job that can be performed within your manufacturing facility. A manufacturing task is associated with the work center in which the task is completed.

Map File

A file that defines the relationship between fields in a third-party system and PeopleSoft Demand Planning tables.

Mapper Type

This defines whether you are mapping actual or budgeted general ledger line items to resource ID within PeopleSoft Enterprise Performance Management.

Marginal Tax Rate

The tax rate that applies to the next dollar of income generated.

Market Compensation

A compensation review process in which you match your company's jobs to similar jobs found in published market compensation surveys, for the purpose of establishing new target market rates. Also referred to as Market Based Pricing or Market Analysis.

Market Capitalization

The value of a corporation as determined by the fair market value of its issued and outstanding common stock. It is calculated by multiplying the number of outstanding shares by the current fair market value of a share. Analysts look at market capitalization in relation to book, or accounting, value for an indication of how investor's value a company's future prospects.

Market Rate

Compensation rates, usually for regular base compensation or total cash compensation, found in published salary surveys. You use the Market Compensation module in PeopleSoft Workforce Rewards to age and weight this data, to create market rates you can compare against your organization's current pay rates.

Mark-to-Market (MTM) Model

In the financial services industry, the reevaluation of a portfolio's position at current market levels.

Market Variance

A comparison of the difference between an individual's, or group's, actual compensation, and available market compensation data for a comparable population in industry. Market compensation data is usually tied to job codes, and comparisons are usually made between similar jobs. Although the variance to market can be evaluated for any of the compensation components in the Compensation tree hierarchy (such as Total, Direct, or Base), market compensation data is most typically available for, and used in evaluating Base Pay (Base Salary). The main point of reviewing the market variance is to evaluate how well your workforce is paid in comparison to both prevailing compensation in industry, and your own organization's compensation strategy.

Mass Adjustment

A process of applying an amount or percentage change to one or many line item budgets at once. PeopleSoft Budgeting-specific.

Mass Cancellation of Requisitions and Purchase Orders

A utility that allows you to select and cancel groups of requisitions and purchase orders. You can use this utility during the year as well as at year-end in preparation for closing. The utility enables you to specify ChartField criteria for selecting documents for cancellation. For example, you can select all requisitions or purchase orders for a particular fund and organization, which have a remaining balance. Then you may select a subset of those records to approve for cancellation.

Mass Change

A user-configurable entity that defines the movement of data between the tables that store your business information. Mass Changes enable you to define the criteria by which you move or replace data in your tables. Based on the configuration of your system, Mass Change dynamically builds data access and gives you complete control over your system processing.

Mass Change Template

The foundation for defining mass changes. Mass change templates enable you to control which fields will be available for the operator to specify when defining a mass change, and whether those fields will be used as selection criteria or defaults.

Mass Change Type

The building blocks used in defining mass change templates. Mass change types specify which records the resulting mass change will select From the database, alter, and subsequently write back To the database. They also set up system field defaults that run behind the scenes to ensure that this mass change is processed correctly.

Mass Validate Metadata Utility

A PeopleSoft Enterprise Warehouse utility that enables you to validate, but not compile, Metadata objects. Mass Validate certifies all “as of dates” created for Filters, Constraints and DataSets for the specified run date. This utility helps ensure that your Metadata is valid at run time and increases your chance of a successful engine run.

Match

A process in PeopleSoft Workforce Planning, by which the system compares the roles, competencies, and accomplishments in the current competency inventory, with the requirements of a given competency strategy.

Matched Punches

A period between two consecutive punches during which some activity happens measured intervals.

Match-Funding

In the financial services industry, Match Funding refers to funding an asset with a like (term to maturity) liability. This helps an organization apply the appropriate funds transfer price. Although the actual asset might be funded with shorter-term liabilities, it does provide a better measure of financial performance for that asset, such as Risk Adjusted Return on Capital.

Material Costing

An inventory accounting method that assigns a cost to items in inventory. These costs can be assigned equally across all items or tracked individually for each item.

Material Issue

An event that triggers stock fulfillment requests for items in inventory.

Material News

Company news that could be expected to affect the value of a company's securities or influence investors' decisions. Material news includes information regarding corporate events of an unusual and non-recurring nature, news of tender offers, unusually good or bad earnings reports, and a stock split or stock dividend.

Material Release

A PeopleSoft Manufacturing process that—after material has been picked—decrements on hand inventory balances for the inventory storage areas and increments inventory to the WIP

locations defined by the routing or production area. The process also changes the production ID's or production schedule's status from Released to In Process.

MAX Method

See Maximum Method Policy.

Maximum Compensation Hours

The greatest number of hours to be paid for a specified TRC (see Time Reporting).

Maximum Lead-Time Usage

In PeopleSoft Inventory Planning, a policy control value that sets the safety stock level to the maximum quantity required during the lead time. This method is normally used when the demand for an item is low or intermittent but sufficient stock must always be available.

Maximum Method Policy

In PeopleSoft Inventory Planning, a policy that controls the way in which the system determines a reasonable high limit for the maximum inventory level of an item. The system provides warning messages when the inventory level exceeds the maximum level.

Maximum Taxable Wage Base

An annual earnings threshold used for Social Security purposes. Pension plans sometimes provide different levels of pension benefits for earnings above and below the Maximum Taxable Wage Base.

Measure

A measure represents the amounts brought into a cube—the numerical data.

In data warehousing, a Measure is a field type used interchangeably with fact. Measures are types of amounts. Any numeric field you want to apply a Data Manager rule against should be a measure.

Measure ID

In the Define Market Compensation module of PeopleSoft Workforce Rewards, a Measure ID is the identification code for a measure. For market compensation surveys, the Measure ID describes the percentile for each type of pay, as well as the regression statistic type. In PeopleSoft Workforce Analytics, for Benchmark Surveys, the Measure ID describes the type of benchmark.

Measure Value

In PeopleSoft Workforce Rewards, Measure Value is the calculated market rate value from market compensation surveys for a given percentile of a market rate, and for a given scenario and job code. This is the annual rate you compare against the compensation paid for similar jobs in your company. The Measure value can also be the regression statistic value used for

Regressing Market rates. In PeopleSoft Workforce Analytics, for Benchmark Surveys, the Measure Value is the delivered Benchmark Ratio.

Member

A member is the OLAP equivalent of a node or detail value on a PeopleSoft tree. A member is a single item within a dimension, such as a single product name, department ID, or part number. Member names must be unique, even across dimensions. Cube Manager uses the term Dimension Field Mapping to identify members, dimension parents, and label mappings.

Merchant

In PeopleSoft eStore and Mobile Order Management, a level of online (web or wireless device) display and order processing controls set by the seller. See also Merchant Variant.

Merchant Variant

Associated with a user ID, a subordinate level of merchant controls that enable customers to access different variations of the same PeopleSoft eStore website. In PeopleSoft Mobile Order Management, variations are primarily used to differentiate order processing options and fulfillment methods or locations.

Merit Matrix

In PeopleSoft Workforce Analytics, a matrix used to define the amount rules for base compensation increases for your workers. A Merit Matrix provides the salary increase parameters for each review rating in a rating scale. The salary increase parameters are expressed in terms of a percentage. The percentage increase amounts are usually structured to express the company's pay strategy relative to employee performance, and the employee's degree of range penetration in their salary range.

Merit Matrix Increase

In PeopleSoft Workforce Analytics, an increase to an employee's base pay awarded based upon a Merit Matrix.

Message definition

The object definition specified in Application Designer which contains message information for PeopleSoft's Application Messaging system.

Metadata

Information about data. Metadata is the information a database or application stores to describe your business data. At its simplest, metadata defines the structure of a data field—its data type and size, for example. Metadata can also describe more complex data relationships, such as the rollup structure for a chart of accounts. Reporting and analysis tools should be able to use this metadata to let users access data just as they would from within the application, without having to understand how it is stored.

For Enterprise Performance Management, metadata is used to describe the data stored in the PeopleSoft Enterprise Warehouse. There are different types of metadata, for example, TableMaps, DataMaps, and constraints. You typically define these when you set up the warehouse; however, Metadata (particularly constraints and DataMaps) is used to develop business rules that manage aspects of the dimensional models. Metadata enables technical users to define relationships between warehouse tables and enables business users to easily identify the data that interests them without having to know the database structure.

Meta-SQL

Meta-SQL: Meta-SQL constructs expand into a platform-specific SQL substrings. They are used in functions that pass SQL strings, such as in SQL objects, the SQLExec function, Application Engine programs, and so on.

Metastring

Metastrings are special expressions included in SQL string literals. The metastrings, prefixed with a percent (%) symbol, are included directly in the string literals. They expand at run time into an appropriate substring for the current database platform.

Method

A method can only be executed from an object, using dot notation. You have to instantiate the object first, before you can use the method.

For Enterprise Relationship Management, a method is the algorithm or formula that defines how the budget amount for a line-item budget is calculated or how it is derived if a calculation is not necessary. Types of methods include amount per FTE, itemization, annual percent growth rate based on a historical figure, and number of units multiplied by cost per unit. PeopleSoft Budget Planning-specific.

Method Amount

The amount for a line item budget resulting from the application of a method. It represents the budget amount after the method is applied to a line-item budget but before any adjustments or allocations are applied. PeopleSoft Budgeting-specific.

Method Base

The defined value to which a method is applied, if applicable. Not all methods require a base. PeopleSoft Budgeting-specific.

Method Driver

The factor used in a method's algorithm. For the method, amount per FTE, FTE is considered the driver. PeopleSoft Budgeting-specific.

Method of Payment

In PeopleSoft Grants, designates whether a payment is to be through a cost invoice or a Letter of Credit.

Method Parameter

A defined and derived value within a method, which drives an expense or revenue calculation. For the method, Amount per FTE, the number of FTEs is considered the driver parameter. PeopleSoft Budgeting-specific.

Metric

A metric is a calculation of facts. A metric is usually a number, but can be anything you want to measure.

Metric Object Security

Metric Object Security determines whether an individual can see a metric object in a Data Mart.

Midpoint (Pay Range Midpoint)

In PeopleSoft Workforce Analytics, the middle value in a pay range, halfway between the minimum and the maximum, calculated as $(\text{Minimum} + \text{Maximum})/2$.

Midpoint Progression

In PeopleSoft Workforce Analytics, the percentage difference from one grade midpoint to the next higher-grade midpoint, calculated as $(\text{Midpoint2} - \text{Midpoint1})/\text{Midpoint1}$.

MIN Method

See Minimum Method Policy.

Minimum Benefit

See Grandfathered Benefit.

Minimum Compensation Hours

The lowest number of hours to be paid for a specified TRC (see Time Reporting).

Minimum Method Policy

In PeopleSoft Inventory Planning, a policy that controls the way in which the system determines a reasonable low limit for the minimum inventory level of an item. The system provides warning messages when the inventory level drops below the minimum level.

Missed Punch

A punch that is not entered at the scheduled time (see Time Reporting).

mkvdk

Verity's command-line tool used to index a collection, insert new documents, perform simple maintenance tasks like purge and delete a collection, and control indexing behavior/performance.

MLS

Multilingual support.

Modal transfer

Modal transfers allow you to transfer an operator from one component to another component (the modal component) modally; that is, requiring the operator to OK or Cancel the modal component before returning to the originating component.

Modal transfers give you some control over the order in which the operator fills in pages. They are useful for finite tasks related to the main transaction. They are particularly useful in cases where data in the originating component can be derived from data entered by the operator into the modal component.

Model Equivalency Factors

In PeopleSoft Demand Planning, factors that adjust model errors to allow a fair comparison. During the Model Reset process, the errors associated with each of the models are multiplied by their associated factors. The factored errors are then compared to select the model with minimum errors.

Models

In the PeopleSoft Enterprise Warehouse, Models enable replication of an organization's business processes for analysis of cost flow through customers, departments, and channels.

Model Recalculation

In PeopleSoft Business Planning, users may checkout slices of the entire model for their appropriate role. This requires the entire model to be periodically recalculated to incorporate the users changes for dependencies in other areas of the model.

Morphing

Morphing is a technique of automatically transforming the look and feel of an interface based on the needs of an active object. The Application Designer toolbar and menus dynamically transform based upon the type of object definition that is active.

Mortality Table

A table showing rates of death by age. Mortality tables are part of a pension plan's actuarial assumptions.

Moving Average

In PeopleSoft Demand Planning, a model that averages a selected number of the most recent demand periods and creates a forecast of demand for the next and subsequent periods.

Multibook

A functionality supporting the requirement of a company to carry one set of books in their local currency (functional currency) and another set of books in the currency of their parent company (reporting currency). In PeopleSoft General Ledger, multibook functionality is multiple ledgers having multiple-base currencies defined for a business unit, and the option to post a single transaction to all base currencies (all ledgers) or to only one of those base currencies (ledgers). Also commonly known as dual-book.

Processes in PeopleSoft applications that can create both application entries and general ledgers denominated in more than one currency.

Multicurrency

The ability to process transactions in a currency other than the business unit's base currency.

Multidimensional Analysis

A type of analysis that enables you to look at data from many different dimensions, or attributes. You identify the dimensions of the data, then combine the dimensions in various ways. For example, you might identify five dimensions of your sales data: sales, region, channel, product line, and time. Once you've identified the dimensions, you can "slice and dice" the data based on combinations of these dimensions, such as sales in the Western region for the last quarter.

Multidimensional Database (MDDb)

A database that stores data for multidimensional analysis in a proprietary multidimensional format. Users access MDDBs exclusively for reporting and analysis, never transaction processing, so they are optimized for retrieval speed.

For Enterprise Performance Management, a Multidimensional Database stores data for multidimensional analysis in a proprietary multidimensional format. These databases are used exclusively for reporting and analysis, and never transaction processing, so they are optimized for retrieval speed.

Multiple Jobs

Multiple jobs allow you to hire an employee into more than one concurrent job and have them processed through Payroll, Benefits, and Pension. In order to enable this feature, the Multiple Jobs check box must be selected in the PeopleTools Options page.

Multiple-table dynamic tree

The user drills down through a hierarchy of parent and child records.

Multivariate Forecasting Techniques

In Enterprise Planning and Simulation, this is a forecasting method that uses both the recorded history for the target value and the history and forecasts for other variables (causal factors) to infer, not only a forecast for the target value, but also a functional relationship between the causal factors and the target value.

N**National Association of Securities Dealers, Inc. (NASD)**

Self-regulatory organization of the securities industry responsible for the regulation of The NASDAQ Stock Market and the over-the-counter markets. The NASD operates under the authority granted it by the 1938 Maloney Act Amendment to the Securities Exchange Act of 1934.

National ID Number

Different countries track some form of National ID for payroll, identification or benefits purposes. For example, German workers are assigned a Social Insurance Number, UK workers have a National Insurance Code, and US laborers have a Social Security Number. Each of these different types of National IDs has unique formatting requirements associated with them as well.

Nature Of Action (NOA) Code

Indicates the type of personnel action being processed.

Nature Of Action Description

Describes the NOA code.

Nature Of Action Effective Date

The date the personnel action is effective.

Negative Amortization

Occurs when a loan payment does not cover the interest due on the loan payment, resulting in an increase of the principal amount.

Net-To-Zero Adjustment

A prior period adjustment where no compensation affecting fields on the pre-existing (original) record are changed by the adjustment.

New Hire Report

In the United States the Personal Responsibility and Work Opportunity Act of 1996 (the so-called Deadbeat Dads law) requires employers to report new hires to specified agencies within a pre-determined number of days from the hire date.

Next Level Item

In PeopleSoft Demand Planning, the **Forecast Item** at the next level that contains the current item as a child. This is the key of the group item at the next level up and is always within the same view.

Next Year

PeopleSoft Benefits term referring to the next open enrollment processing year.

NIC (Numéro Interne de Classement) Code

In France NIC numbers identify the entities inside the same enterprise, and represent an Internal Filing Number.

No Control

A target control that allows the user to submit a budget even if it is not within the planning target and the tolerance levels. The system tracks the budget against the defined planning targets but does not generate any warnings or validations. Users can still compare their planning targets against their budget amounts on the Planning Targets page in Line Item Budgeting.

Node

An individual item on a tree. Nodes summarize detail values or other nodes, and may or may not roll up into other nodes or levels.

Node

A node is a name that you can use to refer to some source of HTML content. In more practical terms, a node is a URI string that defines the database and server to be used when the portal servlet attempts to retrieve content, proxy addresses, and assemble pages.

Non-Base Pay

A pay component not included in the job comp rate calculation. It is used by payroll only in the paysheet calculation. For example, non-base pay can be set up for additional work, holiday pay, bonuses, and so on.

Non-Benchmark Jobs

See Benchmark Jobs.

Noncompetitive Action

An appointment or placement in a position in the competitive service that is not made by selection from an open competitive examination, and that is usually based on current or prior Federal service. A noncompetitive action includes:

- All of the types of actions described under inservice placement, above
- Appointments of non-Federal employees whose public or private enterprise positions brought into the competitive service under Title 5 CFR 316.701; and
- Appointments and conversions to career and career-conditional employment made under special authorities covered in 5 CFR 315, Subpart F.

Nondiscrimination Tests (NDT Tests)

Tests used to help employers ensure that their organization's 401(k), 401(m), and Section 129 dependent care reimbursement plans do not discriminate in favor of highly compensated employees. See Highly Compensated Employees.

Non-Employee

Those workforce resources hired to perform a specific job and/or hired for a specific period of time. Although non-employee time will be entered into Time and Labor for the purposes of managing their Task time, non-employee earnings will not be updated to Payroll and they will not be paid through the Payroll system.

Non-HR Employee [Time and Labor]

An individual employed by the corporation who is administered outside of the PeopleSoft Human Resources system.

Non-Job Event

Actions which result in changes to an employee's personal or demographic information that also affect benefit program and plan eligibility—such as an a state or postal code change, a family status change like a divorce, or a birthdate change. Used by PeopleSoft Benefits Administration. See Event Class.

Non-Productive Time

Any employee scheduled work time spent on tasks (or non-tasks) other than those which the employee was hired to perform. This could include time spent in training, time spent in meetings, travel time, and time spent reporting time.

Non-Qualified Dependent

Dependents such as domestic partners, their children, and other people who do not meet the definition of qualified dependents as presented in IRS Section 152. PeopleSoft Benefits applications enable the creation of benefit programs that offer health and life coverage to non-qualified dependents.

Nonqualified Plan

A plan that doesn't conform to ERISA rules. Employers cannot take a tax deduction for contributions to a nonqualified plan; instead, plan benefits are generally paid directly from the employer's assets.

Nonqualified Stock Option (NQ)

Any option that does not satisfy the conditions of a statutory stock option under the Internal Revenue Code and therefore does not qualify for preferential tax treatment. Generally, companies can design nonqualified options in almost any way they like. Features are:

- The grant price may be less than fair market value (with some exceptions under state law).
- Grants are not limited to employee of the company or subsidiary.
- No taxable income is recognized at the time of grant.
- Options can be granted to anyone (Employees, Consultants and Board of Directors).
- Difference between the fair market value on the date of exercise and the grant price is treated as compensation income.
- In the U.S., withholding tax obligation arises at the time of exercise.
- Company receives a tax deduction equal to the compensation income recognized.

Nontaxable Benefits

Any employer contributions that are not subject to Federal Withholding Tax, such as an employer's portion of a 401(k) plan.

Normal Form of Payment

The payment form associated with the amount calculated by the benefit formula. Pension Administration uses it as a basis for converting to optional forms of payment.

Normal Hours

The hours an employee is normally expected to be at work for any given workweek.

Normal Line Of Promotion (Career Ladder)

The pattern of upward movement from one grade to another for a position or group of positions in an organization.

Normal Retirement Date (NRD)

The date on which an employee is eligible to retire and begin receiving pension benefits. Eligibility for normal retirement is typically based on age only.

Normalized database

A normalized table adheres to certain standards designed to improve the productivity of the database user. Normalization makes the database much more flexible, allowing data to be combined in many different ways.

The standards for a normalized database are called forms, such as first normal form, second normal form, and so on.

Normalized Loss

In the financial services industry, Normalized Loss is the expected loss on a loan and is netted out of the profit and loss statement for management accounting or profitability measurement purposes. Similar to the bank's loan loss reserve, it enables the institution to analyze and account for expected losses on a more detailed level, by financial product.

Northern Ireland Report

In the United Kingdom the Fair Employment (Northern Ireland) Act of 1989 requires private sector employers with more than 10 employees to submit the Northern Ireland report to the Fair Employment Commission annually. The report indicates the religious composition (referred to as Community Background—Catholic, Protestant, Other) of the workforce, job applicants and appointees.

Not To Exceed (NTE) Date

Types are as follows:

- Appointment NTE Date: Indicates the length of time a person may serve in a position.
- Classification Temporary NTE Date: Established temporary date that is used for a temporary classification of a unique position.
- Hospitalization coverage.
- LWOP NTE Date: NTE date is the last day the employee is in leave without pay status. The employee is scheduled to return to duty the next workday.
- Position NTE Date: Indicates the length of time a position is available for use.
- Promotion NTE Date: Specific NTE Date: Specific time for an increase in grade on a temporary basis.
- Suspension NTE Date: Specific time an employee is to be on suspension. No salary is paid for the period.

nPlosion

A PS/ nVision feature that enables you to expand rows and columns in your spreadsheet to underlying details, as in drilldown.

Numeric constant

Numeric constants are any decimal number used in PeopleCode.

O**Object-Based Modeling**

Object-Based modeling technology enables you to create parent and child models. In the PeopleSoft Enterprise Warehouse, you set up such models using the Scenario Manager.

Object reference

An object reference is one that uses the current object. For example, in the case of a component, pages within the component are related objects. The menus that use the component are its object references.

Occupant Of Position/Vice

Indicates new position or former occupant of a position.

Occupational Series Code

Designates a grouping of positions similar in work and qualification requirements. They are designated by a title and four digit number (e.g., the Accounting Series, GS-0510).

Off Date

A specific date that is defined as an off day (see Scheduling).

Off Day

A 24-hour period rounded by daybreaker with no associated shifts (see Scheduling).

Off Day Type

A classification of off days (i.e. holiday, plant shutdown) (see Scheduling).

Off-Cycle Processing

The process of calculating and creating a paycheck for one or more employees aside from the normally-scheduled (*on-cycle*) payroll run for their pay group. You typically use off-cycle payroll processing for employees who are being terminated, new hires who weren't entered into the system in time for the last on-cycle payroll run, and employees who received an incorrect paycheck during a normal on-cycle payroll.

Offer Period

This is the period of time in which an employee's ESPP share price is determined.

Officer

An insider who sits on the Board of Directors and who is also an employee of the corporation. Examples include CEO, CIO, CTO, CFO, COO, Corporate Secretary, and Treasurer.

Official Forwarding Address

An employee's mailing address following separation.

Official Languages Act (OLA)

Canadian federal institutions are required to report on the official languages used in their departments, in accordance with the Official Languages Act (OLA).

Official Personnel Folder (OPF)

The repository of a Federal employee's official documents related to Personnel history.

Official Personnel Folder (OPF) Address

Indicates the address where the Official Personnel Folder is maintained.

Off-Invoice Discount (OI)

A per unit discount deducted from the customer invoice and given by a manufacturer for promotional activities. Off-invoice discounts can originate from a National Allowance or Customer Promotion, and are passed to PeopleSoft Order Management so the discounts are applied correctly during order entry.

Offset Plan

A pension plan where the benefit formula includes an offset of a portion of the participant's Social Security benefits.

OLAP

Online Analytical Processing. OLAP is the multidimensional analysis of application data, performed interactively. The acronym contrasts with OLTP (Online Transaction Processing), which is what most production business application systems do.

OLTP (Online Transaction Processing)

OLTP refers to the applications that perform the business transactions that keep your company running, such as processing invoices or enrolling employees in benefits programs.

Ontario Employment Equity Commission (OEEC)

The OEEC requires employers in Ontario to complete workforce surveys.

Open Enrollment

The scheduled annual re-enrollment of plan participants into appropriate benefit programs and, within those programs, benefit options.

Open Price

The price at which a security starts a trading day.

Open Season

A time period during which Federal employees are open to re-enroll in a specific benefit plan and option. Open Seasons can be scheduled at varying times throughout the year and multiple Open Seasons can occur concurrently with each other. For FEHB processing, it is generally the time period from mid-November through mid-December. For Thrift Savings Plan (TSP) processing, these are semi-annual and are generally held from May 15 - July 31 and November 15 - January 31. Open seasons for FEGLI are infrequent and special notification from the OPM would be issued to all Federal employees should they occur.

Open Transaction

A transaction that has not yet been processed in PeopleSoft Asset Management.

Operation

In PeopleSoft Manufacturing, a job or task performed in a specified amount of time, done in one work center, and using one or more resources.

Operational Data Store (ODS)

A staging area in PeopleSoft Enterprise Warehouse for source application data and pre-processed data for tables optimized for reporting.

OPF Code

Indicates where the OPF is maintained.

OPM

Office of Personnel Management.

Optimization

In PeopleSoft Demand Planning, the process of evaluating and improving forecast model parameters.

Optimize

The process of creating a new PeopleSoft Planning schedule by repairing the violated constraints in a schedule automatically. The Optimizer can be prioritized for meeting due dates, for minimizing overtime costs, and so on.

Option

A contractual right that gives the individual the option to purchase a specified number of shares of stock through an Equity Compensation Plan. Also known as a grant. Regulatory agencies also refer to an option as the right to purchase stock in an employee stock purchase plan. These options are considered granted on the offering begin date.

Option Adjusted Cost (OAC)

In the financial services industry, the difference in the average expected return between an instrument without embedded options that are otherwise identical to the fully loaded instrument and the instrument fully loaded with embedded options.

Option Adjusted Spread (OAS)

In the financial services industry, the average return expected for an instrument, over the short-term risk-free rate, for all projected interest rate paths generated using Monte Carlo simulation.

Option Types

Types of stock options. PeopleSoft Stock Administration supports the following stock option types. Incentive Stock Options (ISO), Nonqualified Stock Options (NQ), Tandem Incentive Stock Options/Stock Appreciation Right (ISO/SAR), Tandem Nonqualified Stock Option/Stock Appreciation Right (NQ/SAR), Restricted Stock Award (RSA).

Optional Forms of Payment

Any alternative forms of payment available to a participant retiring under a pension plan. These can include: annuity options paid over the participant's (and possibly a beneficiary's) lifetime; certain term options paid over a specified number of years; and lump sum options paid out in a single payment.

Options Outstanding

The total number of option shares held by optionees. It is the number of Grants less the number of Exercises, Cancellations, and Expirations.

Order Group

Order groups link order terms that default into sales orders and quotes when you select an order group code.

Order Line Number

The line associated with an order identification number. The order line identifies an item and the requested quantity.

Order Quantity Policy

In PeopleSoft Inventory Planning, a policy that determines how replenishment order quantities are calculated for an item. For example, you can use a static number, provide upper and lower limits, or use an economic order quantity calculated by the system.

Ordinary Income Tax

An individual's tax on earnings from wages, tips, and all other sources except capital gains. Includes option profits upon exercise of non-statutory options.

Origin ID

A code that identifies the location of a payment deposit in PeopleSoft Billing. Origin ID also distinguishes the method of the payment's entry: online, external, or lock box interface. Billing origin ID identifies the remit to origin for billing. This function is mainly used for specifying where the customer should send payment.

Original Option

A stock option that is eligible for repricing. This option has a grant price greater than the current FMV.

OSHA 200 Occupational Injury and Illness Recordkeeping Log

In the United States this record-keeping logbook meets reporting requirements for reporting occupational injuries and illnesses to the Occupational Safety and Health Administration (OSHA). It lists the case numbers and details of each injury and illness that occurred during a calendar year.

Out Punch

Indicates the end of a shift.

Out-of-the-Money

A term used to describe an employee stock option when the current market price is below the option grant price. When an option is out-of-the-money, it costs more to exercise than the underlying stock is worth. Such options are also described as being "underwater."

Output Result Tables

Refer to the database tables that are populated with information at the end of each pay calculation.

Output VAT

VAT collected on sales or outputs.

Outside Scope of VAT

A transaction determined as not subject to VAT. No VAT code is associated with this type of transaction. The transaction is still logged in the VAT transaction table, but no tax is applied.

Outside The Register Appointment

An appointment in the competitive service made under an agency's applicant supply system because either there is not a sufficient number of eligibles on the appropriate register or no competitor inventory exists. Agencies are also authorized to make temporary limited appointments outside the register at grades GS-12 and below.

Outstanding Option

A stock option that still has unexercised (vested or unvested) uncanceled or unexpired shares. Options with a "pending" status are not included. Only options with a status of 'active' or 'suspended' are considered outstanding.

Overlapping Promotions

Multiple customer promotions related to the same customer, and the same product, at the same time.

Override Rate

Cost per hour or unit reported with time used to replace the time reporter's default rate. (see Time Reporting)

Override text

Text not derived from field descriptions.

P***Package level***

The top level of organization is the package level. The package is the entire transaction set file, addressed to your company much as a mail package would be.

Page

A page defined in Application Designer as part of a PeopleSoft Internet Architecture application.

Page Assembly

Page assembly is one of the functions of the portal servlet. Page assembly involves intercepting the user's content request, retrieving the content, and properly formatting it using a pre-defined portal template. To complete the page assembly process, the portal servlet merges content from any HTML documents that it retrieves along with the defined template

HTML. The assembled page is then sent back to the user's web browser as a single HTML document.

Page buffer

Consists of rows of buffer fields that hold data for the various records associated with page controls, including the primary records, related display records, derived/work records, and translate table records.

Pagelet

A page designed to appear on a customized homepage. A pagelet is smaller than the typical page dimensions in many PeopleSoft applications. It can be based on either a page designed in Application Designer or on an iScript.

Paired Punches

Two punches for the same employee in chronological order that exists for the purpose of determining the duration between the punches.

Par Value

The nominal or face value of a security. It establishes a price floor below which shares may not be issued. With common stock, the company issuing the stock sets par value. Par value has no relation to fair market value. Some companies issue no par value stock.

Parallel Processing

In the PeopleSoft Enterprise Warehouse, parallel processing is a system function that "locks in" the information you use for processing. This enables the system to run identical or similar processes at the same time without impacting your results. Running concurrent processes greatly reduces the amount of time it takes to run within the system.

Parent Budget

In commitment control, you can build a hierarchy between different budgets, such as summary and detail budgets. Specifying a relationship of parent and child between a summary and a detail budget for purposes of budget inquiries enables you to retrieve information about either budget through the other.

Parent/Child Models

Object based modeling technology enables you to create parent and child models. In the PeopleSoft Enterprise Warehouse, you set up such models using the Scenario Manager.

Parent node

A tree node linked to lower-level nodes or details that roll up into it. A node can be a parent and a child at the same time, depending on its location within the tree.

Parent Task

A higher-level Planning task in a schedule's hierarchy that drills down into subtasks. Its start time is the start time of its earliest subtask, and its end time is the end time of its latest subtask. Planning tasks are distinct from Manufacturing tasks.

Partial Pay

The pay processed whenever a job record has an effective date in the middle of a pay period. Typically, this happens whenever you hire, terminate, transfer, or change the rate of pay for an employee mid-period.

Participants

Individuals who elect to participate in the stock purchase plan.

Participation

The PeopleSoft Pension Administration function that determines whether an eligible employee has met the plan's rules for joining the plan. Generally, these rules are based on age and service criteria.

Participation ID

In the financial services industry, this is a lookup code used by the financial analytic applications to identify the participants (syndicators) involved in, or responsible for, a financial instrument or group of instruments.

Passive Control

A target control allowing the user to submit a budget even if it is not within the planning target and tolerance rules. The system responds by sending an email to the user of the next budget center level indicating that the budget exceeded planning target tolerance levels.

Passive Events

Events that are initiated by a change that has taken place over time, rather than by a direct data entry action. Events are actions that potentially change benefit coverage eligibility. Examples of passive events include an employee's reaching the age of retirement. See Event Class for more information.

Pattern Reporting

A Time and Labor process that enables you to report a start and stop date, a pattern of one or several time reporting codes, associated hours, amounts, or units and task information once for an employee. The system transforms the information into instances of daily time for each scheduled employee work day based on the employee's schedule.

Pay

Types of "pay" are as follows:

- **Basic Pay:** generally, the total amount of pay received during any one calendar year at the rate fixed by law or administrative action for the position held by the employee or judicial official prior to any deductions and not including any special payments or premium pay.
- **Gross Pay:** total compensation earned by an employee, annuitant, or survivor of a judicial official prior to any deductions. Includes basic pay plus locality pay; availability pay (if any) for LEOs; special payments (if any); an annuity (if any); plus awards (if any).
- **Premium Pay:** pay provided to an employee as a regular addition to basic pay (e.g., administratively uncontrollable overtime (AUO), availability pay, overtime, night differential, holiday pay, etc.).

Pay Basis

A code indicating the principal condition in terms of time, procedures or criteria, that serves as a basis for computing an employee's pay.

Pay Calculation

Formula that calculates an employee's gross to net.

Pay Calendar

Payroll processing cycle for a given pay group.

Pay Components.

Rows in the compensation record. They build the compensation packages in the compensation record.

Pay Confirmation

Process in which the system updates all to-date totals on the database for earnings, deductions, and taxes for pay groups assigned to a given Pay Run ID.

Pay Entity

A pay entity is the organization responsible for making payments to payees. You can also use a pay entity to define the type of currency to be used when processing calculations. The pay entity is a legal definition of an organization from a payroll perspective. In many cases, an organization and a pay entity are the same. However, PeopleSoft Global Payroll does not define a relationship between an organization and a pay entity.

Pay Frequency

Defines how often employees in a pay group are paid—weekly, biweekly, monthly, and so on.

Pay Group

A set of employees grouped together for payroll processing. It's a way of "bundling" payees for more efficient processing. A pay group is made of payees that the system processes at the same time during a pay run.

Pay Period

The established time segments for which employees in a pay group are paid. Pay Periods are defined by their beginning and ending dates.

Pay Plan

A code that denotes the pay schedule under which an employee is paid , e.g., JS, UG, UJ, etc.

Pay Slip

Either an actual check or an advice notice of a direct deposit. You build these to match your organization's needs. A pay slip is the details of a payment you've made.

Pay Structure

In PeopleSoft Workforce Analytics, Pay Structure consists of a series of pay ranges or grades, each with a minimum and maximum. You develop pay structures to support and reinforce your company's pay strategy (for example, to target the market 50th percentile).

Payable Date

The date that a corporate distribution, such as a dividend, is payable to the record holders of a corporation's securities.

Payable Time

Time that is ready to be collected by the payroll system (see Time Reporting).

PayCycle

A set of rules that define to PeopleSoft Payables the criteria by which it should select scheduled payments for payment creation.

Payee

Any payroll recipient. A payee can be an employee or a non-employee of an organization.

Payee Process Stat Record

A record created for each payee during the payroll process. The system creates one Process Stat record per payee for each calendar.

Payee Section

Type of section that can be added to a process list. A payee section defines a set of elements that is to be resolved for a particular payee.

Payline

Record containing standard payroll information for an employee, such as the amount of regular pay, number of regular hours, additional pay (if any), and tax information and job data.

Payment Interface

An Application Engine process that loads payment information from the Banks Statement tables and the Staging tables to the Application tables and performs various checking and default operations.

Payment Predictor

PeopleSoft Receivables' automatic cash application feature that pairs open items with unapplied payments based on predefined algorithms.

Payment Schedule

A schedule of payment dates for leased assets.

Payment Selection

A process by which PeopleSoft Payables selects scheduled payments that are eligible to be paid in a pay cycle.

Payment Worksheet

The work space in which open items are paired with unapplied payments.

Payroll Certifying Officer

The individual with the delegated authority for approving all items relating to payroll for those employees under his/her authority.

Payroll Process Tables.

Records holding data necessary to process a payroll, such as employee, company, and tax information.

Paysheets

Repository for the raw data necessary to calculate pay for employees, including earnings, hours, deductions, taxes, and accounting data.

PBGC Rates and PBGC Grading

The interest rates published monthly by the Pension Benefit Guaranty Corporation. There is an "immediate" rate that applies once benefits commence as well as a series of "graded" rates—calculated based on the immediate rate—that are used during the time between benefit determination and a deferred benefit commencement.

Pegged Chain

A method the PeopleSoft Enterprise Planning and Production Planning solvers use to determine feasible plans. The method ties tasks together in order to explicitly record which supplies are being used to satisfy which demands. Through this process, the Planning engine first determines which independent demand has the highest priority. Then, it determines the lateness preference ranking for dependent demand tasks.

Penalty

A user-assigned value for constraints that can be violated, determining how the schedule's score will be calculated. Setting the penalty configures the constraint to your priority. Use the Control Page to assign a higher penalty to violations that are more critical to your schedule or a lower penalty to constraints that you can deal with externally.

Pending Exception

Any known exception to an employee's scheduled workday. Pending Exceptions are future dated (future is defined to be for a date under report beyond the last date of the employee's current pay period).

Pending Item

Information in PeopleSoft Receivables that has been entered in or created by the system, but hasn't yet been posted. During the Receivable Update process, the system uses the pending items to update customer balances—either by creating new items or by adding item activity lines to existing items.

Pending Time

Time that has been reported or is assumed to have been reported (based on employee work schedule and calendar date) that has not been used by the business entity. Pending Time may be for past, current, and future pay periods. It is the label for those time transactions that are waiting to be used by the business (for example, approved and unapproved time not yet updated to Paysheets).

Pension Status

An employee's standing with regard to a particular pension plan. For example, employees can be active participants, terminated deferred vested, or in pay status.

PeopleCode

PeopleSoft's proprietary language; it is executed by the PeopleSoft Application Processor. PeopleCode generates results based upon specific actions, based upon existing data or the

actions of a user. Business Interlink Objects are executed by calling the `execute()` method from PeopleCode. This makes external services available to all PeopleSoft applications wherever PeopleCode can be executed.

PeopleCode Event

An action that an end-user takes upon an object, usually a Record Field, that is referenced within a PeopleSoft page.

PeopleSoft Activity-Based Management (ABM)

A PeopleSoft Analytic Application that aligns organizational costs with operational activities, enabling a coordinated approach to expense and PeopleSoft Activity-Based Management. PeopleSoft Activity-Based Management identifies and assigns operational activities to products, customers, or services.

PeopleSoft Analytic Applications

These are applications within Enterprise Performance Management (EPM) that help you enrich the data in the PeopleSoft Enterprise Warehouse and perform forward looking simulations and scenarios. These applications include: PeopleSoft Activity-Based Management (ABM), PeopleSoft Asset Liability Management (ALM), PeopleSoft Balanced Scorecard (BSC), PeopleSoft Funds Transfer Pricing (FTP), PeopleSoft Risk Weighted Capital (RWC), and PeopleSoft Workforce Rewards. PeopleSoft Funds Transfer Pricing and PeopleSoft Risk Weighted Capital are applications that target the financial services industry (FSI).

PeopleSoft Asset Liability Management (ALM)

PeopleSoft Asset Liability Management provides financial service institutions with the analytical tools to define, measure, monitor and manage interest rate risk, liquidity risk, options risk, and to some extent exchange rate risk. The primary audience for PeopleSoft Asset Liability Management is the financial institution's Asset/Liability Committee (ALCO).

PeopleSoft Balanced Scorecard (BSC)

PeopleSoft Balanced Scorecard converts an organization's vision and strategy into a comprehensive set of performance and action measures that provide the basis for a strategic management system.

PeopleSoft Budgeting

A budgeting application that is a combination of Education and Government (E&G) Budget Planning and Budgets (commercial). This application resides on the EPM database, primarily using the ODS layer of the PeopleSoft Enterprise Warehouse for its data.

PeopleSoft Business Analysis Modeler (BAM)

A multi-dimensional modeling tool used to support several analytic applications.

PeopleSoft Business Planning

A planning application that enables financial executives to model various alternatives and set corporate financial targets to achieve their strategic goals. PeopleSoft Business Planning integrates with PeopleSoft Analytic Forecasting, PeopleSoft Activity Based Management, PeopleSoft Workforce Analytics, and PeopleSoft Budgeting applications.

PeopleSoft Customer Behavior Modeling

A PeopleSoft application that enables you to: a) create a customer profile by extracting customer data from the Enterprise Warehouse; b) create segments and samples from the profile to efficiently target marketing campaigns and further analyze customer behavior; c) append to it additional data from external sources such as demographic, credit or psychographic information; d) use a data mining tool to create a predictive model; e) score the customers in your profile using the predictive model. You can then publish the results to another transactional application.

PeopleSoft Customer Scorecard

A product that provides a pre-defined set of customer-oriented key performance indicators (KPIs), to help you build a scorecard specific to your organization. This scorecard facilitates the measurement and communication of customer satisfaction, customer activity, and objectives across your organization.

See also PeopleSoft Balanced Scorecard

PeopleSoft Enterprise Performance Management (EPM)

Enterprise Performance Management is a comprehensive, integrated analytic business solution designed to increase the value of organizations by enabling people to make better decisions. The PeopleSoft Enterprise Performance Management product line consists of the PeopleSoft Enterprise Warehouse and optional analytic applications and Data Mart products.

PeopleSoft Enterprise Warehouse (EW)

PeopleSoft's data warehousing solution. The PeopleSoft Enterprise Warehouse provides the tools necessary to query, analyze, and present information to provide the optimal environment for business intelligence. It is the central repository for data that will be used with the analytic applications in the Enterprise Performance Management product line, and can also serve as a standalone data warehouse. The PeopleSoft Enterprise Warehouse consists of dimension, fact, reference, and error tables, reporting and ETL tools (Informatica PowerMart), as well as the Operational Data Store (ODS). The tables in the PeopleSoft Enterprise Warehouse are maintained separately from your transaction-based systems to allow for comprehensive analysis of data originating from any Online Transaction Processing (OLTP) or legacy system.

PeopleSoft Funds Transfer Pricing (FTP)

A PeopleSoft Analytic Application that enables an institution to accurately measure and tune profitability. PeopleSoft Funds Transfer Pricing is an interest rate that represents the value of an asset or liability to the institution. PeopleSoft Funds Transfer Pricing is based on market rates, adjusted for risk and cost variables, specific to the institution. By assigning PeopleSoft Funds Transfer Pricing to each item on the balance sheet, the institution can remove the

effects of interest rate volatility from business units, so that profitability measurements are based on factors within their control, that is, credit quality, pricing and product strategy.

PeopleSoft Operations Data Stores (PODS)

See Data Warehouse Tables

PeopleSoft Marketing Insight

A tool that helps you analyze your marketing campaigns and activities. It helps you determine the effectiveness of marketing events based on factors such as number of leads generated, profiles of respondents, campaign return on investment, and campaign forecasted costs to complete.

PeopleSoft Risk Weighted Capital (RWC)

A PeopleSoft Analytic Application that enables the financial services industry to accurately measure capital that has accounted for risk. RWC allocates capital to various levels within a financial institution according to risk, providing the opportunity to measure performance based on how well each business unit, product, customer, or transaction generates income given its perceived level of risk as quantified by the allocation of capital.

PeopleSoft Sales Activity Insight

A tool that helps you analyze key components of the sales process, such as pipeline status, discount analysis, and sales process.

PeopleSoft Support Insight

A tool that helps you determine the effectiveness of your customer service organization. It helps you answer questions such as: Are we effectively handling customer issues? Has contact center performance changed from last year to this year? Which product quality issues are most prevalent?

PeopleSoft Workforce Analytics (WFA)

PeopleSoft's complete workforce analysis solution, which includes the PeopleSoft Workforce Rewards analytical application and the PeopleSoft Workforce Analysis Insight. The complete solution set includes the PeopleSoft Enterprise Warehouse and PeopleSoft Balanced Scorecard products. It helps to manage strategic employee compensation, goals, and competencies, as well as retention.

PeopleSoft Workforce Rewards

PeopleSoft Workforce Rewards is an analytical application you use to align your workforce compensation and retention initiatives with the strategic objectives of your organization. Modules include Market Compensation, Base Pay Structure, Compensation Planning, Workforce Simulation, and Retention Management. You integrate data from multiple internal and external sources, enrich it using rules you define based on any data in your PeopleSoft Enterprise Warehouse, and simulate multiple scenarios of future workforce compensation and

retention activity. You then analyze and evaluate your scenarios, and make actionable decisions you can communicate back to your PeopleSoft ERP systems for execution.

PeopleSoft Portal

The portal bundled with every PeopleSoft 8 application. It provides a simple navigation system, based on existing menu definitions that have been imported into the portal registry. Navigation to content outside of PeopleSoft applications is not provided.

Percent Cycles Without a Shortage

In PeopleSoft Inventory Planning, a method used with safety stock policies. The value is derived from the percentage of replenishment cycles that will complete without a stockout.

Percent Demand Fill

In PeopleSoft Inventory Planning, a method that can be used with safety stock policies. This method defines the percentage of the total quantity ordered that must be filled without a backorder.

Percentage Tolerance

The acceptable percentage difference between expected cycle count quantities and actual quantities counted in PeopleSoft Inventory. This value provides a margin of error for an item during cycle count reconciliation count quantities.

Performance Appraisal Code

Indicates the level of performance of an employee.

Performance Appraisal Due Date

Date established based on the WGI or LEI for the yearly appraisal of an employee.

Period Closing Offset

In Time and Labor, the closing date beyond which this pay period is not considered current any longer, if the period's closing date is different from its end date. You can enter a positive or negative number of days.

Period Method

In PeopleSoft Inventory Planning, a method used to determine how a single static policy value is to be calculated from time-phased results with static policies.

Period of Interest

The maximum period of time containing the data needed to run all the rules in a rule program (see Batch Processing)

Period Segmentation

When an element (like compensation rate) changes mid-period, requiring all other elements in the process list to be calculated multiple times on either side of the date on which the change took place, period segmentation is used. The system calculates each element more than once, using the components that were effective during the different time slices. The system keeps the results of these calculations separate with the object of creating two gross-to-net result sets.

Periodic Processing

In PeopleSoft Pension Administration, any of several batch processes that a plan administrator must run on a regular basis—for example, consolidation of payroll data.

Personal List

A user-created list of products defined in PeopleSoft eStore, used to quickly populate the shopping cart when creating a new order in either PeopleSoft eStore or Mobile Order Management.

Personal Register (Registre Unique du Personnel)

In France, companies are required to be able to produce, at any given time, a Personal Register. For a given establishment, this report lists current employees and employees who left up to 5 years ago.

Personnel Action

Personnel actions are changes to employee data or status resulting from such activities as promotions, transfers, terminations, salary increases, and leaves of absence.

Personnel Representatives (Délégés du personnel)

In France it is mandatory for companies with more than 11 employees to elect personnel representatives who will represent all of the employees before management.

Perspective

In PeopleSoft Balanced Scorecard, a category for organizing critical success factors and key performance indicators. Usually there are four: financial, customer, internal process, learning and growth.

PF Ledger

The PF Ledger (PF_LEDGER_F00) is an important fact table within the PeopleSoft Enterprise Warehouse. The primary function of the PF Ledger table is to support PeopleSoft Enterprise Performance Management reporting. The data that gets posted to the PF Ledger must be accurate and clean.

PF Business Unit

PF Business Units differ from other PeopleSoft Business Units in that they represent functional or strategic areas of an organization, rather than separate legal entities.

Phase Type

A label for the different phases you want to define for a project. Costs can then be calculated by project phases. Examples include planning, startup, construction, and cleanup.

Physical Accounting

The PeopleSoft Inventory feature that updates tables based on count result input, regardless of how the count was created or the data collected.

Physical Inventory Process

In PeopleSoft Asset Management, the process by which you extract asset data from the Asset Management database to load into your bar code scanning device. You then scan the assets and load the data gathered during the actual physical inventory into Asset Management, enabling you to generate physical inventory results for review. You perform matching and generate transactions to reconcile the data in Asset Management with the results of your physical inventory.

PIA

PeopleSoft Internet Architecture. This is the fundamental architecture on which PeopleSoft 8 applications are constructed, consisting of an RDBMS, an application server, a web server, and a browser.

Piece Work

Method of compensating time reporters based on units completed rather than hours worked

PIN

Technical term for an element. In PeopleSoft Global Payroll, PIN is often referred to in the online object names and within the batch code. PIN stands for Pay Item Name.

Placeholder

A temporary location designator in an engineering bill of material for a component item that has yet to exist. These temporary placeholders have to change into approved items before transferring engineering bills of material (EBOM) to manufacturing bills of material (MBOM).

Plan Administrator

The person selected by the employer to perform the administration of a plan under PeopleSoft Pension Administration.

Plan Eligibility

The PeopleSoft Pension Administration function that uses job data to determine whether an employee may participate in a pension plan. An employee can be eligible based on job data but not be participating because of an unmet service or age requirement.

Plan Type

A unique ranges of codes used during payroll calculation to determine deduction processing rules. See also Benefit Plan Type.

Plan Year

The annual period that a pension plan uses to measure service, earnings, and benefits. Generally, the pension plan year will match the fiscal year of the plan sponsor.

Planning Item

A non-inventory item designated as being used for planning purposes only. It can represent a family or group of inventory items. It can have a planning bill of material or planning routing, and can exist as a component on a planning bill of material. A planning item cannot be specified on a production or engineering BOM or routing, nor used as a component in production. Quantity on hand will never be maintained.

Planning Level

The level on a dimension's tree used for planning. Typically, a customer will choose not to plan at the lowest level of available detail, such as the individual product level. Instead, the individual products are mapped to their corresponding product group and the planning is done at the product group level.

Planning Targets

The amount the budget must equal, such as a budget spending limit or cap for expenses where users can not exceed the amount. Planning Targets are presented at a summary level. This term is interchangeable with Spending Limits.

Planning Target Tolerance

The percentage and/or amount a user can be over or under the planned budget target.

Planning Target Control

Values set at the user role level, including no control, active control, and passive control. For more information see no control, active control, and passive control.

Planning Task

Any activity in PeopleSoft Planning that creates a schedule.

PODS (PeopleSoft Operations Data Stores)

See Data Warehouse Tables

POI. Personnel Office Identifier.

Also known as Submitting Office Number (SON). These are codes assigned by the OPM to the office(s) delegated authority within an agency to process personnel actions on Federal employees.

Pointers

A pointer is an "address" of a driver quantity, or value, within the Enterprise Performance Management product line. Pointers are used as a means of defining where driver quantities exist in tables that reside in the PeopleSoft Enterprise Warehouse. Pointers enable you to extract values from any location in the warehouse and then use these values as driver quantities. There are three different kinds of pointers: explicit, implicit, and multidimensional.

Policy Control Group

In PeopleSoft Inventory Planning, a feature for setting up order quantity, safety stock, reorder quantity, and maximum and minimum policies. The **Control Group** is assigned to a set of **Planning Item**. The policies of the associated planning items can be set explicitly or defaulted from the policies on the policy control group.

Policy Generation

In PeopleSoft Inventory Planning, a set of run options used to control the functions and behavior of the Policy Generation program.

Policy Item

An item record which is related to a location and for which **Inventory Policy** is held. A policy set, **Planning Item** ID, and Location ID uniquely identify a policy item. The combination of an item and a location is called a stockkeeping unit.

Policy Set

Defines a set of the items for which **Inventory Policy** is to be calculated. Each policy set is assigned a unique ID and includes information that defines, for example, the associated **Forecast View**, time periods, and planning horizon.

Policy Simulation

In PeopleSoft Inventory Planning, a feature that simulates the effects of various stocking scenarios, compares current policy with simulated policy, and determines the best inventory investment strategy.

Population

A Population is defined on top of DataMaps created using Enterprise Warehouse metadata. The Population builder allows you to easily format a SQL statement, using filters, to select

rows from one or more tables joined together in a DataMap. You can view the results of a Population directly from the browser.

Pop Up Payment Option

A variation on a joint and survivor payment option under which the benefit payable to the participant is increased if the beneficiary should die prior to the participant.

Portal

A portal is a web site that helps you navigate to other web-based applications and content. Users often consider a portal their “entry point”—the place they typically visit first after launching their web browser.

Portal Registry

The portal registry is a tree-like structure in which content references are organized, classified, and registered. It is a central repository that defines both the structure and content of a portal through a hierarchical, tree-like structure of “folders” useful for organizing and securing content references.

Portal Registry API

The Registry API is provided for accessing each portal registry from PeopleCode, COM, Java, or C programs. Providing the same kind of registry management capability as the online administration pages, it can be used by external systems to update the registry to reflect changes in the content reference URL, taxonomy, and effective dates. The Registry API is fully described in the PeopleCode documentation.

Portal Servlet

A Java servlet that runs on a web server. The portal servlet intercepts user requests for content, retrieves content, and builds a single HTML document to be displayed in the user’s browser.

Portal Solutions

Portal Solutions are separate product offerings from PeopleSoft that consist of pre-built, packaged solutions focused at different audiences (customers, suppliers, and employees). Because they are both pre-built, supported application products, Portal Solutions can be deployed swiftly and easily, saving significant resources when compared to other custom-built solutions.

Position

The officially assigned duties and responsibilities that make up the work performed by an employee. Positions are linked to Job Codes, which can be considered the electronic version of the Position Description. There can be a many-to-one relationship between the Position and Job Code.

Position Budgeting

The budget amounts (salary, benefits, and earnings) associated with positions within an organization. Position budgeting can be calculated based on position information loaded from a human resource system. Position budgets are used to generate line-item budgets for personnel costs. PeopleSoft Budgeting-specific.

Position Change

A move by an employee to another position during the employee's continuous service under the same appointment within the same agency.

Position Date Created

Date the position was created for use in the agency.

Position Description (PD)

In accordance with OPM guidelines, an official description, authorized and approved by an agency official, describing duties and responsibilities to be performed. Position classification standards are used to describe the work, classify the work components by occupational series, and factors (e.g. supervisory control, scope, complexity, competencies required) are used to determine the grade level (i.e., salary range) for the position.

Position Description Number

A number assigned to identify various types of Position Descriptions.

Position Description Required

Identifies those positions for which a position description must be maintained.

Position Number

A number that identifies an authorized Position.

Positive Input

Data such as hours worked or a bonus amount entered for elements that change each pay period. Positive input can be entered manually, generated by the system, or received from other applications.

Positive Task Reporting

A method of time reporting in which all required task elements must be provided (see Time Reporting)

Positive Time Reporting

A method of time reporting in which all elements of time must be provided (see Time Reporting)

Post Differential Percent

Additional compensation that may be paid to certain employees who work in Guam or the Northern Mariana Islands.

Post Differential, Non-Foreign

A differential payable to an employee at a location in a non-foreign area if conditions of environment differ substantially from conditions of environment in the contiguous United States and warrant its payment as a recruitment incentive.

Post-56 Military Deposit

The OPM provides guidelines to Federal agencies on how to calculate and process these voluntary employee deductions from pay toward the employee's current retirement fund for those periods of eligible military service.

Posting

In PeopleSoft the process by which accounting entries are created or updated based on user transaction input and accounting entry templates. In PeopleSoft Receivables, posting is also known as Receivable Update.

These posted accounting entries in the feeder systems, such as accounts receivable or accounts payable, must be further processed by the Journal Generator to create journal entries before posting to the General Ledger occurs.

Post-Tax Deductions

See After-Tax Deductions

Pre-encumbrance

An encumbrance that occurs before an employee/employer relationship exists. You encumber funds for an employee you have on staff; you pre-encumber funds for an employee that you anticipate hiring. For example, you would pre-encumber funds for a new position that has just been approved but not filled.

Pre-encumbrance Ledger

Stores pre-encumbrance amounts. Updated by posting pre-encumbrances, such as purchase- or hiring-requisition source transactions (including journal entries).

Premium

Any additional compensation for extra hours worked, often expressed in terms of factor-above-normal-per-hour pay, such as time and a half (where one-half is premium pay), double time or triple time. Also, any additional pay provided to a time reporter based on compensation rules (see Time Administration in your *PeopleSoft Time and Labor* PeopleBook).

Prenote

A prenotification or waiting period requested by banks before processing payroll direct deposits.

Pre-Retirement Survivor Annuity (PRSA)

A benefit paid to a beneficiary if a pension plan participant dies before commencing benefits. Qualified plans must offer a pre-retirement survivor annuity, although the employee can be required to pay for the coverage with a reduction in the benefit.

Pre-Tax Deductions

See Before-Tax Deductions

Previous Day's Close

The previous trading day's last reported trade.

Price Break

Defined in PeopleSoft Order Management, and linked with price sets, a price break defines the actual adjustments that are made to the list price. They are valid only within a time frame you establish.

Price Rule

Used in conjunction with price sets, rules are essentially a decision tree that establishes the search order the system uses in reviewing sets for a match on the variables they reference for price adjustments.

Price Set

Linked with price breaks, a price set specifies the parameters for your price adjustment. It establishes selection criteria, determines whether the break is based on quantity or price, and defines how the adjustment is applied.

Price Source

A service provider or publication that reports the trading activity for a stock traded on stock exchanges. Examples include Wall Street Journal and Bloomberg.

Primary event code

Primary event codes, also called purpose codes, specify the status of the transaction: whether it's a new transaction, a cancellation, a duplicate, a status request, and so on. Every transaction has a primary event code assigned to it.

Primary Insurance Amount (PIA)

The benefit amount calculated under the Social Security benefit formula.

Primary scroll record

Primary scroll records are the principal SQL table or view associated with a page scroll level. A primary scroll record uniquely identifies a scroll level in the context of its page: each scroll level can have only one primary scroll record; and the same primary scroll record cannot occur on more than one scroll at the same level of the page.

Prior Period

In Time and Labor, any payroll period before the current one.

Prior Period Adjustment

A change or correction to previously reported time or task information, or an insertion of time or task information. Often requires the original report to be offset (reversed) and the correct information to be recorded. (see Time Reporting)

Priority Rank

The numeric value assigned to inventory **Demand Priority Rules**. The lower the number, the higher priority of the rule and the orders matching that rule.

Private

A tracking method used by a privately held company to track their daily prices. The Board of Directors typically establishes a price for a period of time. Stock of a privately held company is not traded on an exchange.

Private Views

User-defined views available only to the user who created them. For more information, *see* Budget Views.

Process

See Batch Processes.

Process Definition

Process Definitions are created in the Process Scheduler Manager pages to define each specific run request. A Process Definition is comprised of a variety of variables including panels associated with a request, Process Groups, output parameters, run locations, and many more.

Process Group

Used to associate specific Process Definitions with a Class Profile in Security Administrator. This allows administrators to restrict an operator's ability to initiate requests.

Process Instance

A unique number that identifies each process request. This value is automatically incremented and assigned to each requested process when the process is submitted to run.

Process Job

Multiple Process Definitions can be logically linked into a job request to process each request serially or in parallel, and optionally initiate subsequent processes based on the return code from each prior request.

Process List

The set of instructions the system uses during a payroll process to determine which elements to resolve. A process list is comprised of sections that identify the sets of elements to be resolved. You build process lists and attach them to calendars.

Process List Manager

The program used during batch processing that reads the Process List and calls the PIN Manager to resolve elements on the list.

Process Request

A single "run request," such as an SQR, a COBOL program, or a Crystal report that you run through Process Scheduler.

Process Run Control

A PeopleTools variable used to retain Process Scheduler-defined values needed at runtime for all requests referencing a run control ID. This is not to be confused with application run controls, which may be defined with the same run control ID, but only contain information specific to a given application process request.

Process Scheduler

A PeopleTool that performs tasks behind the scenes of your application. It can run several kinds of processes, such as COBOL, SQR, and Application Engine programs. You can schedule processes to run on a regular schedule or at your request. Processes can run on your workstation or on a server.

Process Scheduler Server Agent

The server-based program (PTPUPRCS) that manages the selection, validation, and initiation of all queued requests for each defined server within your batch environment (Process Scheduler).

Process Type

A global process definition under which related process definitions are grouped. This allows for easy maintenance of Process Definitions that share common parameters.

Processing group

In order to partition application processing between the client and the application server, it is necessary to define units that, as a whole, run in one location or the other. We call these units processing groups.

Processing groups can encompass one or more PeopleCode events. Some processing groups can run either on the client or on the application server, such as Component Build and Component Save.

Product

A commodity primarily defined in PeopleSoft Order Management. It may be: 1) The Order Management view of an inventory item that has attributes the same as or different from those of its inventory counterpart. 2) A commodity that is not a stocked inventory item such as a product kit or a service. 3) A tangible commodity that is drop shipped from another vendor and is never stocked in inventory.

Product Alternate

Alternative products that can replace the product ordered when it's out of stock or a problem with a particular product temporarily prevents shipment.

Product Catalog

A configurable list of available products that may be of interest to a specific customer. You can create two types of catalogs—inclusive catalogs that contain all the products you want made available to a customer and exclusive catalogs that contain the products you want to withhold from your customer. By attaching product catalogs to a Sold To customer, you define the products they can buy.

Product Definitions

This support module describes how other support modules process the instruments that belong to a particular product. For example, different products have different cash flow characteristics and may be stratified differently, or may react differently to changes in interest rates. This module enables you to specify each of these actions for each instrument.

Product Kit

A commodity that consists of a fixed set of components that are sold together. It appears as a single line on an order, but is represented by multiple lines on shipping documents. Product kits may comprise inventory items, non-stockable products, or a combination of both.

Product Kit Component

A commodity that is part of a product kit. It may be an inventory item or a non-stockable product such as a service.

Product Pricing Model

In the financial services industry, this defines models that describe indices upon which future rates are paid or charged for an individual product.

Product Tree

A user-defined graphical representation of a company's product structure. A product tree defines how products are promoted and determines what users have authority to promote those products.

Production Maintenance Spreadsheets

A set of spreadsheets generated by an nVision process, containing production ID and production schedule data extracted from PeopleSoft Production Management. You can add and maintain production quantity data using these spreadsheets and then import the data back to Production Management.

Production Option

Effective-dated combinations of BOM codes and routing codes. You can create multiple effective-dated BOM code/routing code combinations (or production options) for an item. These combinations can be extracted to PeopleSoft Production Planning. They enable the specification of multiple production variations for an item and provide control of seasonal variations by effective date.

Production Option Cost

A cost based on a specific BOM/routing combination (also known as a production option). The Production Option Cost utility rolls up production options costs based on specific BOM/routing combinations, enabling you to cost individual production options and later to have the DataLink pass this cost to the Planning engine. If this utility isn't run, the Planning engine will use the standard item cost based solely on the primary BOM and routing instead.

Productive Time

Employee scheduled time spent performing any task for which a position was created; work performed on behalf of a business entity that is required for that entity to fulfill its business purpose. Employees doing the work they or someone else was hired to do.

Productive Unit

In Italy employers organize employees into productive units based on agreements between the unions and the employer.

Profile

A data set that you aggregate from the Enterprise Warehouse, according to the filters you specify, the Key Performance Indicators you select, and the 3rd party demographic data you include.

Profile Factor

In PeopleSoft Demand Planning, the weight index assigned to each **Forecast Period** to take into account seasonal fluctuations in demand. The factor or index typically measures the percentage of difference between the base demand and the expected actual demand in the period.

Profiles

Group of employees defined according to a list of job codes and departments. You can use these profiles to ascertain training demands within your organization based on set criteria.

Profit Manager

The Profit Manager is a set of integrated tools that enable true profitability reporting. Profit Manager features are tightly integrated with the PeopleSoft Analytic Applications and provide you with ways to ensure data integrity, edit data, and post data to the Performance Ledger table.

Project

The highest level of hierarchical organization within PeopleSoft Projects. Projects provide the structure to which activities and resources are added. Each node on a Projects tree represents a project. Projects can contain other projects as well as activities and resources. This provides a hierarchical relationship between projects and facilitates cost roll-ups.

In Enterprise Performance Management you use a Project to create or modify a Profile. A Project contains pointers to data elements that you include in a Profile.

Project

In PeopleSoft Time and Labor, a specific endeavor undertaken to achieve a specific goal. Typically, projects are approved and undertaken with level of cost, schedule, and performance already agreed upon. A project is composed of a set of tasks, each of which requires staffing, provisioning, and/or scheduling. Project progress is often measured in terms of task completion.

Project ID

The name or number by which a project is to be identified in all tables and pages.

Project Type

A user-definable grouping of projects. Project types are optional.

Projected Run Date

In PeopleSoft Demand Planning, a calculation made that projects a life volume for a period based on a calculated run rate or performance ratio.

Projection.

An estimated pension benefit calculated as of a future date or any estimated data used as the basis for such a calculation.

Promotion

- For positions under the same type job classification system and pay schedule, a promotion changes the employee to a higher grade level or makes permanent a Promotion NTE;
- When the old and new positions are under different job classification systems and pay schedules, a promotion changes the employee to a position with a higher rate of basic pay or makes permanent a Promotion NTE.

Promotion Pattern

In PeopleSoft Demand Planning, an **Event** function that enables you to apply weights to promotions across a range of **Forecast Period**.

Prompting Profile

A task profile usually used by account managers as a way of creating task profiles for employees who report task time differently by customer. For example, you might have an account manager who has fifty customers; when the account manager comes in each day to report time, the system will display all the customers, and indicate which customers it will use as a default if she doesn't manually report time.

Prorated

In Enterprise Planning and Simulation, prorated is when the computed forecast and the summarized forecast are two different versions of the statistical forecast. In addition, the forecast at the product family level can be allocated down to the individual products. Usually this allocation is done in proportion to the calculated product forecasts at that level. This version of the (statistical) forecast is called the allocated or prorated statistical forecast.

Pro-Rate Purchase

A purchase in which the number of shares to be purchased is prorated according to a specified factor. This may occur when the total number of shares to be purchased is greater than the number of shares allocated to the stock plan from the treasury.

Prorated Forecast

In PeopleSoft Demand Planning, a forecast developed by factoring the group forecast down one level at a time to make the sum of the item forecast equal to the aggregate forecast. The prorated forecast tends to be more accurate than the **Statistical Forecast**.

Proration Rule

Element that defines how you want to prorate an item. You use proration rules in numerous places—for instance you could prorate an earning, deduction, or many of the elements that make up an earning or deduction.

Proration Threshold Ratios

In PeopleSoft Demand Planning, the upper and lower ratios used as thresholds for **Reasonableness** checks when a forecast is developed using proration.

Provider

An entity, such as an insurance company, that provides one or more of the benefits your company offers. For example, Metropolitan Life Insurance Company is a provider to companies that use a Metropolitan life plan.

Proxy Person

A highly compensated executive. Corporations must include information regarding the most highly compensated executive officers in their proxy reporting.

Proxy Statement

The document that must accompany a solicitation of proxy appointment under SEC regulations. The purpose of a proxy statement is to provide shareholders with the appropriate information to make an intelligent decision.

PSADMIN

A PeopleSoft utility providing a menu interface to create, configure and administer application server domains and the Process Scheduler Server Agent (Batch Server).

PSADMIN

A PeopleSoft utility providing a menu interface to create, configure and administer application server domains and the Process Scheduler Server Agent (Batch Server).

PSAPPSRV

PSAPPSRV is the main server process running within a domain. PSSAPPSRV performs the functional requests, such as building and loading components. It also manages the memory and disk-caching for PeopleTools objects on the application server. Each PSAPPSRV process maintains its own memory and disk cache.

It provides authentication services for incoming users. For instance, it checks the PeopleSoft OPRID against the directory server or PSOPRDEFN table.

PSQCKSRV

Essentially, PSQCKSRV is a copy of the PSAPPSRV. It performs quick, read-only SQL requests. It is an optional Server Process designed to improve performance by handling items in the PSAPPSRV transaction request queue.

PSQRYSRV

Like the PSQCKSRV server process, PSQRYSRV is designed to alleviate the workload of PSAPPSRV. PSQRYSRV is designed to specifically handle all user-generated queries submitted by PeopleSoft Query (PSQED.EXE). This server process is designed to improve overall application server performance whether or not you have PSQCKSRV configured. It is specifically, and exclusively designed to process PeopleSoft Query transactions, which can be very SQL intensive.

PSSAMSRV

It processes conversational SQL transactions primarily for Application Designer.

Public Company

A company that has held an initial public offering and whose shares are traded on a stock exchange or in the over-the-counter market. Public companies are subject to periodic filing and other obligations under the federal securities laws.

Public Views

Coordinator-defined views, available to anyone using the application. For more information, *see* Budget Views.

Publish/Subscribe

Publish/Subscribe type messaging is performed with PeopleTools Application Messaging technology. You can send data from one PeopleSoft system to another in an asynchronous mode—meaning the two systems don't have to be sending and receiving at the same time. This is possible because the message transfer is accomplished through a Web server with an "http: gateway."

Pull List

Similar to a pick list, a pull list contains multiple replenishment requests, including the location, quantity, and item quantity required in a specific sorting sequence. You use pull lists in PeopleSoft Flow Production with Inventory replenishment.

Pull Ticket

A document containing the details of a single request replenishment request, including Kanban ID, item, quantity, source, and To locations. You use pull tickets in PeopleSoft Flow Production with Inventory replenishment.

Punch

Precise instances of date and time recorded for a user and measured in seconds, minutes, hours, day, month and year and time zone (see Time Reporting)

Punch Duration

Length of time between two punches in increments of hours or partial hours (see Time Reporting)

Punch Matching

Area of the application which converts paired punches to punch duration by processing rounding rules and assigning the tasks to the appropriate logical day based on rules established by the user

Punch Restriction

The facility to constrain a time reporter's ability to create a punch that deviates from the schedule (see Time Reporting)

Punch Type

A user defined classification of punches, i.e. In, Out, Start, Stop (see Time Reporting)

Purchase

The issuance or purchase of shares through a stock purchase program. The purchase is made using current contributions from a participant and any carry-forward remaining for the participant from previous purchases.

Purchase Price

The discounted price paid for the shares at the end of a purchase period.

Purchase Price Variance

A PeopleSoft Payables matching feature that compares purchase order and inventory prices for any variance in the prices.

Purge Rules

The rules that define criteria to clear data you no longer need from previous open enrollment processing cycles in PeopleSoft Benefits Administration.

Pyramiding

A computer calculation enabling an individual owner of one share of stock to use the stock-swap technique to exercise a stock option of any size without using cash. Not many corporations permit pyramiding.

Q

QDRO

See Qualified Domestic Relations Order.

QDRO Alternate Payee

A former spouse who is entitled to a portion of a participant's pension benefits as a result of a court order.

QJSA (Qualified Joint and Survivor Annuity)

A post-retirement death benefit for a spouse. Plans subject to this requirement must provide an annuity for the life of the participant with a survivor annuity for the life of the participant's spouse.

QMCSO (Qualified Medical Child Support Order)

A QMCSO is a court order that requires a group health care plan to provide benefits to the child of a participant as part of a child support arrangement on the behalf of that participant. Base Benefits enables the tracking of QMCSOs for dependents.

Qualified Domestic Relations Order (QDRO)

A court order ordering a division of a participant's pension benefits. This is normally the result of a divorce and gives a portion of the pension benefits to the former spouse.

Qualified Plan

A pension plan for which the employer can take tax deductions for contributions to the plan. Investment income of the plan trust fund is not taxable to the employer. Tax law places restrictions on the plan rules.

Qualifying Dispositions

A transaction whereby a participant sells shares acquired through a stock purchase plan two years after the grant date and one year after the purchase date.

Quality Function

A level of configuration that enables you to define the fields and attendant information that provides a base level for inspection plan and integration. Quality functions enable you to map process-specific field information into PeopleSoft Quality for identification, tracking, and analysis.

Quality Ranking Factors

Knowledge, skills, and abilities that could be expected to enhance significantly performance in a position, but are not essential for satisfactory performance. Applicants who possess such

KSAs may be ranked above those who do not, but no one may be rated ineligible solely for failure to possess such KSAs.

Quality Server

A PeopleTools-based analysis and graphing program.

Quality Step Increase (QSI)

A step increase awarded to an employee for sustained high quality performance.

Quantity Allocation Method

In PeopleSoft Inventory, the method used to determine how available quantity will be allocated to prioritized demand lines when using the online reservations page.

Quantity Precision Rules

A set of rules specifying whether item quantities for a given unit of measure are expressed as whole numbers or as decimals. Quantity precision is set at the inventory **SetID** and item-UOM levels.

Query

A set of data members that are selected from a Class catalog (provided by the Business Interlink Plug-in) as well as a generic form of Criteria. The criteria are composed of <left-hand-side> <Relational Operator> <right-hand-side> statements that can be concatenated using a set of logical operators. All operators and class catalogs are dynamically provided through the Business Interlink Plug-in.

R

Race And National Origin Code

A code that identifies the employee's basic racial and national origin category.

Range of Dates Reporting

A Time and Labor process that enables you to report a start and stop date, a time reporting code and task information for a single employee. The system transforms the information into instances of daily time based on the employee's schedule or default work schedule, replacing the scheduled time with the entered Time Reporting code and the number of scheduled hours on a day-to-day basis.

Range Penetration

In PeopleSoft Workforce Analytics, Range Penetration is the degree to which an employee's actual pay has progressed through their salary grade, and is expressed as a percentage. The calculation is:

Range penetration = (Employee Base Pay – Range Minimum)/(Range Maximum – Range Minimum).

Range Width

In PeopleSoft Workforce Analytics, the difference between the maximum and the minimum values of the pay range calculated using the following formula (and expressed as a percentage):

(Maximum – Minimum)/Minimum.

Rapid Time Entry

The process that enables you to enter daily time for single employees without the system editing your field entries. The system populates temporary tables, which are used by a batch process that reads, edits and moves the data into the appropriate time and labor tables. You cannot prompt for valid values in any of the fields, and the online system does not edit any of the data you enter against other tables.

Rate Code

Alphanumeric reference to the cost per hour or unit of time reported to a specific TRC.

Rate Code [Global Payroll]

IDs for pay components. Rate codes define rates of pay and are set up in the Comp Rate Code table. Rate codes are then used to represent pay components in pages and when you configure compensation packages in the compensation record.

Rate Code Group

A rate code group is a number of pay components (represented by rate codes) bundled into a subset of a compensation package. The rate code group is used to calculate percentage-based components that do not apply the percentage to all pay components in the compensation package. Rate code groups are constructed on the Rate Code Groups page.

Rate Code Type

Defines how the monetary value of the rate code is calculated. The compensation rate code type is defined on the Comp Rate Code table. Valid values are Flat Amount, Hourly Rate + Flat Amount, Hourly Rate, Percent, and Points.

Rate Combinations

The combination of rate types and conversion rates with account types that is linked to your budgeting model. Typical rate types are current, commercial, floating, average, and historical. Effective dates define different rates for different budget periods. There are several conversion rates for any pair of currencies including not only the current rate, but others rates such as average, historical, asking, and so on. These different types of rates are appropriate for different types of accounts.

Rates

The arrays of values used to calculate the cost of a plan to an employee. Rates can be age-graded, service-related, or general, depending upon the benefit plan type. Any number of benefit program and benefit plan combinations can use each set of rates.

In Enterprise Performance Management, a rate is determined by the user and specifies the dollar amounts to be calculated for each model. This is a financial services industry term.

Rating Model

The scale used by your company to measure competency proficiency. The default rating model is the PSCM (PeopleSoft Competency Management) Rating Model that PeopleSoft delivers with your PeopleSoft Human Resources System.

Raw Punches

See Actual Punch; typically this is distinguished from a rounded punch (see Time Reporting)

Reason Code

Reason Codes provide explanations for occurrences such as returned stock and changes to order headers, lines, or schedules.

Reason Code

A code describing employee time such as comments for sick time or travel time.

Reasonableness

In PeopleSoft Demand Planning, a technique that checks the trend and projected annual growth to make sure that a forecast is realistic. If a forecast falls outside either boundary, the system automatically adjusts it and sends a warning message.

Reassignment

Change of an employee from one position to another without promotion or change to lower grade.

Recalculate Forecast

In PeopleSoft Demand Planning, a forecasting feature that uses the existing model and its associated parameters to create a new forecast.

Recalculate VAT at Payment

Allows the VAT amount to be adjusted at the time of payment if an early payment discount is taken. This calculation option is only valid when VAT is calculated at Gross.

Receipt Cost Method

Determines how you cost receipts. Receipt cost methods include Actual, Non-Cost, and Standard.

Receipt Line

A line associated with a Receipt ID that identifies an item and quantity. If the respective tracking is activated, the lot, lot suffix, and serial number are also identified.

Receivable Update

See **Posting**

Receivables Item

An individual receivable. An item can be an invoice, credit memo, or debit memo. Items and payments combined comprise a customer's balance.

Reconciliation

Within PeopleSoft Enterprise Performance Management, reconciliation differs slightly when it is performed within the PeopleSoft Enterprise Warehouse and when it is performed within the PeopleSoft Analytic Applications.

In PeopleSoft Funds Transfer Pricing (FTP) and PeopleSoft Risk Weighted Capital (RWC), reconciliation identifies differences between Performance Ledger balances and the instrument or position balances, which are risk weighted according to the basis rules you have assigned. The first step in reconciling basis rule balances is to reconcile the individual balances for accounts, instruments, and positions. Reconciling the total balances is the second step. This means that you reconcile the difference between Account/Instrument balances, and the difference between Account/Position balances.

In the PeopleSoft Enterprise Warehouse, reconciliation is a period-end process that posts journal entries to the Performance Ledger for the discrepancies found when you reconciled the individual balances. Typically, you'll run the PF Reconciliation engine after a period to compare "to and from amounts" between tables such as REVENUE_F00 and PF_LEDGER, or the GL_LEDGER and the PF-LEDGER.

Record Date

The date a stockholder must officially own shares in order to vote at the meeting or to derive an adjustment resulting from a stock split or a stock dividend. The Board of Directors sets the Record Date.

Record Definition

A logical grouping of data elements.

Record field

Once a field is inserted into a record definition it becomes known as a Record Field within the record.

Record Group

A set of logically and functionally related control tables and views. Record groups exist for two basic reasons:

- To save you time—with Record Groups, TableSet sharing can be accomplished quickly and easily, eliminating an enormous amount of redundant data entry
- To act as a safety net—Record Groups ensure that that TableSet sharing is applied consistently across all related tables and views in your system.

Record Input VAT Flag

Within PeopleSoft Purchasing, Payables, and General Ledger, this flag indicates that you are recording input VAT on the transaction. This flag, in conjunction with the Record Output VAT Flag, is used to determine the accounting entries created for a transaction and to determine how a transaction is reported on the VAT return. For all cases within Purchasing and Payables where VAT information is being tracked on a transaction, this flag is always set to Yes. This flag is not used in Order Management, Billing, or Receivables, where it is assumed that you are always recording only output VAT, or in Employee Expenses, where it is assumed that you are always recording only input VAT.

Record Output VAT Flag

For certain transactions within PeopleSoft Purchasing, Payables, and General Ledger, it may be necessary to record both input VAT and output VAT on the same transaction. Generally, this would be a situation where the purchaser was required to self-assess VAT. Within these situations, this flag indicates that you are recording output VAT on the transaction. This flag, in conjunction with the Record Input VAT Flag, is used to determine the accounting entries created for a transaction and to determine how a transaction is reported on the VAT return. This flag is not used in Order Management, Billing, or Receivables, where it is assumed that you are always recording only output VAT, or in Employee Expenses, where it is assumed that you are always recording only input VAT.

Record Owner

The "Stockholder of Record" of the stock. This may be different from the "Beneficial Owner" of the stock.

Record Suites

Record suites are temporary tables that enable the system to track how many processes are running. These temporary tables leave the fact tables accessible for processing other jobs simultaneously without impacting your processing.

Reduction In Force (RIF)

Method used to reduce the number of government workers in an agency.

Reemployed Annuitant

An employee who has retired from Federal employment and is receiving an annuity. His/her salary is reduced by the amount of the annuity.

Reference Designators

A user-defined alphanumeric identifier that determines where a component is placed in an assembly.

Reference Transaction

In People Soft commitment control, a reference transaction is a source transaction that is referenced by a higher-level (and usually later) source transaction, in order to automatically reverse all or part of the referenced transaction's budget-checked amount. This avoids duplicate postings during the sequential entry of the transaction at different commitment levels. For example, the amount of an encumbrance transaction (such as a purchase order) will, when checked and recorded against a budget, cause the system to concurrently reference and relieve all or part of the amount of a corresponding pre-encumbrance transaction, such as a purchase requisition.

Referential Integrity

Issues that occur when an update to an instance of one object invalidates one or more instances in a related object. In other words, when you make a change to one area of the application, referential integrity makes sure the changes do not adversely affect another area of the application.

Refresh Time

The process that retrieves the appropriate current version of objects related to employee time (such as task profiles or work schedules) and associates them with that time.

Region Codes

Regions may or may not be physical entities, they may simply be another way to geographically categorize an area. When a region does represent a physical entity, the region code has the same characteristics as a business, that is, an address and a language spoken.

Register of Separations and Transfers (ROST)

The ROST is a regulatory compliance document used by federal agencies to summarize the information in an employee's Individual Retirement Record (IRR). The ROST is a one-page cover sheet that accompanies a batch of IRRs being submitted to the Office of Personnel Management (OPM) at the time of an employee's separation from a federal agency. Employees covered by the CSRS retirement plan require SF-2807. Employees covered by the FERS retirement plan require SF-3103.

Registration

The name or names that appear on the stock certificate to indicate who owns the stock.

Registration Statement

The document that must be filed to permit registration of an issue of securities under the Securities Act of 1933. A major component of the registration statement is the prospectus that is to be supplied to prospective purchasers of securities.

Regression Analysis

A statistical technique that determines the relationship between two or more variables. Regression predicts the value of one variable (the dependent variable) based on one or more independent variables.

Regular Base Compensation

In PeopleSoft Workforce Analytics, the annualized, quoted, compensation rate for a job. Consists of fixed compensation, does not include variable compensation.

Regular Time

An employee's normal (scheduled/shift) work hours.

Regular Time

In PeopleSoft Workforce Analytics, an employee's normal (scheduled/shift) work hours.

Regulation T

Federal Reserve Board regulations governing the extension of credit by brokers or dealers, including their participation in same-day sale transactions and sell to cover exercise.

Regulatory Region

The Regulatory Region functionality in PeopleSoft HRMS is designed for use in performing regulatory and regional edits. You'll use Regulatory Region to drive PeopleCode edits, perform set processing, and control what codes and values the operator sees. A Regulatory Region can be any country (or province or state) where there are specific laws and regulations addressed by functionality in PeopleSoft HRMS.

In Enterprise Performance Management, a Regulatory Region is a region with a common regulatory framework; such as a country (CAN for Canada), or a smaller state or provincial entity (CANBC for British Columbia).

Related Education

Education above the high school level that has equipped the applicant with the KSAs to perform successfully the duties of the position being filled. Education may relate to the duties of a specific position or to the occupation, but must be appropriate for the position being filled.

Release

An industry standard term associated with the lifting of a company's Repurchase Option from a portion or all shares from a Restricted Stock Award (RSA). RSA's are subject to release schedules, similar to vesting schedules.

Relevant Constraint

A constraint PeopleSoft Planning considers when it calculates a score and when it optimizes the schedule. See also **Scorecard** and **Optimize**.

Reloads

Some stock option plans provide for the grant of a "reload" stock option in connection with stock option exercises, typically by means of stock swaps. A reload option feature provides that upon a stock exercise, the employee will receive an automatic grant of a new stock option at the then-current fair market for the shares that they exercised or for the shares that they used to swap.

Remark Codes

Codes that cause the printing of pre-set text passages on a notice of action form. Some passages are general purpose and others are specific to the personnel action being processed.

Remit From Customer

A customer who is responsible for payments billed to other customers. During cash application, it's useful to look at open items for the Remit From group.

Remittance Worksheet

A work space in PeopleSoft Receivables used to select drafts for remittance to the bank.

Reorder Point

The identifier that automatically locates a replenishment need for an inventory item. When the physical quantity in a location falls below the reorder point, a replenishment request can be created.

Reorder Point Policy

In PeopleSoft Inventory Planning, a policy that determines when a replenishment order is launched for an item. The policy has several methods that include days supply, lead time demand, and **Fixed Quantity**.

Replacement Option

The "new" "replacement" stock option that will replace the original stock option. This option will have a grant price lower than the original stock option.

Replenish

A process that indicates when items need to be resupplied from external sources. In PeopleSoft Inventory, the process can occur on an ad hoc basis or at predefined reorder points.

Replenishment Request

In PeopleSoft Flow Production, an online request for material made when the material is needed. You can generate replenishment requests manually or automatically using backflushing. You can communicate that request using pull lists, pull tickets, or Workflow notifications.

Report Scope

A feature that creates multiple instances of an nVision report using a single report request. Each instance contains data specific to an individual ChartField, such as a business unit or department, or for a group of ChartFields, such as all sales departments. Using Scope, each report instance can share the same layout while containing data unique to these particular ChartFields.

Reported Time

Clock time or elapsed time provided to the system by the user (see Time Reporting)

Reporting Person

An insider that is regularly considered by the SEC to have material information and policy-making authority for the corporation. These individuals are subject to the reporting requirements promulgated by Section 16 of the Securities Exchange Act of 1934. Reporting Persons typically include Directors, Officers, and shareholders with 10% holding interest in the equity of the registrant's securities.

Repricing

An agreement between the corporation and the optionee that allows the optionee to cancel an outstanding high-priced, usually "Out-of-the-Money" stock options for lower-priced options.

Repricing Election

Eligible optionees can choose (elect) to accept the corporation's repricing offer or choose to decline the offer.

Repurchase

The reacquisition of shares of stock from an individual by a corporation. This usually occurs when an individual fails to meet the vesting requirements on a RSA or option that is exercised before it vested. The corporation might pay the original cost of the shares to the individual or the fair market value of the shares at the time of repurchase.

Repurchase Option

An irrevocable, exclusive option to repurchase up to the number of shares that constitute Unreleased Shares at the original purchase price per share. The Company shall exercise said option. The repurchase of outstanding shares is regulated under the laws of all states (except Massachusetts). Under some laws, as under the Model Business Corporation Act, the repurchase is prohibited unless the corporation remains solvent, in both the equitable and bankruptcy senses of insolvency and after taking any liquidation preferences of other outstanding stock into account.

Repurchase Right

A company's contractual right to buy back from an employee any stock resulting from the exercise of the option. The buy back can be at fair market value, book value, or the original purchase price.

Reservation Method

The method used to reserve soft reservation items — either batch COBOL reservations or on-line allocation and reservation.

Reserved

A flag indicating that the inventory item is reserved for stock fulfillment in the inventory business unit.

Reserved Orders

Orders that have been reserved against on-hand available quantity at the business unit-item level. Reserved orders are found in the DEMAND_INV table.

Reset

In PeopleSoft Demand Planning, a function of the **Forecast Calculation Process** that determines which forecast model will produce the best forecast, meaning the model with the lowest ratio of error.

Resolution

An activity that closes or partially closes a deduction, such as matching it to a deduction authorization, writing it off, or sending it back to PeopleSoft Receivables.

Resolution Entry Type

Code that identifies how to process activities for items in PeopleSoft Deduction Management and how to create accounting entries.

Resolution Method

A set of rules that defines how to automatically match or write-off deductions in PeopleSoft Deduction Management.

Resolution Worksheet

The workspace in which deduction items are paired with offset items and resolved or written off in PeopleSoft Deduction Management.

Resource

In PeopleSoft Manufacturing, any crews, machines, and tools that can optionally be used at work centers to complete tasks. In PeopleSoft Performance Measurement, any homogeneous grouping of general ledger line items.

Resources

Resources are the economic elements that are required to perform the activities associated with your business. Resources are consumed in the performance of these activities, and thus denote operating costs. In PeopleSoft Activity-Based Management, resources are typically regarded as the groupings of one or more general ledger accounts. In a service business, resources might include salaries, office rentals, and costs of capital such as information systems, depreciation, real estate taxes, and other associated costs.

Resource Amount

The monetary amount of a single, specific resource transaction. The Resource Amount maps to the Posted Total Amount when posted to the general ledger.

Resource Category

A field for defining individual resource types more specifically. For example, if you have a resource type of labor but want to break it down further for tracking purposes, you might define resource categories of architect hours, carpenter hours, plumber hours, and electrician hours. Resource categories are optional.

Resource Driver

In Activity-Based Management, a Resource Driver defines the quantity of resources used by an activity.

Resource Group

A category of resource types. You can define relationships between the resource types within a resource group to facilitate analysis of project costs. For example, if you had resource types for standard labor and overtime labor, you could group them together in a resource group to calculate total labor.

Resource Planning

In PeopleSoft Activity-Based Management, Resource Planning focuses on resources allocations that create expected results like driver rates and cost object costs.

Resource Quantity

A field on each resource transaction that identifies nonmonetary quantity. For example, on a resource line for 12 ball valves the quantity would be 12.

Resource Source

A field on each resource transaction that identifies the system in which the cost originated. For example, PeopleSoft Payables would be the resource source for a resource transaction created from a voucher in that system.

Resource Subcategory

A field for defining individual resource types and categories more specifically. For example, if you have a resource type of labor, and resource categories of architect hours, carpenter hours, and plumber hours, you might want resource subcategories of regular hours and overtime hours. Resource subcategories are optional.

Resource Supplied

An attribute that enables you define a resource as committed or flexible. A committed resource is one that will not likely change in the short term. A flexible resource is more likely to change within the short term.

Resource Transaction

An individual cost line within PeopleSoft Projects. It is through resource transactions that individual costs and types of costs are tracked. Each resource transaction contains a cost and a quantity and as many identifiers of that cost as necessary. Resource transactions are created when you receive information from other systems, run allocations with project resources as the target, or perform internal transactions such as billing, project closure, or adjustments.

Resource Type

The resource transaction field in PeopleSoft Projects that identifies the resource associated with a given cost. Resource types may be very general or very specific depending on your needs; they are used in conjunction with resource categories, resource subcategories, and resource groups.

Resource Use

Resource Use defines the behavior of a resource within PeopleSoft Enterprise Performance Management. An intermediate resource is a grouping of general ledger line items that may be allocated to another intermediate resource or to a primary resource.

Restricted Punch

A punch which is not accepted because it occurs outside of the predefined number of hours and minutes before or after a scheduled (Understanding Time Collecting Device)

Restricted Securities

Securities issued privately by the company, without the benefit of a registration statement. Restricted securities are subject to a holding period before they can be sold under Rule 144.

Restricted Stock Awards (RSA)

An award of shares of stock to an individual, typically granted at the par value or for no consideration. The shares are awarded on the basis of some future performance goal, either the passage of time (vesting) or the attainment of a specific goal. When the goal is achieved, the vesting occurs. The individual, typically, has all other shareholder rights over these shares such as, voting and dividend rights. The shares are issued in the name of the individual at the time of the award and are held in escrow until vesting occurs. If an employee terminates prior to the vesting of the shares then the company normally repurchases the unvested shares.

Retained Grade Effective Date

Date employee became eligible or began receiving a retained grade and pay.

Retained Grade Expiration Date

Expiration date of an employee's retained grade and pay.

Retest Date

In PeopleSoft Inventory, the date a lot should be inspected to determine whether it is still acceptable for fulfillment or consumption. (Retest Date = Creation Date + Retest Lead Time)

Retirement

Types of retirement are:

- Mandatory Retirement.
- Disability Retirement.
- Voluntary Retirement.
- Special Option Retirement.
- ILIA (In Lieu of Involuntary Action) Retirement.

Retirement Coverage Code

A code used to denote an employee's retirement coverage. The major ones include the following:

- Civil Service (CSRS)
- Federal Employees Retirement System (FERS) and FICA
- Foreign Service (FS)

- CSRS Offset
- CSRS - Special (for LEOs)
- FERS and FICA - Special (for LEOs)
- Social Security System
- None

Retroactive Benefits/Deductions

Deductions taken or benefits granted due to a recalculation of previous benefits and deductions. Late or modified union contracts, late paperwork, and delays in benefit enrollment processing may all result in a need for benefit/deduction recalculation.

Return Type Code

A designator on returned material authorizations (RMAs) that indicates what actions the return initiates. This may include replacement of the product or the creation of a credit memo in PeopleSoft Billing.

Reverse Split

A reduction in the number of outstanding shares of a corporation's stock, with a corresponding increase in the stock's value.

Reversionary Annuity

A form of pension payment where the retiree foregoes all benefit during his or her lifetime so that the entire benefit is paid as an annuity to a beneficiary after the retiree's death. If the beneficiary predeceases the retiree, the benefit is forfeited.

RIDDOR (Reporting of Injuries, Diseases, and Dangerous Occurrences Regulations)

Health and safety regulations in the United Kingdom requiring employers to report certain types of health and safety incidents to the Health and Safety Executive (HSE).

Rider

A special court-ordered or regulatory provision that may be applied to an enrollment to expand or limit any dependent or beneficiary coverage.

Risk Function

In Risk Weighted Capital, this is a user-defined formula that the system uses to derive risk weights.

Risk RuleSet

Used to assign a number of rules to a basis, for processing by the PeopleSoft Risk Weighted Capital Application. Used to group together a number of rules that frequently apply to the same type of balance.

Risk Type

In Risk Weighted Capital, this defines the types of risk associated with your business or activity. For example catastrophic, credit, legal, operational, regulatory, foreign exchange, market, interest rate.

Risk Weight

In Risk Weighted Capital, the risk weight is assigned by risk type, and is used to calculate capital allocation or normalized loss for the account or activity.

Risk Weighted Capital (RWC)

See PeopleSoft Risk Weighted Capital

RIZIV Code

This code is for Belgian employers to track the Federal Institute for Illness and Disability Insurance category.

ROE (Record of Employment) Reason Codes

ROE codes are defined by the Canadian government for employers to record employment actions such as Return to School or Pregnancy Leave.

ROLAP (Relational Online Analytical Processing)

ROLAP refers to the analytical processing and analysis of a relational Data Mart cube. ROLAP, is a form of OLAP that leverages the power and flexibility of relational databases.

Role

A role consists of a designated set of tasks, competencies and accomplishments required for a job code or a position.

Role user

A PeopleSoft Workflow user. A person's Role User ID serves much the same purpose as their Operator ID does in other parts of the system. It allows the system to uniquely identify the user and to determine what data the user has access to.

PeopleSoft Workflow uses Role User IDs rather than Operator IDs because it needs different kinds of user information than the rest of the system does. Specifically, it needs to know how to route work items to the user---an email address, for example---and what role the user plays in the workflow. Plus, you can include role users in your workflow who aren't PeopleSoft application users and who don't have Operator IDs.

Roles

Roles describe how people fit into the workflow. A role is a class of users who perform the same type of work, such as clerks or managers. Your business rules typically specify what user role needs to do an activity.

Roll Forward

In commitment control, rolling budget balances forward from the budget ledger you are closing (the source budget ledger) into the new (target) budget ledger.

In PeopleSoft Enterprise Planning and Production Planning, a utility that moves tasks from the past to a valid point in the future using time periods rather than fixed dates.

Roll Up

The act of totaling sums based on information tree hierarchies. You can roll up data for any group of details that you have defined as dependent with the Tree Manager.

Rolling Plan

An ESPP offering period where the purchase date is measured from the offer start date. If at the purchase date, the current stock price is lower than the last stock purchase price, you may elect to reset your employees to the new lower purchase price. The offering period is now based of the new purchase date.

Rollup

In PeopleSoft Demand Planning, the process of adding up the demand and forecast **Array** from one level to the next from child to parent. Information such as caption, description, and unit of measure can also be rolled up. The process is also referred to as summarization.

Rounded Punch

A punch that has a company's rounding requirements applied to it (see Time Administration)

Rounding Rule

Defines a rounding rule. You use rounding rules in numerous places—for instance you could round an earning, deduction, or many of the elements that make up an earning or deduction.

Routing (Manufacturing and Engineering)

A set of information detailing the method to manufacture a particular item. It consists of sequentially numbered operations that reference the task to be performed, the work center in which the task is to be performed, the resources to be used, and the time required to complete the task. Engineering Routings differ from Manufacturing Routings in that they are not visible within Production Planning or Production Management and are isolated from Manufacturing.

Routing Option

In PeopleSoft Planning, a valid method for replenishing supply for an item. There are two types of routing options: build options and purchase options. An item may have more than one routing option.

Routing Transit Number (RTN)

A number that identifies the financial institution to which an electronic payment should be sent for deposit.

Routings

Routings connect the activities in the workflow. They are the system's means of moving information from one place to another, from one step to the next. Routings specify where the information goes and what form it takes—email message, electronic form, or worklist entry.

RSZ (Rijksdienst Sociale Zekerheid) Category Codes

These government defined Social Insurance category codes are used to maintain social security records for your Belgian employees. RSZ Categories are associated with a Contract Type, Statute and Substitute for Claey's Formula calculations.

Rule

Representation of a company's compensation, task allocation, or exception requirements (see Creating Rule)

Rule 10b-5

A SEC rule that prohibits trading by insiders on material non-public information. This is also the rule under which a company may be sued for false or misleading disclosure.

Rule 144

A SEC rule that applies to public re-sales of restricted securities as well as all sales by affiliates. The requirements include (1) current public information about the issuer, (2) a one-year holding period for "Restricted Securities," (3) unsolicited brokers' transactions, (4) an amount limitation. the greater of 1% of the outstanding stock or the average weekly trading volume may be sold during any three-month period, and (5) a Form 144 filing.

Rule Actions

Functions that can be used in the creation and application of a rule (see Time Administration)

Rule Elements

Customer defined pieces of information which are passed to Time Administration in order to apply and evaluate rules (see Time Administration)

Rule Period

A Time & Labor period used in the evaluation and application of a rule (see Time Administration).

Rule Program

Specifies the set of rules the Time Administration process will execute and the order in which it will execute the rules.

Rule Results

Net effect of the application of a rule; for instance, the creation of time, initiation of workflow, modification of reported tasks (see Time Administration)

Rule Templates

Templates used to quickly create a variety of rules for the Time Administration program to execute when processing reported and/or scheduled time. Some examples are. compensation rules for overtime and holidays, notification rules for irregular attendance, and rules for just about any other time-reporting situation that requires special processing.

Rules

Rules are your company's business practices captured in software. Rules determine what activities are required to process your business data.

Rule Set

Rule Sets enable you to apply basis rules to your PeopleSoft Analytic Application in the sequence that you prefer. This is particularly helpful if there are multiple basis rules for the same account node, product node, or position source code. Rule Sets can control the execution sequence of your rule combinations, filter combinations, or both. The first occurrence on the node will be applied and any other occurrence will be ignored. Rule Sets are also used with the Data Manager, and with the Currency Conversion engine.

Rules/Time Administration

A physical implementation or execution of a company's compensation, exception and task rules (see Time Administration)

Run Control

A run control is a type of online page that is used to begin a process, such as the batch processing of a payroll run. Run control pages generally start some type of program that manipulates data in some way.

Run Control ID

A unique ID to associate each operator with their own run control table entries. Process Scheduler.

Run ID

Code that uniquely identifies a Run Control for batch processes.

RWC (Risk Weighted Capital)

See PeopleSoft Risk Weighted Capital

S**Safety Stock Policy**

In PeopleSoft Inventory Planning, a policy that determines how safety stock quantities are calculated for an item. The policy has several methods that include days supply and percentage of demand fill.

Salary

Rate of compensation received by an employee.

Salary Group

Part of a group of defaults assigned to job codes. A salary group may include items such as steps and grades dependent on individual company parameters.

Salary Plan

A plan of salary defaults, grades, and step components

Salary Step Components

Pay components assigned to a salary step by entering the corresponding rate codes on the Salary Step Components page.

Salvage Value

An estimate of the amount of money one might receive upon selling an asset once that asset reaches its useful life. Salvage value is used in several depreciation calculations, including Straight Line.

Sales Order Rebate and Penalties

Rebates or penalties that are calculated against sales orders independent of **Buying Agreement**.

Sales Person

A required field used in PeopleSoft Receivables, Billing, Order Management, and Deduction Management when working with items. Each item must be assigned to a credit analyst. If no

credit analyst is assigned to an item, the credit analyst assigned to the customer is used as the default.

Sales UOM

The only units of measure that can be referenced on sales orders and quotes. You establish them on the Product Attributes by UOM page.

Same-Day Sale

An exercise and sale occurring on the same day. The exercise of the option and sale of the underlying shares take place simultaneously. The broker uses the proceeds of the sale to pay the company the exercise price and any tax withholding and the optionee the net cash (less any brokerage commission/fees).

Sample Method

One method of entering characteristic readings for a quality control plan. Using this method, for one given control plan you inspect all the characteristics for the first sample, then all the characteristics for the next sample and so on.

Scale

On a Goals Matrix (In PeopleSoft Workforce Analytics), a scale that defines the lowest, middle, and highest levels of performance needed to achieve associated minimum, midpoint, and maximum levels of compensation pay out. These are referred to as the Threshold, Target, and Stretch levels, respectively. The scale can be used to standardize multiple performance goals to a common scale of measurement.

Scenario

A scenario is a particular outcome you are analyzing when you run in Scenario Manager. Scenarios enable you to study various changes in organization models you created. For each parent and child model you want to study, you create a scenario ID that you use with all run controls.

There are two types of scenarios defined in the Scenario Manager: Historical and Forecast. In the case of an Historical Scenario all future looking fields will be display only and the Scenario Manager component serves as a wrapper to run any analytic engines. In the case of a Forecast Scenario, the Scenario Manager refers to all the business rules, forecasts, and economic assumptions that make up the scenario.

In PeopleSoft Budgeting a scenario is a ChartField used in PeopleSoft Budgeting to identify different budget iterations that use different assumptions.

Schedule

Specific task, date, and time to be worked by a Time Reporter (see Scheduling)

Schedule 13D or 13G

Disclosure forms required to be filed with the SEC and the company by a shareholder (or shareholders) that own(s) more than 5% of a public company. Schedule 13G is a short-form version of the 13D and may generally (but not always) be used only by institutional investors.

Schedule Group

A category of employees or employee groups associated for purposes of time scheduling.

Schedule Line Number

The line associated with an Order ID. The schedule line identifies an item and scheduled ship quantity that may be different from the requested quantity due to item availability.

Schedule Number

A number identifying the salary table form that an employee's pay is computed. Also has a second meaning related to the Payment Voucher processing for the ECS.

Schedule Reconciliation

In PeopleSoft Payables, the process of reconciling scheduled payments by Payment Schedule ID. Schedule Reconciliation helps U.S. federal agencies meet their requirement to schedule or group together payment orders for submission to the Treasury Disbursing Office.

Schedule Template

An ordered pattern of workday(s) and/or off day(s) used in scheduling (see Scheduling)

Schedule Type

In PeopleSoft Payables, an indicator of the nature of items purchased with a Payment Schedule.

Scheduled Punch.

A time reporter's expected punch (see Scheduling)

Scheduling

A function of PeopleSoft Time and Labor and PeopleSoft Global Payroll that enables you to create work schedules and assign them to employees.

Scorecard

A weighted sum of constraint violations in a schedule that evaluates the schedule's validity (that is, acceptability). The score is calculated by adding the value for each relevant violated constraint. See also **Penalty**, **Weight** and **Relevant Constraint**.

In PeopleSoft Balanced Scorecard, views of a strategy tree's components and Key Performance Indicators with red, yellow, or green scores that show its assessments.

Scrap

Any material outside of specifications and possessing characteristics that make rework impractical.

SearchIndex

A set of objects that give the programmer the ability to create, delete, insert, and update a search index and the items within it. Search index items contain a set of statistics about the document that has been indexed (keywords, number of occurrences, proximity to other words, and so on) as well as a key that can be used to point to the document (a URL, database key, or file path).

SearchQuery

A set of objects that allow the programmer to pass a query string and operators to the search engine and receive a set of matching results with keys to the source documents from the search index in return.

Seasonal Index

In PeopleSoft Demand Planning, measures the amount by which a forecasting period is influenced by seasonal effects. The index typically measures the percentage of difference between the base demand in the period and the expected actual demand in the period. An index of 100 indicates an average period in a seasonal cycle.

Seasonal Profile

In PeopleSoft Demand Planning, identifies the weight index assigned to a forecast time period to take in account seasonal fluctuations in the demand.

Seasonality Group

In PeopleSoft Demand Planning, a group of **Forecast Items** with a repetitive pattern of demand from year to year where some periods are higher than others. Typically a group of items is designated as a contributor to the seasonality group. Contributors are chosen because they are representative of the group, are stable, and have at least two years of demand history. The seasonality group profile is more stable than individual profiles of the contributors because the Aggregation process smoothes out random errors.

In Enterprise Planning and Simulation, a Seasonality group is a group of items with similar seasonal patterns. To determine if a forecast element is seasonal or nonseasonal, by averaging their history values over a year and determining where they were above and below average. Seasonality groups capture means seasonal behavior among related products.

Secondary COBRA Events

COBRA qualifying events that extend the amount of time a participant is eligible for COBRA coverage. For an event to qualify as a secondary COBRA event, it must fulfill the following qualifications: The participant must already be enrolled in COBRA coverage as a result of an initial COBRA event, the initial COBRA event must be one that is associated with a change to the employee's job status (such as a reduction in hours, termination, or retirement), and the

secondary event must be one of the COBRA event classifications that involves loss of coverage for the dependent (such as divorce, marriage of dependent, or death of employee). See COBRA and Initial COBRA Events.

Secondary event code

Secondary event codes, also called transaction codes, specify the type of transaction in detail. For example, a transaction's secondary event code could say that the transaction is a catalog order, a rush order, or a request for a sample. Not all transaction types include secondary event codes.

Section

A set of logically related elements that are to be resolved during the payroll process. You define your payroll process by creating sections and adding them to process lists. You can create four different types of sections: standard, generate positive input, sub-process, and payee.

Section 16(a)

Provision of the Securities Exchange Act of 1934 that requires company insiders to file changes in beneficial ownership of the company's equity securities and periodic reports disclosing their holdings.

Section 16(b)

Provision of the Securities Exchange Act of 1934 that requires that any profit realized by a company insider from the purchase and sale, or sale and purchase, of the company's equity securities within a period of less than six months must be returned to the company. It is also known as the "short-swing profit" rule.

Section 423

The Internal Revenue Code section that regulates Employee Stock Purchase Plans.

Section 83(b) Election

A tax filing within 30 days of grant that allows employees granted restricted stock to pay taxes on the exercise date, rather than the date when restrictions lapse. If an employee files the election, taxes are based on the fair market value on the exercise date, with any future appreciation taxed as a capital gain. If the employee does not file an election, taxes are based on the fair market value on the date the restrictions lapse, which will be higher assuming the stock has appreciated in value.

Securities Act of 1933

Often referred to as the "truth in securities" law, the act requires that investors receive financial and other significant information concerning securities being offered for public sale; and prohibits deceit, misrepresentations, and other fraud in the sale of securities.

Securities Exchange Act of 1934

The Congressional act that created the Securities and Exchange Commission. The Act empowers the SEC with broad authority over all aspects of the securities industry. This includes the power to register, regulate, and oversee brokerage firms, transfer agents, and clearing agencies as well as the nation's securities self regulatory organizations (SROs). The various stock exchanges, such as the New York Stock Exchange, and American Stock Exchange are SROs. The National Association of Securities Dealers, which operates the NASDAQ system, is also an SRO. The Act also identifies and prohibits certain types of conduct in the markets and provides the Commission with disciplinary powers over regulated entities and persons associated with them. The Act also empowers the SEC to require periodic reporting of information by companies with publicly traded securities.

Security Clearance

Security Clearances (Classified, Secret, Top Secret) are granted to employees by government agencies and are usually associated with jobs that bring employees into contact with classified government projects or sensitive technologies.

Security Event

In commitment control, events that trigger security authorization checking, such as budget entries, transfers, and adjustments; exception overrides and notifications; and inquiries.

Segmentation

You can “segment” components of pay based on such events as changes in compensation, employee status, or job changes during a pay period. For example, if an individual changes jobs in the middle of a pay period and your organization has a practice of separating components earned in the first job from those earned in the second job, you can set up your system to trigger segmentation of earnings results on the pay slip when there is a change to the job change action/reason field.

Selective Factors

Knowledge, skills, abilities or special qualifications that are in addition to the minimum requirements in a qualification standard, but are determined to be essential to perform the duties and responsibilities of a particular position. Applicants who do not meet a selective factor are ineligible for further consideration.

Selective Merge

In PeopleSoft Enterprise Warehouse the selective merge allows for an additional WHERE clause when you delete a merge.

Self-Service Application

Another name for PeopleSoft's HRMS and ERP applications accessed using a browser.

Sell to Cover Exercise

When an optionee sells a portion of the option shares to cover the exercise cost and any applicable taxes.

Seniority Pay

A premium paid for seniority or for the length of time an employee works for an organization.

Seniority Pay Components

Pay components whose rate codes are assigned to the seniority rate code class SENPAY (provided by PeopleSoft) on the Comp Rate Code page, allowing you to access the seniority pay functionality.

Seniority Rate Codes

A premium paid for seniority or for the length of time an employee works for an organization.

Separate Debit and Credit

A feature in PeopleSoft General Ledger that captures and reports in greater detail the accounting information that resides in balance sheet accounts. It shows the gross debit/credit balances in addition to the net balance for each account stored in the ledger. This feature also supports reversing—debit and reversing—credit journal entries for error correction.

Server Process

A server process is executable code that receives incoming client requests on the application server. The server process carries out a client request by making calls to a service that executes SQL against the database.

Service

A service performs a particular task of an application. Examples of services are MgrGetObj, SQLAccess, RemoteCall, and so on. When a client workstation sends a request to the application server, it sends a service name and a set of parameters, such as "MgrGetObject + parameters". The application server associates the service request with the appropriate server process to complete the transaction.

Service

The PeopleSoft Pension Administration function that determines how much service credit an employee has accrued.

Service Buy Back

The process by which an employee repays a pension plan in order to restore service credit that was forfeited when the employee withdrawal previous contributions. Typically, employees withdraw contributions upon termination and initiate service buy back processing upon rehired.

Service Purchase

The process by which an employee gets additional pension service credit for periods not normally considered eligible. The employee “purchases” this service by paying into the plan.

Service Schedule

A table showing how much service an employee earns based on the number of hours the employee worked during the year or month.

SetID

The label that identifies a TableSet.

Sex Code

Used to indicate gender.

Shape

For a transaction, the set of inputs and outputs for that transaction. For a class, the data members of that class.

Share

A share of a company's stock. Stock options give you the option to purchase a certain number of shares of company stock.

Share Price

The price per share of a company's stock. See, also, "stock price."

Shareholder

Owner of one or more shares of stock in a corporation. Also known as a stockholder or investor.

Shares Available to Issue

The total number of shares authorized, less shares granted, plus cancellations that revert to the Plan pool.

Shares Cancelled

This is usually triggered by a specific event, such as termination of employment in which the unvested shares as of the date of termination are no longer available for future vesting and exercise. These shares are therefore canceled from the option and can be returned to the plan, retired to treasury or allocated back to a group.

Shares Exercisable

The number of shares that are vested and available for exercise.

Shares Exercised

The number of shares purchased upon exercise of a stock option.

Shares Expired

Option shares that no longer are exercisable at the end of the option term. The length of the option term is defined in option agreement. This date is usually the earlier of the exercise period for vested shares after termination of employment or the full length of the option term.

Shares Outstanding

The number of company shares currently held by shareholders, as tracked by the transfer agent

Shift [Time and Labor]

The block of hours that an employee works in a day, such as nine to five, four to eleven, or ten to six. In PeopleSoft Time and Labor, Shift is used as a template of clock hours for scheduling an employee or group of employees to be at work or available to work (on call). Shifts may be constant, rotating, repeating, and/or split; any given shift may or may not have an associated Shift Differential or Bonus. A shift is always associated with a Work Schedule, and consists of clock hour Start and Stop times (two to allow for split shifts), meal periods (two) and relief periods (two).

In PeopleSoft Workforce Analytics, the block of hours that an employee works in a day such as nine to five, four to eleven, or ten to six.

Shift Bonus

A fixed amount (either a flat dollar figure or stated in terms of an employee's rate) paid for working a particular Shift.

Shift Code

A numerical shift identifier that is unique within a SetID.

Shift Differential

Additional compensation paid an employee for time worked during certain shifts. Typically, shift differential is administered as a flat amount per shift, hour worked, and/or as a percentage of the amount paid per shift hour or shift worked.

In PeopleSoft Workforce Analytics, a premium paid for work over regular pay, for which employees on certain shifts may be eligible, such as double-time for late night shifts. Shift differentials are usually stated as an additional rate or factor.

Shift Elements

Individual components of a shift such as TRC start and stop time, duration (see Scheduling)

Shift Name

Customer defined nomenclature for a shift (see Scheduling)

Shift Type

A customer-defined classification associated with a shift. The shift type can be used in the evaluation of rules or exceptions (i.e. On Call) (see Scheduling)

Shift Type [Time and Labor]

Time and Labor defined classification of shifts. Valid shift categories include Flex, General and Elapsed. Shift categories are used in the creation of time reporter schedules (see Scheduling)

Shipping Priority Code

Shipping Priority Codes act as tie breakers during order reservation in PeopleSoft Inventory when different orders are scheduled for the same delivery date and time. When the reservation process in Inventory encounters a situation where there are more orders than available stock, the system reserves the order with the highest priority. If schedules are encountered with the same priority, orders are then considered by order date.

Short Sale

The sale of a security that is not owned or is not delivered at the time of the trade, necessitating its purchase or delivery some time in the future to "cover" the sale. A short sale is usually made with the expectation that the stock value will decline, so that the short seller can eventually cover at a price lower than the original sale, thus realizing a profit. At the time of the short sale, the broker borrows stock to deliver on the settlement date. A short sale can be "naked," in which case the seller does not deliver the shares being sold short and must provide the broker with collateral. Or the short sale can be "against the box," in which case the seller delivers the shares being sold short for the broker to hold "in the box" until the seller chooses to close out the short position.

Short-Swing Transaction

A purchase and sale, or sale and purchase, of the issuer's equity securities by an insider within a period of less than six months. See "Section 16(b)" above.

Short-term Variable Compensation

In PeopleSoft Workforce Analytics, this is cash compensation paid to a worker that is not fixed; includes bonuses and commissions.

Sibling

A tree node at the same level as another node, where both roll up into the same parent. A node can be a sibling, parent, and child all at the same time, depending on its location in the tree.

Sibling

A tree node at the same level as another node, where both roll up into the same parent. A node can be a sibling, parent, and child all at the same time, depending on its location in the tree.

Sibling

A tree node at the same level as another node, where both roll up into the same parent. A node can be a sibling, parent, and child all at the same time, depending on its location in the tree.

Sick Leave

Sick leave is accrued by full-time permanent/seasonal employees at the rate of 4 hours every biweekly pay period; for part-time permanent/seasonal employees, it is accrued at one hour for every 20 hours worked.

Simulated Workforce

In the PeopleSoft Workforce Rewards product, Manage Compensation module, the calculated Simulated Workforce = Existing Employees + New Simulated Employees + Reduced-Employees.

Single Life Annuity

A benefit payable during the lifetime of the participant, with no payments made after the death of the participant. Also referred to as a “life only annuity” or a “straight life annuity.”

Single Signon

This refers to the process by which a user can, after being authenticated by one PeopleSoft application server, access a second PeopleSoft application server without entering a user ID or password.

SIREN Code (Système Informatique pour le Répertoire des Entreprises)

This stands for the Electronic List of Enterprises. The SIREN code is assigned to a company when it registers as a business with the French government, and identifies the purpose of the establishment for regulatory reporting purposes in France.

SIRET (Système Informatique pour le Répertoire des Établissements)

This stands for Electronic List of Entities. In France the SIRET is an identifying number given to a French business by the INSEE, an official statistics and economics organization in France. The SIRET number is a combination of the SIREN and NIC numbers. This number is used by the tax and social security authorities to identify a business enterprise and its entities.

Site Tree

In PeopleSoft eStore, a hierarchical structure that controls navigation, as well as content and behavior within the header, footer, and left margin areas of the web page template.

Slice

The span of time into which an element is segmented as a result of element segmentation. Unlike a segment (or period), a slice does not represent a separate gross-to-net process since it affects only a limited set of elements within a period or segment. Like segments, slices have their own begin and end dates.

Slice Dimension

A model dimension used to restrict user access to the system. For example, a product manager's access to the system can be restricted to only the products he or she is responsible for by defining "Products" as a slice dimension, and assigning this person the members of the "Products" dimension he or she can access.

Slice and Dice

Another term for multidimensional analysis. When your data has three (or more) dimensions, you can think of it as being arranged in a cube (or hypercube), with each side representing a dimension. When you analyze the data, you "slice" off part of the cube or "dice" it to get to an individual cell.

Slotting

In PeopleSoft Workforce Rewards, a process by which the system establishes the target market compensation rates to use for compensating workers in non-benchmark jobs.

Social Security Number

Nine numeric digits assigned to an individual by the Social Security Administration. Also known as a Taxpayer Identification Number (TIN).

Source

The Source table stores valid journal entry and posting sources. These can include job titles (such as CFO), user IDs (such as CLERK123), PeopleSoft General Ledger processes (such as Consolidations), or other applications (such as PeopleSoft Payables).

Source Transaction

In commitment control, any transaction generated in a PeopleSoft or third-party application that is integrated with commitment control, and which can be checked against commitment control budgets. For example, a pre-encumbrance, encumbrance, expenditure, recognized revenue or collected revenue transaction.

Sparsity/Density

A multi-dimensional concept of whether data exists at intersections of dimensions. If a cube has many dimensions, but little or no data in some of those dimensions, the cube is considered sparse. Sparse cubes take up unnecessary disk space and reduce calculation performance. The goal is to create dense cubes and only use dimension intersections where data actually exists.

Special Accumulator

A device that accumulates earnings from different sources for a specific purpose. 401(k), pension and retirement plans use special accumulators. A 401(k) plan might use a special accumulator to calculate a deduction using regular, vacation, and overtime earnings. Special accumulators can add to or subtract from a pool of earnings.

Special Payments

A payment that occurs once or under special circumstances (e.g., back pay interest, lump sum leave, bond refund, longevity bonus, compensatory time reimbursement, death payment, severance pay, separation bonus, etc.).

Special Rates

Higher salary rates for specific grade levels and occupational groups determined by OPM for employees working in specific geographic areas. Each area is assigned a separate Schedule Number.

Specialist

A member of a stock exchange who maintains a fair and orderly market in one or more securities. A specialist or specialist unit performs two main functions. executing limit orders on behalf of other exchange members for a portion of the floor broker's commission, and buying or selling for the specialist's own account to counteract temporary imbalances in supply and demand, preventing wide swings in stock prices.

Specialized Experience

Experience that has equipped the applicant with the particular knowledge, skills, and abilities to perform successfully the duties of the position and is typically in or related to the work of the position to be filled.

SpeedChart

A user-defined shorthand key designating several ChartKeys to be used for voucher entry. Percentages can optionally be related to each ChartKey in a speedchart definition.

SpeedType

A code representing a combination of ChartField values. SpeedTypes simplify the entry of ChartFields commonly used together.

Spending Limits

See Planning Targets.

Split and Join

In PeopleSoft Demand Planning, the process of subdividing a forecast so that multiple users can make changes to their portions of the forecast. After changes are complete, the portions are joined back into a single forecast.

Split Deduction

Deduction that you create by splitting an existing deduction into two deduction items. The new deduction retains the original item ID with an added suffix number.

Split Shift

Periods of productive time split up by period of non-working time; example. a time reporter comes to work as a busboy for the lunch shift from 12-2 p.m. and then returns to work from 6-8 for the dinner shift (see Scheduling)

Split Stream Processing

The matching of a payment's cash information with the payment advice information when they have been received as separate transmissions through EDI and lockbox.

The uniting of the payment cash with the payment advice when they have been received as separate information through EDI.

Spokesmen Committee (Sprecherausschusse)

In Germany the Spokesmen Committee represents the interest of the management in your company before the ownership. The Spokesmen's Committee is consultative in nature, although they play a co-determination role on individual employment contracts, hiring, and dismissals. They also play a role in monitoring employment fairness, equity, and non-discrimination in terms of nationality, race, religion, sex, and age.

Spouse Demonstration J&S

In the PeopleSoft Pension Administration system, an informational-only form of pension payment that tells what the spouse's total benefit would have been if the retiree had chosen the spouse as the beneficiary rather than a nonspouse beneficiary. You cannot pay pension benefits based on this form because it is informational only.

Spouse Eligibility Alias

In PeopleSoft Pension Administration, a Custom Statement that defines any criteria that must be met before the plan will provide an Automatic Spouse Benefit. For example, the plan may require that the employee and spouse be married a full year before they are eligible for an automatic spouse benefit.

Spouse Eligibility Statement

See Spouse Eligibility Alias.

Spread

Depending on the context, either (1) the difference between the bid and asked prices for an over-the-counter stock, or (2) the difference between an option's exercise price and the market price at the time of exercise (i.e., the profit component of the exercise).

SQL Objects

Used to create rules that are more complicated than templates or actions and conditions allow—select statements, insert statements, table joins, and sub-queries

ST (Strategic Trust)

See Strategic Thrust

Staged Date

The date an item was received into the inventory business unit.

Staging ID

An identifier for a putaway plan. The inventory system sequentially assigns Staging IDs when it creates the putaway plan.

Standard Cost

A predetermined, fixed cost associated with an **Inventory Item** or **Forecast Item**, representing detailed estimates of each element of cost entering into the purchasing or manufacturing of an item. Standard cost is used when minor variations in an item's cost are not needed. The use of standard costs also enables management to determine how much an item should cost (Standard), look at how much it does cost (Actual), analyze the differences between the two and their causes (Variances), and compute economic order quantity.

Standard Form (SF)

A standardized form for interagency use by the Federal government. The SF prefix is the most common but not exclusive one in usage.

Standard Price

In PeopleSoft Demand Planning, the standard selling price associated with a **Forecast Item**. The price can be introduced into the system directly in forecast item maintenance or indirectly using the demand transfer interface. At higher levels in the view where there is no standard price available, the summarization function can be set up to develop one.

Standard Unit of Measure

The smallest unit of an item that a PeopleSoft application tracks.

Startup Data

In PeopleSoft Pension Administration, accrued Service, Cash Balance Account, or Employee Account data loaded into the system in the form of an opening balance and “as of” date. The alternative would be to load the entire accrual history.

State Record

The State Record is a PeopleSoft record, keyed by process instance, that must be created and maintained for each Application Engine program. The State Record defines the fields that an Application Engine program uses to pass values from one SQL statement to another.

Static Group

An employee group in Time and Labor that enables you to control its creation and maintenance. The group remains the same at all times until you change it.

Static Policy Controls

Determines how a static (versus time-phased) **Inventory Policy** is to be calculated. Static controls use period and average methods and their arguments.

Statistical Account

An account that has an associated unit of measure, used for tracking and monitoring statistical data. For example, the Workstations account uses EA (each) as a generic unit of measure, while the Floor Space statistical account might use square feet and the Work Days account would use days.

Statistical Code

The unit of measure used for tracking and monitoring statistical data. For example, using a statistical code of WS may represent the number of Workstations.

Statistical Forecast

In PeopleSoft Demand Planning,, a forecast developed at each level of the forecast pyramid and that considers the item’s history in isolation.

Status Checking

In PeopleSoft Projects, a control feature that can be applied to transactions coming into Projects from cost feeder systems. If the incoming transaction does not conform to predetermined status and analysis conditions, an online warning will display or the transaction will be rejected.

Status Position Code

A code that identifies the various conditions of a position, e.g., frozen, classified, etc.

Statutory Account

Account required by a regulatory authority for recording and reporting financial results. In PeopleSoft, this is equivalent to the Alternate Account (ALTACCT) ChartField.

Step

A secondary level or subcategory within the primary pay level (depending upon pay plan, different employees may have a different number of steps within their primary pay level).

Step Progression

In PeopleSoft Workforce Analytics, a pay increase granted to an employee or group whose salary plan includes steps within grades. Each step increase is a step up the pay range for the employee.

Stock

In corporate finance, the form in which an owner's interest is represented, distributed in units known as shares.

Stock Administrator

An individual who administers and manages the corporation's benefits and/or equity compensation plans. This individual serves as the contact for transfer agent and broker inquiries. Stock Administrators manage Stock Option Plans, Employee Stock Purchase Plans, Restricted Stock Award Plans, and Stock Bonus Plans.

Stock Appreciation Rights (SAR)

A contractual right to receive, either in cash or employer stock, the appreciation in the value of the employer's stock over a certain period of time. A SAR can be used alone or in tandem with Incentive Stock Options (ISO/SAR) or Nonqualified Stock Options (NQ/SAR). PeopleSoft Stock Administration supports only tandem SAR's.

Stock Awards

Stock allocations that are processed in the Manage Variable Compensation business process. Stock Administration creates stock grants from finalized stock awards.

Stock Exchange

An organized marketplace in which bonds, stocks, and common stock equivalents are traded by members of the exchange, acting as agents (brokers) and as principals (dealers or traders). Such exchanges have a physical location where brokers and dealers meet to execute orders to buy and sell securities. Each exchange sets its own requirements for membership.

Stock Option

A contractual right granted by the company, generally under a stock option plan, to purchase a specified number of shares of the company's stock at a specified price (the exercise price) for a specified period of time (generally five or ten years). Assuming that the exercise price is the

same as the fair market value on the grant date, the option will become more valuable if the fair market value goes up, because the option effectively gives the optionee the right to buy stock in the future at a discount.

Stock Price

The price per share of a company's stock. See, also, "share price."

Stock Purchase Participant

An individual who participates in the corporation's Stock Purchase Plan.

Stock Purchase Plan

A type of broad-based stock plan that permits participants to use payroll deductions accumulated over a period of time to acquire stock from the company.

Stock Split

A change in the capitalization of an issuer that increases or decreases the number of securities outstanding, and adjusts the value of the securities accordingly, without a corresponding change in the assets or capital of the issuer. For example, if an employee has options to purchase 25 shares at \$10 per share and the company has a 2-for-1 stock split, the employee thereafter has the option to purchase 50 shares at \$5 per share.

Stock Swaps

A payment method that can be used to cover the cost of the exercise price and taxes depending on whether it is allowed by the plan. When an employee elects to exercise a stock option by means of a stock swap, they surrender already-owned shares of stock to pay the total required option exercise price and/or taxes for the option being purchased. The surrendered shares are usually valued at the fair market value of the company's stock on the date of exercise.

Stock Trading Symbol

The three or four letter symbol used to identify a company's stock on the stock exchange where it trades. Also known as a "ticker symbol".

Stock Withholding

A cashless method of satisfying the withholding taxes due upon the exercise of a stock option by authorizing the company to withhold from the shares being exercised a number of shares equal to the taxes.

Stockholder of Record

Person or entity, often a broker or the Depository Trust Company, named on the issuer's or transfer agent's stock record books as the owner of shares held in "street name." The stockholder of record acts in part as a way of safekeeping stock certificates that might otherwise be lost by the beneficial owner, and also in order to keep the identity of the beneficial owner confidential from the company.

Stock-In Probability

A replenishment option for defining transfer parameters for PeopleSoft Demand Planning or Inventory Planning upload files. The option is the percentage of time you want to have the item on hand for the **Business Unit** and is used to calculate safety stock.

Stop Time

Out punch

Storage Area

A division of a **Business Unit** used to store material and to track **Inventory Transaction**. Storage areas might include shipping and receiving docks, staging areas, warehouse zones, and inspection and quality control departments. Each storage area can be divided into a maximum of four levels, with each level representing a physical subdivision of the area.

Storage Level

A hierarchical subdivision of a storage area.

Storage Location

The combination of a storage area and that area's most detailed storage level. This is the smallest definable physical space within an **Inventory Business Unit**.

Strategic Initiatives

In PeopleSoft Balanced Scorecard, actions the organization must take to implement strategy. May be temporary or short-term in nature.

Strategic Thrust (ST)

In PeopleSoft Balanced Scorecard, four to five statements or paragraphs that summarize the core components of an organization's strategy. Strategic thrusts describe the key areas across which a scorecard is balanced. They are themes or goals your organization is striving to achieve; more specific descriptions of what you must do to achieve that goal are defined by critical success factors. Key performance indicators may be attached to strategic thrusts as long as there aren't critical success factors below them, but typically strategic thrusts aren't directly associated with key performance indicators.

Strategy Tree

In PeopleSoft Balanced Scorecard, the hierarchical relationships of the objectives your organization is striving to achieve. Used as the foundation for a scorecard, and typically balanced across four major categories: Financial, Customer, Learning and Growth, and Internal Processes. These are made up of Vision, Strategic Thrusts, and Critical Success Factors.

Stratification Engine

A support module that structures the volume of financial accounts and balances at a large financial institution to a manageable scale for processing by the PeopleSoft Funds Transfer Pricing (FTP) and PeopleSoft Risk Weighted Capital (RWC) applications. It categorizes data by a range of values and summarizes data based on rules you define for FTP and RWC.

Stratification Wizard

Stratification Wizard is a tool you can use to quickly create new stratification rules or update the existing rules. Stratification Wizard prompts you for each of the possible source and destination fields, grouping operations, and summarization actions to be performed. Stratification Wizard enables you to stratify your data according to tiers, discrete values, periodic increments, and numeric increments. It also enables you to leave the data aggregated.

Streams

An optional feature that enables you to reduce processing time by processing groups of payees simultaneously.

Street Name

See "Stockholder of Record".

Stretch

In a Goals Matrix performance scale (In PeopleSoft Workforce Analytics), this is the level of performance for which an employee achieves maximum pay out. Performance above this level receives no greater pay out.

Strike Price

The price per share which must be paid in order to exercise the stock option. The strike price is typically the fair market value of the stock on the grant date. Also known as the "exercise" or "grant" price.

String constant

String constants are delimited in PeopleCode by using either single (') or double (") quote marks.

Strip Funding

One of several methodologies used by PeopleSoft Funds Transfer Pricing (FTP) to derive maturity when calculating FTP rates based on matched maturity funding. This approach matches the projected cash for the instrument in each time period, with a specific cost of funds rate for that cash flow. The FTP rate for the instrument is then calculated by weighting the cost of funds rate for the cash flow in each time period by the term of the cash flow.

Structured Query Report (SQR)

A type of printed or displayed report generated from data extracted from a PeopleSoft SQL-based relational database. PeopleSoft applications provide a variety of standard SQRs that summarize table information and data. You can use these reports as is, customize them, or create your own.

Style File (Verity)

Collection style refers to a set of configuration options that are used to create the indexes associated with a collection. A collection has one collection style and it is defined in a set of style files before creating the collection.

SubCustomer Qualifier

A value that groups customers into a division for which you can generate detailed history, aging, events, and profiles.

Sub-Process Section

A type of section you can add to a process list. Sub-process sections are especially useful for performing iterative processes such as gross ups (calculating the gross amount for a given net amount). You can include conditional logic within a sub-process section.

Subscription

The process of mapping fields, selecting data parameters and submitting the information to an outside vendor.

Substitute Item

In PeopleSoft Manufacturing, an item that can be used when there are no primary components available in inventory or when there is a long-term shortage of the original item. The substitute item can be defined at three levels: setID, business unit/item, and bill of material/engineering bill of material.

Subtask

A lower-level Planning task in a schedule's hierarchy that rolls up into a parent task. For example, an operation performed on a production ID would be a subtask of the production order.

Summarization Process

See Rollup.

Summary ChartField

A feature for creating summary ledgers that roll up detail amounts based on specific detail values or on selected tree nodes. When detail values are summarized using tree nodes, summary ChartFields must be used in the summary ledger data record to accommodate the maximum length of a node name (20 characters).

Summary Forecast

In PeopleSoft Demand Planning, a type of forecast that results from adding up the adjusted forecast totals from the next lower level, meaning the sum of the children's forecasts for the parent. The summary forecast at level one (1) is always zero since there is not a logical lower level.

Summary Ledger

An accounting feature used primarily in allocations, inquiries, and PS/nVision reporting to store combined account balances from detail ledgers. They increase the speed and efficiency of reporting by eliminating the need to summarize detail ledger balances each time a report is requested. Instead, detail balances are summarized in a background process according to user-specified criteria and stored on summary ledgers. The summary ledgers are then accessed directly for reporting.

Summary Tree

A tree used to roll up accounts for each type of report in summary ledgers. In effect, summary trees enable you to define "trees on trees." In a summary tree, the detail values are really nodes on a detail tree or another summary tree (known as the *basis* tree). A summary tree structure specifies the details on which the summary trees are to be built.

Summed Adjustment Type

When the system finds multiple summed discounts, they are added together, and applied once.

Super Tree Utility

A PeopleSoft Enterprise Warehouse utility that enables you to combine multiple effective dates of a tree into one. This super tree contains all tree changes for a certain period of time enabling you to analyze effective dated trees more easily.

Supplemental IRR

An IRR type used when a federal employee's retirement deductions were originally under-reported. An agency can create a Supplemental IRR to report the difference in the retirement deduction amount to the Office of Personnel Management (OPM).

Supplemental Tax Method

A payroll tax calculation method that uses a straight percent rather than allowances. The percentage depends on state requirements.

Supply Chain Warehouse

See Warehouses.

Support Costs

Activity costs not directly connected to production.

Support Modules

The support modules are a collection of engines and analysis models that derive values, rates, financial calculations, and prices. PeopleSoft Analytic Applications take this data and use it for further processing. The support modules perform processes that are used in the financial services industry. They are only utilized by two of the PeopleSoft Analytic Applications: Risk Weighted Capital (RWC) and Funds Transfer Pricing (FTP).

Support Team

A group of people working together to sell to and/or support a customer. You can assemble support teams and associate them with ship to customers, quotes, sales orders, and **Buying Agreement**.

Supporting Element Overrides

Provide a mechanism to override various supporting element types, such as brackets, dates, durations, formulas, and variables, at various different levels.

Supporting Elements

Supporting elements are building blocks for other elements. In PeopleSoft Global Payroll, they are used in combination with other elements to create rules. They are not stand-alone. Typical supporting elements are arrays, brackets, rounding rules, and fictitious calculations.

Suspend Exercise

As a condition of a leave of absence, a company may stipulate to restrict the exercise of shares during the leave or for a period of time. Only applicable if the Stock Action is LOA.

Suspend Vesting

As a condition of a leave of absence, a company may stipulate to suspend vesting of shares. Only applicable if the Stock Action is LOA. If a company does not Suspend Vesting then the Vest Deferral Grace Period and Service Rule are not applicable.

Suspended Item

In PeopleSoft Demand Planning, an item suspended by the system. The suspension is due to the lack of demand for the number of periods defined for the item's **Control Group**.

System Element

In PeopleSoft Global Payroll, system elements are delivered and maintained by Peoplesoft. There are two types of system elements: database system elements and system-computed elements. Database system elements contain payee-related data that can be used frequently in a calculation, such as department ID, location, and personal data. System-computed elements are automatically populated by the payroll process.

System Functions

A list of all activities that the system supports, along with their associated General Ledger distributions.

System-Defined Count

The PeopleSoft Inventory feature that employs user-defined criteria to begin the count creation process.

System-Defined History

Any statistical information updated by the posting and aging programs, maintained to reflect customer credit standing.

T**T+3**

The obligation in the brokerage business to settle securities trades by the third day following the trade date. "Settlement" occurs when the seller receives the sales price (less the broker's commission) and the buyer receives the shares.

Table

The underlying format in which data is stored by columns (fields) and rows (records, or instances).

Table Lookup

A utility in PeopleSoft Pension Administration that finds an unknown value based on a known one in your data set. For example, you can look up an interest rate based on a year, or an actuarial factor based on an employee's age.

TableMaps

In the PeopleSoft Enterprise Warehouse, TableMaps define the physical relationships between related tables. TableMaps allow you to define "families" of related tables and the columns that define the key relationships between the tables.

TableSet

A group of rows across control tables identified by the same SetID.

TableSet Sharing

Specifying the control table data for each business unit so that redundancy is eliminated.

Tardy

The circumstance when a time reporter reports for work after the scheduled start time

Target

In a Goals Matrix performance scale (In PeopleSoft Workforce Analytics), this is the performance level your organization establishes as the norm for performance and pay out.

Target Cost

A desired target cost (for production, engineering or marketing) is found by subtracting the desired profit margin from a competitive or estimated price.

Target Currency

The value of the entry currency or currencies converted to a single currency for budget viewing and inquiry purposes.

Target End Date

The intended end date for employee schedules in Time and Labor reporting. You establish a target end date, and depending on the work schedule templates, the application determines what the actual end date should be.

Target Grade

Highest obtainable grade for a position.

Target Matrix

In PeopleSoft Workforce Rewards, the Target Matrix defines the level of award to be paid based upon a predetermined level of performance that a defined measure is evaluated against.

Target Plan

In PeopleSoft Workforce Rewards, a plan for distributing compensation awards, in which the level of the award is linked directly to a predetermined level of performance that a defined measure is evaluated against. For example, a Target Plan for a Sales group might be as follows: “As a group, increase the business unit’s net income by 10% and each member of the group will be eligible for an award equal to 5% of base pay.”

Target Rate

In PeopleSoft Workforce Rewards, calculated market rates you choose to use as new target pay rates for the jobs in your organization. Think of these target rates as pay guidelines, in support of your company’s overall pay strategy.

Target View

In PeopleSoft Demand Planning, the of the views being reconciled during the **Cross-View Reconciliation** process. During the process, fields for reconciliation are defined for both a source and target view.

Tariff

In Germany a Tariff is a contract between the employee's unions and the employers' association, defining labor agreements on issues such as standard working hours, income, and vacation. This contract is valid for all of Germany for the business or industrial sector the company is working in (such as **Banking** or **Metal**).

Tariff Area

In Germany, additional labor agreement terms beyond those in the Tariff, such as salary plans or employee reviews, can also be applied based upon the Tariff Area. The Tariff Area is often split along regional lines (such as **Bavaria** or **Berlin**).

Task

See **Manufacturing Task** and **Planning Task**

Task

A piece of work assigned to or demanded of a person; a unit of work (see Time Reporting)

Task Entity

Individual component of a task; for example, Project ID, Activity ID, Work order, Department, Company, Business Unit (see Time Reporting).

Task Profile

A way of viewing or establishing where to allocate employee task information for a day and time. The task profile fields that appear on the page are established by the *Task Profile Template*.

Task Profile

Entity that establishes the default values for optional and required task elements. This can be for single or multiple tasks. (i.e. default values based on hours allocation, percentage distribution, equally distributed or by prompt) (see Time Reporting)

Task Rules

A methodology that is applied to scheduled, reported and payable time to allocate or redistribute task assignments (see Time Administration)

Task Transfer

Department transfer

Task Values

The customer defined value for a specific task element i.e. Customer 1, project 1, etc (see Time Reporting)

Taskgroup

Identifies the default time reporting templates, task template, and task profile(s) for time reporters that share the same task reporting requirements.

Taxable Benefits

Any employer contributions that are subject to Federal Withholding Tax.

TDS (Transfert de Données Sociales)

In France the TDS is a social security transfer report, submitted on magnetic media to the government.

Team Member

An individual who is part of a support team. Each team member may be in a commissionable or non-commissionable role.

Technical Scenario

In PeopleSoft Enterprise Warehouse technical scenarios allow you to set up the object type values that the Resolver uses to chunk the record/TableMap you'll resolve. Technical scenarios allow technical users to define chunking criteria that enable chunks to run in parallel. This allows for parallel data processing.

Template

A portal template is simply HTML code, associated with a web page, to define the style and layout of the page. Templates allow a developer to build an HTML page by combining HTML from a number of sources. Templates do two basic things: define the layout of the page, and define where to get HTML for each part of the page.

Template Pagelet

One piece of an overall template. For example, in a given template, there may be one template pagelet for the universal navigation header and one template pagelet for the target content.

Temporal Constraint

A relationship between Planning tasks that defines their sequence and timing in a schedule. Temporal constraints cannot be violated by the Optimizer. PeopleSoft Planning constraints include finish to start, start to start, finish to finish, start to finish.

Temporary Continuation of Coverage (TCC)

The TCC program, as prescribed by the OPM, requires Federal agencies to provide to separating Federal employees the opportunity to temporarily continue their FEHB coverage for up to 18 months (unless involuntarily separated because of gross misconduct), provided the individual pays the full cost of coverage, including both the employee and government share and a two percent administrative charge. Agencies may elect to provide this service in-house or enter into cross-servicing agreements with another Federal agency.

Tenor

Used by the PeopleSoft Funds Transfer Pricing (FTP) application to refer to the maturity of an instrument. It represents the length of time an instrument is available as either a source or use of funds. The FTP application calculates the transfer price for an instrument, based on the marginal cost of funds of similar liquidity and tenor.

Term Certain and Continuous Payment Option

See Certain and Continuous Payment Option.

Term Certain Payment Option

See Certain Only Payment Option.

Termination

A transaction in which an employee ceases to be an employee of the corporation.

Threshold

In a Goals Matrix performance scale (In PeopleSoft Workforce Analytics), this is the minimum threshold for adequate performance, the designated level of performance below which it is inappropriate to pay incentives.

Think-time process

Think-time functions suspend processing either until the user has taken some action (such as clicking a button in a message box), or until an external process has run to completion (for example, a remote process).

Three-Tier

A three-tier architecture introduces an intermediary application server between the client workstation and the database server to improve performance. Within PeopleSoft, the application server sends the SQL to the database and then returns results to the client in the form of lightweight Tuxedo messages.

Threshold Checking

In PeopleSoft Projects, a control feature that can be applied to transactions entered directly into Projects or integrated into Projects via the INTFC_PROJ_RES table. If the transaction exceeds a predefined tolerance, a warning will display or the transaction will be rejected.

Thrift Savings Plan (TSP)

A voluntary retirement savings and investment plan for Federal employees administered by the Federal Thrift Investment Board.

Ticker Symbol

The three or four letter symbol used to identify a company's stock on the stock exchange where it trades. Also known as a "stock trading symbol".

Tiers

In the financial services industry, Tiers are ranges that you set up for stratifying your instrument data (such as amounts, rates, and numbers) into specific groups. You define Tiers within Stratification Engine.

Time Administration

A process which provides four [separate] different online tools for creating, maintaining, and applying an organization's compensation, task, and exception rules to both reported and scheduled time. templates, actions and condition, SQL objects, and user exits

Time and Labor Period

A distinct, configurable period of time used by all the PeopleSoft Time and Labor processes (see Time Reporting)

Time and Labor User

Either a Time Reporter or a Time Manager

Time Capture Device

Third party system or methodology for collecting elapsed or time capture device time, i.e., time capture device, IVR, Fax, etc. (see Time Reporting)

Time Collection

A Time and Labor feature that collects positive and exception time reports, applies appropriate business rules and edits to the reported time to ensure validity and reasonableness, and returns errors and questionable items to the time reporter for correction or scrutiny. Time collection is also responsible for scrutinizing future (previously) posted time information for correctness when those reports are ready for use.

Time Collection Device

A group of time collection device lumped together and named for ease of assignment to employees. In other words. clock group 1 is made up of clock 1, and clock 2. Employee 123 is assigned to clock group 1 and can then punch in at either clock 1 or clock 2. (see Understanding Time Collection Device)

Time Collection Device time

Reporting time by recording actual starts and stop times (see Time Reporting)

Time Dimension

Determines how date-related information is presented in a **Cube View**. This dimension defaults to a two-level hierarchy consisting of the **Inventory Policy** year and a standard period, such as monthly.

Time Fence

In PeopleSoft Planning, a user-defined parameter that specifies the business rules to be used in the generation of the plan. PeopleSoft Planning time-fence types include start of time, end of time, planning close date (demand time fence), purchase order fence, leveling fence, action message cutoff, and planning time fence.

Time Manager

An individual who supervises Time Reporters

Time Period

A period of Time used in Time and Labor rules processing. You can categorize time periods in terms of days, weeks, or months. You establish day, week, or month-type periods for use when you apply rules for compensation, holidays, and so on.

Time Report

A payroll time and/or labor distribution time report for an employee for any date within the employee's current period.

Time Reporter

Any employee or contractor for who time is reported or generated in PeopleSoft Time and Labor.

Time Reporter Information

Values associated with the Time Reporter that are displayed when entering or viewing reported time and facilitate the processes of Time Reporting and Time Management (see Time Reporting)

Time Reporting

Any information required by a business unit that can be attributed to an individual employee (worker/contractor) and can be expressed in hours.

Time Reporting Code

A hybrid of two PeopleSoft objects: the Payroll Earnings Type and the Human Resources Absence Type. The Time Reporting Code represents the level at which a business actually needs to track employee time to support all of its administrative and compensation needs.

Time Reporting Code Type

Categorization of a time reporting code. Valid categories include: units, amounts, hours or a combination of hours and amounts (see Time Reporting)

Time Reporting Group

See Group [Time and Labor].

Time Segment

For Service, Cash Balance Accounts, and Employee Accounts, employees can accrue benefits differently at different times. The period of time during which employees use a particular rule is that rule's time segment.

TimeSpans

Relative periods, such as year-to-date or current period, that can be used in various PeopleSoft General Ledger functions and reports when a "rolling" time frame, rather than a specific date, is required. TimeSpans can also be used with Flexible Formulas in PeopleSoft Projects.

Tolerance

In PeopleSoft Projects, a value that is defined at either the project or activity level as either a percentage of a project's funding or an actual amount.

In commitment control, the percentage over budget that you allow, excluding revenues applied to increase budget limits, before the system creates an exception.

Total Authorized But Unissued

The combined total number of shares from Shares Available to Issue plus Total Options Outstanding.

Total Compensation

In PeopleSoft Workforce Analytics, this is generally the officially recognized compensation provided an employee in the course of their employment with an organization; includes both direct compensation and benefits compensation.

Total Compensation Management

The ability to track and report on all types of cash programs, non-cash programs, benefits and deferred compensation for all current workers, ex-workers and individuals associated with ex-workers who receive compensation due to the employment of the ex-worker.

Total Non-Compensation

In PeopleSoft Workforce Analytics, generally, this is the often unrecognized compensation an employee receives in the course of their employment with an organization; it includes Learning and Development compensation such as training, and Workplace Environment Compensation such as telecommuting privileges or other prerequisites.

Total Options Outstanding

The number of company shares currently held by shareholders as tracked by the transfer agent. Derived by using the number of Prior Outstanding plus Grants, less Exercises and less Cancellations.

Total Rewards

In PeopleSoft Workforce Analytics, this is the total rewards provided to an employee by their employment with an organization; it includes their officially recognized total compensation, and less often recognized total non-compensation.

Tour of Duty

The scheduled days and hours per day of attendance at a duty station for an employee.

Tracking Signals

PeopleSoft Demand Planning, a forecasting tool that detects bias in the forecast and provides an early warning of an unstable forecast. There are six tracking signals associated with each **Forecast Item** that correspond to the six most recent historical periods.

Trade Payment

An authorization for a customer deduction in a Promotion application.

Training Report 2483

The Training Report 2483 is a French regulatory report used to declare vocational training your company has provided to your employees. It is also known as the Declaration 2483 Report. The purpose of the report is to receive tax deductions from the government based upon the amount of money your company has spent on training.

Transaction

A named command with optional named and typed inputs and outputs. The associated external system or the Business Interlink Plug-in understands this command. The types of inputs and outputs are based on a set of generic types.

Transaction

A named command with optional named and typed inputs and outputs. The associated external system or the Business Interlink Plug-in understands this command. The types of inputs and outputs are based on a set of generic types.

See also **Inventory Transaction** or, for PeopleSoft Projects, **Resource Transaction**.

Transaction catalog

Lists transactions used to interface to the external system.

Transaction Code

In PeopleSoft Projects, an additional field on each resource transaction that is used in conjunction with accounting entry templates. Transaction codes enable you to deal with exceptions to your accounting entry templates without having to create additional transaction types. You can set up separate accounting entry templates for resource transactions containing the transaction codes you create. The accounting entry templates for those resource transactions lines can then use the same transaction types, but specify different accounts.

In PeopleSoft Asset Management, transaction codes identify special asset transactions and are used in conjunction with transaction type to create accounting entries.

Transaction Code

Identifies what action has taken place against the position.

Transaction Costing

See Multidimensional Costing

Transaction Currency

In the financial services industry, the original currency in which a company conducts its business activities. When a company has multinational operations, it may use different transaction currencies. These are translated to the base currency for consolidation and reporting of financial results.

Transaction Date

The date a transaction actually occurred as opposed to the date the transaction is recognized—the accounting date (although the two dates can be the same).

Transaction Dated

Data aggregated over a date range.

Transaction group

The package can contain one or more transaction groups. Each transaction group is a set of transactions of the same type, with the same trading partners involved.

Transaction Loader

The SQR in PeopleSoft Asset Management that transfers load lines from the Loader tables into the PeopleSoft Asset Management Tables as assets and open transactions.

Transaction Tables

In the PeopleSoft Enterprise Warehouse, these are tables that contain dynamic information and are keyed by business units.

Transaction Type

The building blocks of accounting entry templates in PeopleSoft Asset Management and Projects. For each transaction type you create you define specific transaction lines. The transaction lines are then transferred into accounting entry templates. In the accounting entry templates each transaction line is assigned a specific general ledger account.

Transactional System

A business application for performing the business transactions that keep your company running. Transactional applications, and the databases that support them, are optimized for quick transaction processing. Because they are constantly changing and are not optimized for data retrieval, transactional system databases are not usually the best source of data for analysis.

Transfer Agent

An individual or firm who that keeps a record of your shareholders and the number of shares they own. Transfer Agents also issue new share certificates and cancel old certificates. Unlike Brokers, Transfer Agents are not responsible for selling stocks. Instead they are primarily concerned with maintaining records on all stocks which your company has issued.

Transfer Forecast

In PeopleSoft Inventory Planning, a Generation process option that transfers the forecast from the target view in Demand Planning forecasts. The process only transfers items from Demand Planning that have been set to update the **Inventory Policy**.

Transfer Punch

The start of a work period that specifically denotes a change in task and usually compensation-related characteristics

Transfer Type

An interunit transfer setting PeopleSoft Production Planning and Enterprise Planning use to determine where it will obtain item data for transfer tasks. If the type is a supply or demand transfer task, the Planning engine only processes the transfer item for a single location, reducing the time for plan processing. If the transfer type value is both, the Planning engine processes the transfer item using data from both the To and From units.

Transfer Worksheet

A work space for transferring an open item from one customer to another.

Transferable Stock Options

Options that may be transferred by the optionee, generally only to a family member or to a trust, limited partnership or other entity for the benefit of family members, or to a charity.

Translate Table

A system edit table that stores codes and translate values for the miscellaneous fields on the database that do not warrant individual edit tables of their own.

Translate Table

A system edit table that stores codes and translate values for the miscellaneous fields on the database that do not warrant individual edit tables of their own. In most cases PeopleSoft maintains the Translate Table.

Transport Rate

The Transport is a statutory deduction in France. Each establishment has a rate, and the URSSAF notifies establishments of this rate on a yearly basis. This deduction is used by the region to subsidize transportation, and maintain and build roads.

Transportation Lead Times

The transportation lead time is the in-transit interval from the date and time a shipment leaves your warehouse (**Inventory Business Unit**) to the date and time it arrives at your customer's receiving dock. The transportation lead time is used in calculating the scheduled shipment and scheduled arrival dates on the order when you enter either a requested arrival date or a requested shipment date.

Travel And Relocation Date

Length of time an employee must remain in the Government after the Government has paid to relocate him/her from one official duty station to another or for initial appointment.

TRC Program

A program that runs the level at which an organization actually needs to track employee time to support all of its administrative and compensation needs. TRCs are assigned to TRC Programs, which are ultimately assigned to workgroups. Multiple Workgroups can share these TRC Programs.

Treasury Interface files

These are DOS-based files generated by PeopleSoft in accordance with FMS file layouts for transmission of payment data to one of the FMS' Regional Financial Centers.

Treasury Position Code

In the financial services industry, this is a lookup code used for off-balance sheet treasury position accounts, such as foreign exchange, derivatives, precious metals, or any other account position that is the result of trading room and treasury operations.

Treasury Stock

Shares of a company's stock that have been repurchased or otherwise reacquired by the company and are "held in treasury." Whether the treasury shares count as "issued" or as "outstanding" shares of the company is a matter of state corporate law. Generally, a company may not vote its own shares held in treasury.

Treasury Stock Method

The method of calculating primary and fully diluted earnings per share when common stock equivalents such as unexercised stock options exist. Required under generally accepted accounting principles.

Tree

The graphical hierarchy in PeopleSoft systems that displays the relationship between all accounting units (for example, corporate divisions, projects, reporting groups, account numbers) and determines roll-up hierarchies.

Tree Compare Utility

A PeopleSoft Enterprise Warehouse utility that enables you to compare effective dates for trees. The results page shows nodes that have been added, deleted, or moved from one parent to another. You may also view the detail objects that have changed.

Tree control

Tree Control is a hierarchical search tool that you can embed in a panel. Tree Controls give the user a view of hierarchical data structures and enable them to drill down through the hierarchy to a particular row of data.

Tree Denormalizer

The Tree Denormalizer Application Engine process converts trees into multi-column data format so they can be used by third-party OLAP or ROLAP tools.

Trigger

See Event Trigger.

Trustee Extract

A PeopleSoft Pension Administration data extract containing data that a third party needs in order to produce pension checks.

Turnover Costing

In PeopleSoft Workforce Analytics, this is a calculation of the cost to the organization of employee turnover, in dollars.

Turnover Rate

In PeopleSoft Workforce Analytics, the rate that employee's are leaving the company.

TUXEDO

BEA's middleware product used to manage transaction queues, server process initiation, system administration, time-outs, data encryption, compression, logging and other application server processing.

Two-Tier

A two-tier architecture refers to the traditional client/server model in which a client workstation connects to and sends SQL directly to the database server.

Type of Appointment

Indicates the specific type of appointment, e.g., part-time permanent, full time temporary, etc.

U**Underlying Security**

The security underlying a stock option that an optionee has the right to buy, or the security underlying a convertible security.

Underpayment Adjustment Limit

The maximum amount or percent above which underpayment adjustments are not allowed for a given business unit.

Underwater Option

When the current market price is below the option exercise price. When an option is underwater, it would cost more than the underlying stock is worth to exercise the option. Such options are also described as being "out-of-the-money."

Underwriter

An investment banking firm that actually buys the shares from the company in a public offering and then resells them (at a slightly higher price) to its customers.

Unexpected Losses

In the financial services industry, these occur when the economic capital is exhausted and the insolvency rate is exceeded. Unexpected losses are determined by a targeted insolvency rate (confidence level); for example, a 99.7% confidence level indicates that there is a 0.03% estimated probability that the unexpected losses will exceed economic capital (or shareholder equity).

Union Code

Part of a group of defaults assigned to job codes. Union code may be used by human resources to group similar jobs or bargaining units together, dependent on individual company parameters.

Unit Code

In the financial services industry, Unit Code is used as an alternate means of measuring the relative size of companies participating in external surveys. A typical measure would be the number of employees in a company. The concept of unit is generic enough that the units can be other measures besides number of employees. For example, in the hospital industry the unit could be the number of hospital beds. Or in the hotel industry the unit could be the number of rooms.

Unit of Measure (UOM)

A type of unit used for quantifying in PeopleSoft systems. Depending on the application, units of measure might describe dimensions, weights, volumes, or amounts of locations, containers, or business activities. Examples include inches, pounds, workhours, and standard cost dollars.

Unit of work

Each transaction group includes one or more individual units of work. A unit of work is a single transaction that you want to commit or rollback as a whole.

Unitize Assets

The process of unitizing a single load line, usually originating from a different application, into multiple assets in PeopleSoft Asset Management.

Univariate Forecasting Technique

In Enterprise Planning and Simulation, the Univariate Forecasting Technique is a forecasting method that uses only the recorded history for the value to forecast its future.

Universal Navigation Header

Every PeopleSoft portal includes the universal navigation header, intended to appear at the top of every page as long as the user is signed on to the portal. In addition to providing access to the standard navigation buttons (like Home, Favorites, and signoff) the universal navigation header can also display a welcome message for each user.

Unscheduled Punch

A punch that is made by a time reporter who was not scheduled (see Time Reporting)

Unvested Shares

Unvested stock options are options that have not vested and, therefore, are not exercisable.

URI

A URI does not include the query string (the text following a ? on the URL). You can think of it as a subset of the URL that points to the resource, but does not include any parameters being passed to that resource. From the above example, the URI portion of the URL is as follows:

`http://serverx/InternetClient/InternetClientServlet`

URL

In this document, the term URL refers to the entire query string. The following is an example of a URL:

`http://serverx/InternetClient/InternetClientServlet?ICType=Script&ICScriptProgramName=W
EBLIB_BEN_401k.PAGES.FieldFormula.iScript_Home401k`

URSSAF Code

The URSSAF is the body responsible for ensuring payment of Social Security contributions by all French employers.

Useful Life

The amount of time an asset may be depreciated.

User Data

PeopleSoft Demand Planning, data held in user-defined fields. These fields provide for storage of additional data that is not supplied by the standard set of fields in the system. The fields can also become part of the key for the **Forecast Item** at each level within the **Forecast View**.

User-Defined History

A summary of customer receivables activity that is defined by the user.

User-Field Code

PeopleSoft Demand Planning, a definition of a set of user-defined fields that contain data specific to the installation.

V**Valuation**

The way a company represents the value of a non-monetary award such as stock.

Value allocation

A process in PeopleSoft Workforce Planning, by which you assign an overall monetary value to a competency strategy for your organization, and assign a weight or importance to the roles,

competencies and accomplishments in the strategy. The system then allocates a breakdown of the value to roles, competencies, and accomplishments in the strategy based on their relative weighting or importance.

Value Object

In the PeopleSoft Enterprise Warehouse, Value Objects are a metadata layer that provides descriptive information about fields and values. Value Objects are used as constants in Data Manager target object definition.

Variable

Temporary storage for use or defined information used in the creation and application of rules (see Time Administration)

Variable [Global Payroll]

An element type that defines and stores values such as a character, date, or number. You can use variables to create generic formulas for situations where you use the same values over and over again in a calculation.

Variable Compensation

In PeopleSoft Workforce Analytics, direct compensation that is not fixed, that is paid out in variable amounts, such as bonuses and commissions.

Variable Plan.

A plan in which either the number of shares and/or the price at which they will be issued is not known on the grant date.

VAT Account Type

A code that identifies the different types of accounting entries that must be created for VAT transactions. These codes are also used to categorize transactions in the VAT transaction table. The account type is used in conjunction with the VAT code and VAT transaction type to determine the VAT ChartFields used for a given VAT accounting entry.

VAT Apportionment

For mixed activity, VAT apportionment is the mechanism that allows you to specify the ratio of taxable activity to exempt activity for individual ChartFields.

VAT Calculation Method

Options are Net or Gross. When calculating VAT at net, the early payment discount is applied to the goods amount before calculating the VAT. The amount of VAT calculated using this method is the amount that is to be paid, regardless of whether the early payment discount is actually taken at time of payment. When calculating VAT at gross, the VAT is initially calculated based on the gross transaction amount. The early payment discount is not taken

into account at this point. However, in some countries an adjustment is made to the VAT amount at the time of payment, if the early payment discount is taken.

VAT Calculation Type

Options include Exclusive or Inclusive. If exclusive, the VAT amount is stated separately from the goods amount. If inclusive, the VAT is not stated separately but is included with the goods amount.

VAT Code

The tax code used to define a percentage the system uses to determine the VAT amount. The VAT code is similar to the sales and use tax code, with a few exceptions. The tax authority tied to the VAT code generally consists of a single authority, and the ChartFields for a VAT code don't reside with the tax authority but are determined by the combination of the VAT code, VAT account type, and VAT transaction type.

VAT Declaration Point

When VAT transaction information is declarable for reporting purposes. Options include Invoice or Payment. If you choose invoice, the system will recognize VAT at invoice time; if you choose payment, the system will recognize VAT at the time of payment.

VAT Entity

The level or entity within an organization at which VAT reporting is performed. VAT entities can be registered for VAT in multiple countries, but only one country can be designated as the VAT entity's home country. VAT and Intrastat reporting information and VAT default information are defined for each country in which the VAT entity is registered. You may also specify any VAT exceptions—either exoneration or suspension from paying VAT—for any country in which the entity is registered.

VAT Exempt Supply or Purchase

A transaction where the product or item is non-taxable or exempt from VAT. No VAT code is associated with the transaction. Although no tax is applied to the transaction, the transaction is still logged in the VAT transaction table.

VAT Exonerated

A transaction where the purchaser has been determined as not subject to VAT. For these cases, there may be an exoneration certificate number tied to the purchasing entity (either the customer or the VAT entity) as proof of exoneration. A zero-rated VAT code should be associated with transactions where exoneration applies. The transaction is still logged in the VAT transaction table, but no tax is applied.

VAT Rebate Percent

Within Canada, for Public Service Bodies, the percentage of VAT that is not normally recoverable but which may be refunded in the form of a tax rebate.

VAT Recoverability Percent

The percent of VAT that's recoverable.

VAT Registration Countries

Country codes associated with a VAT registration number for a particular customer or VAT entity.

VAT Transaction Table

Stores detailed transaction information for VAT reporting. It is the primary source of information for all VAT reports. Each application is responsible for writing to this table and also to a cross-reference table used to link entries in the VAT transaction table with entries within each application.

VAT Transaction Type

Used to categorize VAT transactions according to particular VAT accounting and reporting requirements. The VAT code and the VAT transaction type are used in conjunction with the VAT account type to obtain the ChartFields for accounting entries.

VAT Treatment

A description of how the transaction must be treated for VAT purposes. This is used to determine how VAT defaults are applied, what accounting entries are required, and how and if the transaction is reported on the VAT return.

VAT Use ID

A code used to identify the type of activity in which a purchased good or service will be used, and therefore to determine a recoverability percent and a rebate percent (when applicable) that will be applied to a transaction line. Activities are categorized as taxable, exempt, or mixed. Where activity is mixed, you may associate either the ratio of taxable activity to exempt activity directly with the Use ID, or you may indicate that this ratio is determined at the ChartField level.

VdkVgwKey

A key within a Verity BIF file for every document to be indexed. VdkVgwKey values must be unique across all collections that will be searched in any one application.

Vendor Draft

A draft issued by a vendor. PeopleSoft Receivables generates vendor drafts, provides a flexible worksheet environment for approval management, and enables discounted or standard submission for bank processing. PeopleSoft Payables receives vendor drafts and associates the appropriate vouchers.

Verity

The third-party search engine integrated with the PeopleSoft Portal.

Verity Fields

Verity fields are stored in the collection for retrieval and searching, and can be returned on a results list. Fields are defined in the BIF file and stored in the collection for retrieval and searching, and can be returned on a results list. Fields, like date and numeric fields can be used with the comparison operators (<,<=,>,>=).

Verity Thesaurus

The custom thesaurus consists of lists of synonyms defined in a synonym control file and can be used for synonym searching. After defining synonym lists in the control file, you use the mkisyd utility to create a custom thesaurus (a control file which has the .syd extension) that the search engine uses.

Verity Topics

Verity applications can provide end users with predefined search criteria called *topics*. A topic is a named object that represents a concept, or subject area and can be used for synonym searching. It consists of words and phrases grouped together using the Verity query language in a tree-like structure. When provided, topics can be shared by all users.

Verity Zones

Zones are specific regions of a document to which searches can be limited. When the zone filter is used, the Verity engine builds zone information into the collection's full-word index. The index, enhanced with zone information, permits quick and efficient searches over zones. Searching a zone is faster than field searching. Zones are defined in the DAT file. The contents of a zone cannot be returned in the results list of an application.

Version

There can be up to five budget versions for each Budget Center level in a Budgeting Model. Budget versions are used to perform what-if analysis and comparisons of budget amounts before the user selects one version to submit as the Budget Center's budget plan. PeopleSoft Budgeting-specific.

Vest Deferral Grace Period

The specified period of time within which an optionee must return from leave to avoid having the vesting deferred. Only applicable if the Stock Action is LOA. Suspend Vesting must be selected for this rule to be applicable.

Vest Deferral Grace Period Service Rule

If the company provides a vest deferral grace period, they may stipulate that only certain individuals are eligible for the grace period based on service with the company. Only applicable if the Stock Action is LOA. Suspend Vesting must be selected for this rule to be applicable.

Vest Immediately

A stock option plan may provide that upon specific types of terminations, or upon a case by case scenario, all unvested shares held by an individual can be made immediately vested as of a specific date. Some companies' plans provide that under certain circumstances, such as retirement, the vesting of option shares accelerates upon termination of employment. When this occurs, you must modify the vesting schedule before you terminate the individual.

Vested Shares

Option shares that are free of any ownership restriction. Generally, vested exercised shares are fully owned by the optionee, free from restrictions and freely tradable.

Vested Termination

The termination of an employee who has a vested benefit. The benefit is deferred until the participant reaches retirement age. The employee is considered "Terminated Vested," "Term Vested," or simply "TV."

Vesting

The method by which a granted option becomes free of all restrictions and the Optionee has full rights to the shares.

Vesting Schedule (Template)

A convenient way to set up the framework for a vesting schedule that can be uniformly applied to individual options. When you grant stock options, you define a vesting schedule to determine the default-vesting schedule for the option.

Vesting Service

The service used to determine an employee's vesting percentage. Rules for accruing vesting service may be different from rules for accruing other plan service credits.

VETS-100 Federal Contractor Report

This report is required of employers in the United States. It lists federal job classifications, and the number of employees and new hires in the last 12 months who are special disabled military veterans or Vietnam era military veterans. It also provides totals for each job classification of both veterans and non-veterans who hold these jobs.

View

PeopleSoft Demand Planning, a multilevel forecast structure. Each view is associated with a unique view ID and includes information that defines the view and structure type. The three types of views are working, disbursement, and dynamic.

For PeopleSoft Budgeting see Budget View.

Virtual Tasks

In Time and Labor, Virtual task data is associated with a taskgroup profile that defines common characteristics for a given Taskgroup and Task Profile ID. A single row of data is linked to multiple Earnings records for multiple employees. By minimizing the physical storage of daily task data we provide enhanced performance without limiting its functionality.

Vision

In PeopleSoft Balanced Scorecard, the overall mission of an organization. Usually the highest level on a strategy tree. Vision is optional; you aren't required to have a vision component on each strategy tree.

See also Strategy Tree

Volume

Total share volume traded in a stock during market hours.

W**WA (Workforce Analytics)**

See PeopleSoft Workforce Analytics

Waiver Of An OPM Qualification Standard

Involves setting aside requirements in a published standard to place an employee in a particular position, usually to avoid some kind of hardship to the employee, such as in cases of RIF or administrative error on part of the agency. Extra training and/or skills development may be needed to help the employee adjust to the new position. Waivers are granted by OPM or an agency, as appropriate, on a case-by-case basis, and do not directly affect other positions in the organization.

Warehouses

A warehouse reporting and analysis solution that supports the specific PeopleSoft business application that warehouse is using. It consists of predefined ETL maps, data warehouse tools, and Data Mart definitions. The warehouses we deliver are: PeopleSoft Financials Warehouse, PeopleSoft HRMS Warehouse, PeopleSoft CRM Warehouse, and PeopleSoft Supply Chain Warehouse.

Warning Exception

A transaction that exceeds the available funds but is allowed to continue to be posted against the budget. Warnings are informational only.

Warrant

A type of security, usually issued together with a bond or preferred stock, that entitles the holder to buy a proportionate amount of common stock at a specified price, usually higher than the market price at the time of issuance, for a period of years or to perpetuity. A warrant is usually issued as a sweetener, to enhance the marketability of the accompanying fixed income securities. Warrants are freely transferable and are traded on the major exchanges.

WCB

In Canadian provinces the Worker's Compensation Board (WCB) operates as an independent board, and thus would have different requirements in each province. For example, in British Columbia the organization is called the Worker's Compensation Board of British Columbia and in the Province of Quebec, the board is known as Commission de la Santé et de la Sécurité du Travail (CSST).

Weight

In PeopleSoft Planning, a user-defined value for the constraints that can be violated, determining how the schedules score will be calculated. Violations that are more critical to your schedule merit a higher weight.

Weight and Volume Pricing

You can price shipments by weight or volume to create price prices. Weight and Volume pricing requires using estimated shipments.

Weighted Average Cost of Funds

The projected principle payments for an instrument are used to derive a series of matched maturity funding rates, which in turn are used to calculate the overall base PeopleSoft Funds Transfer Pricing (FTP) rate. The Weighted Average Cost of Funds (WACF) method calculates a weighted average FTP rate where each of the funding rates is weighted by the principle payment amount and the term to maturity of the payment.

WFA (Workforce Analytics)

See PeopleSoft Workforce Analytics

WGI Due Date

Identifies the date of an employee's next within grade increase. Current policy is that the step increase is implemented on this date automatically unless prevented by the processing of an unsatisfactory performance appraisal.

WGI Non-Creditable Days

Total number of days that cause the WGI due date to be adjusted forward.

Whole Calendar Month

An instruction telling the system to use every day in each month for this time period. The system fills in the last day of the period according to the information you have entered.

Wildcard

You can replace the right-hand characters in a search field with a percent (%) wild card to query a range of values beginning with the remaining, left-hand characters. For example, by entering '2%' in a six-character field, you will receive a range of available values, such as 200000 through 299999 or 2aaaaa through 2zzzzz.

Window Period

The ten-day period, from the third to twelfth day after public release of a company's financial statement, when insiders may exercise their stock-appreciation rights without violating Securities and Exchange Commission rules for short-term trading.

Windows Client

Traditional PeopleSoft 32-bit client. Windows clients connect to the application server domain (Tuxedo) using a port number (or connection string) specified in PeopleSoft Configuration Manager.

WIP Replenishment Method

Designates how the PeopleSoft Flow Production request is communicated. For a replenishment method of Inventory, the Workflow, Pull Ticket, and Pull List replenishment methods are available. With feeder line replenishment, you can only use Pull Tickets.

WIP Replenishment Mode

Determines how PeopleSoft Flow Production is triggered to generate a replenishment request for an item. Replenishment options include Backflush, Manual, and Kanban Card.

WIP Replenishment Source

Determines where you send your PeopleSoft Flow Production replenishment request and what source supplies your WIP location. Options include Feeder, Inventory, and Vendor.

Withdrawal

An election not to continue participation in a stock purchase plan.

Withdrawal of Contributions

In a pension plan, the act of returning pension contributions, with interest, to an employee who is terminating. An employee who withdraws contributions typically forfeits all service associated with those contributions. If the employee is later rehired, repayment of contributions and interest typically reinstates the forfeited service.

Withholding

A deduction taken by employers out of taxable income of an individual. Typical withholding taxes include federal income taxes, federal social security, Medicare taxes, and state and local income taxes.

Within Grade Increase (WGI)

A longevity-based increase in salary based on predetermined time in grade requirements and acceptable performance.

Work Council (Comité d'Enterprise)

In France it is mandatory for companies with more than 50 employees to elect a Work Council to represent the employees in negotiations with management.

Work Effort

See Activity Type.

Worker

In PeopleSoft Workforce Analytics, workers are defined as anyone who performs functions for the organization, and receives compensation from the organization's operating expense funds in return. Workers can be direct employees or independent contractors. This includes individuals contracting business directly from the company or through an agency.

Work Group

In PeopleSoft Enterprise Performance Management, the work group is a grouping of employees that share a similar activity profile.

Work Period

A Days On/Days Off template; the smallest unit of time that a business uses to communicate with their employees regarding when to be and/or not to be at work (that is, time working and time not working). The work period can be any number of hours. Until clock hour reporting is implemented, the application does not care about the number of hours. The initial Time and Labor product will apply the work period to a calendar day.

Work Queue

In PeopleSoft Demand Planning and Inventory Planning, a feature for reviewing and working with exceptions created during the processing of forecasting and inventory data.

Work Schedule

A template consisting of a sequence of work periods (days) on and off, and the number of scheduled hours per work period. Work Schedules and Work Periods should not be confused with calendar days.

Worksheet

A way of presenting data to the user through a BAM interface that enables users to do in-depth analysis using pivoting tables, charts, notes, and history information.

Work Templates

Work templates describe your employee's work patterns. Work templates could apply to individuals or entire organizations. For instance, 9 AM to 5 PM, Monday through Friday is a fairly standard working week in organizations.

Workday

A 24-hour period rounded by daybreaker with one or more associated shifts (see Scheduling)

Workday Override

A function that allows a Time Manager to override a Time Reporter's schedule for a single workday. For example, Jane's long-term schedule assignment is Monday – Friday, 8.00 to 17.00. Due to an increase in production demand, her manager needs to schedule her to work 7.00 to 18.00 on Thursday, 16 March 2000. Her manager needs to be able to make this change to her schedule in the PeopleSoft Time and Labor system, so when Jane checks her schedule for this week, she'll see the revised schedule.

Worker

Workers can be defined as anyone who performs functions for the organization and receives compensation from the organization's operating expense funds in return. Workers can be direct employees or independent contractors (includes individuals contracting business directly from the company or through an agency).

Workers Compensation

The days an employee is on LWOP due to sustaining an injury or illness while on the job.

Workflow

The background process that creates a list of administrative actions based on your selection criteria and specifies the procedure associated with each action.

Workflow

The background process that creates a list of administrative actions based on your selection criteria and specifies the procedure associated with each action.

Workforce Monthly Report (Déclaration Mensuelle Obligatoire des Mouvements de Main D'oeuvre)

In France, companies that employ 50 or more employees are required to submit the Workforce Monthly Report to the Administrative Division of the Ministry of Work and Social Relations. The report contains workforce information for a given establishment of a company, including

the total number of employees and details of employees who have joined or left the establishment during the month.

Workgroup

A user-defined group of employees who share identical compensation rules. A workgroup may be equivalent to all the employees in a business enterprise, all employees in a Paygroup, all employees belonging to the same Union or Union Local, or all employees who work at a specific work location.

Worklist

The automated "to do" list that Workflow creates. From the Worklist you can directly access the panels you need to perform the next action, and then return to the Worklist for another item.

Worklist

The automated "to do" list which Workflow creates. From the worklist you can directly access the pages you need to perform the next action, and then return to the worklist for another item.

Works Councils (Betriebsrat)

In Germany, the works councils for your company are internal committees elected by the employees that represent the interests of salaried and hourly paid employees, other than management. Every work location in your company has its works council (this would be the local works council) and the company as a whole has a central works council.

Work-Study Program

Government or non-government programs supervised work experience related to a student's course of study and are a part of, or a supplement to, education. Federal student-trainee programs are examples of such programs.

X

Y

Yearly Maximum Pensionable Earnings (YMPE)

Amount set by the government upon which Canadian Pension Plan (CPP) contributions are made.

Z

Zero-Based Budgeting

A budgeting option that builds a budget from the ground up starting with zero values. This is in contrast to an incremental budget that is based upon using prior year actual or budget values as starting point. PeopleSoft Budgeting-specific.

Zero-Rated VAT

A VAT transaction with a VAT code that has a tax percent of zero. Used to track taxable VAT activity where no actual VAT amount is charged.

Zip Code

The term for postal codes in the United States.

Index

#

#DEBUG 2-55
#DEFINE 2-92, 2-121, 2-122
#ELSE 2-99, 2-118
#end-file 2-213
#ENDIF 2-102, 2-118
#END-IF 2-102
#IF 2-118
#IFDEF 2-121
#IFNDEF 2-118, 2-122
#INCLUDE 2-122

:

:#INCLUDE command 2-122
:BOX and SHADE 2-181
:-Cnn 2-31
:COMMIT command 2-48
:-DBdatabase 2-32
:-LOCK 2-32
:-NR 2-32
:ROLLBACK command 2-215
:-SORTnn 2-32

A

abs 2-129
absolute value
abs 2-129
absolute value 2-129
acos 2-129
ADD 2-2
AFTER 2-21, 2-22
AFTER qualifier, ON-BREAK argument 2-192
AFTER-BOLD argument
DECLARE-PRINTER command 2-80
AFTER-BOLD argument
PRINT command 2-180
AFTER-PAGE argument
DECLARE-PROCEDURE command 2-84
DECLARE-TOC command 2-88
USE-PROCEDURE command 2-233
AFTER-REPORT argument
DECLARE-PROCEDURE command 2-84
USE-PROCEDURE command 2-233
AFTER-TOC argument
DECLARE-TOC command 2-87

ALTER-CONNECTION 2-4, 2-115, 2-219
ALTER-LOCALE 2-6
ALTER-PRINTER 2-13, 2-76, 2-82
ALTER-REPORT 2-15
arccosine 2-129
arcsine 2-130
arctangent 2-130
array 2-139
array elements 2-52
arrays
ARRAY-ADD command 2-16
ARRAY-DIVIDE 2-16
ARRAY-DIVIDE command 2-16
ARRAY-MULTIPLY 2-16
ARRAY-MULTIPLY command 2-16
ARRAY-SUBTRACT command 2-16
maximums 2-52
ascii 2-139
asciic 2-139
asin 2-130
ASK 2-19, 2-121, 2-122
atan 2-130
AT-END argument, NEXT-COLUMN command 2-171

B

BATCH-MODE, INPUT command 2-125
BEEP argument, SHOW command 2-222
BEFORE 2-21, 2-22
BEFORE qualifier, ON-BREAK argument 2-192
BEFORE-BOLD argument
DECLARE-PRINTER command 2-79
BEFORE-BOLD argument
PRINT command 2-180
BEFORE-PAGE argument
DECLARE-PROCEDURE command 2-84
DECLARE-TOC command 2-87
USE-PROCEDURE command 2-233
BEFORE-REPORT argument
DECLARE-PROCEDURE command 2-84
USE-PROCEDURE command 2-233
BEFORE-TOC argument, DECLARE-TOC command 2-87
BEGIN-DOCUMENT 2-20
BEGIN-FOOTING 2-24
BEGIN-HEADING 2-25
BEGIN-HEADING command see also Page Header 2-25
BEGIN-PROCEDURE 2-27
BEGIN-PROGRAM 2-30

BEGIN-SELECT 2-21, 2-30
 BEGIN-SETUP 2-34
 BEGIN-SQL 2-35
 bind variables 2-32
 BLINK argument, SHOW command 2-222
 -Bnn
 BEGIN-SELECT command 2-31
 BOLD argument
 PRINT command 2-180
 SHOW command 2-222
 BOTTOM-MARGIN argument
 DECLARE-LAYOUT command 2-72
 BOX argument
 GRAPHIC command 2-116
 PRINT command 2-181
 BREAK 2-40
 BREAK argument, EVALUATE command 2-104
 break processing, field changes 2-191

C

CALL 2-40
 CALL SYSTEM USING 2-40
 CAPTION argument, PRINT-BAR-CODE 2-199
 ceil 2-130
 CENTER argument, PRINT command 2-181
 charts 2-201
 CHART-SIZE 2-59
 CHAR-WIDTH argument, DECLARE-LAYOUT command 2-73
 CHECKSUM argument, PRINT-BAR-CODE 2-199
 chr 2-140
 CLEAR-ARRAY 2-46, 2-53
 CLEAR-LINE argument, SHOW command 2-222
 CLEAR-SCREEN argument, SHOW command 2-222
 CLOSE 2-46
 -Cnn argument
 BEGIN-SELECT command 2-31
 BEGIN-SQL command 2-37
 CODE-PRINTER argument, PRINT command 2-181
 ALTER-COLOR-MAP
 ALTER-COLOR-MAP command 2-2
 COLOR argument
 DECLARE-PRINTER command 2-78
 DECLARE-COLOR-MAP
 DECLARE-COLOR-MAP command 2-65
 GET-COLOR
 GET-COLOR command 2-113
 SET-COLOR
 SET-COLOR command 2-217
 columns 2-231
 COLUMNS 2-47, 2-171
 COLUMNS argument, POSITION command 2-177

COMMAND 2-21, 2-22
 COMMIT 2-48
 compiler directive 2-118
 compiler directives 2-99, 2-102, 2-118, 2-121, 2-122
 CONCAT 2-49
 concatenation 2-226
 cond 2-140
 conditional processing 2-104, 2-120, 2-235
 CONNECT 2-50
 CONNECTION 2-21
 connection_name 2-3, 2-5, 2-67, 2-113, 2-217
 cos 2-131
 cosh 2-131
 cosine 2-131
 CREATE-ARRAY 2-51, 2-211
 CREATE-COLOR-PALETTE 2-54

D

data input 2-123
 data types supported
 by INPUT command 2-123
 DATA-ARRAY argument, PRINT-CHART command 2-204
 DATA-ARRAY-COLUMN-COUNT argument, PRINT-CHART command 2-204
 DATA-ARRAY-COLUMN-LABELS argument, PRINT-CHART command 2-204
 DATA-ARRAY-ROW-COUNT argument, PRINT-CHART command 2-204
 database cursors 2-50
 DATE argument
 DECLARE-VARIABLE command 2-90
 DISPLAY command 2-94
 MOVE command 2-165
 PRINT command 2-182
 SHOW command 2-222
 date edit format characters 2-187
 dateadd 2-141
 datediff 2-141
 DATE-EDIT-MASK argument, ALTER-LOCALE command 2-8
 datenow 2-141
 DATE-SEPARATOR argument, ALTER-LOCALE command 2-8
 datetostr 2-142
 DAY-OF-WEEK-CASE argument, ALTER-LOCALE command 2-9
 DAY-OF-WEEK-FULL argument, ALTER-LOCALE command 2-9
 DAY-OF-WEEK-SHORT argument, ALTER-LOCALE command 2-9
 DB2 2-48, 2-215
 -DBconnectionstring argument, BEGIN-SELECT 2-33, 2-38
 DEBUG 2-118

- DEBUG 2-121, 2-122
- DECIMAL argument, DECLARE-VARIABLE command 2-90
- DECIMAL-SEPARATOR argument, ALTER-LOCALE command 2-8
- DECLARE-CHART 2-56, 2-202
- DECLARE-CONNECTION 2-21, 2-66, 2-67
- DECLARE-IMAGE 2-68
- DECLARE-LAYOUT 2-69
 - arguments with 2-72
- DECLARE-PRINTER 2-58, 2-76, 2-180
- DECLARE-PROCEDURE 2-83
- DECLARE-REPORT 2-85
- DECLARE-TOC 2-87
- DECLARE-VARIABLE 2-89
- default database 2-37
- DEFAULT-NUMERIC argument
 - DECLARE-VARIABLE command 2-90
- DEFAULT-NUMERIC argument, DECLARE-VARIABLE command 2-90
- deg 2-131
- delete 2-137
- destination field 2-164
- DISPLAY 2-93
- DISTINCT argument, BEGIN-SELECT command 2-31
- DIVIDE 2-97
- DO 2-29, 2-30, 2-98
- DO argument, EXECUTE command 2-106
- document marker 2-20, 2-177
- DOCUMENT paragraphs 7-3
- DOT-LEADER argument, DECLARE-TOC command 2-87
- DSN 2-3, 2-4, 2-5, 2-65, 2-67
- dynamic query variables 2-32

E

- edit 2-142
- EDIT argument
 - PRINT command 2-182
 - SHOW command 2-222
- edit masks
 - case sensitivity 2-187
 - changing dynamically 2-188
 - uses 2-185
- edit types 2-182
- EDIT-OPTION-AD argument, ALTER-LOCALE command 2-9
- EDIT-OPTION-AM argument, ALTER-LOCALE command 2-8, 2-9, 2-37
- EDIT-OPTION-BC argument, ALTER-LOCALE command 2-9
- EDIT-OPTION-NA argument, ALTER-LOCALE command 2-8
- EDIT-OPTION-PM argument, ALTER-LOCALE command 2-9

- ELSE 2-100
- ENCODE 2-100
- END-DECLARE 2-65, 2-67, 2-101
- END-DECLARE 2-67
- END-DOCUMENT 2-101
- END-EVALUATE 2-101
- END-EXECUTE 2-21
- END-FOOTING 2-101
- END-HEADING 2-101
- END-IF 2-102
- ending a query 2-108
- END-PROCEDURE 2-102
- END-PROGRAM 2-102
- END-SELECT 2-21, 2-102
- END-SETUP 2-102
- END-SQL 2-102
- END-WHILE 2-102
- ERASE-PAGE argument
 - NEXT-COLUMN command 2-171
- EVALUATE 2-40, 2-103
- Examples 2-4
- EXECUTE (SYBASE and Microsoft SQL Server) 2-105
- exists 2-138
- EXIT-SELECT 2-108
- exp 2-132
- exponents 2-132
- expressions 2-127
- EXTENT argument, LOAD-LOOKUP command 2-160
- external source files 2-122
- EXTRACT 2-109

F

- FIELD argument, CREATE-ARRAY command 2-53
- fieldsSee* columns 2-53
- file number 2-174
- file-related functions
 - delete 2-137
 - exists 2-137, 2-138
 - rename 2-138
- files, reading 2-212
- FILL argument 2-190
- FIND 2-110
- FIXED argument, OPEN command 2-174
- FIXED_NOLF argument, OPEN command 2-175
- flagging SQR 1-5
- FLOAT argument
 - DECLARE-VARIABLE command 2-90
- floor 2-132
- FONT argument
 - DECLARE-PRINTER command 2-79
- fonts 2-82
 - HP Laser Jet 2-80
 - PostScript 2-80

- Windows printers 2-81
- FONT-TYPE argument
 - DECLARE-PRINTER command 2-78
- FOOTING 2-24
 - footings 2-24
- FOR-APPEND argument, OPEN command 2-174
- FORMFEED argument
 - DECLARE-LAYOUT command 2-72
- FOR-READING argument
 - OPEN command 2-174
- FOR-REPORTS argument
 - DECLARE-PRINTER command 2-77
 - DECLARE-PROCEDURE command 2-84
 - DECLARE-TOC command 2-87
 - USE-PROCEDURE command 2-233
- FOR-REPORTS, BEGIN-HEADING command 2-26
- FOR-WRITING argument, OPEN command 2-174
- FROM PARAMETER 2-22
- FROM ROWSET 2-22
- FROM ROWSETS 2-21
- functions
 - arguments 2-129
 - transcendental 2-137
- functions, miscellaneous
 - array 2-138
 - ascii 2-139
 - asciic 2-139
 - character 2-140
 - cond 2-140
 - edit 2-142
 - get environment variable 2-142
 - isblank 2-144
 - isnull 2-144
 - substrb 2-151
 - substrp 2-152
 - to_char 2-153
 - to_multi_byte 2-153
 - to_number 2-153
 - to_single_byte 2-154
 - wrapdepth 2-155

G

- GET 2-111
- GETDATA 2-21, 2-22
- getenv 2-142
- GOTO 2-115
- GOTO-TOP argument, NEXT-COLUMN command 2-171
- GRAPHIC VERTLINE 2-115
- GRAPHICBOX 2-115
- GRAPHICHORZLINE 2-115

H

- HEADING 2-26
- headings 2-26
- Hebrew language support 2-197
- HEIGHT argument, PRINT-BAR CODE command 2-199
- HORZ-LINE argument, GRAPHIC command 2-117
- HPLASERJET 2-77
- hyperbolic cosine 2-131
- hyperbolic sine 2-135
- hyperbolic tangent 2-136

I

- IF 2-120
- IF nested within a WHILE loop 2-236
- IFDEF 2-118
- images 2-68
- images printing 2-209
- IMAGE-SIZE argument
 - DECLARE-IMAGE command 2-69
 - PRINT-IMAGE command 2-210
- IN 2-21
- INDENTATION argument
 - DECLARE-TOC command 2-87
- Informix 2-48, 2-215, 2-225
- Ingres 2-215
 - BEGIN-SELECT command, -Cnn 2-31
 - BEGIN-SQL command, -Cnn 2-37
 - Cnn 2-31
 - COMMIT command 2-48
 - ROLLBACK command 2-215
- INIT-STRING argument, DECLARE-PRINTER command 2-77
- INOUT 2-21
- INPUT 2-123
- INTEGER argument, DECLARE-VARIABLE command 2-90
- INTO argument, EXECUTE command 2-107
- isblank 2-144
- isnull 2-144

K

- KEEP 2-85
- KEY argument, LOAD-LOOKUP command 2-160

L

- labels 2-115
- LAST-PAGE 2-126

LAYOUT argument, DECLARE-REPORT command 2-85
 LEFT-MARGIN argument
 DECLARE-LAYOUT command 2-72
 length 2-144
 lengthb 2-145
 lengthp 2-145
 lengtht 2-145
 LEVEL argument, TOC-ENTRY command 2-228
 LEVEL qualifier, ON-BREAK argument 2-192
 LINE-HEIGHT argument, DECLARE-LAYOUT command 2-73
 LINEPRINTER 2-77
 LINE-WIDTH argument, DECLARE-LAYOUT command 2-72
 loading an internal table 2-159
 LOAD-LOOKUP 2-158, 2-162
 local procedures 2-28
 LOCALE argument, ALTER-LOCALE command 2-8
 -LOCK
 BEGIN-SELECT command 2-32
 BEGIN-SQL command 2-37
 locking types defining 2-37
 log 2-133
 log base 10 2-133
 log base e 2-133
 log10 2-133
 logical expressions 2-120
 LOOKUP 2-159, 2-162
 LOOPS argument, BEGIN-SELECT command 2-33
 lower 2-146
 LOWERCASE 2-163
 lpad 2-146
 ltrim 2-146

M

MATCH argument, PRINT command 2-190
 MAX-COLUMNS argument, DECLARE-LAYOUT command 2-72
 MAXLEN arguments, INPUT command 2-124
 MAX-LINES argument, DECLARE-LAYOUT command 2-73
 MBTOSBS 2-164
 mod 2-133
 MONEY argument
 DISPLAY command 2-94
 MOVE command 2-165
 PRINT command 2-191
 SHOW command 2-222
 MONEY-EDIT-MASK argument, ALTER-LOCALE command 2-8
 MONEY-SIGN argument, ALTER-LOCALE command 2-8

MONEY-SIGN-LOCATION argument, ALTER-LOCALE command 2-8
 MONTHS-CASE argument, ALTER-LOCALE command 2-9
 MONTHS-FULL argument, ALTER-LOCALE command 2-9
 MONTHS-SHORT argument, ALTER-LOCALE command 2-9
 MOVE 2-164
 multiple reports 2-14
 MULTIPLY 2-168

N

NAME 2-4
 NAME argument 2-46
 CREATE-ARRAY command 2-52
 LOAD-LOOKUP command 2-160
 natural log base e raised to x power 2-132
 NEED argument, NEXT-LISTING command 2-173
 nesting
 arguments 2-129
 IF command 2-120
 nesting levels
 #INCLUDE command 2-122
 NEW-PAGE 2-169
 NEW-REPORT 2-170
 NEXT-COLUMN 2-47, 2-171
 NEXT-LISTING 2-172
 no recovery mode 2-37
 NO-ADVANCE argument, NEXT-LISTING command 2-173
 NO-DUPLICATE 2-5, 2-68
 NOLINE argument
 DISPLAY command 2-95
 SHOW command 2-222
 -NOLIS 2-85
 NOP argument, PRINT command 2-191
 NOPROMPT arguments, INPUT command 2-125
 NORMAL argument, SHOW command 2-222
 NOWAIT, CALL command 2-43
 -NR argument
 BEGIN SQL command 2-37
 BEGIN-SELECT command 2-32
 NULL 2-21
 NUMBER argument
 DISPLAY command 2-94
 dst_any_var command 2-165
 format_mask command 2-165
 MOVE command 2-165
 PRINT command 2-191
 SHOW command 2-222
 NUMBER-EDIT-MASK argument, ALTER-LOCALE command 2-8
 numeric functions 2-129
 10 raised to x power 2-132

- absolute value 2-129
- arccosine 2-129
- arcsine 2-130
- arctangent 2-130
- cosine 2-131
- degrees 2-131
- hyperbolic cosine 2-131
- hyperbolic sine 2-135
- hyperbolic tangent 2-136
- largest integer 2-132
- log base 10 2-133
- log base e 2-133
- natural log base e raised to x power 2-132
- power 2-134
- radians 2-134
- round 2-134
- sign 2-135
- sine 2-135
- square root 2-136
- tangent 2-136
- nl 2-147

O

- ODBC 2-48, 2-215
- ON-BREAK
 - in PRINT command 2-191
 - SET qualifier 2-192
- ON-BREAK , AFTER qualifier with 2-192
- ON-BREAK, BEFORE qualifier with 2-192
- ON-ERROR 2-21
- ON-ERROR argument 7-2
 - BEGIN-SELECT command 2-33
 - BEGIN-SQL command 2-39
 - CONNECT command 2-51
 - DIVIDE command 2-97
 - EXECUTE command 2-106
- OPEN 2-174
- open a new report 2-170
- operands 2-127
- operators 2-128
- Oracle 2-31, 2-37, 2-48, 2-215
- ORIENTATION argument
 - DECLARE-LAYOUT command 2-72
- OUTPUT argument, EXECUTE command 2-107

P

- page numbering 2-126
- page overflow 2-170
- PAGE-DEPTH argument, DECLARE-LAYOUT command 2-72
- PAGE-NUMBER 2-176
- PAPER-SIZE argument, DECLARE-LAYOUT command 2-72
- PARAMETER 2-21

- PARAMETER_LIST 2-22
- PARAMETERS 2-5, 2-21, 2-67
- PASSWORD 2-5, 2-67
- PC 2-42
- PeopleBooks
 - printed, ordering xv
- PITCH argument, DECLARE-PRINTER command 2-78
- PL/SQL 2-36
- POINT-SIZE argument
 - DECLARE-PRINTER command 2-78
- position
 - PRINT-IMAGE 2-209
- POSITION 2-177
- POSTSCRIPT 2-77
- power 2-134
- precision 2-127
- PRINT
 - definition 2-178
 - format commands 2-178
- PRINT qualifier, ON-BREAK argument 2-192
- PRINT-BAR-CODE 2-197
- PRINT-CHART 2-57, 2-200
- PRINT-DIRECT 2-208
- PRINTER
 - WP 2-81
- PRINTER qualifier 2-208
- printer type 2-232
- PRINTER-TYPE argument
 - DECLARE-REPORT command 2-86
- PRINT-IMAGE
 - definition 2-209
 - position 2-209
- printing
 - Hebrew 2-183
 - on Windows 2-82
- PROCEDURE 2-21, 2-22
- procedures
 - running 2-98
 - USE-PROCEDURE command 2-233
- prompt 2-123
- PUT 2-210

Q

- QUIET argument
 - LOAD-LOOKUP command 2-161
 - STOP command 2-216, 2-225

R

- rad 2-134
- range 2-148
- READ 2-212
- RECORD argument, OPEN command 2-174
- rename 2-138

- replace 2-149
- report layout 2-70
- reserved variables
 - #sql-status 2-28
 - \$sql-error 2-28
 - \$username 2-50
- RESET-STRING argument, DECLARE-PRINTER command 2-78
- RETURN_VALUE argument, LOAD-LOOKUP command 2-160
- REVERSE argument, SHOW command 2-222
- reversed characters 2-197
- RIGHT-MARGIN argument, DECLARE-LAYOUT command 2-72
- ROLLBACK 2-215, 2-225
- roman 2-149
- round 2-134
- ROUND argument
 - ADD command 2-2, 2-97, 2-169
 - SUBTRACT command 2-228
- ROWS argument, LOAD-LOOKUP command 2-160
- ROWSET 2-22
- rpad 2-149
- RSV 2-21
- rtrim 2-150

S

- SAVE qualifier
 - ON-BREAK argument 2-192
- SBTOMBS 2-215
- SCHEMA 2-21
- screen I/O 2-220
- SECURITY 2-216
- sequential processing, FOR-READING 2-174
- SET qualifier, ON-BREAK argument 2-192
- SET-GENERATIONS 2-219
- SET-LEVELS 2-219
- SET-MEMBERS 2-220
- SETUP 2-65, 2-66
- SETUP section 2-34
- SHADE argument
 - PRINT command 2-194
- SHOW 2-220
- sign 2-135
- sin 2-135
- sine 2-135
- sinh 2-135
- SIZE argument, CREATE-ARRAY command 2-52
- SKIPLINES argument, NEXT-LISTING command 2-173
- SKIPLINES qualifier, ON-BREAK argument 2-192
- SORT argument
 - BEGIN-SELECT command 2-32
 - BEGIN-SQL command 2-37
 - SORT argument, LOAD-LOOKUP command 2-161
 - sort buffer size 2-37
- SOURCE argument
 - DECLARE-IMAGE command 2-69
 - PRINT-IMAGE command 2-210
- source field 2-164
- SQLBase 2-37
 - Cnn 2-37
 - COMMIT command 2-48
 - DBdatabase 2-37
 - LOCK 2-37
 - NR 2-37
 - ROLLBACK command 2-215
 - SORTnn 2-37
- SQR page buffer 2-208
- sqrt 2-136
- square root 2-136
- STARTUP-FILE argument
 - DECLARE-PRINTER command 2-78
- STATUS 2-21
- STATUS argument
 - INPUT command 2-125
 - OPEN command 2-175
- STOP 2-225
- stored procedures
 - Oracle 2-36
- STRING 2-226
- string functions
 - extract 2-109
 - find 2-110
 - instr 2-143
 - instrb 2-143
 - length 2-144
 - lengthb 2-145
 - lengthp 2-145
 - lengtht 2-145
 - lower 2-146
 - lowercase 2-163
 - numeric position 2-143
 - substr 2-151
 - substring 2-151
 - substringb 2-151, 2-152
 - substrt 2-152
 - transform 2-154
 - translate 2-154
 - unicode 2-155
 - UNSTRING 2-229
 - upper 2-155
 - UPPERCASE 2-230
- string values 2-138
- strtodate 2-150
- subroutines
 - calling 2-42
 - writing 2-40
- substitution variables
 - #DEFINE command 2-92

- #IF command 2-118
- #IFDEF command 2-121
- #IFNDEF command 2-122
- ASK command 2-19
- substring 2-109, 2-229
- substrt 2-152
- SUBTRACT 2-227
- SYBASE
 - COMMIT command 2-48
 - ROLLBACK command 2-215
 - STOP command 2-225
 - USE command 2-230
- SYBASE CT-Lib 2-31
- SYBASE DB-Lib 2-32, 2-37
- SYMBOL-SET argument
 - DECLARE-PRINTER command 2-78
- SYSTEM argument, CALL command 2-43

T

- TABLE argument, LOAD-LOOKUP command 2-160
- tabular reports
 - NEXT-LISTING command 2-172
 - ON-BREAK argument 2-191
- tan 2-136
- tangent 2-136
- tanh 2-136
- TEXT argument
 - DECLARE-VARIABLE command 2-90
 - PRINT-BAR-CODE 2-199
 - TOC-ENTRY command 2-228
- text edit format characters 2-182
- THOUSAND-SEPARATOR argument, ALTER-LOCALE command 2-8
- TIME-SEPARATOR argument, ALTER-LOCALE command 2-8
- to_char 2-153
- to_multi_byte 2-153
- to_number 2-153
- to_single_byte 2-154
- TOC argument, DECLARE-REPORT command 2-85
- TOC-ENTRY 2-228
- TOP-MARGIN argument
 - DECLARE-LAYOUT command 2-72
- transcendental functions 2-137
- transform 2-154
- translate 2-154
- trunc 2-137
- truncate 2-137
- TYPE argument
 - DECLARE-PRINTER command 2-77
- TYPE argument
 - DECLARE-IMAGE command 2-69, 2-210
 - INPUT command 2-125
 - PRINT-BAR-CODE 2-198

U

- UCALL.C 2-41
- UFUNC.C 2-156
- UNDERLINE argument
 - PRINT command 2-195
 - SHOW command 2-222
- unicode 2-155
- Unix 2-42, 2-43
- UNSTRING 2-229
- upper 2-155
- UPPERCASE 2-230
- USE 2-230
- USE-COLUMN 2-47, 2-231
- USE-PRINTER-TYPE 2-232
- USE-PROCEDURE 2-232
- USER 2-4, 2-5, 2-67
- USE-REPORT 2-234

V

- value determination 2-104
- values of STATUS argument 2-124
- VARY argument, OPEN command 2-175
- VERT-LINE argument, GRAPHIC command 2-117
- VMS 2-42, 2-43

W

- WAIT, CALL command 2-43
- WHEN argument, EVALUATE command 2-104
- WHEN-OTHER argument, EVALUATE command 2-105
- WHERE argument, LOAD-LOOKUP command 2-160
- WHILE 2-40, 2-235
- Windows
 - CALL command 2-42
- WITH RECOMPILE argument, EXECUTE command 2-107
- WRAP argument
 - ON 2-196
 - PRINT command 2-195
- wrapdepth 2-155
- WRITE 2-237
- writing to a page 2-178

X

- XC argument, EXECUTE command 2-106
- XP argument
 - BEGIN-SELECT command 2-32
 - BEGIN-SQL command 2-37

Z

ZIP+4 Postnet 2-200

