HP ALLBASE/BRW Reference Manual

for MPE/XL

HP Part No. 35360-90051
Printed in U.S.A.

E0192

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD PROVIDES THIS MATERIAL "AS IS" AND MAKES NO WARRANTY OF ANY KIND, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. HEWLETT-PACKARD SHALL NOT BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE OR USE OF THIS MATERIAL WHETHER BASED ON WARRANTY, CONTRACT, OR OTHER LEGAL THEORY.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

SWT/SOFTWARE TECHNOLOGY CENTER
8000 FOOTHILLS BOULEVARD
ROSEVILLE, CA 95678, U.S.A.

Printing History

New editions of this manual will incorporate all material updated since the previous edition.

The software product part numbers printed alongside the date are the versions and update levels of the software products at the time the manual edition or update was issued.

First Edition	April 1988	35360 A.00.00
First Update	December 1988	35360 A.00.01
Second Edition	January 1992	35360 A.01.20

Preface

This manual documents the A.01.20 release of HP ALLBASE/BRW on HP 3000 computer systems running under the MPE XL operating system.

This manual assumes that the MPE operating system is correctly installed and operating. It also assumes that you have HP TurboIMAGE/XL correctly installed and that you have access to the sample **database**, TOYDB.

The *HP ALLBASE/BRW Reference Manual* contains 22 chapters, a glossary, and an index. The manual is divided into four parts:

- Concepts
- Tasks
- Reference
- Administrative

The Concepts Part discusses general information about HP ALLBASE/BRW, defines and describes tables, and logging on to the MPE XL system. The Tasks Part has step-by-step instructions on each task used in building a report. The Reference Part has two alphabetically arranged chapters on Screens and Calculations. The Administrative Part contains information most likely to be used by the System Administrator, Database Manager, or by a programmer.

Some chapters have a "Special Cases" section that provides examples of sophisticated uses of HP ALLBASE/BRW.

The Glossary defines the meaning of some HP ALLBASE/BRW, HP ALLBASE/SQL, and HP TurboIMAGE/XL terms.

Conventions Used in This Manual

The typographic conventions used in this manual are summarized below.

Notation	Description
user input	When necessary for clarity, user input is indicated like this. For example: BAMBOO: HELLO MGR.ITF3000
`COMPUTER` font	In examples, command names, options, and parameters are shown in uppercase letters, in computer font. For example: `SUM i = 1 TO ?month OF customers.turnover (i)`
bold font	Terms that appear in the Glossary at the end of this manual are printed in **bold** font the first time they are introduced in each chapter or appendix.
Italics font	Words in italics font in examples denote a parameter that you must replace with a suitable name. For example: `DIGIT_OF (`*number*`, `*number*`)`
`shading`	Shaded text represents inverse video on the terminal screen. Also, function keys are depicted in shaded text. For example: Choose `other keys` means to press the function key (for example, f4) corresponding to the highlighted label on the screen.
⎵	The symbol ⎵ indicates a key on the terminal keyboard. For example, Spacebar indicates the spacebar.
␣	When necessary for clarity, the symbol ␣ is used to indicate a required blank or an exact number of blanks.

Contents

1

About HP ALLBASE/BRW

An overview of HP ALLBASE/BRW

At a Glance

HP ALLBASE/BRW is a high performance **business report writing system** for information management professionals. Its ease-of-use significantly reduces the time needed for report development. It runs on any HP 3000 running MPE XL.

HP ALLBASE/BRW is available in two versions: the full developer system, and the run-time version. The developer system contains all the programs and utilities necessary to develop reports and to run them. The run-time version (available on the MPE XL Fundamental Operating System tape) provides everything necessary for application users to run developed reports.

Note

HP BRW/V is the same report writing system for the MPE V operating system. See Chapter 21 for more information about HP BRW/V.

Features of HP ALLBASE/BRW

Report tuning

You can tune report generation to optimize performance and conserve system resources.

Layout features

The sample layout feature of HP ALLBASE/BRW allows you to verify the format and design of the report without compiling, so you can see how your report will look as you design it.

Calculation language

HP ALLBASE/BRW has a powerful calculation language that lets you perform both simple and complex arithmetic operations. Operations can be performed on numeric data, string values, and date and time figures. The calculation language allows nested operations, conditional execution, **array** calculations, and string manipulation. You can also define your own functions or use the predefined functions supplied with HP ALLBASE/BRW.

Interface to HP DeskManager

HP ALLBASE/BRW has an interface to HP DeskManager so that you can distribute your reports electronically.

Multiple application support

HP ALLBASE/BRW supports multiple data manager applications. You can access data in:

- HP ALLBASE/SQL **tables**
- HP TurboIMAGE/XL **data sets**
- **files**
- other **reports**

Multiple pass reporting

You can output data from a specific report to an **intermediate report file** and then use the report file to generate further reports.

This chapter covers

- elements of HP ALLBASE/BRW
- requirements
- HP ALLBASE/BRW files
- sample databases

Elements of BRW

The Developer Version

The Report Writer

The report writer is the main part of the HP ALLBASE/BRW developer system. You use the report writer to specify, compile, and execute reports.

BRWCONV

BRWCONV converts existing reports, created with Report/V or Inform/V, into HP ALLBASE/BRW format.

BRWCONV translates all Inform/V features; no manual modifications are necessary. BRWCONV does not translate all Report/V features; some are not available or not necessary in HP ALLBASE/BRW. See Chapter 22 for more details.

Run-time Only Version

The run-time only version of HP ALLBASE/BRW is part of the Fundamental Operating Software (FOS) for MPE XL. It consists of two programs:

BRWSTART — The BRWSTART program is included with the MPE XL operating system. BRWSTART allows you to request a report in batch mode, accessing the Request Report screen from outside HP ALLBASE/BRW.

BRWEXEC — With BRWEXEC, you can run a report successfully outside of HP ALLBASE/BRW, even if the HP 3000 you use does not have HP ALLBASE/BRW installed. The BRWEXE program is part of the MPE XL operating system and is all you need to run a compiled HP ALLBASE/BRW report.

See Chapter 11, "Requesting Reports" for more information about BRWSTART and BRWEXEC.

HP BRW-Desk

HP BRW-Desk/XL is an add-on product to HP ALLBASE/BRW that delivers reports electronically to one or more people using HP DeskManager. A single report can be split electronically, at specified levels, and each part distributed to a list of HP DeskManager users.

Requirements

The following sections describe the hardware and software requirements for using HP ALLBASE/BRW.

What You Need

To use the developer version

- Have the MPE operating system correctly installed and operating.
- Have the developer version of HP ALLBASE/BRW installed and configured.
- For non-SQL reports, you need the BRWDIC file, which is used to create and compile non-SQL reports. To create the BRWDIC file, you need:
  - the HP Dictionary/V, HP System Dictionary, HP Application Dictionary, or HP Access Server Dictionary (formerly the HP Access Central Dictionary).
  - or, you can generate a dictionary source file from which to create the BRWDIC file. (See Chapter 18).
- For SQL reports, you need the HP ALLBASE/SQL production database (or a test database with the same table structures, owner names, and access rights).

To use the run-time only version

- Have the MPE operating system correctly installed and operating, including the BRWEXEC and BRWSTART programs.
- Have the report execution file created by the developer version of HP ALLBASE/BRW.
- Have the database from which the information will be extracted.
- Have any intermediate report files required by the report.

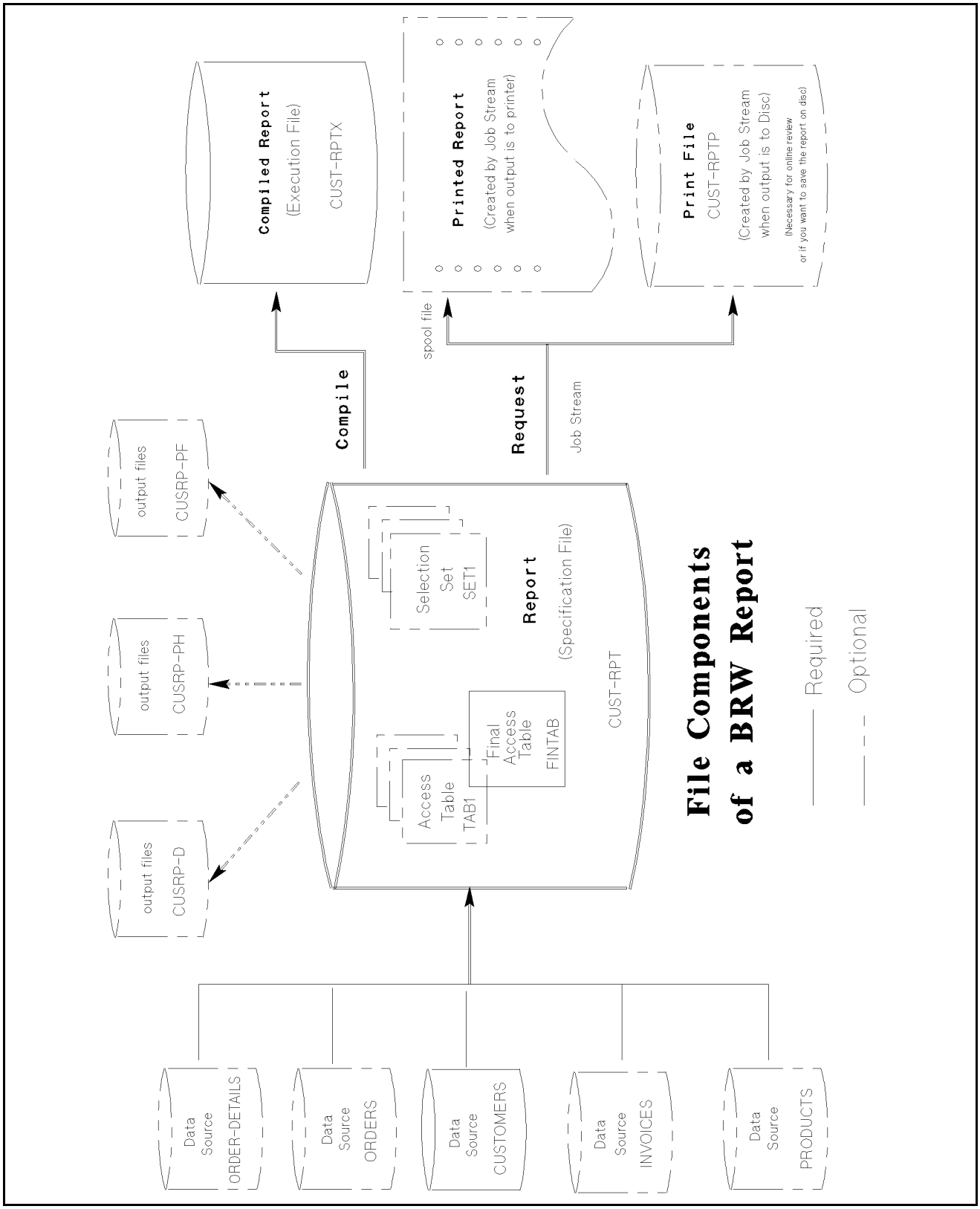
Note

The version of the operating system on which the report is run must always be the same as, or later than, the version of the operating system on which the report was defined.

To use HP BRW-Desk

- Have the BRW-Desk XL software installed.
- Have HPDESKManager version B.03.00 or later installed and running.
- Have the HPDESK INTRINSICS on the MPE XL system. The HP Desk Intrinsics are part of HP DESKManagerPlus.

See Chapter 13 for details about BRW-DESK XL.


File Components
of a BRW Report
Required
Optional
Compiled Report
(Execution File)
CUST-RPTX
Printed Report
(Created by Job Stream
when output is to printer)
spool file
Print File
CUST-RPTP
(Created by Job Stream
when output is to Disc)
(Necessary for online review
or if you want to save the report on disc)
Compile
Request
Job Stream
Report
(Specification File)
CUST-RPT
Selection
Set
SET1
Access
Table
TAB1
Final
Access
Table
FINTAB
output files
CUSRP-PF
output files
CUSRP-PH
output files
CUSRP-D
Data
Source
ORDER-DETAILS
Data
Source
ORDERS
Data
Source
CUSTOMERS
Data
Source
INVOICES
Data
Source
PRODUCTS

BRW Files

When you create a report with ALLBASE/BRW, you use as well as create files on the MPE XL system. The figure shown on the previous page represents the basic files used or created by ALLBASE/BRW and also shows some logical components of the ALLBASE/BRW report file.

This section describes these files and logical components:

- data sources
- created files
- temporary files
- logical components of the specification file

Data Sources

You build a table using from 1 to 14 source tables. Source tables can be any of the following:

- IMAGE data sets
- HP ALLBASE/SQL tables
- KSAM files
- MPE files
- ALLBASE/BRW table, which is another table defined in the report, for example, when using a series of tables
- intermediate report file (output file) from another ALLBASE/BRW report

When you create a ALLBASE/BRW report, you need to know the names of these data sources.

Created Files

Some files are created as a result of creating, compiling, or requesting the report. These files are saved in the MPE group and account where you started ALLBASE/BRW, or you can put them in another group by adding a group name to the path.

- Specification file (the report specifications)
- Execution file (the compiled report)

  When you compile the report, this MPE file is created containing the compiled version of the report.

- Print file

  If you request a report and designate `DISC` as the Print Device, you create an MPE file that contains a printer-ready version of the report. You can print it later to the printer, or you can use the Online Review feature to preview the report.

- Output file (intermediate report file)

You can save certain **linesets** from your report by sending them to an output file. The output file contains only the **items** from the lineset. Output files and formats are discussed in Chapter 12.

Temporary Files

While ALLBASE/BRW is creating and processing your report, some temporary files, like workfiles and sortfiles, are created. These files are eliminated when the processing is complete.

Logical Components of the Specification File

Within the specification file, you create some logical entities that are collections of report specifications. You can give these entities unique names within the report (like table1 and set1) but they don't need to be unique in your group and account. They are not files and do not appear when you use LISTF. Some of these entities include:

- Access tables

  The access tables are created from the data sources. They do not physically contain data, but give a logical definition of one or more source tables.

  You can add one or more access tables to the report. Access tables added to the report can, in turn, be used as source tables (in that report) for more efficient report tuning. These tables are not files and do not appear in the MPE directory. Before you can compile your report, you must consolidate all the tables into one **final access table** containing all the data you need for the report.

- Selection sets

  A selection set is an optional specification for determining parameters needed for data selection, job scheduling, and printing characteristics such as number of copies, name of the print device, and so on.

  You can define multiple selection sets, or you can choose not to define any selection set. Selection sets are discussed in greater detail in Chapter 9.

Sample Databases

HP ALLBASE/BRW is supplied with one sample database (TOYDB), an HP TurboIMAGE/XL database. Another database (EXDBE) is an example database for HP ALLBASE/SQL data. The EXDBE database is not supplied with HP ALLBASE/BRW.

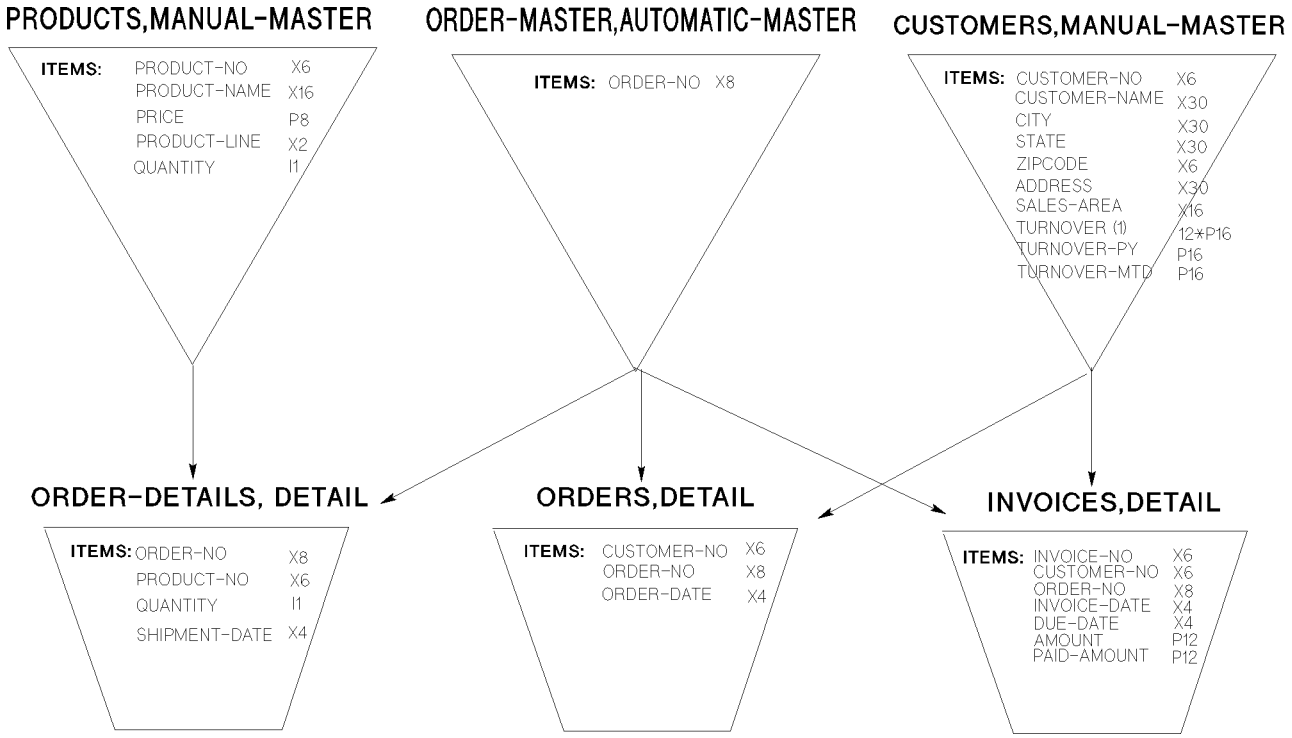
Note

In this manual, the term, IMAGE database, is used to represent both IMAGE and HP TurboIMAGE/XL databases.

The TOYDB Database

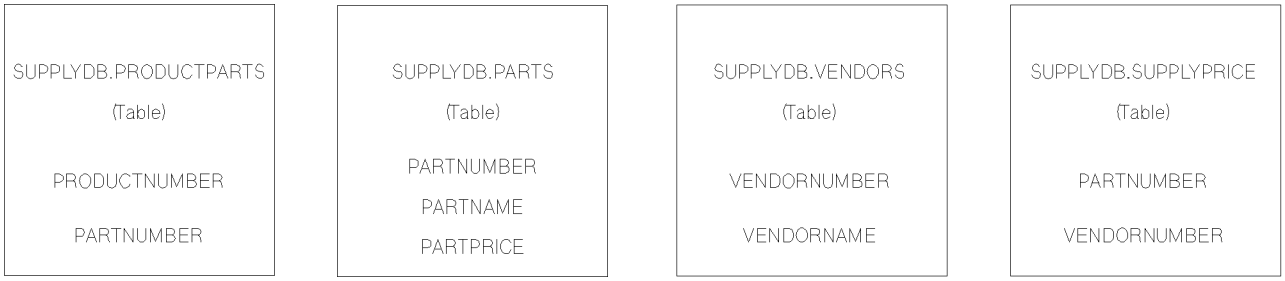
This sample HP TurboIMAGE/XL database has been designed for use in the HP ALLBASE/BRW training course and is the basis for many of the example screens shown in this manual. It is usually located in TOYDB.PUB.ITF3000, but ask your system manager or database manager to be sure that is the correct location for your system.

TOYDB contains three master data sets: PRODUCTS, ORDER-MASTER, and CUSTOMERS; and three detail data sets: ORDER-DETAILS, ORDERS, and INVOICES. The diagram below represents the TOYDB database. Master data sets are shown as triangles. All the items in the sample are shown with their respective types and lengths.



The EXDBE Database

In this manual, a database called EXDBE is used as the basis for several HP ALLBASE/SQL examples. This database is an example only, and does not exist as part of HP ALLBASE/BRW. The following figure shows the organization of this sample database.



If you would like to try some sample reports from a real HP ALLBASE/SQL database, use the PARTSDBE database provided with HP ALLBASE/SQL.

The PARTSDBE Database

The HP ALLBASE/SQL database PARTSDBE.SAMPLEDB.SYS is composed of tables that contain data on parts and vendors. The database is supplied with HP ALLBASE/SQL. It is organized like this:

Limits on HP ALLBASE/BRW

Item Types

HP ALLBASE/BRW supports the following item types:

Strings

Strings with up to 255 characters. Items greater than 255 characters are truncated.

Numeric

Packed decimals, zoned decimals, and display numerics in the range +/-9,999,999,999,999.999999. HP ALLBASE/BRW internally converts them into binary-coded decimal numbers with a total of 19 digits, including 6 decimals. Zoned decimals (type Z) must be Trailing Overpunch or No Sign. See Chapter 18.

HP Business BASIC floating-point decimal numbers. HP ALLBASE/BRW converts these to packed decimal numbers with a total of 19 digits, including 6 decimals. An overflow is set to the OVERFLOW exception value, an underflow to zero.

64-bit integers are supported, and are converted to numeric. An overflow is set to the OVERFLOW exception value.

Integer

16-bit and 32-bit integers. HP ALLBASE/BRW internally converts 16-bit integers into 32-bit integers.

Real

32-bit and 64-bit real numbers of either HP Real or IEEE Real format. HP ALLBASE/BRW converts 32- bit real numbers internally to IEEE-type 64-bit real numbers.

Date

Date types, as described in Chapter 18. HP ALLBASE/BRW internally converts them into strings of the form YYYYMMDD.

Time

Time types, as described in Chapter 18. HP ALLBASE/BRW internally converts them into binary-coded decimal numbers with a total of 19 digits including 6 decimals. These numbers contain time values expressed as seconds.

Database and File Types

HP ALLBASE/BRW supports the following types of databases and files:

IMAGE Database — **IMAGE and TurboIMAGE databases**. The database passwords supplied with the datasets specified in HP ALLBASE/BRW reports must allow access to all items from these datasets, unless Item Level Security is specified in the Tune Access screen.

HP ALLBASE/SQL Database — **HP SQL database**. Access authorization is checked by HP ALLBASE/SQL for MPE logon user. Users developing or executing a report must have CONNECT authority to the DBEnvironment and SELECT authority on all tables or views specified.

KSAM Files — **KSAM files with fixed length records**. All types of KSAM keys (no duplicate; duplicate; random, etc.) are supported.

MPE Files — **Standard sequential MPE files with fixed length records**. If a file has the CCTL attribute, the carriage control character must be reflected in the format of that file (first byte in the format). RIO, CIR and MSG files are not supported.

Printers

HP ALLBASE/BRW supports the following printers:

- Spooled printers which support the following carriage control codes:

%40	single space
%53	no space, return
%61	page eject
%101	set pre-space option
%103	set single space option without automatic page eject
%202 - %277	skip 2 - 63 lines
%320	no space, no return

You can avoid using carriage-control codes by specifying NOCCTL for printfiles.

Environment files can be used if the printer supports it.

Note

Printers connected to terminals (non-spooled printers) are *not* supported.

Report Characteristics

HP ALLBASE/BRW reports have the following characteristics:

- up to 300 characters in the report width
- up to 999 lines in the page length
- up to 9 sort levels
- up to 50 parameters and MATCHES statements in formulas (that is, both numbers together cannot exceed 50)

What Next?

Chapter 2 contains concept information about relational data, including tables.

If you need help using editing keys or using the on-line help, read Chapter 3, "Communicating With HP ALLBASE/BRW." Chapter 4 describes how to log on to your MPE XL system and how to start HP ALLBASE/BRW.

If you are ready now to start creating, compiling, or running HP ALLBASE/BRW reports, use chapters 5 through 12 of Part I for task instructions. To look up reference information, use Part II. Part III is useful information for system managers and programmers.

2

Understanding Relational Data and Tables

Describing the relational model

At a Glance

HP ALLBASE/BRW uses a **relational model** for manipulating data. The relational model looks at data as if it were in a table, with rows and columns. The model is concerned with **data structure**, **data integrity**, and **data manipulation**. All data is represented by explicit value.

The table below represents a typical relational model for data:

CUSTOMER-NO	CUSTOMER-NAME	CITY	SALES-AREA	STATE	ZIPCODE
223	Nancy Watson	1411 Oak	Roseville	CA	95999
789	Mike Barnes	232 Benton Dr.	Cool	CA	95888
456	Frank Heartney	111 Tudor Lane	Citrus Heights	CA	95601

When you work with the data, you select columns and qualify rows in one or more tables to create another table containing a specific set of rows and columns of values.

In HP ALLBASE/BRW the rows are called **records** and the columns are **items**. For instance, in the sample table shown above, the column containing customer numbers is an **item** and each row containing customer information is a **record**. Wherever an item and a record meet, you have a **value**. For instance, *Frank Heartney* is the value for CUSTOMER-NAME in the record for customer number 456.

However, unlike an ordinary table, the columns and rows are not arranged in any particular order in a relational database model. You define the relationship you want to exist in your table.

Tasks for adding tables and defining relations are discussed in Chapter 6.

This chapter covers

- about tables
- about relations

About Tables

There are two main kinds of tables: source tables and access (result) tables.

Source Tables

You build an access table using from 1 to 14 source tables. Source tables can be any of the following:

- IMAGE data sets
- HP ALLBASE/SQL tables
- KSAM files
- MPE files
- HP ALLBASE/BRW tables—That is, other access tables defined in the same report. These tables are useful when you are using a series of tables in the same report.
- intermediate report file (output file)—Files from another HP ALLBASE/BRW report.

With HP ALLBASE/BRW, you define *which* data and *what* calculations or qualifications you want separately from *how* you get the data. This lets you tune data access for best performance after the report is completed, without the risk that the report does not print the correct data if you change access methods.

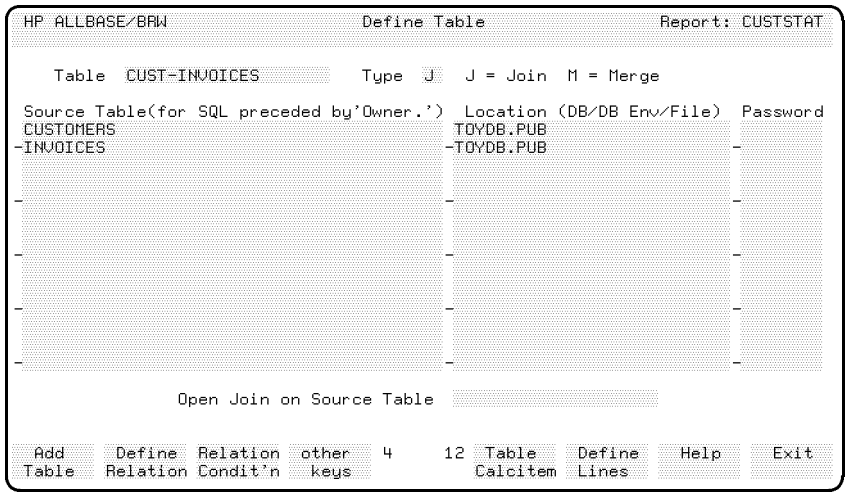
Single-source Tables

Single-source tables access data from only one source, that is, one data set, table, or file. To specify a single-source table, you define it as a joined table on the Define Table screen, but you only list that one table. The table below is an example of a table built from one source table, the CUSTOMERS data set.

CUSTOMER-NO	CUSTOMER-NAME	CITY	SALES-AREA	STATE	ZIPCODE
C2	Boot Inc.	Miami	Western	Florida	46670
C7	Gambler	Reno	Southern	Nevada	23348
C12	Eiffel SA	Paris	Eastern	France	99999
C27 . . .	Shoe Spa	Rome	Eastern	Italy	77777
C9999	Last's Ties	FA	Northern	Alaska	66743

Multiple-source Tables

Multiple-source tables access data from more than one data set, table, or file. Multiple-source tables are listed on the Define Table screen. You can **join** or **merge** a multiple source table.



Joining and merging tables is completely described in Chapter 6.

Access Tables

You use **relational data access** to combine one or more source tables into a new table, called an **access table** (or result table that contains data items from the source tables that you need in your report. You can define the following:

- Which source tables you need to combine into the final access table and then use that table to create a report.
- How you combine source tables: by joining, open joining, or merging them.
- Which common items you want to use to join the source tables.
- Which table-specific selections you need to make. Only those records satisfying the relation condition are selected from the source tables.
- Which items you transfer (project) from the source tables to the final table. Only items that are projected can be used in reports.
- What calculations you need to make using the items. Items are calculated when data is retrieved from the source table and the result is included in the new table.

The table built as a result of these above operations can itself be used as a source table. Using the result table as a source table in another table is called nesting tables. When you have all the data that you require for your report in one table, that table is used in the report and is known as the **final access table**.

About Relations

A **relation** specifies which records from the source tables are combined in the new table. Each relation consists of an item that is common to several source tables. The new, joined table consists of records in which the common item entries of all source tables have identical values.

Relations can be thought of as filters which extract from all possible combinations only those records which match the relation (that is, have the same value).

Relations need not affect all source tables. For example, if you joined four tables, you could define a relation that affects two, three, or all four tables. Whether or not you define a relation, you can use relation conditions (described in Chapter 6) to further restrict records written to a joined table.

When you define a relation between two source tables, you are saying: "I want a record to be written to the joined table when *item1* of *table1* = *item2* of *table2*."

What Next?

For basic starting information such as logging on to MPE and running HP ALLBASE/BRW, see Chapter 4.

If you are ready to create reports, Chapter 5 has specific task instructions.

Tasks for defining relations are described in Chapter 6.

3

Communicating with HP ALLBASE/BRW

Moving from screen to screen, entering data, editing fields, understanding messages and prompts, and getting online help

At a Glance

If you are just learning HP ALLBASE/BRW and have not used another report writing system or other Hewlett-Packard database management product, this chapter can help you communicate with HP ALLBASE/BRW.

Note

Information in this chapter assumes that you are using HP ALLBASE/BRW from an MPE XL terminal and are using a standard terminal keyboard.

If you are using another keyboard or if you access the MPE XL system over a network from another system, you may need to make some adjustments. For instance, on a PC keyboard, the Return key may be marked Enter. You should not assume that this is equivalent to the terminal Enter key. If the PC keyboard has a keypad with another Enter key, that will probably be the equivalent to the terminal Enter key.

It is up to you to use the proper keyboard for optimum success.

This chapter covers

- using HP ALLBASE/BRW screens
- using softkeys
- using Quick Browse
- getting online help
- understanding messages and prompts

Using HP ALLBASE/BRW Screens

HP ALLBASE/BRW is screen driven. You define a report by moving from screen to screen and entering the specifications for the report.

HP ALLBASE/BRW data screens have one or more input fields that allow you to enter and edit data. Data screens can also have display-only fields on them. These fields display system or application information. You cannot change the information in display-only fields.

Entering and Editing Data

When HP ALLBASE/BRW first displays a data screen, it positions the cursor at the first column of the first input field.

You can type data into any input field. To enter the data, press Enter or a specific function key.

Editing Fields

You can use any of the following terminal keyboard keys to edit data or to move the cursor within a data entry field:

- The Insert Char key toggles insert mode on or off. When you are in insert mode, typing any character inserts the character in front of the cursor, between existing characters. This causes all characters to the right of the cursor to move across the field by one character.
- The Delete Char key deletes the character at the cursor.
- The Backspace key moves the cursor one column to the left within the current field. Moving the cursor back past the left edge of the field has no effect on the field.
- The ◀ key moves the cursor one column to the left within the current field. Moving the cursor back past the left edge of the field has no effect on the field.
- The ▶ key moves the cursor one column to the right within the current field. Moving the cursor past the right edge of the field has no effect on the field.
- The ▲, ▼, ◀, and ▶ keys can move the cursor to a position outside the current field. If you enter any data, the cursor is immediately returned to the start of the next field in the tabbing sequence that follows the cursor position and the new data is inserted there.
- The Tab key moves the cursor to the next input field.
- The Shift Tab combination moves the cursor to the start of the previous input field if the cursor is at the first column of the current field. If the cursor is not at the first column, pressing Tab moves the cursor to the start of the current field.
- The ▼ key moves the cursor to the first input field on the current screen.
- The Clear Line key clears the current field from the cursor position to the right end of the field.

- The Clear Display key clears all input fields from the current cursor position to the bottom of the screen.

Note

The Return key has no function in HP ALLBASE/BRW. If you press Return, the cursor moves to the first column of the terminal screen.

Using Softkeys

Eight labels (softkeys), displayed at the bottom of all HP ALLBASE/BRW screens, indicate the functions that you can access from the screen. The softkeys correspond to the eight function keys on the terminal keyboard, f1 to f8. Softkeys are indicated in this manual by a shaded rectangle (for instance, `Define Table`). These softkeys correspond to the terminal funtion keys as shown below:

Note

To select a function, this manuals asks you to "Choose `softkey`." That means you need to press the corresponding function key on the terminal keyboard.

A blank softkey means that the function key associated with that softkey has no function for the current screen.

The `other keys` Softkey

Some HP ALLBASE/BRW screens have more than one set of softkeys (for example, the Define Lines screen has four sets). When there are other sets of softkeys for the same screen, the `other keys` softkey is available so you can switch between the sets.

To access other keys

1. Choose `other keys`.

   The next set of softkeys appears.

2. Choose `other keys` again to display additional sets of softkeys or return to the first set.

Using Quick Browse

If a screen displays `Next ...` and `Prev ...`, you can skip directly to a particular element in a list.

To skip to a particular element in a list

1. In the highlighted field, type the name of the element (item, table, parameter, and so on) you want to see.
2. Choose `Next ...` or `Prev ...`.

HP ALLBASE/BRW searches for the next (or previous) occurrence of the element with the name you entered and displays it.

For example, suppose you are on the Project Items screen and the list of items extends for several pages of screens. The item you want to view is not displayed on the current page of the screen. You would type the name of the item in the highlighted field and press `Next Items`. The item that you request appears at the top of the list.

Using Quick Copy

HP ALLBASE/BRW provides a quick method of copying functions, tables, items, and parameters. You can use Quick Copy at most of the definition screens throughout HP ALLBASE/BRW.

To copy

1. Type the name of the new function, table, item, or parameter on the appropriate screen and press the (CLEAR DSPLY) key on the terminal keyboard.
2. Choose `Add xxxx`. (Where xxx is Table, Item, Function, or Param.)
3. When the system prompts, type the name of the function, table, item, or parameter you want to copy.
4. Press (Enter). The table, item, function, or parameter (and all the definition that goes with it) is copied to the new name.

Getting Online Help

Online help is available throughout HP ALLBASE/BRW. When you press the `Help` function key, a help screen with information about that screen appears.

Often, the help screens have more than one "page."

HP ALLBASE/BRW help screens display these softkeys to help you move through the information.

- (f5) `Prev Page` HP ALLBASE/BRW displays this key if there is more than one page of help for the current screen, and you are on the second or subsequent page. Press this key to redisplay the previous page of help.
- (f6) `Next Page` HP ALLBASE/BRW displays this key if there is a further page of help linked to the current help screen. Press this key to display the next page of help.
- (f8) `Exit HELP` Press this key to restore the original HP ALLBASE/BRW screen and resume processing at the point where you asked for help.

Note

(Page Up) and (Page Down) on your keyboard do not function on the Help screens.

Understanding Messages and Prompts

HP ALLBASE/BRW displays several types of messages on the screen (usually on the last line):

- system-active messages
- grant messages
- warning messages
- error messages

System-Active Messages

HP ALLBASE/BRW displays a system-active message while it performs the action you requested. System-active messages are displayed in normal blinking video. For example, when you enter a report name on the Select Report screen and press Enter, HP ALLBASE/BRW displays the following system-active message:

```
Loading specifications
```

Grant Messages

HP ALLBASE/BRW displays a grant message when an action has completed. For example, when you make changes to your report specification and press Enter, HP ALLBASE/BRW displays the following grant message:

```
Report modified
```

Warning Messages

Warning messages tell you that your request may cause a problem. Note that HP ALLBASE/BRW executes the requested action even though the warning message is displayed. For example, if you request an empty report specification file on the Select Report screen, HP ALLBASE/BRW displays the following warning message on the Define Report screen:

```
WARNING: Specification file is empty
```

Error Messages

HP ALLBASE/BRW displays an error message if you enter invalid data in a screen field and press Enter, press an incorrect key, or attempt an incorrect operation. For example, if you press a function key that has no label, HP ALLBASE/BRW displays the following error message:

```
Invalid key pressed
```

If an error is caused by entering invalid data in a screen field, you must correct the error before you can continue entering data or proceed to the next step.

Error messages for formula and conditional screens are displayed *above* the formula.

For more information about error messages and a list of common messages for different tasks, see the index section of this manual under "errors."

Prompts

On occasion, when you need to enter information, HP ALLBASE/BRW displays a prompt on the screen. You can do any of the following when a prompt appears:

- Type an answer and press (Enter) or press the function key that initiated the action.
- Press the `Cancel Prompt` function key and resume working on the screen.
- Press a different function key to cancel the prompt and initiate a new action.

When a prompt appears, `Exit` changes to `Cancel`. For example, if you press `Exit BRW` on the Select Report screen, HP ALLBASE/BRW prompts you to confirm or cancel your action. While the prompt is displayed, `Exit BRW` is changed to `Cancel`.

What Next?

If you are ready to start HP ALLBASE/BRW read Chapter 4. If you are ready to create HP ALLBASE/BRW reports, read Chapter 5.

To look up reference information, use Part III. Part IV is useful information for system administrators and programmers.

4

Getting Started With HP ALLBASE/BRW

Logging in to the MPE operating system and starting HP ALLBASE/BRW

At a Glance

This chapter covers

- logging in to MPE
- running HP ALLBASE/BRW
- leaving HP ALLBASE/BRW

Logging in to MPE

To log in to MPE

1. At the operating system prompt, type:

   : HELLO *username.acctname,groupname* Return

   For example:

   : HELLO MGR.ITF3000,BRWSPEC Return

2. If the system manager has assigned a password, MPE prompts you to enter it. When you see the password prompt, type your password and press Return.

   ```
   ENTER USER PASSWORD:
   ```

   MPE does *not* display the password characters as you type.

When you have logged in to MPE, you can run HP ALLBASE/BRW.

Additional notes

If you have other information about your system logon, write it in the space provided here.

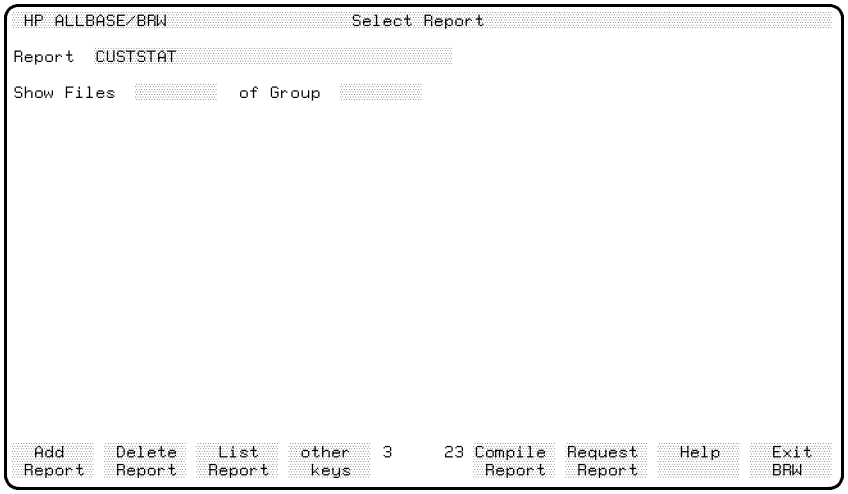
Running HP ALLBASE/BRW

To run HP ALLBASE/BRW

1. At the MPE prompt, type:

   : RUN BRWXL.PUB.SYS (Return)

   The Select Report screen appears.



   The Select Report screen is the main screen for HP ALLBASE/BRW All report tasks begin here.

Additional notes

If you have other information about starting HP ALLBASE/BRW (such as UDCs), write it in the space provided here.

Leaving HP ALLBASE/BRW

To leave HP ALLBASE/BRW

1. Press `Exit` until you arrive at the Select Report screen.
2. At the Select Report screen, press `Exit BRW`.
3. Press (Enter) to confirm leaving HP ALLBASE/BRW.

What Next?

The next chapters describe HP ALLBASE/BRW tasks you do for creating reports.

For information about creating, compiling, or running the report, see the following chapters:

To create a report, go to Chapter 5 through Chapter 9.

To compile a report, go to Chapter 10.

To run a report, go to Chapter 11.

For help with specific screens, see Chapter 16.

5

Report Writing with HP ALLBASE/BRW

Creating a report specification file

At a Glance

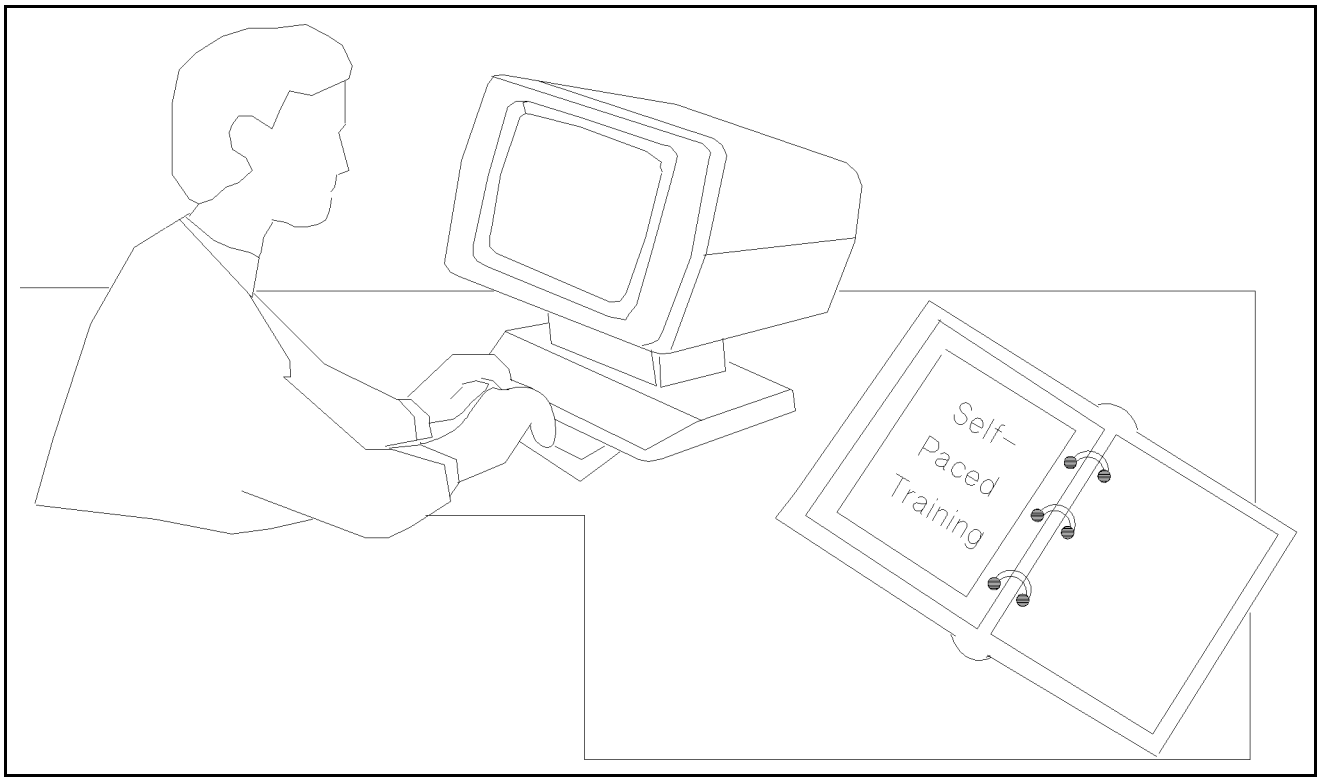
This chapter is a general overview of the process of creating a report.

- First, define the general report characteristics, such as report width and page length, on the Define Report screen.
- Supply data access information.
- Next, on the Define Breaks screen, specify how the report is to be sorted and paginated.
- Define calculations and conditions.
- Then, compose the report layout by positioning text and items for each lineset on the bottom portion of the Define Lines screen.
- Format item values on the Item Edits screen.

When you have finished creating the basic report, you have a **report specification file** that contains all the specifications for your report.

If you are not familiar with any report writer tool, or have not gone through the HP ALLBASE/BRW tutorial, this chapter will give you an idea of how to use HP ALLBASE/BRW for writing business reports.

We highly recommend going through the HP ALLBASE/BRW tutorial, *HP ALLBASE/BRW Self-Paced Training*, if you are new to HP ALLBASE/BRW. This chapter, as well as all other chapters in this reference manual, will make more sense if you have had some HP ALLBASE/BRW hands-on experience.



This chapter covers

- planning ahead
- defining a report
- defining a table
- defining relations

Planning Ahead

Before you attempt to create a report with HP ALLBASE/BRW, you need to have a plan for the report. The following table shows the steps you need to take to plan the report and shows how the plan can be implemented in HP ALLBASE/BRW.

Planning and Implementing

Planning Steps	Implemention in HP ALLBASE/BRW
Find out where the information you need is stored. HP ALLBASE/BRW can use IMAGE data sets, HP ALLBASE/SQL tables, KSAM files, MPE files, or other HP ALLBASE/BRW tables and intermediate files as data sources for the report. To effectively plan your report, you need to know how the data source tables are structured: For instance, the types of data in each column of the table (customer name, address, and so on).	Use the Define Table screen to identify the data source(s) for your new table. You need a complete path to the data. In addition to the path to the data, you also need to confirm that the necessary data sources are defined in your data dictionary.
Decide if more than one data source is needed.	The sources can be combined on the Define Table by indicating a (J)oin or a (M)erge. **Join** and **Merge** are discussed in Chapter 2. To join data sources, you need to find a relation between the data sources by identifying one ore more common items (columns) in each table: for example, the customer name. Define the common items on the Define Relation screen.
Decide on which information you need from the tables. For instance, do you want records on customers who have past due accounts, or records on customers with orders over $1000?	You can qualify the data you select by specifying a relation condition on the Relation Condition screen. Only those records that satisfy the condition are selected from the data source(s).
Plan how you want the data to appear on the printed page: Do you want **headings** or **footings**? How many lines of data do you want? Do you want the customers to appear in alphabetical order, or in order by customer number?	You can use the Define Lines screen, the Define Breaks and Pagination screen, and other report layout screens to manipulate the data in many ways. Use the Review Layout screen to look at the resulting layout.
Do you want to perform calculations on the data? Do you want running totals? Do you want to create a new column from two more more existing items or from an existing item and a formula?	There are four screens that let you perform calculations on your report: Layout Calc Item screen, Layout Calc Item Formula screen, Layout Function screen, and Layout Function Formula screen.

Creating a Report

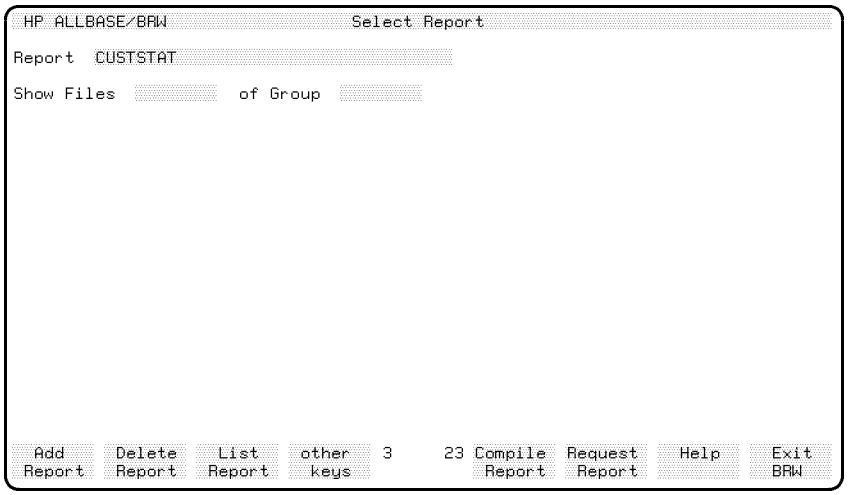
After you have planned the report to your satisfaction, start HP ALLBASE/BRW and create the report.

To create a report

1. Start at the Select Report screen (the first screen when you start HP ALLBASE/BRW).

   (For instructions on starting HP ALLBASE/BRW, see Chapter 4).

2. Type the name of the report you want to create in the `Report` field. This will be the name of the report specification file that HP ALLBASE/BRW creates.



3. Choose `Add Report`.

4. At the prompt that appears, enter the name of the report form (if any) that you want to copy, or press Enter to create a new report.

   ```
   Copy from report      or press ENTER.
   ```

   The Define Report screen appears.

To define the report

1. Type a password in the `Password` field if you need security on this report.

2. Type a report description in the `Description` field. The description can be anything that will help you identify the report. This field is optional.

3. Check the global report characteristics (described in the next section) and make sure that they are correct for your report.

```
HP ALLBASE/BRW                  Define Report                  Report: CUSTINV

 Report Specification Password   MONEY

 Report Description    Customer Invoice Report

 Report Width          132         Number of Horizontal Pages  1

 Page Length           60

 Special Paper

 String Sorting Sequence          A  ( A = ASCII, E = EBCDIC, N = National )

 SQL Transaction Isolation Level     ( RU=Read Uncommitted,RC=Read Committed,
                                       CS=Cursor Stability,RR=Repeatable Read )

 USE Access of Report  INVTAB.SPEC

 Final Access Table

Define  Define  Define  other   20* 36  Define  Output   Help    Exit
Table   Lines   Breaks  keys      n     Select'n File
```

4. Press Enter to record the report description. The message: `Report modified` appears.

Global Report Characteristics

On the Define Report screen, there are several global report characteristics that have a default value already assigned. These characteristics are described below to help you select the correct ones for your report.

Paper Characteristics

Paper characteristics are:

- Report width (1 - 300)
- Page length (1 - 999)
- Optional special paper to be used for printing a report such as pre-printed check forms. A message requesting that special paper be mounted on the printer is sent to the operator console each time the report is printed. After mounting and aligning the special paper, the operator replies that printing can start now (see the *Console Operators Guide* for your system).
- Number of Horizontal Pages. You can set the number of horizontal pages for each physical page. HP ALLBASE/BRW can print a number of pages, side-by-side, on one physical page.

String Sorting Sequence

You can specify whether to sort strings according to ASCII, EBCDIC or National Language Rules. ASCII is fastest.

SQL Transaction Isolation Level

This option lets you determine whether other users can modify the table while you are reading it. If you allow others to modify the table

while you are reading the data, you may get only partially updated information. RC (read committed) is the default. If this field is blank, the execution value in the BRW Configuration File is used.

Data Access

Data access can be defined in either of these two ways:

- **USE Access** by typing the data access definition of another report at the Select Report screen (in the `USE Access of Report` field).
- **Self-Defined Access** by setting up a data access definition within the report you are defining at the Define Table screen.

To use the data access of another report

1. Type the specification file name of the other report in the `USE Access of Report` field on the Define Report screen, as shown on the following sample screen:

```
HP ALLBASE/BRW                  Define Report                  Report: CUSTINV

 Report Specification Password   MONEY

 Report Description    Customer Invoice Report
 Report Width          132         Number of Horizontal Pages  1
 Page Length           60
 Special Paper
 String Sorting Sequence          A  ( A = ASCII, E = EBCDIC, N = National )
 SQL Transaction Isolation Level     ( RU=Read Uncommitted,RC=Read Committed,
                                       CS=Cursor Stability,RR=Repeatable Read )
 USE Access of Report  INVTAB.SPEC
 Final Access Table    CUSTOMER-INVOICE

Define  Define  Define  other  4     35 Define  Output   Help    Exit
Table   Lines   Breaks  keys              Select'n File
```

2. Press Enter.

To set up self-defined data access

Self-defined data access uses the Tune Access and the Parameter screens and is discussed in Chapter 6.

What Next?

If you want to know more about data access, read Chapter 6.

If you are ready now to start defining the details of the report. See chapters 6 through 10 of Part II for complete task information.

To look up reference information, use Part III. Part IV is useful information for system managers and programmers.

6

Defining Data Access

Adding tables, defining relations, placing items, and tuning access

At a Glance

Data access includes all the methods you use to access the data in source tables or files. These methods can be one or more of the following:

- adding tables
- joining or merging tables
- defining relations
- deciding on which items to use in the final report
- tuning the data access

At the simplest level, you access one data source and use all the items in the final report. But you can also choose more than one data source, join or merge data sources, and define *access tables* for the report that consist of joined or merged data sources. Once defined, access tables can appear as source tables in subsequent access tables for that report. (Access tables do not exist as logical files outside of the report specification.)

When you join tables, you combine them in such as way that each record in the resulting access table can have a value for every item in the original tables. When you join two or more source tables, you must define a **relation** between the records in each table, based on a common item.

When you merge tables, you *append* the records of additional tables to the end of the records for the first table. There does not need to be any relationship between the tables merged. That is, there does not need to be a common relation item.

You may not want to use all the combined items that result from a join or a merge. Selecting only the items that are necessary in your report is an optional part of defining data access.

To access the data in the most efficient (that is, time-effective) way, you can use methods that will effectively *tune* your data access.

This chapter covers

- joins and relations
- merged tables
- data from different sources
- items
- precision of numbers
- data access methods
- relation conditions

Joins and Relations

Joining Tables

You join two or more source tables by:

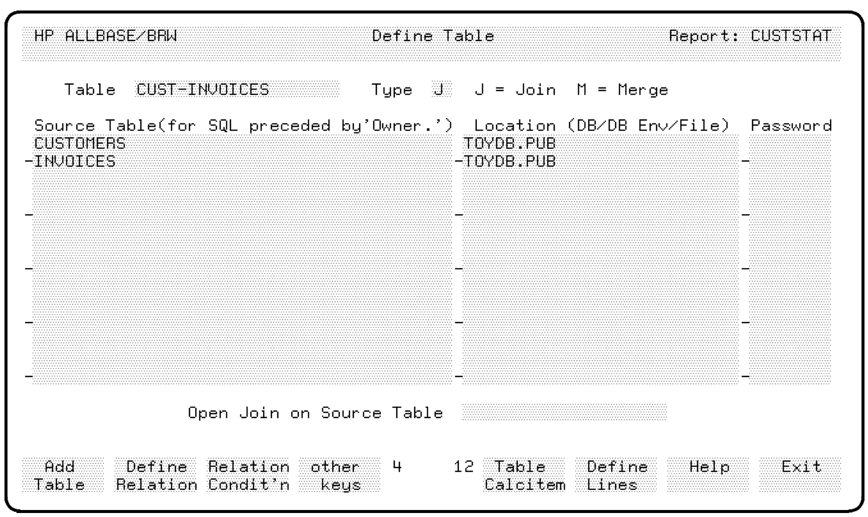
- naming the table
- specifying the type of join
- listing the data sources
- defining a **relation** between the records in each table. The relation is based on a common item.

You may specify an **open join** to further qualify the results you get from the join.

A join is not complete until you have defined the relation between the joined tables.

To join tables

1. On the Define Table screen, type the name for a new table in the `Table` field.
2. Type `J` in the Type field (for join).
3. In the `Source Table` fields, type the names of the source tables from which the data for the new table is to be extracted. Specify *owner_name.table_name* for each HP ALLBASE/SQL table.
4. In the `Location` fields, type the path to each source table.
5. Type any applicable passwords in the `Password` fields.
6. If this is an open join, type the name of the source table you want for the source that can contain NO_VALUE if no records match the relation.



7. Choose `Add Table` to add the joined table as an access table for this report.

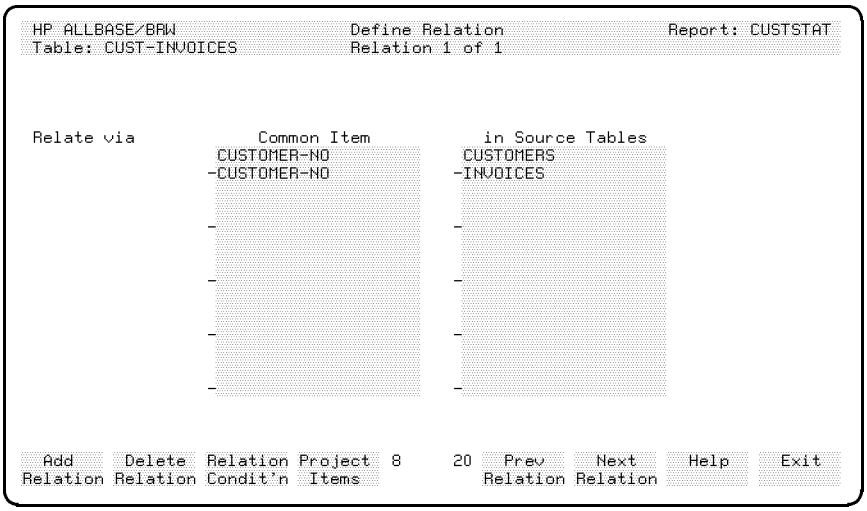
Defining Relations

To define a relation

1. Go to the Define Relation screen by choosing `Define Relation` on the Define Tables screen.

   An example of the Define Relation screen is shown below.

2. Type the name of each item from the source tables that you want to define as a common item.



   If you have one or more items with the same name in different tables, just type one name in the Common Item column.

3. Press Enter.

The Define Relations screen is described in greater detail in Chapter 16.

Common items have these characteristics:

- They must be of the same type, (strings can only be related to strings, dates to dates, and so on). Types are described in Chapter 15. String items do not have to be of the same length.
- They can have different names in the source tables.
- They need *not* be key items (that is, IMAGE search items or KSAM keys).

Note

HP ALLBASE/BRW does not require that the relation is logical.

That is, you could relate, say, CUSTOMER-NO item of the INVOICES table with the SALES-AREA item of the CUSTOMERS table, since both items are the same data type (strings). (If two strings of differing lengths are compared, the shorter is padded with blanks.) A record would then be written to the new table, CUST-INVOICES, when the value for CUSTOMER-NO was equal to the SALES-AREA. This example is, of course, not terribly sensible. It is up to you to make the relation logical so that you get meaningful results in your report.

Caution

If you do not specify a relation, HP ALLBASE/BRW writes all possible combinations of records to the result table. In other words, HP ALLBASE/BRW will relate all possible records in one table with all possible records in another, and you may get an enormous table full of unstructured data.

Example: Join

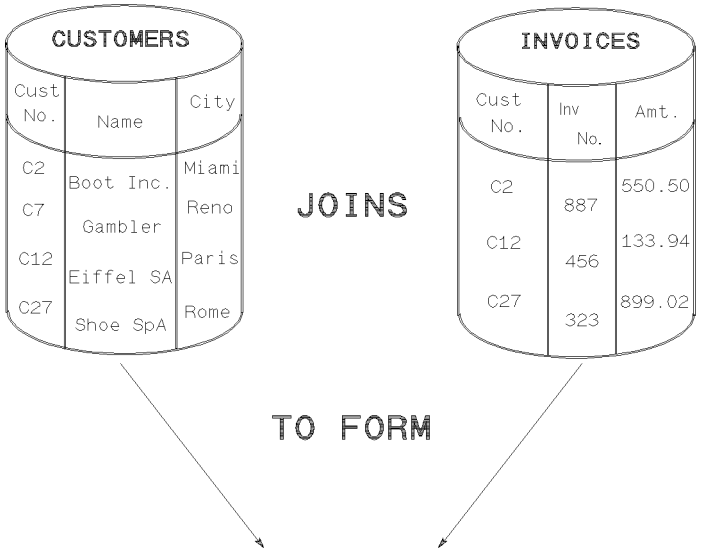
Suppose you want to create a report CUSTINV, using the data sources CUSTOMERS and INVOICES.

You join the source tables CUSTOMERS and INVOICES to create the final access table, CUST-INVOICES. If you define the relation using the common item CUSTOMER-NO, HP ALLBASE/BRW reads each table and writes a record to the result table each time the value of CUSTOMER-NO for a record in CUSTOMERS is equal to the CUSTOMER-NO of a record in INVOICES.

If you did not specify a relation for table CUST-INVOICES, each record in CUSTOMERS would be related to each record in INVOICES. If there were 5000 records in CUSTOMERS and 10000 recods in INVOICES, CUST-INVOICES would contain 50,000,000 records! Since you would only want to have this happen in certain rare cases, make sure that you have defined all the necessary relations before running a report.

Using CUSTOMER-NO as the common item, the record written to the result table contains a value for every item in both source tables (provided that the item names are unique). Because the item CUSTOMER-NO exists in both sources, the CUSTOMER-NO item from only one table is written to each record in the result table. (You can, however, supply an alias name for CUSTOMER-NO in the other table so that both items are written to the result table. This is explained in "Projecting Items").

The join of CUSTOMERS and INVOICES on CUSTOMER-NO to produce the result table CUST-INVOICES is illustrated in the following figure:



Cust No.	Name	City	Inv No.	Amount
C2	Boot Inc.	Miami	887	551.50
C12	Eiffel SA	Paris	456	133.94
C27	Shoe SpA	Rome	323	899.02

CUST-INVOICES

Each record of the table CUST-INVOICES contains invoice information and all the customer information for the customer to whom the invoice applies.

Notice these things about the CUST-INVOICES table:

- Only those records in the source tables CUSTOMERS and INVOICES that have a matching value of the common item CUSTOMER-NO are written to the result table, CUST-INVOICES.
- The customer number C7 has no counterpart in the INVOICES table and, therefore, does not appear in the result table.

For cases of multiple records for one CUSTOMER-NO, multiple records are written to the result table. For instance, if customer number C2 had two invoices listed in the INVOICES table, a record for each invoice would appear in the result table.

Open Joins

Open joins let you join source tables and report all records from one source table, even if these have no corresponding records in the other source table. Items in the open join source table are set to NO_VALUE when there are no corresponding records.

Note

NO_VALUE is an exception condition and is described in Chapter 15. NO_VALUE indicates that an item has no value. For example, a numeric item will be set to NO_VALUE if it is projected from an open join source table and has no value in that table. Note that NO_VALUE is not the same as a numeric item being equal to zero.

This Define Table screen shows how the open join for these tables is defined:

```
 HP ALLBASE/BRW                  Define Table                Report: CUSTSTAT

    Table  CUST-INVOICES        Type  J   J = Join  M = Merge

 Source Table(for SQL preceded by'Owner.')  Location (DB/DB Env/File)  Password
 CUSTOMERS                                  TOYDB.PUB
-INVOICES                                  -TOYDB.PUB                 -

-                                          -                          -

-                                          -                          -

-                                          -                          -

-                                          -                          -

                 Open Join on Source Table  INVOICES

  Add     Define  Relation  other   4     12  Table    Define     Help      Exit
 Table   Relation Condit'n  keys             Calcitem   Lines
```

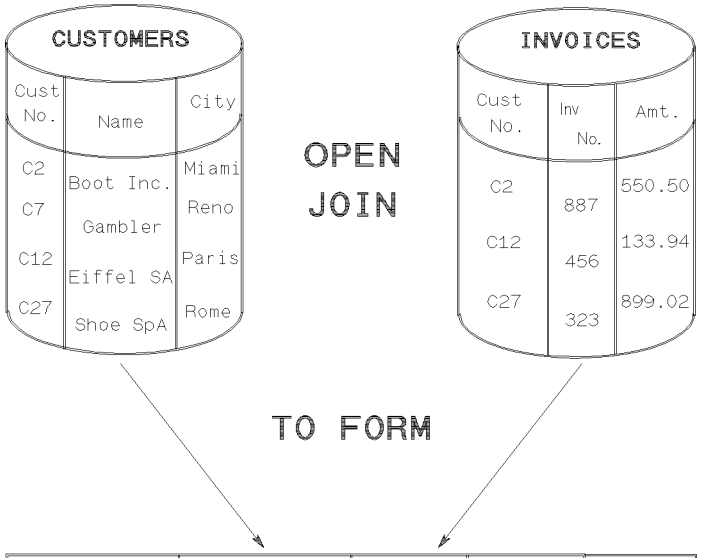
Example: Open Join

Suppose you want to join CUSTOMERS and INVOICES to produce a table CUST-INVOICES, as in the previous example. But now you also want to report on all customers, whether or not they have invoices. The previous example will only write records to CUST-INVOICES when a customer has an invoice, that is, when CUSTOMER-NO in CUSTOMERS equals CUSTOMER-NO in INVOICES. If a customer has no invoices the CUSTOMER-NO in CUSTOMERS cannot match a CUSTOMER-NO in INVOICES and no record will be written. But you can include all customers by specifying an open join on the INVOICES source table.

In this case, when CUSTOMER-NO in CUSTOMERS matches CUSTOMER-NO in INVOICES, a record is written as before. But if a CUSTOMER-NO in CUSTOMERS has no corresponding entry in INVOICES, a record will also be written because of the open join on INVOICES.

You can specify only one source table on each Define Table screen as an open join. If you want open joins on more than one source table, see "Multiple Open Joins" later in this chapter.

The next illustration shows what happens when there is an open join on INVOICES. All items coming from INVOICES will be set to NO_VALUE in the resulting row when no invoice is present.



Cust No.	Name	City	Inv No.	Amount
C2	Boot Inc.	Miami	887	551.50
C7	Gambler	Reno	——	——
C12	Eiffel SA	Paris	456	133.94
C27	Shoe SpA	Rome	323	899.02

CUST-INVOICES

NO_VALUE is signified in this table by a long dash (-------). Note that now a record for customer GAMBLER (CUSTOMER-NO C7) is written to CUST-INVOICES, although there is no record with CUSTOMER-NO C7 in INVOICES.

Multiple Open Joins

Since you can only specify one open join per table, nested tables must be used to specify multiple open joins.

To nest tables

1. Decide how you want to group your source tables as access tables.
2. Define an access table as described in "Joining Tables" earlier in this chapter.
3. Define the relation for the first table.

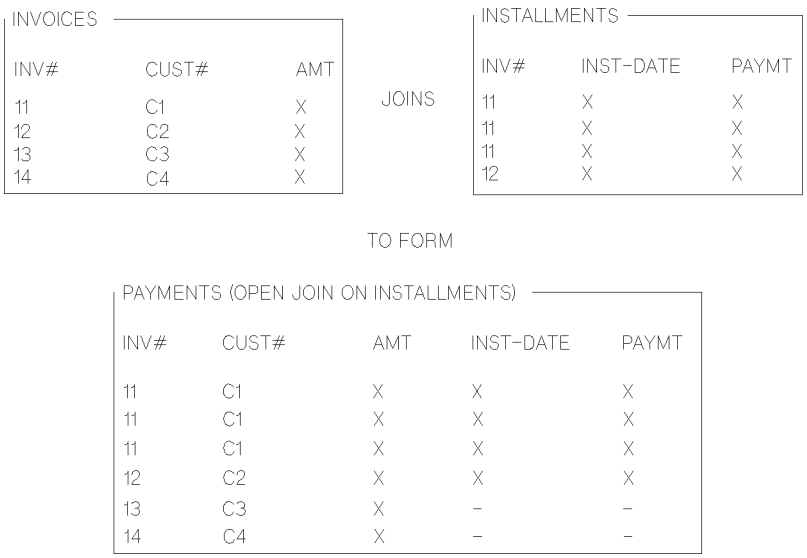
4. Return to the Define Table screen.
5. Type the definition information for the next table. You can type over the previous definition and use the space bar to remove extra characters if you need to.
6. Choose `Add Table`. **Do not** press Enter (instead of `Add Table`).! *Pressing* Enter *will redefine the first table instead of adding a new table.*
7. Define the relation for the new table.

You can repeat the above steps for as many tables as you want to nest. The maximum number of source tables you can use in a single access table is 14.

Example: Nested Tables

Suppose you want to report customers, invoices, and installments and include customers with no invoices and invoices with no installments. The join must be split into two tables because you need to have two open joins.

One table, called PAYMENTS, joins the source tables INVOICES and INSTALLMENTS with an open join on INSTALLMENTS, as shown here:



The Define Table screen for the first open join is shown below:

```
HP ALLBASE/BRW                    Define Table                    Report: CUSTINV

    Table  PAYMENTS                 Type  J   J = Join   M = Merge

 Source Table(for SQL preceded by'Owner.')  Location (DB/DB Env/File)  Password
 INVOICES                                   TOYDB.PUB
-INSTALLMENTS                              -INSTDB.PUB                -

                Open Join on Source Table  INSTALLMENTS
```

The other table, shown next, joins CUSTOMERS and PAYMENTS with an open join on PAYMENTS (the access table created by the previous join).

CUSTOMERS

INV#	CUST#	AMT
11	C1	X
12	C2	X
13	C3	X
14	C4	X

JOINS

PAYMENTS

INV#	CUST#	AMT	INST-DATE	PAYMT
11	C1	X	X	X
11	C1	X	X	X
11	C1	X	X	X
12	C2	X	X	X
13	C3	X	-	-
14	C4	X	-	-

TO FORM

CUSTINV (OPEN JOIN ON PAYMENTS)

CUST#	CUST-NAME	INV#	AMT	INST-DATE	PAYMT
C1	X	11	X	X	X
C1	X	11	X	X	X
C1	X	11	X	X	X
C2	X	12	X	X	X
C3	X	13	X	-	-
C3	X	14	X	-	-
C4	X	-	-	-	-

The Define Table for this open join looks like this:

```
HP ALLBASE/BRW                    Define Table                    Report: CUSTINV

    Table  CUSTINV                  Type  J   J = Join   M = Merge

 Source Table(for SQL preceded by'Owner.')  Location (DB/DB Env/File)  Password
 CUSTOMERS                                  TOYDBE.PUB
-PAYMENTS                                  -                          -

                Open Join on Source Table  PAYMENTS
```

CUSTINV is the final access table.

Open Join on HP ALLBASE/SQL Tables

With HP ALLBASE/BRW, a user can define an open join using HP ALLBASE/SQL tables.

Example: Open join on HP ALLBASE/SQL table

The following example is for a report of all vendors and the parts they sell, including those vendors that do not sell parts. The two data sources are VENDORS and SUPPLYRPICE in the SUPPLYDB DBEnvironment, as shown on this Define Table screen.

```
HP ALLBASE/BRW                    Define Table                    Report: VENDORS

    Table  ALL-VENDORS            Type  J   J = Join  M = Merge

 Source Table(for SQL preceded by'Owner.')  Location (DB/DB Env/File)  Password
 SUPPLYDB.VENDORS                          EXDBE.PUB
-SUPPLYDB.SUPPLYPRICE                     -EXDBE.PUB                  -

                Open Join on Source Table  SUPPLYPRICE
```

The name of the table specified in the `Open Join on Source Table` field is the table (in this case SUPPLYPRICE) for which NO_VALUE is used if there is no corresponding record. In other words, the new table will show a record for a vendor that does not sell parts, since the vendor name is in the PAYMENTS table, but NO_VALUE will appear in the record where the parts sold would appear.

The relation using the common item VENDORNUMBER is shown on this Define Relation screen:

```
HP ALLBASE/BRW                  Define Relation                 Report: VENDORS
Table: ALL-VENDORS              Relation 1 of 1



 Relate via          Common Item          in Source Tables
                  VENDORNUMBER           VENDORS
                 -VENDORNUMBER           -SUPPLYPRICE
```

In a join, you can use the same access sequence for an HP ALLBASE/SQL query, but with an *open join* you cannot. The access sequence is specified on the Tune Access screen. For this open join, the Tune Access screen will look like this:

```
HP ALLBASE/BRW                    Tune Access                   Report: VENDORS
Table: ALL-VENDORS

                                                                 IMAGE
                                                                -------
                     Table Access                          Lock Open Item Level
    Source Table     Type Sequence       Key              Mode Mode Security
 VENDORS             SQL     1
-SUPPLYPRICE        -SQL    -2   -                         -    -    -
```

HP ALLBASE/SQL will create a separate workfile for each HP ALLBASE/SQL query. For more information about tuning the data access, see "Data Access Methods" later in this chapter.

Joining a Source Table To Itself

You can join records from one source table to other records in the same source table. Since the source table names in a table must be unique, nested tables can be used to join a source table to itself, as in the following example.

Example: Self-join

The data set EMPLOYEES contains three items: EMPLOYEE-NUMBER, EMPLOYEE-NAME and SUPERVISOR-NUMBER. Suppose you want to report each supervisor and the employees she manages. These steps are necessary:

- Create the table SUPERVISORS, which only has one source table, EMPLOYEES, as shown on this section of the Define Table screen:

```
HP ALLBASE/BRW                    Define Table                    Report: SUPERS

   Table  SUPERVISORS            Type  J   J = Join  M = Merge

 Source Table(for SQL preceded by'Owner.')  Location (DB/DB Env/File)  Password
 EMPLOYEES                                  EMPDB.PUB
```

- Create the table EMPL-SUPERS which joins EMPLOYEES and SUPERVISORS, as shown here:

EMPLOYEES

EMPL#	EMPLNAME	SUPERV#
1	THOMAS	6
2	PETER	1
3	JEAN	2
6	RICHARD	14

JOINS

SUPERVISORS

EMPL#	EMPLNAME	SUPERV#
1	THOMAS	6
2	PETER	1
3	JEAN	2
6	RICHARD	14

This is what the Define Table screen would look like for this join:

```
HP ALLBASE/BRW                    Define Table                    Report: SUPERS

   Table  EMPL-SUPERS            Type  J   J = Join  M = Merge

 Source Table(for SQL preceded by'Owner.')  Location (DB/DB Env/File)  Password
 EMPLOYEES                                  EMPDB.PUB
-SUPERVISORS                               -                          -
```

- Relate the two tables by the EMPLOYEE-NUMBER from the SUPERVISORS table and the SUPERVISOR-NUMBER from the EMPLOYEES table as shown below:

```
HP ALLBASE/BRW                Define Relation                Report: SUPERS
Table: EMPL-SUPERS            Relation 1 of 1


Relate via          Common Item          in Source Tables
                  EMPL#                SUPERVISORS
                -SUPERV#               -EMPLOYEES
```

- In your new table, you want the item EMPLNAME from the SUPERVISORS table to be called SUPERVISOR-NAME. To rename it, give the item an alias name as shown here:

```
HP ALLBASE/BRW                Project Items                Report: SUPERS
Table: EMPL-SUPERS                 Type: J

        Item              Source Table    Type Proj     Alias Name      NumPrec
EMPLNAME                  SUPERVISORS       S   X  SUPERVISOR-NAME
```

- The final joined table looks like this:

EMPL-SUPERS

EMPL#	EMPLNAME	SUPERV#	SUPERVISOR-NAME
1	THOMAS	6	RICHARD
2	PETER	1	THOMAS
3	JEAN	2	PETER

Joining HP ALLBASE/SQL Tables With Identical Table Names

Within HP ALLBASE/BRW, tables are identified by their table names only. Therefore, HP ALLBASE/SQL table names belonging to one BRW table must be unique. Winthin HP ALLBASE/SQL, however, you may have identical table names with different owners. To use these tables in HP ALLBASE/BRW, use one of the HP ALLBASE/SQL tables to define an HP ALLBASE/BRW table and give it a unique name. You can then join the HP ALLBASE/BRW table and the other HP ALLBASE/SQL table.

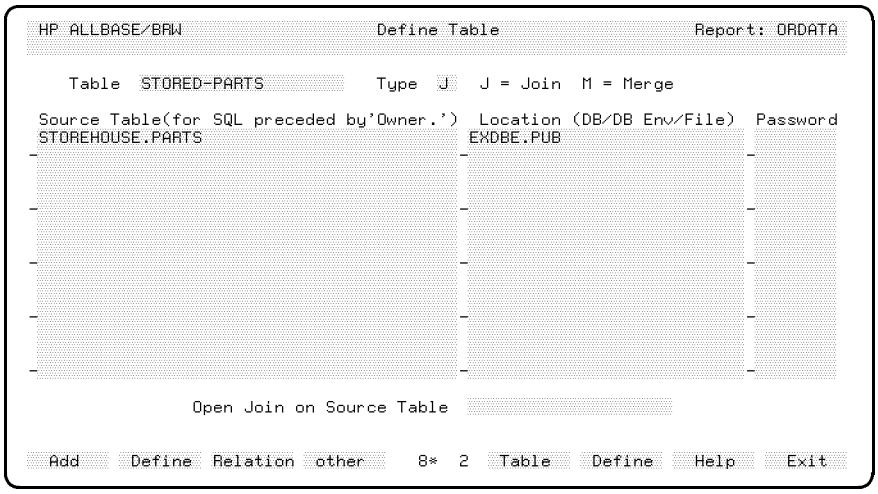
Note

The naming of HP ALLBASE/SQL tables with identical names should be avoided whenever possible.

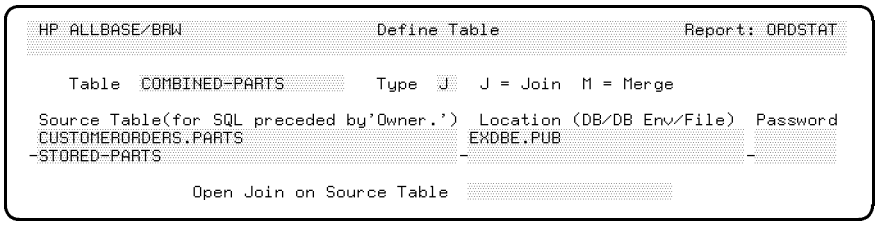
Example: Join with identical table names

In this example, the HP ALLBASE/SQL tables STOREHOUSE.PARTS and CUSTOMERORDERS.PARTS have different owner names (STOREHOUSE and CUSTOMERORDERS), but the same table names (PARTS).

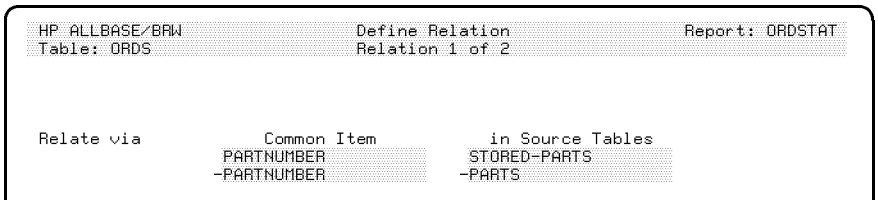
- Create the HP ALLBASE/BRW table STORED-PARTS from the HP ALLBASE/SQL table STOREHOUSE.PARTS.



- Create the HP ALLBASE/BRW table COMBINED-PARTS. This holds the results of joining the HP ALLBASE/BRW table (STORED-PARTS) and the HP ALLBASE/SQL table (CUSTOMERORDERS.PARTS).



- Join the two HP ALLBASE/BRW tables by using the PARTNUMBER in each table.



Merged Tables

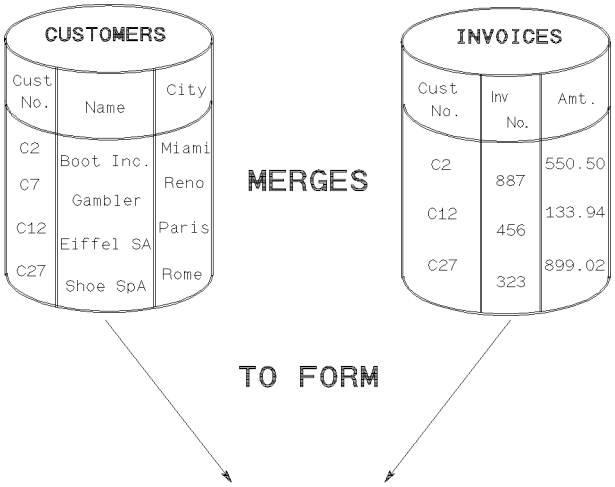
By merging tables, one source table is appended to another to form a new table containing all the information from both tables.

To merge tables

1. On the Define Table screen, type the name for a new table in the `Table` field.
2. Type `M` in the `Type` field (for merge).
3. In the `Source Table` fields, type the names of the source tables from which the data for the new table is to be extracted. Specify *owner_name.table_name* for each HP ALLBASE/SQL table.
4. In the `Location` fields, type the path to each source table.
5. Type any applicable passwords in the `Password` fields.
6. Choose `Add Table`.

Example: Merge

The following example *merges* the source table CUSTOMERS with the source table INVOICES to produce the result table CUST-INVOICES. To see the difference between a join and a merge, look at the diagram of the join (shown earlier in this chapter under the "Joining Tables" section) and compare it to the diagram of a merge (shown below).



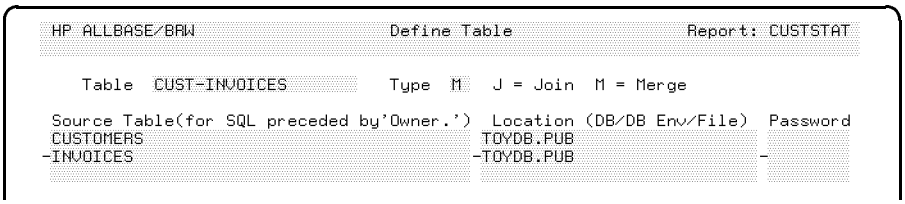
Cust No.	Name	City	Inv No.	Amount
C2	Boot Inc.	Miami	–	–
C7	Gambler	Reno	–	–
C12	Eiffel SA	Paris	–	–
C27	Shoe SpA	Rome	–	–
C2	–	–	887	551.50
C12	–	–	456	133.94
C27	–	–	323	899.02

CUST–INVOICES

In this example, the CUSTOMERS source table does not contain the items INVOICE-NO and AMOUNT which appear in the merged table. Records coming (projected) from this source table will contain NO_VALUEs for these items in the merged table. Likewise, items CITY, NAME, and so on, appear only in source table CUSTOMERS and will be set to NO_VALUE in records projected from source table INVOICES.

Note that records are not related when tables are merged. Thus customer Boot Inc, (CUSTOMER-NO C2), has more than one record in the result table.

The following section of the Define Table screen shows how the above source tables are merged:



Note

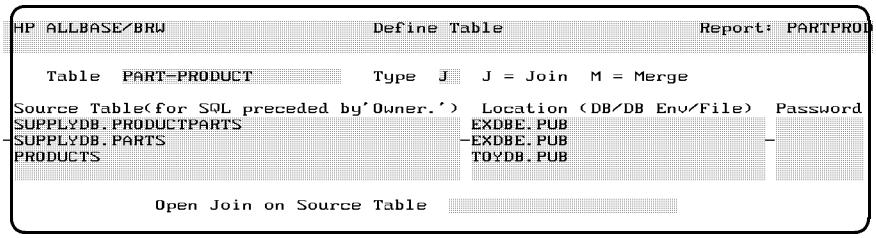
You cannot specify calculated items or relation conditions in a merged table. To use a calculated item in a report with a merged table, first create a joined table with the calculated item and then merge it with the other source tables.

Data from Different Sources

With HP ALLBASE/BRW it is possible to show, on one report, data coming from an IMAGE database, an HP ALLBASE/SQL database, MPE files, and KSAM files.

To combine data from different sources

1. Define the table, relations, and relation conditions just as you would any other table. Remember to specify the *owner_name.table_name* for any HP ALLBASE/SQL tables.



2. Use the Tune Access screen to tune the access for the tables in the most efficient way for the data sources you are using. See "Tuning Data Access" in this chapter for more information about data tuning.

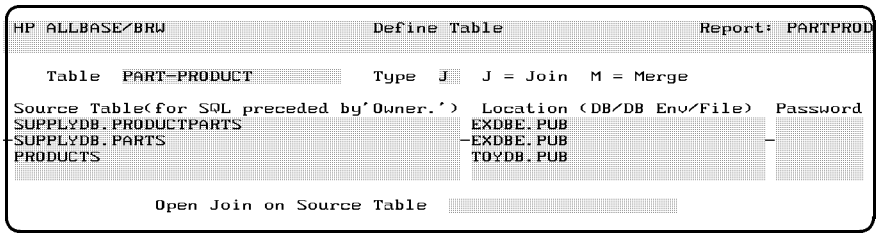
Example: Combine data

The following example combines data from an IMAGE data set and two HP ALLBASE/SQL tables. The sample databases used are shown in Chapter 1.

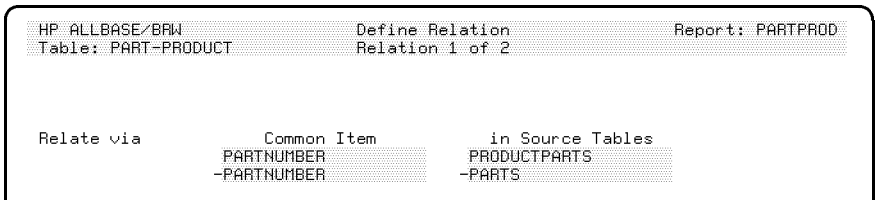
The IMAGE database (TOYDB) has a master data set PRODUCTS, which contains product information (PRODUCT-NO, PRODUCT-NAME, PRICE). The HP ALLBASE/SQL DBEnvironment (EXDBE) has a table SUPPLYDB.PRODUCTPARTS, which lists the parts used to manufacture the product (PRODUCTNUMBER, PARTNUMBER), and a table, SUPPLYDB.PARTS, which contains information on the parts (PARTNUMBER, PARTNAME, PARTPRICE).

The information on the report PARTPROD has to show the part information (PARTNUMBER, PARTNAME, PARTPRICE) for parts that have a part price greater than or equal to 1000.00. For the parts satisfying this condition, the report also has to show the product information (PRODUCT-NO, PRODUCT-NAME, PRICE) for all products that contain this part as long as the product price is less than 5000.00.

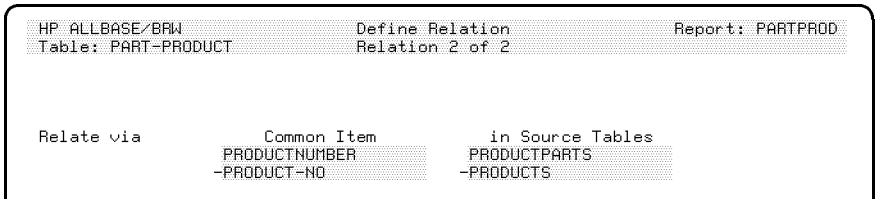
- Create the table PART-PRODUCT. This will be the final access table, which contains all the data items required for the report. Note that for HP ALLBASE/SQL tables, *owner_name.table_name* is specified in the Source Table field.



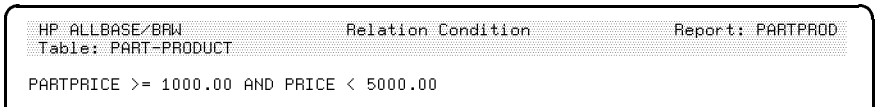
- Join the two HP ALLBASE/SQL tables PARTS and PRODUCTPARTS using PARTNUMBER. This provides the product numbers for each part.



- Join the HP ALLBASE/SQL table PRODUCTPARTS and the IMAGE data set PRODUCTS using the product number. This provides the product information for each product the part is used in.

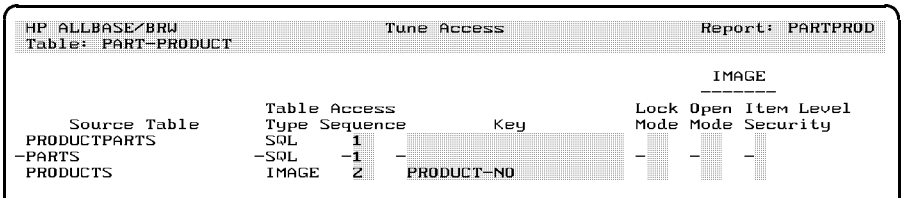


- Create the formula that selects only parts with a part price greater than or equal to 1000.00 and products with a product price less than 5000.00.



- The Tune Access screen shows the table types and the order in which tables and data sets are accessed for this example. The following table show the access sequence number for the tables used in this example, with an explanation about what happens.

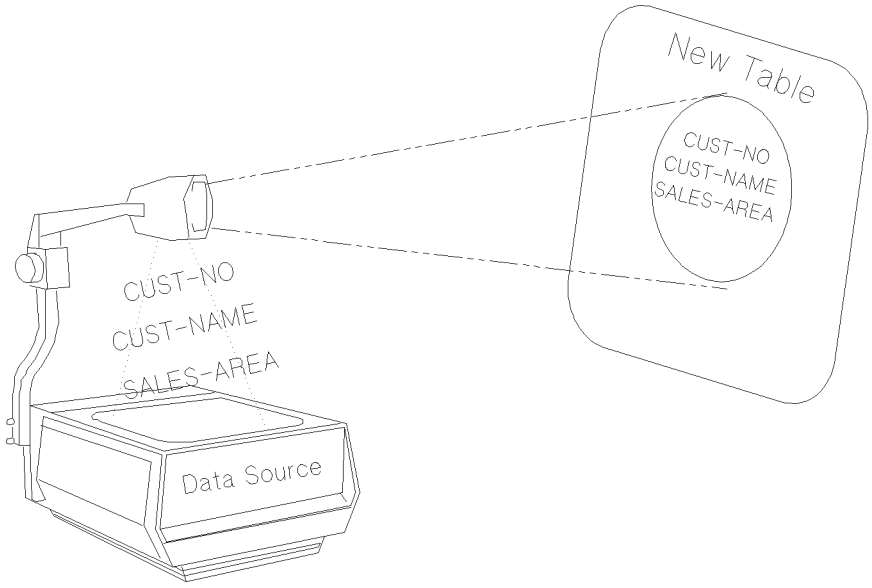
Access Sequence No.	Table	Explanation
1	PRODUCTPARTS and PARTS	HP ALLBASE/BRW generates an HP ALLBASE/SQL query to retrieve parts with a PARTPRICE greater than or equal to 1000.00 where there is a corresponding PRODUCTNUMBER (according to the relation defined for these two SQL tables). For the HP ALLBASE/SQL tables, the default access sequence number is the same if the tables belong to the same DBEnvironment and are listed consecutively on the Define Table screen. When the access sequence number is the same, one HP ALLBASE/SQL query is generated for both HP ALLBASE/SQL tables. Separate queries are passed if the tables do not belong to the same DBEnvironment.
2	PRODUCTS	HP ALLBASE/BRW retrieves the product information from the IMAGE data set PRODUCTS where the corresponding PRODUCT-NO matches the PARTNUMBER taken from the two HP ALLBASE/SQL tables. Note that a keyed access is made to the data set PRODUCTS on PRODUCT-NO. That is, for every PRODUCT-NO retrieved by HP ALLBASE/SQL, HP ALLBASE/BRW uses keyed access into PRODUCTS to find the PRODUCT information.



HP ALLBASE/SQL can be given access sequence numbers that are lower than IMAGE tables because HP ALLBASE/BRW supports keyed reads from HP ALLBASE/SQL into IMAGE, but not keyed reads from IMAGE into HP ALLBASE/SQL.

Items

In HP ALLBASE/BRW, the columns from the data source table are known as **items**. **Projecting an item** means that the values for an item are made available for the new table and can be used in later tables or reports.



Projecting Items

By default HP ALLBASE/BRW projects *all* items with unique names from their source tables to the result table. If items have identical names in a joined table, *only one* of them is projected. Therefore, if you want to project two items with identical names, you must provide alias names (see below) for all items with identical names.

When the report is compiled, HP ALLBASE/BRW de-selects all unnecessary items, so that only those items required in the report are projected. For instance, suppose you accept the default and project all items but only use certain ones in the report layout or for calculations. HP ALLBASE/BRW removes the unused items from the final table. This cuts the size of the tables and improves the report performance.

On the Project Items screen you can change the default projection to:

- Project additional items by giving them unique alias names and marking them for projection. For joined tables, you can give unique alias names to items with the same name so that they project as different items. In a merged table, you can give the same alias name to items with different names and project them as one item.

  Alias names are only in effect *outside* the source table; that is, they do not affect the source tables.

- Exclude items from projection. Excluding items is not necessary for performance but can help maintain security.
- Change numeric precision of an item when it is projected from a table.

To exclude items from projection

1. At the Define Table, Define Relation, or Relation Condition screen, choose `Project Items`.

   All items for all the sources you used for your table appear on the Project Items screen. By default, all the items contain an `X` in the `Proj` (project) field, meaning that all items will be projected unless you change the default.

2. Use the Tab or cursor control keys to move to the `Proj` field for any items you want to exclude from projection.
3. With the cursor under the `X`, press the space bar to exclude that item from projection.

To give a projected item a different name

1. At the Project Items screen, move to the `Alias Name` field for the item you want changed.
2. Type a new name for the item.

To change the numeric precision for a projected item

1. At the Project Items screen, move to the `NumPrec` field for the item.
2. Type a number and a method code for type `N, I,`, or `R` items if you want to change their numeric precision. (The `Type` field on the screen shows `S` for string and `N, I,` or `R` for number.)

Valid numbers are in ranges, depending on the number type for the field:

Item Type	Number
N (fixed-point numeric)	-6 to 12
I (integer)	0 to 8
R (floating-point real)	-90 to 76

The number is a power of 10 or 10th exponent. Precision applies only to the result of a calculation. The computation is made at full precision and any truncation or rounding necessary is performed only after the computation is complete.

Precision is explained in greater detail in Chapter 15.

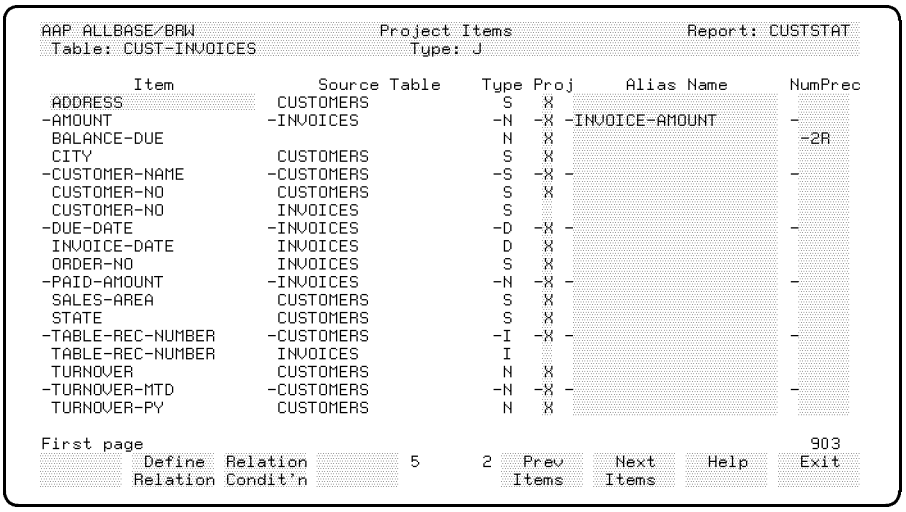
If you change the precision for a number field, you must also tell HP ALLBASE/BRW what to do with digits that are left over. The

letter you type is a method code indicating how any superfluous digits are treated:

Method Code	Digits are:	Explanation
T	truncated	All the digits outside the precision are dropped.
R	rounded	Rounded up if the first digit outside the precision is a 5 or greater and rounded down if the first digit outside the precision is less than 5.
C	ceiled	Always rounded up.

Examples

In this example of a joined table, the item AMOUNT has been given an alias; items without an `X` in the Project field have been excluded from projection, and changed numeric precision has been specified for the item BALANCE-DUE. Only items projected from a table can be used in later tables or in a report.



Definitions on the Project Items screen only affect how the item is used outside its source table; they have no effect inside the table.

Note

HP ALLBASE/BRW always projects a common item from a source table *not* subject to an open join, no matter what table you select for projecting the item. For example: suppose you selected the common item CUSTOMER-NO to be projected from the open-join table INVOICES. If there are no invoices for a customer, that customer's CUSTOMER-NO will contain NO_VALUE, as you projected it from INVOICES, the open-join source table. (All other information held in the CUSTOMERS table would be projected normally.) In this case HP ALLBASE/BRW automatically projects the item CUSTOMER-NO from the source table CUSTOMERS, so ensuring that all customer numbers are present in the report.

In the next example, the Project Items screen illustrates how items can be projected from a merged table:

```
HP ALLBASE/BRW                  Project Items                  Report: CUSTSTAT
Table: PROD-DETAILS               Type: M

        Item              Source Table    Type Proj     Alias Name       NumPrec
 ORDER-NO               ORDER-DETAILS        S   X
-PRICE                 -PRODUCTS            -N  -X -                     -
 PRODUCT-LINE           PRODUCTS             S   X
 PRODUCT-NAME           PRODUCTS             S   X  NAME-OF-TOY
-PRODUCT-NO            -ORDER-DETAILS       -S  -X -                     -
 PRODUCT-NO             PRODUCTS             S
 QUANTITY               ORDER-DETAILS        I   X
-QUANTITY              -PRODUCTS            -I  -  -                     -
 SHIPMENT-DATE          ORDER-DETAILS        D   X
 TABLE-REC-NUMBER       ORDER-DETAILS        I   X
-TABLE-REC-NUMBER       -PRODUCTS            -I  -  -                     -
```

All the items are projected and the item PRODUCT-NAME has been given the alias NAME-OF-TOY.

Precision of Numbers

You can define the precision for any type of number on the Project Items screen if you want the precision changed at this stage of the report. Changing the precision of an item on the Project Items screen only affects how the item is used *outside* the current table.

You can set the precision for these number types:

- Fixed-point Numeric (N)
- Integer (I)
- Floating-point Real (R)

You can change the numeric precision in a report at the following stages:

- with the result of a table calculated item, defined on the Table Calc Item screen
- when an item is projected from a table via the Project Items screen
- with the result of a layout calculated item on the Layout Calc Item screen
- when an item is used in a lineset or with the result of a column calculation on the Define Lines screen
- when an item is to be printed and is defined using scaling/decimals on the Item Edits screen

Note that all parts of the report performed later will use the item with the new precision.

The number indicates the *power of 10*, or *10th exponent* that indicates the precision.

Valid number ranges are as follows:

Value	NumPrec	Result
1024.3849	-2R	1024.38
864186.46	3C	865000.00

Data Access Methods

You define the data access when you define the table. In most cases, you use both the Define Relation screen, the Parameter screen, and the Tune Access screen to define the data access.

This section discusses how you can define:

- the sequence in which the source tables are to be accessed
- which keys (IMAGE search items or KSAM keys) are to be used for accessing which source tables
- which source tables are to be read serially, HP ALLBASE/BRW default method
- which non-SQL source tables are to be read directly via record number
- which HP ALLBASE/SQL source tables are to be combined to one query
- which source tables are to be locked during access
- the database open mode for IMAGE data sets

Note

For better performance:

Check the report's compile listing for the transformation of tables into access blocks and the optimization of conditions and calculated items.

Then, use the access block statistics printed with the report to determine which data access methods are most efficient for the report.

Finally, if necessary, change the data access sequence on the Tune Access screen. See Chapter 19 for details about data access tuning.

Tuning Data Access

You can tune the data access for better efficiency after you have selected the source tables for the report.

Choosing the right data access method can significantly improve the performance level for a report. HP ALLBASE/BRW lets you try different data access methods. Using different access methods does not change the definition of the data in the report.

There are several things you can do to tune your data access for better efficiency:

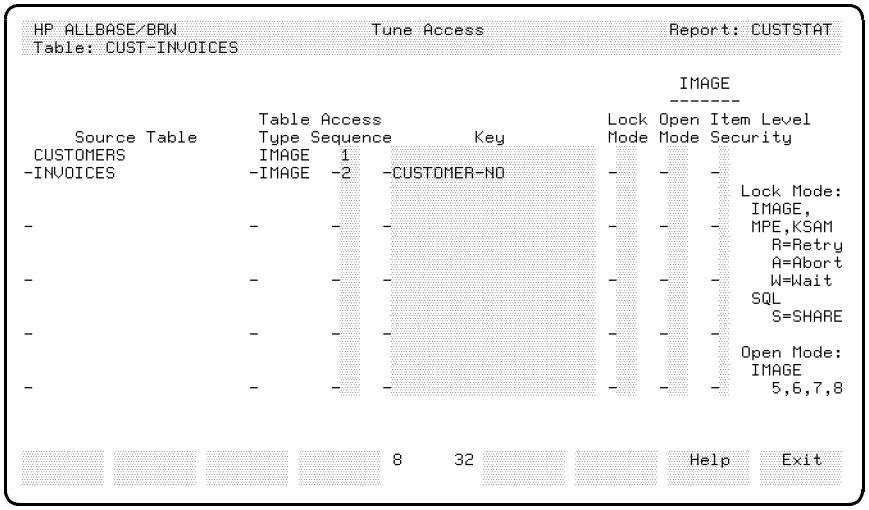
- Define the sequence in which the source tables are accessed.
- Specify keys (IMAGE search items or KSAM keys) to access certain source tables. (The default is a serial read.)
- Read non-SQL source tables by record number using TABLE-REC-NUMBER as a key.
- Combine SQL tables belonging to the same DBEnvironment as one query using the same sequence number.

- Secure locks on IMAGE datasets or files.
- Define Open Mode capabilities such as "Read, allow concurrent modify".

To tune data access

1. Define the source tables, relations, relation conditions, and parameters for your report, as described in previous chapters.
2. From the Define Table screen, choose `Tune Access`.

   The Tune Access screen lists the source tables you have specified and also shows the table type.



3. Type the access sequence for each source table. See the discussion below for more information about access sequence.
4. Type any keys you want to use for non-SQL tables.
5. Type a Lock Mode, Open Mode, or Item Security code. See the *IMAGE Data Base Management Reference* manual for a full description of the DBOPEN modes and capabilities. See also Chapter 17, "Configuration and Security."
6. Press Enter.

Defining Access Sequence

Defining access sequence affects the order in which the tables are accessed. If one of the data sources listed is an MPE file, it is usually most efficient to list it as the first data source. Other considerations when deciding on access sequence include:

- A source table on which an open join is made must be accessed last.
- HP ALLBASE/SQL tables belonging to the same DBEnvironment can be combined into one query by specifying the same access

sequence number for each table. (If the tables are listed consecutively and no open join is indicated, the same access sequence number is assigned by default.)

- If different access sequence numbers are entered for HP ALLBASE/SQL tables, separate queries are generated.
- If keyed access is used for the first source table in the access sequence, a relation must be defined and a single-value or list-of-values parameter must provide the value for that key. Using keyed access to the first source table can improve the performance of the report if you want to select only records for some distinct key values.

Example: Keyed access to first source table

The following is an example of keyed access to the ORDERS and ORDER-DETAILS data set, in which the parameter ?CUSTOMER-NO is used to provide the key values for accessing ORDERS.

The Parameter screen below shows the definition for the parameter ?CUSTOMER-NO:

```
HP ALLBASE/BRW                     Parameter                     Report: ORDSTAT

 Parameter    CUSTOMER-NO             Value Length  8  (if String)

 Result Type  S     S = String     N = Numeric fixed
                    D = Date       I = Integer
                    T = Time       R = Real

        Type  L     P = Comparison Predicate
                    S = Single Value
                    L = List of Values

 Value Required     X

 Upshift Value      X





 Add      Delete                      4     16  Prev     Next     Help     Exit
 Parm     Parm                                  Parm     Parm
```

The value for ?CUSTOMER-NO is provided when the report is executed.

Two relations are defined: CUSTOMER-NO to ?CUSTOMER-NO and ORDER_NO (from the ORDERS table) to ORDER-NO (from the ORDER_DETAILS table, as shown on these two Define Relation screens.

```
HP ALLBASE/BRW                  Define Relation                 Report: ORDSTAT
Table: ORDS                     Relation 1 of 2


Relate via            Common Item           in Source Tables
                   CUSTOMER-NO              ORDERS
                  -?CUSTOMER-NO             -
```

```
HP ALLBASE/BRW                  Define Relation                 Report: ORDSTAT
Table: ORDS                     Relation 2 of 2


Relate via            Common Item           in Source Tables
                   ORDER-NO                 ORDERS
                  -ORDER-NO                 -ORDER-DETAILS
```

The Tune Access screen shows the access sequence and keys used for these source tables:

```
HP ALLBASE/BRW                      Tune Access                 Report: ORDSTAT
Table: ORDS

                                                                  IMAGE
                                                                  -------
                     Table Access                              Lock Open Item Level
    Source Table     Type Sequence        Key                  Mode Mode Security
 ORDERS               IMAGE   1    CUSTOMER-NO                  -    -    -
-ORDER-DETAILS       -IMAGE  -2   -ORDER-NO                     -    -    -
```

Access by Record Number

You can access a record in an IMAGE data set, KSAM file or MPE file directly by using its logical record number (TABLE-REC-NUMBER) in a relation.

In the following example, the file PRODX (format PRODUCT-INDEX) contains an item (PRODUCT-REC#) that relates to the logical record numbers (TABLE-REC-NUMBER) for the records in the data set PRODUCTS.

The common items PRODUCT-REC# and TABLE-REC-NUMBER are specified on the Define Relation screen:

```
HP ALLBASE/BRW                  Define Relation                 Report: ORDSTAT
Table: ORDS                     Relation 1 of 2


Relate via            Common Item           in Source Tables
                   PRODUCT-REC#             PRODUCT-INDEX
                  -TABLE-REC-NUMBER         -PRODUCTS
```

The access sequence is defined on the Tune Access screen:

```
HP ALLBASE/BRW                    Tune Access                  Report: ORDSTAT
Table: ORDS

                                                                    IMAGE
                                                                   -------
                        Table Access                          Lock Open Item Level
    Source Table        Type Sequence        Key              Mode Mode Security
 ORDERS                 IMAGE   1
-ORDER-DETAILS         -IMAGE  -2   -TABLE-REC-NUMBER         -    -    -
```

Once the report is defined, you can experiment separately with ways of accessing the data. You can use serial read or chained/calculated (keyed) reads to see which takes the less time. The access method will not change the result of the report.

HP ALLBASE/BRW verifies that the access methods are compatible with the relational access definition.

Relation Conditions

Relation conditions restrict the records written to the result table to those that match a specific condition. Relation conditions are independent of relations. That is, you do not need to define a relation in order use a relation condition.

Relation Conditions have these characteristics

- They select a subset of records to be included in a table.
- They are calculated from items, calculated items, or functions defined in the same table, parameters or constants.
- They must be *logical-exprs*, according to Syntax Diagram 2, in Chapter 15.
- They are expressed with the calculation language as described in Chapter 15.

To define a relation condition

1. At the Define Report screen, Define Relation screen, or the Project Items screen, choose `Relation Condit'n`.
2. Type the condition formula, using the calculation language described in Chapter 15.

```
HP ALLBASE/BRW                Relation Condition              Report: PARTPROD
Table: PART-PRODUCT

PARTPRICE >= 1000.00 AND PRICE < 5000.00
```

3. Press Enter.

If you made a mistake, the part of the formula that is in error is highlighted. Correct the error and press Enter again. You can also press `Exit Save` to save the formula in spite of the error. This lets you return to some other part of HP ALLBASE/BRW to verify item names or other formula elements and then return to the Relation Condition screen to correct the formula later.

For more details on how to use the Relation Condition screen, see the Relation Condition section of Chapter 16.

Examples

The condition shown below selects only those orders with order numbers that begin with ZJ:

```
SUBSTRING (ORDER-NO,1 ,2) ="ZJ"
```

The next condition selects only those invoices that are overdue:

```
DUE-DATE < DATE
```

This condition selects shipment dates in the month equal to the parameter value provided at run time for `?month`:

```
MONTH_OF (shipment-date) = ?month
```

When you define a relation condition *and* a relation, the relation condition further refines the data selection. For example, suppose you defined a new table (CUST-INVOICES) from the CUSTOMER table and INVOICES table with a relation between the CUSTOMER-NOs for each table. If you also defined the relation condition `DUE-DATE < DATE`, only those records whose CUSTOMER-NOs matched records in each table *and* whose invoices were overdue would be written to table CUST-INVOICES.

What Next?

When you have defined the data access and have all the items you want for your report, you can begin defining the way your final report will look.

See Chapter 7 for information about report layouts. See Chapter 8 for information about sorting, breaking, and pagination.

If you want to perform calculations on the items that you selected, go to Chapter 15 for more information.

7

Defining Report Layout

Defining linesets, and placing and editing items

At a Glance

You define the report layout by filling in fields on the Define Lines screen. The specifications you enter are checked on-line for correctness. As you are defining the layout, you can review what you have already defined for the report by looking at the Review Layout screen. This makes it easy for you to create new reports and to maintain existing ones.

Features of Report Layout

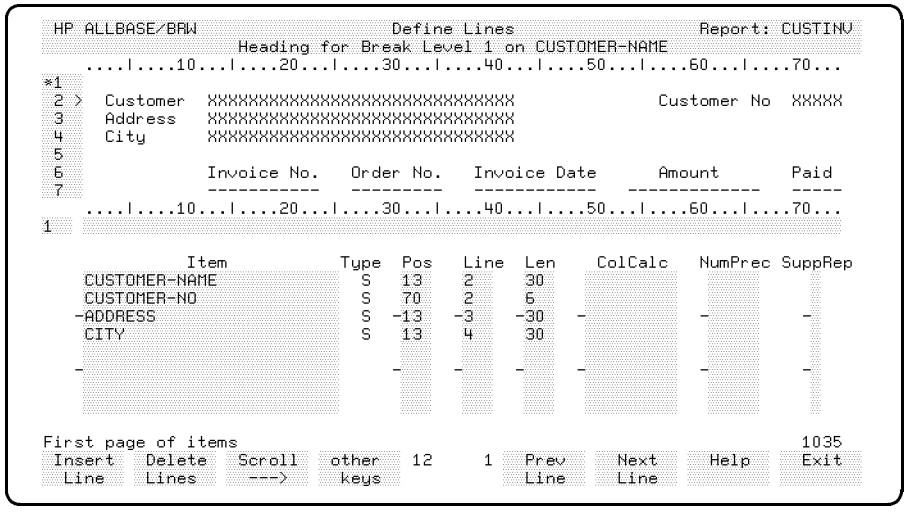
- Easy "fill in the box - see what you get" layout specification
- Break heading and footing linesets for report, page and up to 9 sort levels
- Up to 999 lines per lineset
- Item editing and numeric scaling
- Alternate item edits for zero values
- Row and column calculations
- Suppressed printing of repeating item values
- Control of calculation precision
- Conditional lineset formats
- Page size ranges from 1 - 999 lines long and 1 - 300 characters wide
- Support of horizontal pages
- Optional demand for special paper before printing the report
- New page before or after a lineset is printed
- Positioning a lineset in a fixed place on a page
- Resetting page numbering at specific sort level

This chapter covers

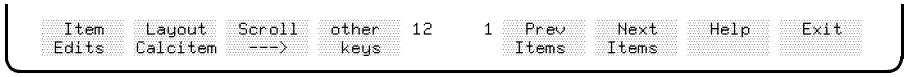
- report layout principles
- defining linesets
  - detail linesets
  - page and report linesets
  - break linesets
- positioning and editing items

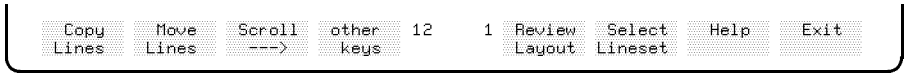
Using the Screen

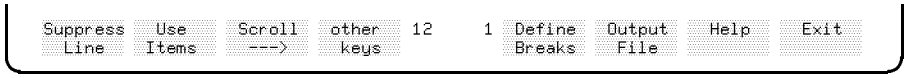
This is one example of the Define Lines screen where the report is laid out. This particular screen shows the lines for the heading (including the column headings) for a first-level break in the report. There is a similar screen for details, report headings, report footings, and break level footings.



In the top half of the Define Lines screen, XXX's and 999's are used to indicate where data from items defined in the bottom half of the screen will be printed. There are three other sets of softkeys available from the Define Lines screen.







These softkeys let you create and revise lines on the report:

`Insert Line`	inserts a line at any position in the lineset.
`Delete Lines`	deletes one line or a line range.
`Move Lines`	moves a line or a line range to an other position in the lineset. Items positioned on that lines are also moved to the new location.
`Copy Lines`	copies a line or a line range from any lineset including the items positioned on the lines, or copies a range of sample layout lines, making it easy to create column headings.
`Review Layout`	displays a sample report on screen.

Here is a sample report layout as seen on the Review Layout screen:

```
HP ALLBASE/BRW                    Review Layout                    Report: CUSTINV

     ....|....10...|....20...|....30...|....40...|....50...|....60...|....70.
  RH17               XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
  RH18               XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 1BH1
 1BH2>  Customer   XXXXXXXXXXXXXXXXXXXXXXXXXXXXX              Customer No  XXX
 1BH3   Address    XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 1BH4   City       XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 1BH5
 1BH6>              Invoice No.   Order No.   Invoice Date      Amount       Pai
 1BH7>              -----------   ---------   ------------   -------------   ---
   D1>                XXXXXX      XXXXXXXX       0m/0d/yy     ZZZZZZZZ9.99    ZZZ
 1BF1>                                                       ------------    ---
 1BF2>                            Customer Total Amounts     ZZZZZZZZ9.99    ZZZ
 1BF3
  RF1
  RF2   Total Invoice Amounts   ZZZZZZZZ9.99
  RF3   Total Paid Amounts      ZZZZZZZZ9.99
  RF4   Total Due Amounts       ZZZZZZZZ9.99


     ....|....10...|....20...|....30...|....40...|....50...|....60...|....70.
Last page                                                                  1222
 Define   Print    Scroll   Scroll  1      1   Prev     Next     Help      Exit
 Lines    Layout    <---     --->                Page     Page
```

Notice that in the example, the words `last page` appear above the `Define Lines` softkey. That means that the screen is showing only the last screenfull of the Report Layout. To see the first part of the layout, you would need to choose `Prev Page`. The numbers and initials in the shaded column on the left of the layout indicate the lineset type, break level, and line number for each line on the report layout.

The initials indicate:

RH	A report heading (at the beginning of the report)
PH	A page heading (at the top of each page)
BH	A break heading (repeated at the beginning of each break)
D	A detail line (repeated for each item in the report)
BF	A break footing (repeated at the end of each break)
PF	A page footing (at the bottom of each page)
RF	A report footing line (at the end of the report)

The number to the left of the break lineset initials is the break level number. In the example, all the break linesets are for a first level break. Second level breaks would show the number 2. The number to the right of all the initials is the line number for that lineset. For example, RH17 and RH18 on the screen indicate the 17th and 18th lines for the report heading.

These lineset identifiers are important because each lineset is defined on a different iteration of the Define Lines screen. To get to the Define Lines screen, you must specify which lineset you want to see.

Defining Linesets

A **lineset** is a unit printed when a certain event occurs in a report, for example, when a detail occurs. A lineset can consist of a single line of text and items, multiple lines, or no lines.

You define linesets at the Define Lines screen. There is a different Define Lines screen for each lineset type. For instance, if you want to define the lines for the page heading, you must go to the Define Lines screen for Page Heading.

To get to the Define Lines screen

1. From the Define Report screen, choose `Define Lines`.

   A prompt appears, asking which lineset you want to define.

   ```
   Specify lineset level (D,P,R,1..9):  and type (H,F):
   ```

2. In the first field of the prompt, type one of these options:

   a. `D` for detail

   b. `P` for page

   c. `R` for report

   d. `1` through `9` for a break level.

3. In the second field, type one of these options (if you are doing a heading or a footing):

   a. `F` for footing

   b. `H` for heading

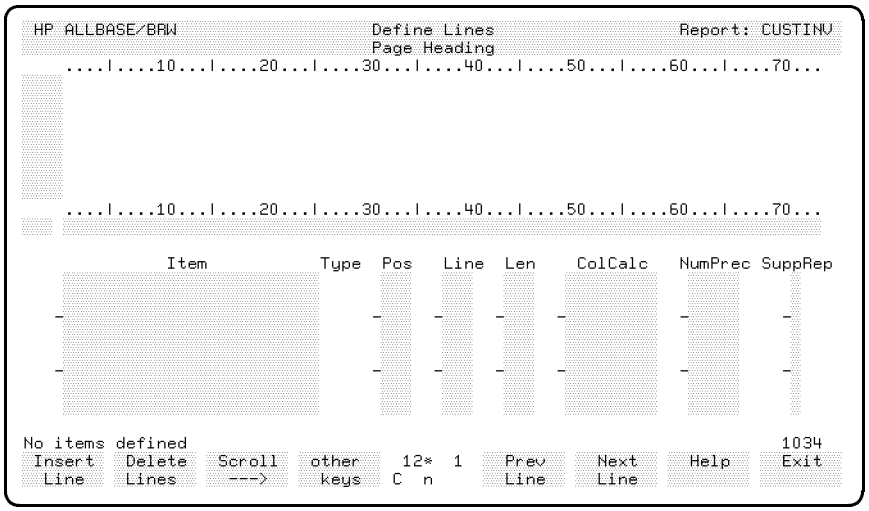
4. Press Enter.

   The Define Lines screen appears for the lineset you request.

To add a lineset

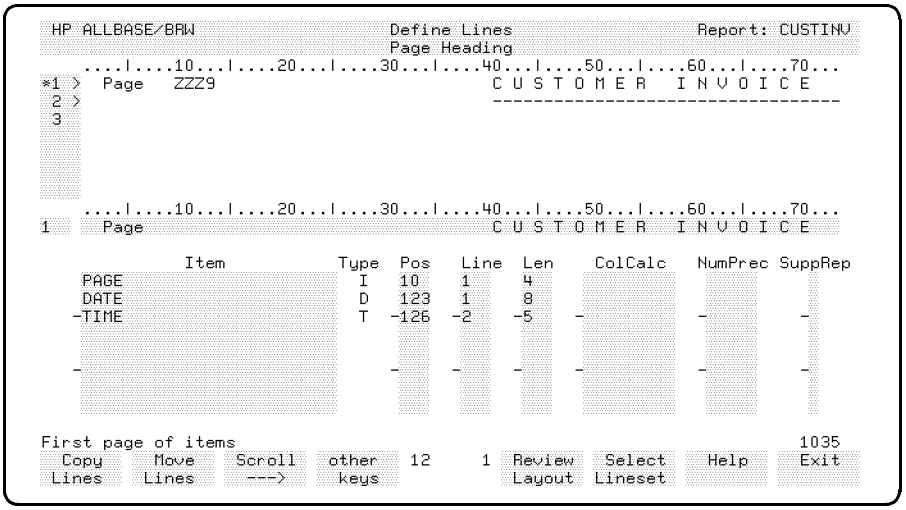
1. Go the the Define Lines screen (as described previously in the "Defining Linesets" section of this chapter).

   Your cursor will be in the lower part of the screen below the page ruler markings.



2. Type a line number for the first line of the page heading in the first field of the line.
3. Type any text you want to appear on the line in the second field of that line.
4. Press Enter.
5. Repeat the three previous steps for every line you want in the page heading.
6. Type the item name and information for any items you want to use in the page heading.

As you enter the information for each line, the top portion of the screen will show what the report layout looks like. An asterisk (*) appears in the line number field for the line number you are currently entering.



To position items on the lines of the lineset

1. Type the name of the item in the `Item` field.
2. In the `Pos` field, type the column number where you want the first character of the field to be placed.
3. In the `Line` field, type the number of the line where you want the item placed. This is optional. If you want to use the current lineset you can leave the number blank. The current lineset is indicated by an asterisk (*) in the left-hand (line number) field of the upper screen.
4. In the `Len` field, type a number of characters you want printed. This is optional. If you want to use the full length of the item, just leave this field blank and HP ALLBASE/BRW will fill it in for you after you press Enter.
5. In the `ColCalc` field, type a column calculation that you want performed, if any.
6. In the `NumPrec` field, type a number and the method code for the numeric precision you want (if the field is a number field).
7. Type a number for 1 to 9 in the `SupRep` field if you want to designate a particular break level (1 to 9) at which to suppress repeat printings of this particular item value.

Items placed within the lineset should not overlap with text or other items.

Note

You can press Enter at any time on this screen. When you do, the lines you have defined or modified will appear at the top of the screen. Use the `Scroll` key to move left and right if your layout exceeds the screen width.

For more details about the Define Lines screen and field descriptions, see the Define Lines section of Chapter 16.

Detail Linesets

The detail lineset is printed for each record in the final access table, that is, for each detail. You use it to print the detail data in your report. For example, the specification of a simple listing report only consists of a detail lineset, one for each record in the report, as shown in this example:

100114	ACME INDUSTRIES	TUSCON	ARIZONA
100213	BLACK & WHITE	HARRISBURG	PENNSYLVANIA
100311	TOOLS EQUIPMENT CORP.	SEATTLE	WASHINGTON
100321	MILLER'S CARS	SANTA BARBARA	CALIFORNIA
100327	DAILY POST	DENVER	COLORADO
100365	LEMWAY INC.	EL PASO	TEXAS
100432	SUR	PARAMUS	NEW JERSEY
100498	C. WINSTON F & S	RICHMOND	VIRGINIA
100537	MACHINERY CORP.	SAN FRANCISCO	CALIFORNIA
100568	HIGGIN'S COMPANY	DENVER	COLORADO
100827	TU CONSULIDATED	LOS ANGELES	CALIFORNIA
100888	REGULAR OIL INDUSTRIES	HOUSTON	TEXAS
100946	BROWN BROTHERS	MIAMI	FLORIDA
100953	BEVERLEY'S SHOP	PRESCOTT	ARIZONA

Data that is not to be repeated for each detail can be printed on other linesets, such as break headings and break footings, or page headings and page footings, described later in this chapter. The layout for each lineset is defined on the Define Lines screen.

Reports Without Detail Lines

You can define reports without detail lines if you want to print only summarized data. See the sections on break footing linesets and column calculations in this chapter.

Page and Report Linesets

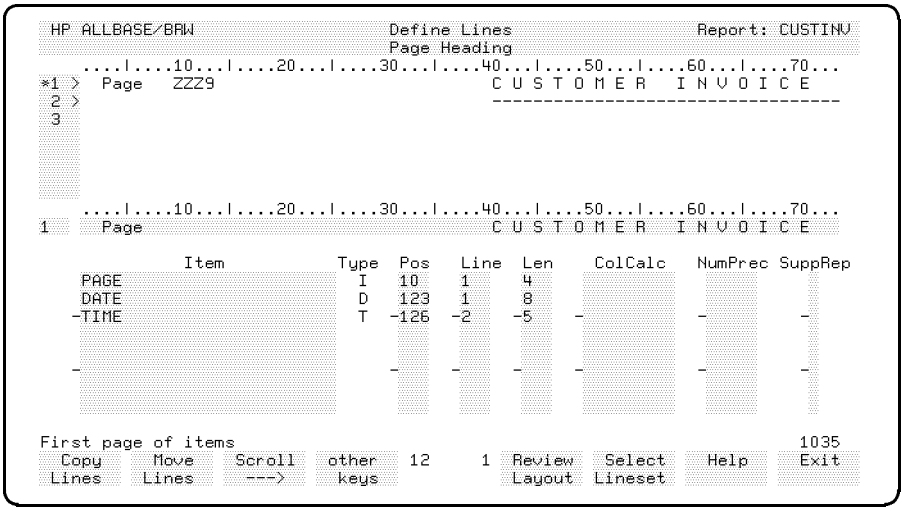
In addition to specifying break heading and footing linesets for a report, you can define page linesets and report linesets.

Page Heading Lineset

The page heading lineset specifies that certain data will be printed on the top of each page.

Example: Printing the page number, date, and a title

This example shows the page number, date, and title being inserted on the page heading.

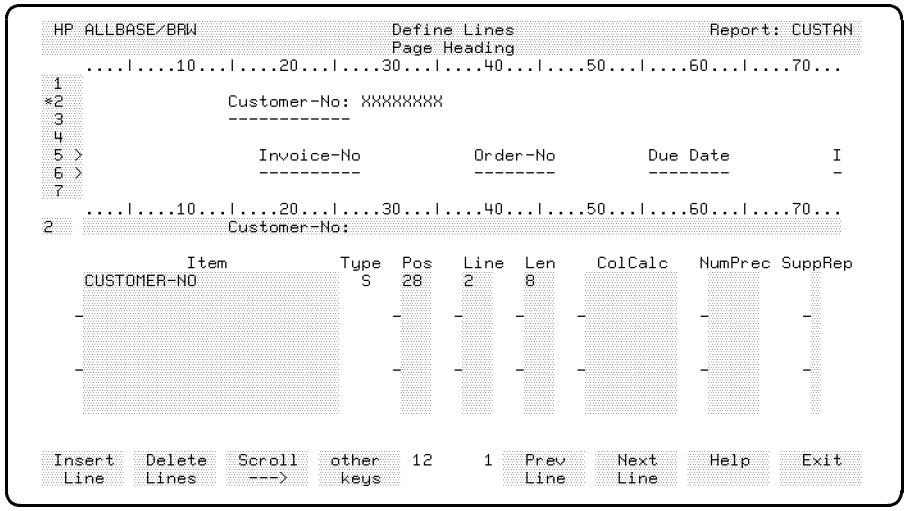


The screen shows the items for PAGE, DATE, and TIME in the `Item` field. The upper part of the screen does not show the DATE and TIME items because they are at positions 123 and 126 (off the screen).

Example: Repeating the break heading on each page

Suppose you want to specify a new page before printing the break heading line set, but you do not want to define lines for that break heading lineset. Instead, use the page heading lineset to print the break heading data.

Look at the following example:



The page heading lineset prints data from the detail that is used to print the next lineset. Column calculation values are printed in the state of the previous page footing lineset, which allows column calculations to be carried forward.

Page Footing Lineset

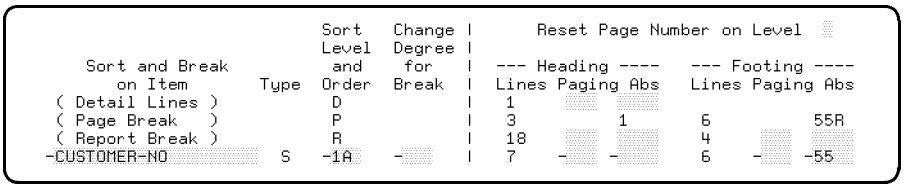
The page footing lineset is printed at the bottom of each page and can contain the page number and summarised data for the page.

The page footing can be replaced by another lineset. You can specify that a break footing lineset is printed instead of the page footing lineset on each page where the break occurs.

Example: Printing a break footing instead of the page footing

At the end of a break, you may want to print only the break footing and not repeat the page footing. You can position the break footing lineset so that it always appears at the bottom of the page. by typing the line number of the first line for the break footing in the `Abs` field.

In this example, the break footing for the break defined when CUSTOMER-NO changes is given an absolute position at line 55 of the page.



Sort and Break on Item	Type	Sort Level and Order	Change Degree for Break	Reset Page Number on Level					
				--- Heading ----			--- Footing ----		
				Lines	Paging	Abs	Lines	Paging	Abs
(Detail Lines)		D		1					
(Page Break)		P		3		1	6		55R
(Report Break)		R		18			4		
-CUSTOMER-NO	S	-1A	-	7	-	-	6	-	-55

The page footing lineset prints data and column calculations from the detail that was used to print the preceding lineset. That is, since no new item values or column calculations are generated just because of a page break, any item or calculation reference in a page footing lineset has the value that was last generated.

Report Heading Lineset

The report heading lineset is printed once at the beginning of the report. You can use it to print global data for the report (see next sample screen). Normally, a report heading lineset is preceded by a page heading lineset.

Example: Suppressing a page heading

You have defined a page heading and a report heading. The report heading contains some of the same information as the page heading (such as the name and date of the report). It would be redundant to have both headings print on the first page of the report.

Here is the Define Lines screen for the Report Heading:

```
HP ALLBASE/BRW                    Define Lines                    Report: CUSTINV
                                 Report Heading
    ....|....10...|....20...|....30...|....40...|....50...|....60...|....70...
 7
 8 >                                       C U S T O M E R   I N V O I C E
 9
*10>              -------------------------------------------------------------
 11>              Description:   XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 12               Date       :   0m/0d/yy
 13               Time       :   HH:0M
    ....|....10...|....20...|....30...|....40...|....50...|....60...|....70...
10                -------------------------------------------------------------

                Item          Type  Pos   Line  Len    ColCalc   NumPrec SuppRep
    REPORT-DESCRIPT            S    30    11    50
    DATE                       D    30    12    8
   -TIME                       T   -30   -13   -5    -            -        -
    REQUEST-DATA1              S    15    15    80
    REQUEST-DATA2              S    15    16    80
   -REQUEST-DATA3              S   -15   -17   -80    -            -        -
    REQUEST-DATA4              S    15    18    80


 Insert   Delete   Scroll   other   12    1   Prev    Next    Help    Exit
  Line    Lines     --->     keys             Line    Line
```

To suppress the page heading lineset on the first page (where the report heading would print), go to the Define Breaks screen and position the report heading lineset at an absolute of line 1 of the page.

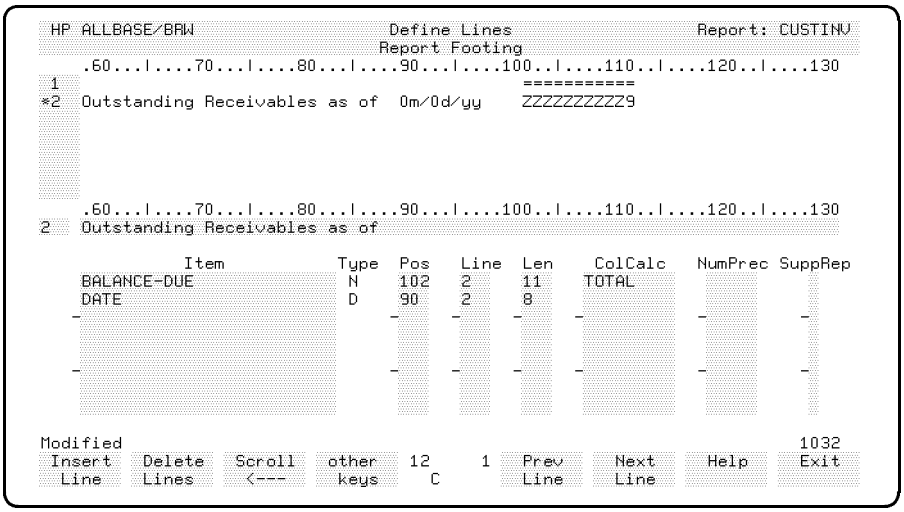
The report heading lineset prints data from the first detail in the report. All column calculations are zero. That is, at the start of the report, nothing has yet been calculated, so if the report heading lineset contains fields for column calculations, the values are zero. Non-calculated items have a value taken from the first detail line. Standard items, such as date and time, are always available.

Report Footing Lineset

The report footing lineset is printed at the end of the report. You use it to print summarized data for the whole report.

Example: Defining a report footing

This screen sample shows the Define Lines screen for a two-line report footing showing the outstanding receivables as of a particular day:



The report footing lineset prints data and column calculations from the last detail in the report. The DATE item appears in the upper portion of the screen as `0m/0d/yy` and the BALANCE-DUE appears as `ZZZZZZZZZ9`. The BALANCE-DUE will be a total of all the BALANCE-DUE values for the report.

Break Linesets

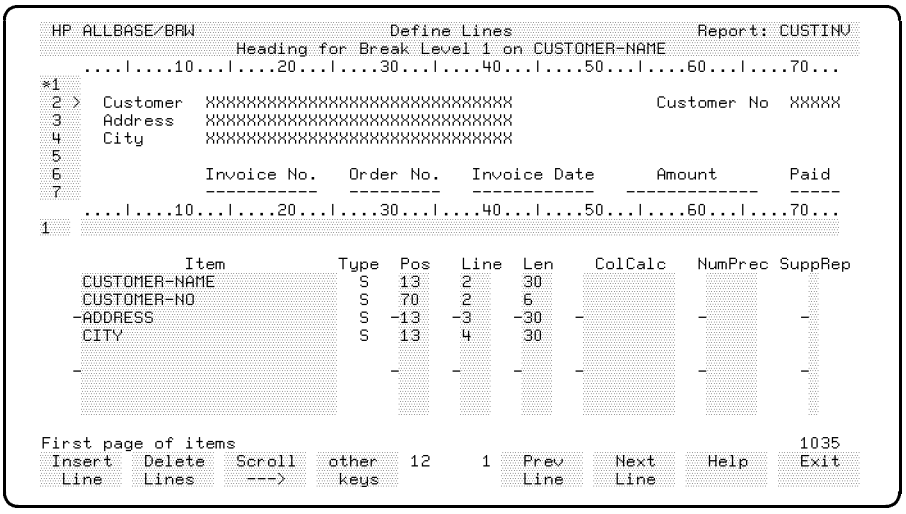
Text, items, or summarized data can be printed each time the value of a sort item changes (breaks). You can create lines that print before the beak (a break heading) or lines that print after the break (a break footing).

Break Heading Linesets

Break heading linesets can be defined for each sort item, so that heading information is printed for the new sort item value.

Example: Defining a break heading

Customer name, address, and column headings for the items are printed for the details following a break, as illustrated in this example of the Define Lines screen:



The break heading lineset prints data from the first detail in the break. However, column calculation values retain the values from previous detail record. This allows column calculations to be carried forward. See Column Calculations, later in this chapter, for details.

Break Footing Linesets

You can define break footing linesets for each sort item, so that summary information of the previous details is printed.

Example: Printing summary information

You can print totals or average values as break footings, as shown here:

```
 HP ALLBASE/BRW                    Define Lines                    Report: CUSTINV
                  Footing for Break Level 1 on CUSTOMER-NAME
    ....|....10...|....20...|....30...|....40...|....50...|....60...|....70...
*1 >                                                 -----------    -----
 2 >                          Customer Total Amounts     ZZZZZZZZ9.99    ZZZZZ
 3
 1
    ....|....10...|....20...|....30...|....40...|....50...|....60...|....70...
1                                                    -----------    -----

              Item            Type  Pos   Line  Len    ColCalc   NumPrec SuppRep
    INVIOCE-AMOUNT              N   54    2     13    TOTAL
    PAID-AMOUNT                 N   70    2     13    TOTAL
   -DUE-AMOUNT                  N  -86   -2    -13   -TOTAL      -         -
   -                                 -     -     -     -          -         -

First page of items                                                      1035
 Insert   Delete   Scroll   other   12     1   Prev     Next     Help     Exit
  Line     Lines    --->     keys                Line     Line
```

In this example, totals for the items invoice-amount, paid-amount and due-amount will be printed as the break footing for each customer.

Instead of (or in addition to) printing the heading information for that break in the break heading lineset, you can print it in the break footing.

The break footing lineset prints data from the last detail in the break and column calculation values in the state of this detail.

Positioning and Editing Items

Item values can be printed anywhere in the lineset, as long as there is no overlap with text or other items. HP ALLBASE/BRW provides a variety of editing capabilities for item values.

You can position and edit items, array items, and parameters.

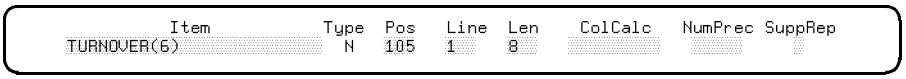
As indicated in the task description earlier in this chapter, HP ALLBASE/BRW provides defaults if you omit line and length in an item specification.

Printing Array Items

You can print array items (also called array components) by adding the array index to the array name when you list the item on the Define Lines screen.

Example: Printing array items

In this example, the array item is TURNOVER and the index is (6).



Item	Type	Pos	Line	Len	ColCalc	NumPrec	SuppRep
TURNOVER(6)	N	105	1	8			

If you omit the array index on the Define Lines screen, the first array component is printed, with the following exceptions:

- If you specify a column calculation for the array name without an index, the aggregate of all the array components is used in the calculation.
- If an array name without an index is included in an output file, the entire array (that is, all the array components) is output.

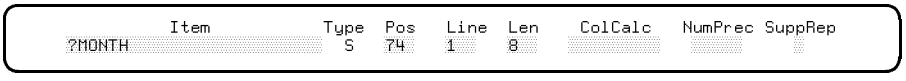
For more information, see "Arrays and Array Items" in Chapter 15, the "Layout Calculations" section in this chapter, and the "About Output Files" section in Chapter 12.

Printing Parameters

You can print single value parameters just as you can print any other item. Specify the parameter name preceded by a question mark.

Example: Printing single value parameters

In this example, the parameter ?MONTH is positioned as shown below:

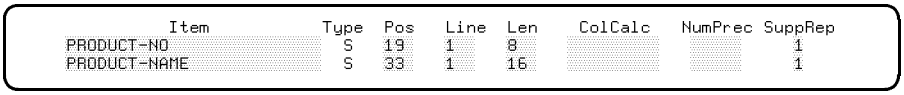


Item	Type	Pos	Line	Len	ColCalc	NumPrec	SuppRep
?MONTH	S	74	1	8			

Suppressing Repeating Item Values

You can make a tabular report more readable by omitting its redundant elements.

For an item in a detail lineset, you can define that it is only to be printed for the first detail within a certain break by typing the break level in the SuppRep field of the Detail Lines screen:



Item	Type	Pos	Line	Len	ColCalc	NumPrec	SuppRep
PRODUCT-NO	S	19	1	8			1
PRODUCT-NAME	S	33	1	16			1

Example: Suppressing lines

Suppose your report, without suppressing lines, looks like this:

```
Sales Area    Customer
----------    ------------------
North         Hillside Market
North         Kingsford Shirt Shop
North         Mineshaft Supplies
South         Better Bedrooms
South         Jack Frost Company
South         Nice 'N Light Creamery
```

You can suppress the SALES-AREA item on the break level that changes when the Sales Area changes. The resulting report would look like this:

```
Sales Area    Customer
----------    ------------------
North         Hillside Market
              Kingsford Shirt Shop
              Mineshaft Supplies
South         Better Bedrooms
              Jack Frost Company
              Nice 'N Light Creamery
```

Editing Items

You can specify rules for editing item values before they are printed. Rules are defined on the Item Edits screen.

To edit an item

1. At the Define Lines screen, choose `other keys` and then choose `Item Edits`.

2. A prompt asks you to type the name of the item to be edited:

   `Type the item edit name`

3. Press (Enter).

If the item is a string, the String Item Edit screen appears.

```
 HP BRW/V                        String Item Edits                     Report: TEST2S
                                       Detail

Item CUSTOMER-NAME      Line 1    Position 5     Length 30


Shift String        U = ALL UPPERCASE
                    L = all lowercase
                    I = Initial Capitals

Edit Mask


           X = character

            any other character (including x) is taken as it is



Justify (Left/Center/Right)              "No Value" Character       -
Prefill with Character                   "Error Value" Character    ?



                  Select   Select    4* 57   Prev     Next     Help     Exit
                   Item    Lineset     n     Item     Item
```

If the item is a numeric, the Numeric Item Edit screen appears.

```
 HP BRW/V                        Numeric Item Edits                    Report: TEST2S
                                       Detail

Item TURNOVER-PY        Line      Position      Length 15

Scaling  0    Decimals  2         Rounding      (Truncate/Round/Ceiling)

                 positive                negative      Adjust
Prefix     *                       *-                     X
Suffix     *                       *

Edit Mask *ZZZZZZZZZZZ9                                  integer part
          *.99                                           fraction part
           9 = digit   Z = digit without leading/trailing zeros

When zero print text
          *                                              integer part
          *                                              fraction part

Justify (Left/Center/Right)                "No Value" Character       -
Prefill with Character                     "Error Value" Character    ?
                                           "Field Overflow" Character <
          *use ^ characters for trailing blanks

                  Select   Select    4* 56   Prev     Next     Help     Exit
                   Item    Lineset     n     Item     Item
```

String item edits and numeric item edits are described in the next sections.

Note

Upper-and lower-case characters are treated literally in edit masks.

Editing items is optional. You need not explicitly supply item edits for all items. HP ALLBASE/BRW copies the default item edits from the configuration file, once you have specified an item on the Define Lines screen. This default item edit is displayed automatically on the Item Edits screen for an item. See Chapter 17 if you want to alter the default edit mask.

Item edits are performed in the sequence in which the edits are displayed on the screen. For instance, in the example of a string item edit (following), a string value is shifted to all upper-case characters, then edited with the edit mask and, finally, right justified.

String Item Edits

To edit string items

1. From the Define Lines screen, choose `Item Edits`.
   - For unedited printing, leave the Edit Mask field blank.
   - For edited printing, you can:
2. specify the letters in the string as all uppercase, all lower-case or initial capitals;
3. use an edit mask to include additional characters in the string.

For example, you can use the edit mask XX-XXX-X (shown on the next screen example) to print the product number 61288f as 61-288-f.

```
 HP ALLBASE/BRW                String Item Edits                 Report: CUSTINV
                                     Detail

Item PRODUCT-NO             Line 2    Position 1     Length 8

Shift String  U   U = ALL UPPERCASE
                  L = all lowercase
                  I = Initial Capitals

Edit Mask
XX-XXX-XX

           X = character

            any other character (including x) is taken as it is


Justify (Left/Center/Right)  R              "No Value" Character      -
Prefill with Character                      "Error Value" Character   ?


                 Select   Select  4     61  Prev     Next     Help     Exit
                  Item    Lineset           Item     Item
```

Note that string values are truncated on the right without warning if the print field is not large enough or if the edit mask does not contain enough X characters.

Numeric and Integer Item Edits

Items of types N and I need an edit mask for printing. However, the default edit mask is often all that is needed. To change the default edit masks, see Chapter 17, "Configuration and Security."

Numeric item edits are defined at the Numeric Item Edits screen.

```
 HP BRW/V                        Numeric Item Edits                  Report: TEST2S
                                       Detail

Item TURNOVER-PY        Line      Position      Length 15

Scaling  0    Decimals  2        Rounding     (Truncate/Round/Ceiling)

                 positive              negative       Adjust
Prefix     *                     *-                     X
Suffix     *                     *

Edit Mask *ZZZZZZZZZZZ9                                        integer part
          *.99                                                 fraction part
           9 = digit   Z = digit without leading/trailing zeros

When zero print text
          *                                                    integer part
          *                                                    fraction part

Justify (Left/Center/Right)                  "No Value" Character        -
Prefill with Character                       "Error Value" Character     ?
                                             "Field Overflow" Character  <
          *use ^ characters for trailing blanks

          Select  Select    4* 56   Prev    Next    Help    Exit
           Item  Lineset     n      Item    Item
```

If you do not want the default item edits, you can explicitly do one of these types of edits:

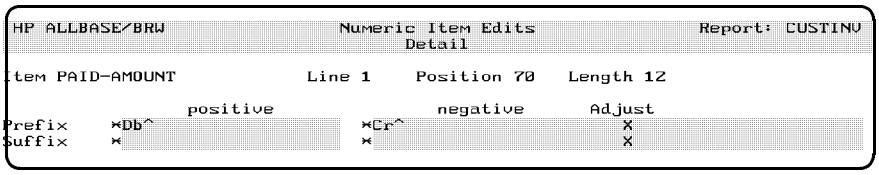
- **Scale the number**. This means moving the decimal point to the left, or to the right (if the scaling is negative). You can also define how many decimal places are to be appended to the scaled number and whether the scaling is to round (R), truncate (T) or ceil (C) the original number.

  This section of the Numeric Item Edits screen illustrates scaling:

```
 HP ALLBASE/BRW                  Numeric Item Edits                  Report: CUSTINV
                                       Detail

Item AMOUNT                Line 1    Position 54   Length 10

Scaling  6    Decimals  1        Rounding  R   (Truncate/Round/Ceiling)
```

Original Values	Edited Values
5893100	5.9
56999067.89	57.0
9989871.21444	10.0
57248971.89	57.2

- **Define a prefix and a suffix** for both positive and negative numbers. For example, this Numeric Item Edits screen shows a prefix assigned to the item for the positive as well as the negative instance of the value.

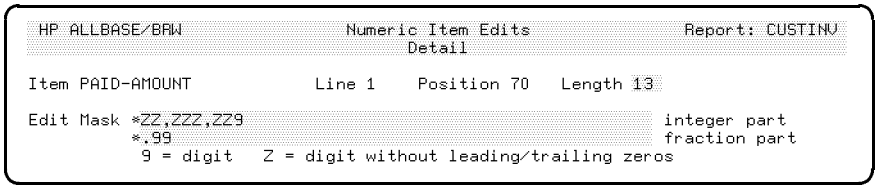


This table shows the result of a suffix edit on a positive and a negative value.

Original Values	Edited Values
-12.37	Cr 12.37
+50.92	Dr 50.92

- **Define an edit mask** to print the number with or without leading and trailing zeros (in the fraction part). You can define symbols for decimal and thousands characters, or you can have the words "millions" and "thousands" written out instead of the single character.

  The following example shows how the edit mask is specified:



Original Values	Edited Values
720.50	720.50
132720.00	132,720.00

- **Specify a value to print a zero number** as dashes or blanks instead of zeroes, as shown below:

```
HP ALLBASE/BAW                Numeric Item Edits                Report: CUSTINV
                                    Detail

Item AMOUNT                 Line 1    Position 54   Length 8

When zero print text
          *-----                                              integer part
          *.--                                               fraction part
```

Original Values	Edited Values
0	———-.—

Real Item Edits

See the screen Real Item Edits in Chapter 16 for a description of how you can print Real Items.

Date Item Edits

Date items always need an edit mask for printing. If you do not want the default item edits, you can specify another edit mask to print the date item, as shown here:

```
HP ALLBASE/BAW                Date Item Edits                   Report: CUSTINV
                                Page Heading

Item DATE                   Line 1    Position 123  Length 8


Edit Mask
dd-mm-yy
```

Time Item Edits

Time items always need an edit mask for printing. If you do not want the default item edits, you can specify another edit mask to print the time item, as shown here:

```
HP ALLBASE/BAW                Time Item Edits                   Report: CUSTINV
                                Page Heading

Item TIME                   Line 2    Position 126  Length 5


Edit Mask
HH:MM
```

Justify And Prefill

You also can specify justification and prefill item edits for all types of items.

Edits performed after the above string, numeric, date or time item edits are:

- The edited item value is justified left, right, or it is centered in the print field.
- The edited item value is prefilled from the left with a character you supply, as shown here:

```
HP ALLBASE/BRW              Numeric Item Edits              Report: CUSTINV
                                  Detail

Item PAID-AMOUNT          Line 1    Position 70   Length 8

Justify (Left/Center/Right)  R              "No Value" Character        -
Prefill with Character       *              "Error Value" Character     ?
                                            "Field Overflow" Character  <
          *use ^ characters for trailing blanks
```

Value	Results In:
50	***50.00

Item Edits For Exception Cases

NO_VALUE and error values (see Chapter 15, "Calculations") do not permit item edits to be performed and require default edits. System defaults for these edits are described in Chapter 17, "Configuration".

These and other exception cases are described below:

- **NO_VALUE** is edited by filling the print field with the default no-value character.
- **Error values** are edited by filling the print field with the default error-value character.
- **Numeric field** overflow occurs if the number is too large to be expressed with the desired edit mask. In that case, the print field is filled with the default field-overflow character.

Below is a sample report in which exception cases are illustrated:

```
      Customer No.    Inv. No.     Amount      AMT-Due   Paid-AMT

      ------------    --------     ------      -------   --------

      10000           IN5378        200.00     ???????   <<<<<<<

      10023           IN5566      1500.50       500.50   1000.00

      10034           ------      -------       -------  -------

      11255           IN2370      1009.00         9.00   1000.00
```

In addition, a warning message, such as those listed below, is printed at the end of the report if one of these exception cases occurs.

```
The following exceptions occurred in this report:

Field overflow occurred with item PAID-AMOUNT          (54*)
Error values occurred with itme AMT-DUE                (24*)
```

The number to the right of the message indicate the number of times the exception occurred in the report.

Note

HP ALLBASE/BRW will print a warning message each time a 'non-printable character' is found. However, you cannot specify a character to be printed as a substitute character for non-printable characters.

Layout Calculations

Layout calculations are those which are performed when the report is formatted. You can define these kinds of layout calculations:

Type of Calculation	Orientation
column calculations	vertical
row calculations (layout calculated items)	horizontal
combination of both—Either:	
column calculations on layout calculated items	vertical
or **layout calculated items that use column calculation results as arguments**	horizontal

Column Calculations

Column calculations summarize data for a break. For example, you can specify a break on the total sales values for a sales area.

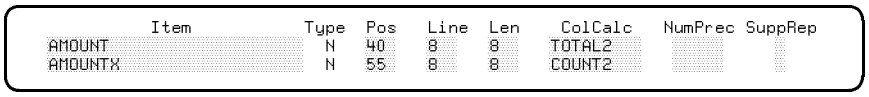
Column calculations can be defined for any numeric item (type N, R, and I) and can be one of these types of calculations:

Calculation	Result
TOTAL	total value
AVG	average value
MIN	least value
MAX	largest value
COUNT	count of the values (NO_VALUEs are not counted)
CALC	row calculation results

To perform a column calculation on an item

1. At the Define Lines screen, type the calculation type in the `ColCalc` field for the item.
2. Type an optional suffix after the calculation type if you want to reset the break level. Use one of these suffixes:

R	To reset on the report break
P	To reset on the page break
1-9	To reset on one of the break levels (1 through 9).
no suffix	The default. The column calculation value is reset on the current break.



Item	Type	Pos	Line	Len	ColCalc	NumPrec	SuppRep
AMOUNT	N	40	8	8	TOTAL2		
AMOUNTX	N	55	8	8	COUNT2		

Running column calculations begin again after being reset.

3. If you want to specify the source level for the calculation, type an optional suffix preceded by a comma after the calculation type. Use one of these suffixes:

,R	To use the report break as the source
,P	To use the page break as the source
,1-,9	To use one of the break levels (1 through 9) as the source
no suffix	The default. The value is taken from the detail line.

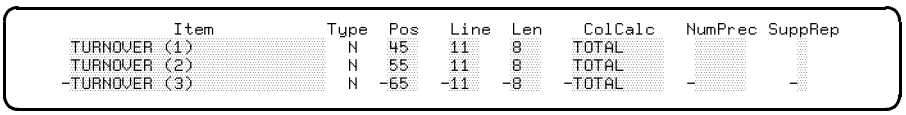
Column calculations with a particular source level are accumulated when the break footing (not heading) of the source level occurs.

4. If you are finished defining the line for that item, press Enter.

If you specify a calculation for an array item that does not have an index number, the column calculation is performed for all array components. If an index is specified, only that array component is used for the column calculation.

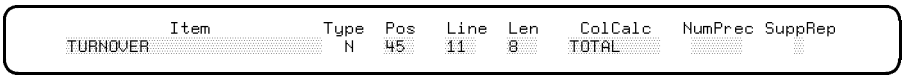
Example: Array items

In the following example of a Define Lines screen, for each array item (TURNOVER (i)), a TOTAL calculation is specified. This prints total values for each specific array item.



Item	Type	Pos	Line	Len	ColCalc	NumPrec	SuppRep
TURNOVER (1)	N	45	11	8	TOTAL		
TURNOVER (2)	N	55	11	8	TOTAL		
-TURNOVER (3)	N	-65	-11	-8	-TOTAL	-	-

By comparison, in the next example the item TURNOVER is specified without an index, and TOTAL is entered for the column calculation. Here, the whole array will be totalled.



Item	Type	Pos	Line	Len	ColCalc	NumPrec	SuppRep
TURNOVER	N	45	11	8	TOTAL		

The item name for column calculations on array items is specified in one of two ways:

With an index	If you want to print only the item.
Without an index	If you use the column calculation result of the array item in a calculation formula, or if you want to write it to an output file. See Chapter 9, Multi-pass Reporting.

Example: Resetting the calculation at break levels

If you specified a column calculation of TOTAL for item TURNOVER-MTD in the level 1 Break Footing Lineset, the value would be reset to zero at each level 1 break. You specify the default by stating the column calculation without a suffix.

You can specify a different reset levels for a column calculation by appending the break level at which you want the value reset to the name of the column calculation. For example, to have a column calculation run through the whole report, you add an R after the column calculation name: TOTAL R.

Example: Using a source level for the calculation

In this example, the Sales Area Report is produced from the following final access table:

Sales-Area	Cust-#	PY-TO	Inv-#	Amt
EAST	100888	2439.75	IN5325	3029.85
EAST	100888	2439.75	IN5331	850.00
EAST	100946	19765.50	IN5335	1775.00
EAST	100946	19765.50	IN5340	2125.00
EAST	100946	19765.50	IN5341	13400.00
EAST	100946	19765.50	IN5342	4137.50
NORTH	100114	85113.85	IN5346	700.00
NORTH	100114	85113.85	IN5326	7745.20
NORTH	100537	800.00	IN3338	600.00

Suppose the column calculation TOTAL is used on the item TURNOVER-PY (turnover for the previous year) without a source level.

The value for TURNOVER-PY will be added for every detail. However, the report requires that the turnover be added only at the break on CUSTOMER-NO, since the turnover has only one value per customer. If it is added at every detail, the turnover would have a value equal to the actual turnover times the number of records for the same customer.

For this report a source level is specified for TURNOVER-PY on the Define Lines screen:

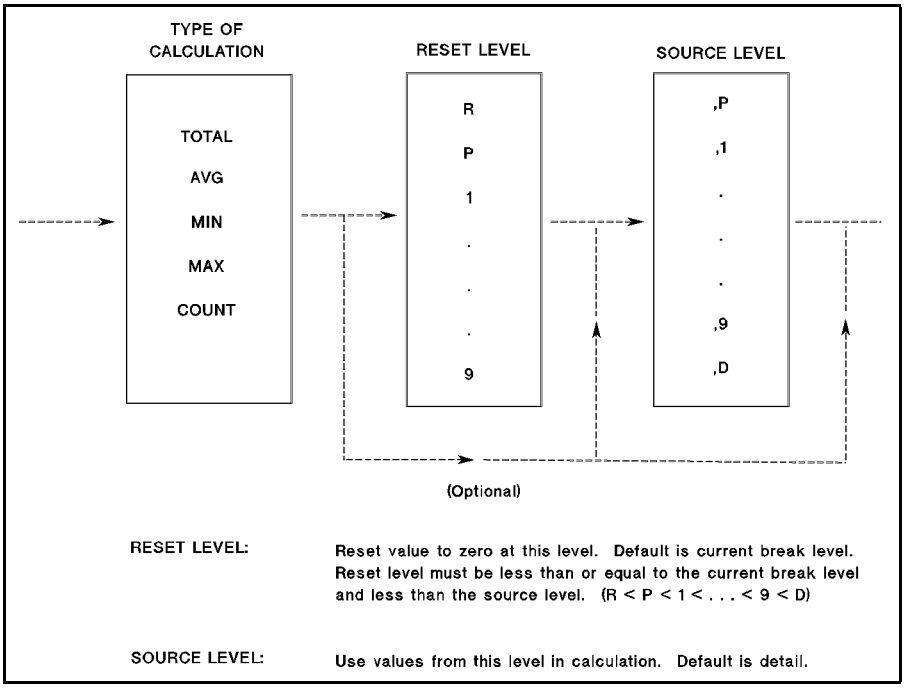
```
 HP ALLBASE/BAW                    Define Lines                    Report: AREAINV
                    Footing for Break Level 1 on SALES-AREA
     ....|....10...|....20...|....30...|....40...|....50...|....60...|....70...
 1                                      --------
*2 >Total Sales Area Invoice Amount: ZZZZZZZ9      Previous-year Turnover: ZZZ
 3
     ....|....10...|....20...|....30...|....40...|....50...|....60...|....70...
2   Total Sales Area Invoice Amount:                Previous-year Turnover:

            Item              Type  Pos   Line  Len    ColCalc   NumPrec SuppRep
    AMOUNT                     N    34    2     8     TOTAL
    TURNOVER-PY                N    72    2     8     TOTAL,2
   _                          _     _     _     _     _         _       _
   _                          _     _     _     _     _         _       _

 Insert   Delete   Scroll   other   12     1   Prev     Next     Help     Exit
  Line     Lines    --->     keys              Line     Line
```

The resulting report looks like this:

```
SALES AREA: EAST
  Customer:  100888

              Invoice-No          Amount
              ----------         -------
               IN5325             3029.85
               IN5331              850.00
                                 --------
        Total Invoice Amount:     3879.85       Previous-year Turnover: 2439.75


  Customer:  100946

              Invoice-No          Amount
              ----------         -------
               IN5335             1775.00
               IN5340             2125.00
               IN5341            13400.00
               IN5342             4137.50
                                 --------
        Total Invoice Amount:    21437.50       Previous-year Turnover: 19765.50

Total Sales Area Invoice Amount: 25317.35       Previous-year Turnover: 22105.25

SALES AREA: NORTH

  Customer:  ...............

                ...........
                ...........
```

Example: Using both reset and source levels

You can specify both the source and reset level for column calculations. The reset level is the first character after the column calculation, followed by a comma, followed by a character for the source level. See the following table:



Restrictions on Column Calculations

Restrictions on column calculations are as follows:

- For break heading/footing and detail linesets, the reset level must be less than the source level.
- Report heading and footing linesets cannot have a reset level.

Row Calculations (Layout Calculated Items)

Row calculations are layout calculated items. They are calculated when the report is formatted.

Layout calculated items have these characteristics:

- They can be any one of these item types:

  N - fixed point numeric
  R - floating=point fixed
  I - integer
  S - string
  D - date
  T - time

- They can be calculated from:

  source table items
  other calculated items or functions, including column calculations
  parameters or constants

- They cannot be used for:

  formulas for table calculated items
  relation conditions
  table functions

- By default, they are calculated for each detail lineset but you can specify that they be calculated for specific linesets.

To define row calculations

1. Choose `Layout Calc Item` on the Define Report screen or on the Define Lineset screen.
2. Type the name of the item, result type, result length, and numeric result precision in the appropriate fields.

```
HP ALLBASE/BRW              Layout Calc Item              Report: CUSTSTAT

  Item  DUNNING-DAYS

  Result Type    I    S = String     N = Numeric fixed
                      D = Date       I = Integer
                      T = Time       R = Real

  Result Length      (if String)

  Numeric Result Precision

  Add      Delete   Define   Parms/   4      9  Prev     Next     Help     Exit
  Item     Item     Formula  Funct'ns           Item     Item
```

3. Choose `Add Item`.
4. Choose `Define Formula`.
5. At the Layout Calc Item Formula screen, type the calculation formula.

```
HP ALLBASE/BRW           Layout Calc Item Formula         Report: CUSTSTAT
Item: DUNNING-DAYS                      Type: I

IF due-date < date
THEN DAYS_DIFF (due-date, date)
ELSE 0

                                    4      1                    Help     Exit
```

6. Press Enter.

To calculate an item on another lineset

1. Go to the Define Lines screen.
2. Type the name of the item in the `Item` field.
3. Type the position and line for the item in the `Pos` and `Line` fields.
4. Type CALC in the `ColCalc` field for that item.
5. Type any other information as appropriate for that item (such as numeric precision or suppression of repetition).
6. Press Enter.

When a layout calculated item is specified with CALC in a lineset, it has the following characteristics

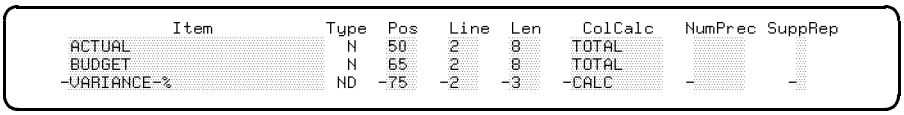
- It is calculated when the lineset for which it is specified is printed.
- It uses a column calculation as an argument in its formula, if the corresponding item and the column calculation are specified in the same lineset.
- Other items used as arguments have the same numeric precision as specified in the lineset containing the layout calculated item.

Example: Row calculations based on column calculations

Suppose you want to calculate the percentage of variance (VARIANCE-%) between two column total values. The layout calculated item and its formula are defined on the Layout Calc Item screen and the Layout Calc Item Formula screen. The formula for VARIANCE-% is:

```
100 * (actual - budget) / budget
```

On the Define Lines screen, the calculated item VARIANCE-% is specified as shown here:



Item	Type	Pos	Line	Len	ColCalc	NumPrec	SuppRep
ACTUAL	N	50	2	8	TOTAL		
BUDGET	N	65	2	8	TOTAL		
-VARIANCE-%	ND	-75	-2	-3	-CALC	-	-

Here is an example of the report using these items:

	Actual	Budget	Variance-%
	500	250	100.00
	300	250	20.00
	900	450	100.00
Totals:	1700	950	78.95%

VARIANCE-% is calculated each time the value of the item account changes. Because CALC is specified in the ColCalc field for this item, the total values of arguments ACTUAL and BUDGET are used to find the percentage of variance of these totals.

In this example, if TOTAL had been specified instead of CALC, the result would be 220%.

Note

On the Define Lines screen, HP ALLBASE/BRW uses the definition of the argument items found in the ColCalc or NumPrec field. For example, a numeric precision for ACTUAL and BUDGET could also be specified in the above example. The percentage of variance would then be calculated from the totals using this precision.

Precision of Numbers

The precision of numbers in column calculation can be controlled by using any number type: fixed-point numeric (N), integer (I), or real (R).

You define numeric precision by specifying a number and method code at one of several screens:

- the Define Lines screen
- the Layout Calc Item screen
- the Table Calc Item screen

The number, as shown in the following table, indicates the power of 10 or 10th exponent that indicates the precision.

Valid number range	for type N:	-6 through 12;
	for type I:	0 through 8.
	for type R:	-90 through 76

Use one of the following method codes to indicate how superfluous digits are to be treated:

Method Code	Digits are:
R	rounded
T	truncated
C	ceiled

T means truncate any remainder. C means round up if the remainder is non-zero. The default method is to round. In the configuration file you specify whether the exact half is to be rounded up or down. See Chapter 17.

Some examples of how to use numeric precision are shown below.

Example: Controlling numeric precision

Suppose you want to obtain total sales values in K$ (where $1000.00 = 1K) for printing or other calculations. In this example, the total is built from the original (unchanged) sales values. First, you specify the item on the Define Lines screen as shown in this break footing lineset:

```
          Item            Type  Pos   Line  Len    ColCalc   NumPrec SuppRep
AMOUNT                     N     54    1     12     TOTAL     3R
```

To print sales values in K$, you specify scaling on the Item Edits screen, as shown here:

```
HP ALLBASE/BRW                Numeric Item Edits                Report: CUSTINV
                                    Detail

Item AMOUNT                 Line 1    Position 54   Length 12

Scaling 3   Decimals 0        Rounding R  (Truncate/Round/Ceiling)
```

To obtain total sales values where the total is built from single sales values changed to K$, specify the item in the detail lineset on the Define Lines screen as shown here:

Item	Type	Pos	Line	Len	ColCalc	NumPrec	SuppRep
AMOUNT	N	54	1	12		3R	

And in the break footing lineset with a TOTAL in the ColCal field as shown here:

Item	Type	Pos	Line	Len	ColCalc	NumPrec	SuppRep
AMOUNT	N	54	1	12	TOTAL		

Example: Controlling numeric precision

As another example, if you wanted to obtain the result of the layout calculated item tax-amount with two decimals, you would specify -2R as the numeric precision.

Suppressing Lines and Linesets

HP ALLBASE/BRW lets you specify condition formulas which control when lines and linesets are printed. Conditions use the Calculation Language described in Chapter 15; see the section in that chapter called Rules For Formulas and Conditions.

Suppress Line Conditions

For each line in a report you can define a condition formula that expresses when that line is to be suppressed. Suppressing line conditions only suppress printing of the line and does not affect other parts of the report, as suppress lineset conditions can.

To suppress a line in a report

1. At the Define Lines screen, choose `other keys` until the `Suppress Line` softkey is available. Choose `Suppress Line`.
2. Answer the prompt for the appropriate line:

   ```
   Suppress condition for Line  :
   ```

3. At the Suppress Line screen, type the formula for the condition.
4. Press Enter.

Example: Suppressing a line

For example, to suppress an address line, use the condition:

```
address2 = " "
```

Before a lineset is printed, HP ALLBASE/BRW first checks that it will fit completely on the page (unless you have defined that lineset split is allowed). Note that suppressed lines are not counted in that check.

Standard items PAGE and LINE can have other values in the suppress line condition than when they are printed in that line, because all suppress line conditions in a lineset are evaluated before pagination and printing of the first line.

Suppress Lineset Conditions

You can specify when a complete lineset is to be suppressed. The suppress lineset condition formula suppresses:

- all lines in the lineset
- any record to be written to the lineset's output file
- column calculations to be counted, if this is a detail lineset or a break footing lineset
- calculation of layout calculated items for this lineset
- values from the standard items prev-lset-level and prev-lset-type

To suppress a lineset

1. At the Define Breaks screen, choose `Suppress Lineset`.
2. Answer the prompt for the appropriate lineset level:

   ```
   Specify lineset level (D,P,R,1-9):  and type (H,F):
   :
   ```

3. At the Suppress Lineset screen, type the formula for the condition.
4. Press Enter.

Example: Suppressing a lineset

Suppose you want to suppress a detail line if there is no order. You could define a suppress lineset condition as shown below.

```
order-no = NO_VALUE
```

Since the suppress lineset condition is evaluated first, all column calculations and standard items page and line have the values from the previous lineset.

Layout calculated items which use a column calculation, page or line as an argument can have different values if they are used for the suppress lineset condition or for another purpose in the lineset.

Special Cases

This section provides some examples for special uses of Report Layout features.

Example: Using alternate lines

Instead of NO_VALUE item edits, you can specify another line in the detail lineset with a phrase to be used when the item has no value. Note, however, that the two lines need complementary suppress line conditions.

The following Customer Order Report has an open join on the orders table. If a customer has no orders, detail line 2 prints:

```
**Customer has no orders**
```

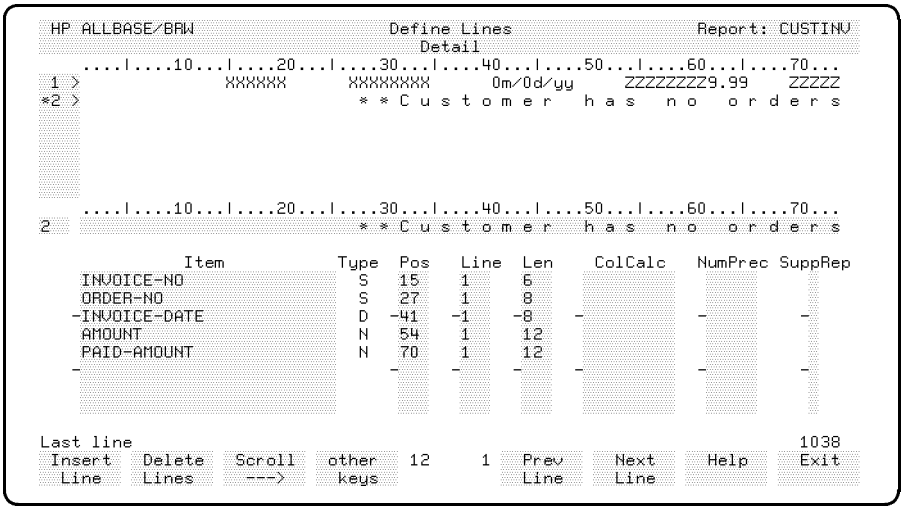
The suppress line condition (line 2) for this is:

NOT (Order-no = NO_VALUE)

In the case of orders from a customer, the suppress line condition for line 1 is:

Order-no = NO_VALUE

The Define Lines screen for this report looks like this:



Example: Bucketing reports

Suppose you want to consolidate several reports into one. In order to do that, you need to rearrange some of the numeric data from one table from a vertical orientation to a horizontal orientation.

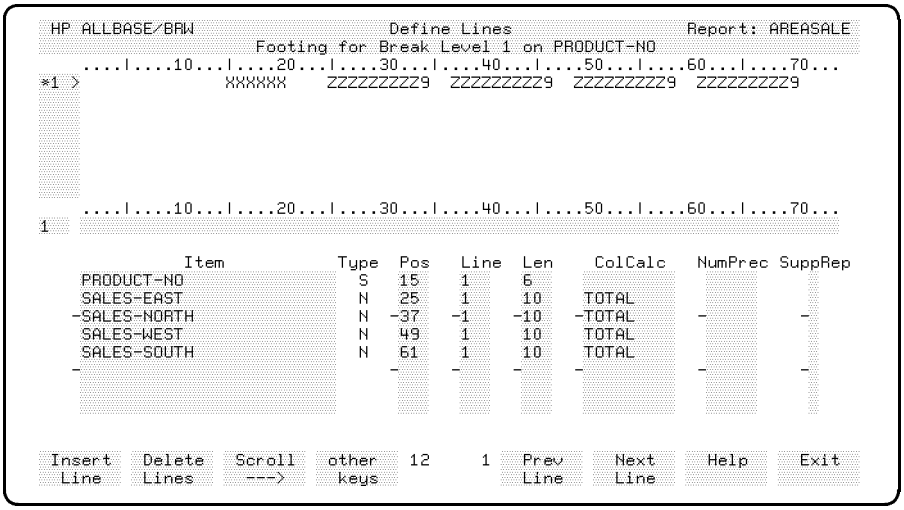
The report below is an example of bucketing, in which numeric data has been rearranged.

```
Report: SALESREP                                    11/11/86  Page  1

                      Area Sales Report by Product
                      ----------------------------

  Product Line:  60

           Product                EAST       NORTH        WEST       SOUTH
           -------              ----------  ----------  ----------  ---------

           47392A                 352.00     9912.75       45.00    2233.50
           14326A                7586.24      789.00     3346.80   24570.00
           47392X                4050.00        0.00     4050.00    8100.00
           78532J                3650.00      500.00     3650.00       0.00
                                ----------  ----------  ---------  ---------


    Total for product line      15638.24    11201.75    11091.80   34903.50


  Product Line:  70

           Product                EAST       NORTH        WEST       SOUTH
           -------              ----------  ----------  ----------  ---------

           47392A                 598.00       50.00      243.95     352.87
           14326A                3246.90      100.00       33.15      33.99
           47392X                4050.98     2435.60     5690.35     220.00
           78532J                9900.10      436.96      445.82    3970.00
                                ----------  ----------  ---------  ---------


    Total for product line      17795.98     3022.56     6413.27    4576.86
```

In order to do this using the sample database TOYDB, you can define four required table calculated items as listed below:

```
SALES-EAST  IF sales-area = "EAST" THEN quantity * price ELSE 0

SALES-NORTH IF sales-area = "NORTH" THEN quantity * price ELSE 0

SALES-WEST  IF sales-area = "WEST" THEN quantity * price ELSE 0

SALES-SOUTH IF sales-area = "SOUTH" THEN quantity * price ELSE 0
```

On the Define Lines screen below, the break footing on product-number is defined. This line is used to print the bucketed data. Note that the report has no detail lines.

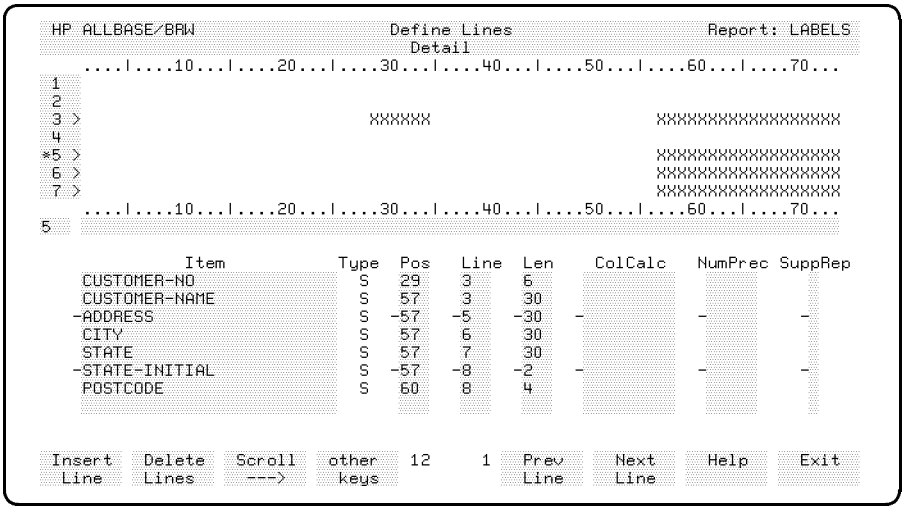


Example: Reports for pre-printed forms

HP ALLBASE/BRW lets you set up a report to print on pre-printed forms, provided you specify the correct positioning of items on the report layout. Follow the steps below to set up a report for a pre-printed form:

1. Set up a template report with columns and row numbers which count across and down the layout, as shown on the next page.
2. Print the template on the pre-printed form paper.
3. Correlate the positions on the form with those on the report template to determine text and item positioning.
4. Compose the report layout based on the findings in step 3.

In the example on the next page, a mailing address labels report has been created. The screen shows the resulting lines and positioned item.

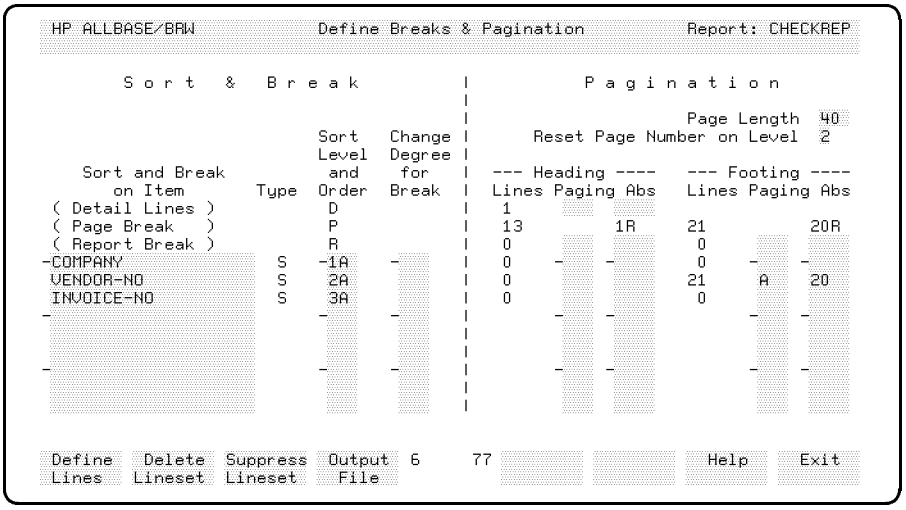


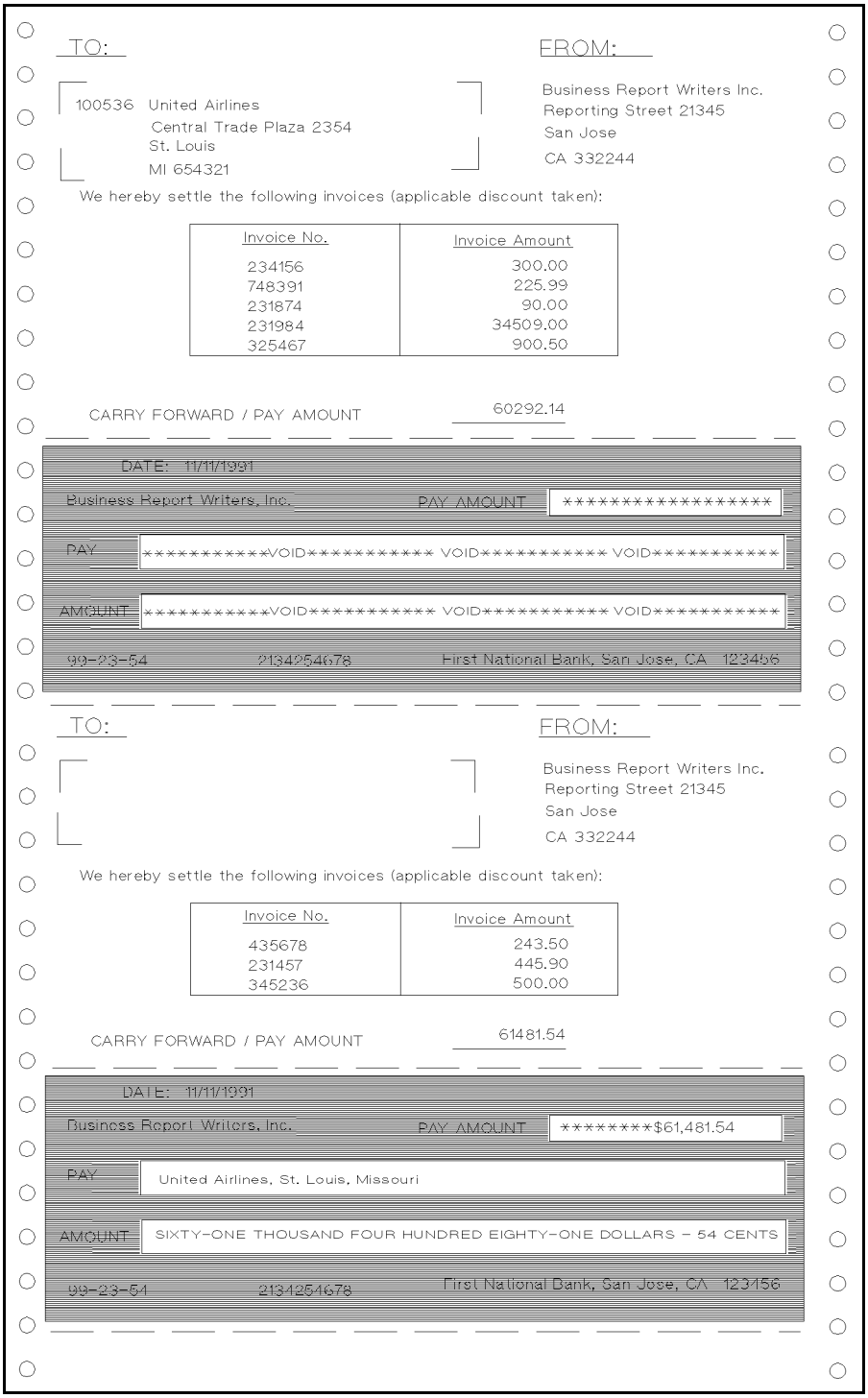
```
10 4567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
 2                                                                                                    2
 3          CUSTOMER-NO: __________       CUSTOMER-NAME: ____________________                       3
 4                                                                                                    4
 5                                        ADDRESS:       ____________________                       5
 6                                                       ____________________                       6
 7                                                       ____________________                       7
 8                                                       -- __________                              8
 9                                                                                                    9
20 4567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
 1                                                                                                    1
 2                                                                                                    2
 3          CUSTOMER-NO: __________       CUSTOMER-NAME: ____________________                       3
 4                                                                                                    4
 5                                        ADDRESS:       ____________________                       5
 6                                                       ____________________                       6
 7                                                       ____________________                       7
 8                                                       -- __________                              8
 9                                                                                                    9
30 4567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
 1                                                                                                    1
 2                                                                                                    2
 3          CUSTOMER-NO: __________       CUSTOMER-NAME: ____________________                       3
 4                                                                                                    4
 5                                        ADDRESS:       ____________________                       5
 6                                                       ____________________                       6
 7                                                       ____________________                       7
 8                                                       -- __________                              8
 9                                                                                                    9
40 4567890123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
 1                                                                                                    1
 2                                                                                                    2
```

Example: Voiding unused check forms

The report shown on the next page and the following series of screens illustrate how to void unused check forms with HP ALLBASE/BRW. The check report, shown below, has two alternate footings, which print the check stub (grey areas). The first footing is a page footing. The second footing is a break footing on vendor.

The Define Breaks screen below shows the break footing as positioned absolutely in the same position as the page footing. When the break footing occurs, no page footing is printed on that page. If a break continues over more than one page, the page footing is printed.




TO:

100536 United Airlines
Central Trade Plaza 2354
St. Louis
MI 654321

FROM:

Business Report Writers Inc.
Reporting Street 21345
San Jose
CA 332244

We hereby settle the following invoices (applicable discount taken):

Invoice No.	Invoice Amount
234156	300.00
748391	225.99
231874	90.00
231984	34509.00
325467	900.50

CARRY FORWARD / PAY AMOUNT 60292.14

DATE: 11/11/1991

Business Report Writers, Inc. PAY AMOUNT **********

PAY **********VOID********** VOID********** VOID**********

AMOUNT **********VOID********** VOID********** VOID**********

99-23-54 2134254678 First National Bank, San Jose, CA 123456

TO:

FROM:

Business Report Writers Inc.
Reporting Street 21345
San Jose
CA 332244

We hereby settle the following invoices (applicable discount taken):

Invoice No.	Invoice Amount
435678	243.50
231457	445.90
345236	500.00

CARRY FORWARD / PAY AMOUNT 61481.54

DATE: 11/11/1991

Business Report Writers, Inc. PAY AMOUNT ********$61,481.54

PAY United Airlines, St. Louis, Missouri

AMOUNT SIXTY-ONE THOUSAND FOUR HUNDRED EIGHTY-ONE DOLLARS - 54 CENTS

99-23-54 2134254678 First National Bank, San Jose, CA 123456

Sample voided check form

The following two Define Lines screens show the page footing lineset and the break footing lineset. The page footing prints text which voids the check stub. The break footing prints the actual check amount.

```
HP ALLBASE/BRW                 Define Lines                 Report: CHECKREP
                               Page Footing
    ....|....10...|....20...|....30...|....40...|....50...|....60...|....70...
 10
 11                                                               **************
 12
 13
*14           **********VOID*************VOID**************VOID************
 15
 16
    ....|....10...|....20...|....30...|....40...|....50...|....60...|....70...
14            **********VOID*************VOID**************VOID************

              Item            Type  Pos   Line  Len   ColCalc   NumPrec  SuppRep
    AMOUNT                     N    44    4     14    TOTAL2
   _                               _     _     _     _          _         _

   _                               _     _     _     _          _         _


 Insert   Delete   Scroll   other   12     1   Prev    Next    Help    Exit
  Line    Lines     --->    keys               Line    Line
```

```
HP ALLBASE/BRW                 Define Lines                 Report: CHECKREP
                  Footing for Break Level 1 on VENDOR-NO
    ....|....10...|....20...|....30...|....40...|....50...|....60...|....70...
 10
 11                                                               99999999999999
 12
 13
*14           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 15
 16
    ....|....10...|....20...|....30...|....40...|....50...|....60...|....70...
14

              Item            Type  Pos   Line  Len   ColCalc   NumPrec  SuppRep
    AMOUNT                     N    44    4     14    TOTAL
    AMOUNTX                    N    12    11    23    TOTAL
   -VENDOR-STRING              S   -12   -14   -61   -          -         -
    AMOUNT-STRING              N    12    17    61    CALC

   _                               _     _     _     _          _         _


 Insert   Delete   Scroll   other   12     1   Prev    Next    Help    Exit
  Line    Lines     --->    keys               Line    Line
```

Example: Using horizontal pages

You can set the number of horizontal pages for each physical page. HP ALLBASE/BRW will then print a series of horizontal pages on one physical page. You set the number of horizontal pages from a field on the Define Report screen, labeled "Number of Horizontal Pages".

For example, if you want to print address labels, and the address label paper is set up so that the labels are three abreast, you can set the number of horizontal pages to 3.

The labels start printing at the left-hand corner of the page and continue down the page to the physical end of the page. At the end of the page, no page eject occurs. Instead, the second column of addresses starts at the top of the same sheet in the center column. Addresses print down the center column until the end of the page. The next addresses start at the top of the page in the right-hand

column. At the bottom of the third column, the paper ejects and the next set of labels starts at the top of the left-hand column of the next sheet.

Note that HP ALLBASE/BRW considers each label to be one report - only the manner of printing changes. The default is one horizontal page, or one page per sheet.

The number of horizontal pages multiplied by (*) the page width cannot exceed the maximum physical page width of 300 characters.

What Next?

Before you save and compile your report, you can define specific data that you want the report to capture. Chapter 9, Defining Data Selection, shows you how to keep your data secure and provide for flexible reports.

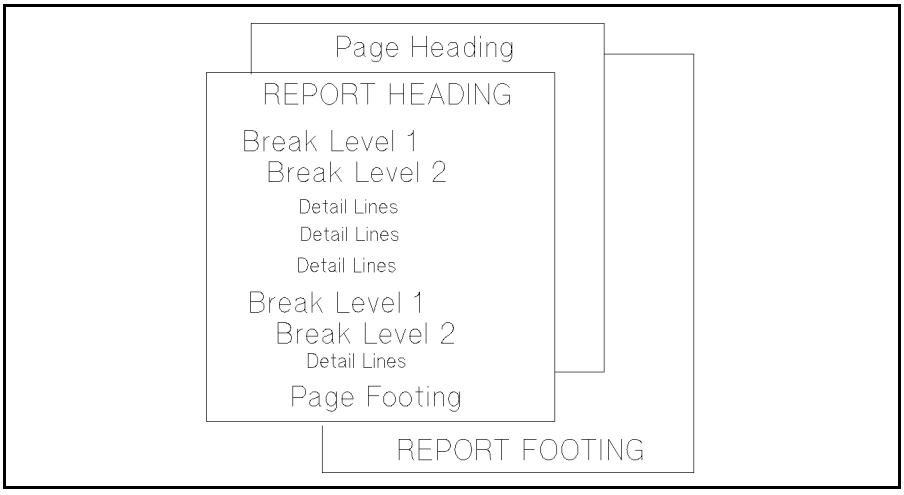
8

Sorts, Breaks, and Pagination

Structuring the basic report

At a Glance

After you decide on the items you want in your report, you are ready to specify which items are to be sorted, and in what order. You can also decide to break the report into logical divisions. You might want the CUSTINV report broken up by sales area, for instance. You can also define when a new page should start.

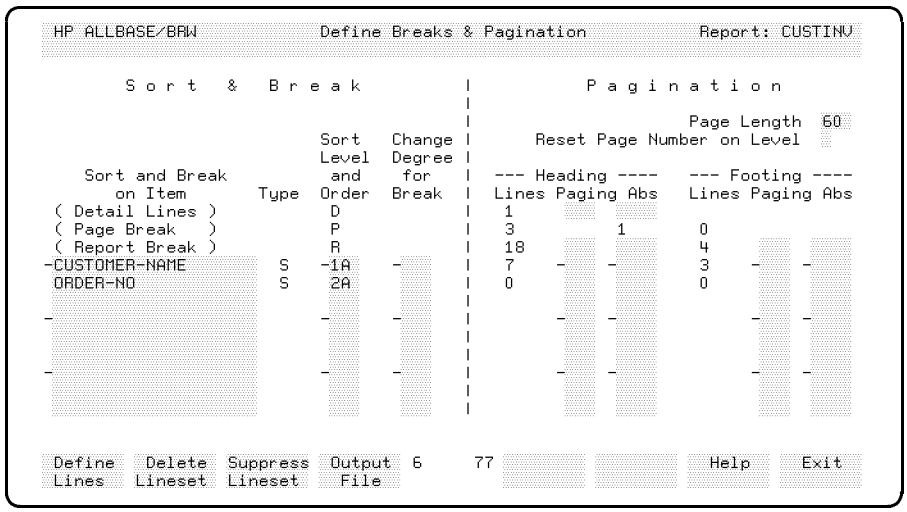


This chapter covers

- using the Define Breaks & Pagination screen
- sorts and breaks
- pagination

Using the Screen

The Define Breaks and Pagination screen is broken into two sides, the Sort & Break side and the Pagination side.



The top part of the screen gives you some important information about the existing linesets, as described below for the example screen show above.

- The (Detail Lines) information shows that there is 1 detail line.
- The (Page Break) information shows that there are 3 page break heading lines, requiring an absolute (`Abs`) page position.
- The (Report Break) information shows that there are 18 lines that print on the top of the page (a heading) for the report break (essentially the cover page of the report) and 4 lines that print on the bottom of the page (a footing) at the end of the report.

This information helps you make other pagination and break decisions.

The bottom part of the screen lets you define sort levels, order, degree of change, and placement of heading and footing lines.

Sorts and Breaks

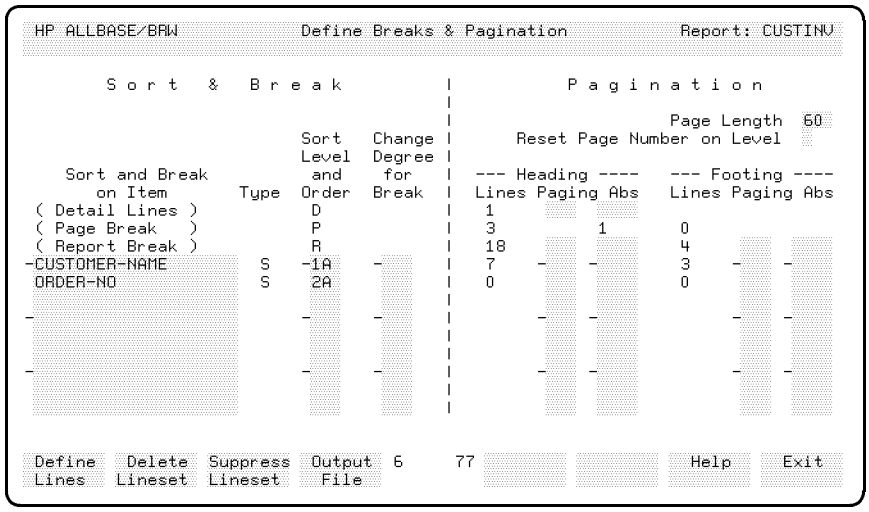
Data of a report is arranged by using up to nine nested sort items, each in ascending or descending sequence.

Breaks occur each time a sort item value changes. For instance, if you sort on CUSTOMER-NAME, all the occurrences of a particular customer name are grouped together. When the next group of customer names is ready to print, that is a break.

When you define the report layout, you can create a break heading or footing so you can have text appear before or after the break occurs. See Chapter 7 for details on break headings.

To sort and break the report

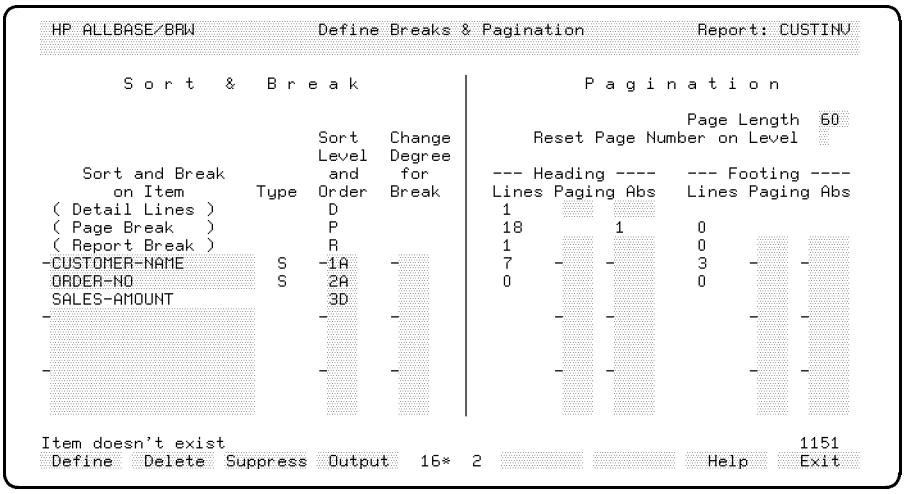
1. From the Define Report screen, choose `Define Breaks`. The Define Breaks and Pagination screen appears.



2. Type the item names to sort in the `Sort and Break on Item` column.
3. Tab to the `Sort Level and Order` field and type a number (1 through 9) for the sort level.
4. Type a letter (next to the sort level number) for Ascending (`A`) or Descending (`D`). Ascending sorts the records from A to Z or from 0 to 9 (and up). Descending sorts the records in the reverse order.

Example: Specifying a sort sequence

In this example, a sort sequence has been specified for the items CUSTOMER-NAME, ORDER-NO., and SALES-AMOUNT.



For each customer name, order numbers are sorted, and for each order number, sales amounts are sorted. Also, for CUSTOMER-NAME and ORDER-NO, ascending sort order is specified, and for SALES-AMOUNT, descending sort order is specified.

Customizing Breaks

Every time the value of a sorted item changes, you can customize your report at the break that occurs. For instance, you can:

- Have additional text, items, or summarized data printed either before a break or after a break. See Chapter 7 for details on how do do this.
- Specify a break to occur when only part of the value changes. You can indicate to what *degree* the value can change before a break occurs. That is, you can create a break if only a certain part of the item value changes. For instance, suppose you want a break in your report every time the month changes in a date item. You specify a "Change Degree" for that sort and break. See the following procedure.

To indicate a change degree for a break

1. At the Define Breaks and Pagination screen, type the instructions for the change degree in the `Change Degree for Break` field. Use the table below as a guide.

For . . .	Type in the `Change Degree for Break` field:
String items	A number for the last character position where you want the break to begin if there is a change. For instance, if the item has 26 characters, and you want to break if there is a change in the first 10 characters, you type `10` in the `Change Degree for Break` field. Count the character position beginning from the left.
Numeric items	A number that is the least significant digit for the break as the 10th exponent of this digit. Valid number ranges are as follows: N(umeric): -6 to 12 I(nteger): 0 to 8 R(eal): -90 to 76
Time items	One of the following: ■ H - to break when the hour changes ■ HM - to break when the hour or minute changes ■ HMS - to break when the hour, minute, or second changes
Date items	One of the following: ■ Y - to break when the year changes ■ YM - to break when the year or month changes ■ YMD - to break when the year, month, or day changes

For more information on change degrees, see the field descriptions for the Define Breaks screen in Chapter 16.

Example: Breaking when the year or month change.

In this example, the report will break when the year or month (YM) of the ORDER-DATE field change.

```
                                Sort   Change |      Reset Page Number on Level
                                Level  Degree |
     Sort and Break                      for   |  --- Heading ----   --- Footing ----
        on Item          Type  Order  Break   |  Lines Paging Abs   Lines Paging Abs
   ( Detail Lines )            D              |   1
   ( Page Break   )            P              |   3          1       0
   ( Report Break )            R              |   18                 4
  -ORDER-DATE              D   -1A    -YM     |   7    -     -       3    -     -
```

Pagination

Pagination lets you control when different groups of lines (linesets) in your report are printed on the pages of the report. For instance, pagination controls these elements:

- How many lines per page?
- Do you want page numbering to start over when a break begins?
- At which line should a new page always start?
- If a lineset won't fit on the remaining page, should the lineset be split, or printed completely on the next page?

You define pagination on the Define Breaks screen. Even if a lineset contains no lines, you can specify pagination for it.

For each lineset (except for page heading and page footing linesets), you can specify that a new page must begin before this lineset is printed and/or after this lineset is printed.

To define pagination

1. At the Define Report screen, choose `Define Breaks`.
2. At the Define Breaks and Pagination screen, type the pagination specifications in the fields provided.

```
                                         |
                                         |                        Page Length  60
                           Sort   Change |      Reset Page Number on Level
                           Level  Degree |
     Sort and Break         and    for   |  --- Heading ----   --- Footing ----
        on Item       Type Order  Break  |  Lines Paging Abs   Lines Paging Abs
  ( Detail Lines )          D            |   2
  ( Page Break   )          P            |   3            1      0
  ( Report Break )          R            |   18                  4
 -CUSTOMER-NO          S   -1A    -      |   7     -B   -        3     -     -
```

`Page Length`	The default is 60 lines. The limits of the length depend on your printer and the size of the paper. For instance, a 8 1/2"x 11" paper can print up to 66 lines per page (assuming 6 lines per inch). The default of 60 lines gives a 1/2 inch (3 lines) margin at the top and bottom.
`Reset Page Number on Level`	You can set the page number to 1 everytime a specific break level occurs. For example, if the report prints data for several departments in one report and you have a break specified for each time the department changes, you can reset the page number to 1 for the first page of each department. A new page is started when the page number is reset.
`Paging`	Use one or any combination of these codes to indicate when a new page should start: `A` = after printing a lineset `B` = before printing a lineset `S` = a lineset should be split The default is that linesets are not split and a new page is started before a lineset if it doesn't fit completely on the rest of the page.
`Abs`	The absolute position for a lineset on the page. In addition to specifying the exact line for the beginning of the lineset (for instance, for a break heading or footing), you can also type an `R` after the number to indicate that the space must be reserved. If a lineset is not used on a particular page, other linesets are not printed in that space, unless they have a absolute position defined. If a lineset with an absolute position overlaps a page heading or footing, the page heading or footing is suppressed on that page.

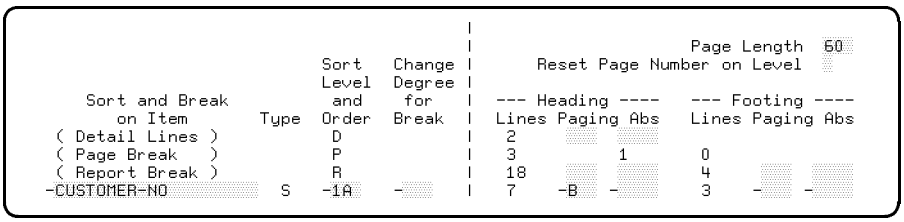
Example: Resetting the page number to 1

If you want a new page and a new page number 1 to start each time the report begins at a sort level 1, type the number `1` in the `Reset Page Number on Level` field. If you want the page numbering to be set at sort level 3, type `3` in the field.

A new page is started automatically when the page number is reset.

Example: Breaking before a lineset

In this example, the break for the CUSTOMER-NO causes a page break *before* the first heading lineset for CUSTOMER-NO begins. This is indicated by the `B` in the `Paging` field for the heading for CUSTOMER-NO.



Sort and Break on Item	Type	Sort Level and Order	Change Degree for Break	Page Length 60 / Reset Page Number on Level	Heading Lines	Heading Paging	Heading Abs	Footing Lines	Footing Paging	Footing Abs
(Detail Lines)		D			2					
(Page Break)		P			3		1	0		
(Report Break)		R			18			4		
-CUSTOMER-NO	S	-1A	-		7	-B	-	3	-	-

Example: Splitting across page boundries

In this example, the footing lineset is split across the page boundry if there isn't enough room for the lineset on the rest of the page. This is indicated by the `S` in the `Paging` field for CUSTOMER-NO footing.

```
                                         |
                                         |                    Page Length  60
                          Sort   Change  |      Reset Page Number on Level
                          Level  Degree  |
     Sort and Break        and    for    |  --- Heading ----   --- Footing ----
        on Item      Type Order  Break   |  Lines Paging Abs   Lines Paging Abs
  ( Detail Lines )         D             |   2
  ( Page Break   )         P             |   3           1       0
  ( Report Break )         R             |   18                  4
 -CUSTOMER-NO          S  -1A     -      |   7     -    -        3    -S   -
```

When a lineset is split, the page footing and page heading linesets are printed between the split parts of the lineset.

Example: Printing a Lineset at an Absolute Page Position

Normally, a lineset is printed directly after the preceding lineset. However, HP ALLBASE/BRW also lets you specify an absolute page position (a fixed position on the page) by typing a number and, optionally, an `R` to reservse the space in the `Abs` field, as shown in this screen:

```
                                         |
                                         |                    Page Length  60
                          Sort   Change  |      Reset Page Number on Level
                          Level  Degree  |
     Sort and Break        and    for    |  --- Heading ----   --- Footing ----
        on Item      Type Order  Break   |  Lines Paging Abs   Lines Paging Abs
  ( Detail Lines )         D             |   2
  ( Page Break   )         P             |   3           1       0
  ( Report Break )         R             |   18                  4
 -CUSTOMER-NO          S  -1A     -      |   7     -    -1       3    -S   -
```

The report based on this definition looks like this:

```
Customer No.  10044


        Inv. No.          Amount          Paid Amount

        --------          ------          -----------

        4458A             700.00           700.00

        5569B             350.00           350.00

        5570B             554.00           554.00

        66723            1079.00          1000.00

                         -------          -------

        Total For This

        Customer         2683.00          1683.00
```

The heading for CUSTOMER-NO always begins on line 1 of the report. Since the space is not reserved, other pages that don't have a break heading can begin on line 1, if necessary.

Note

Two linesets with the same or an overlapping absolute page position will be printed on different pages. The page heading and page footing linesets, however, are suppressed if another lineset occupies the space in which they should be printed. As an example, see Voiding Unused Check Forms in the Special Cases section of this chapter.

If linesets have space reserved, linesets without an absolute page position are only printed in the first space not reserved on the page.

What Next?

After sorting your report to create breaks and after setting the pagination criteria, you can begin the detailed layout of your report. See Chapter 7 for information about defining the report layout.

9

Defining Data Selection

Security and flexibility of data selection

At a Glance

HP ALLBASE/BRW data selection lets you use these features for flexible, but secure data selection:

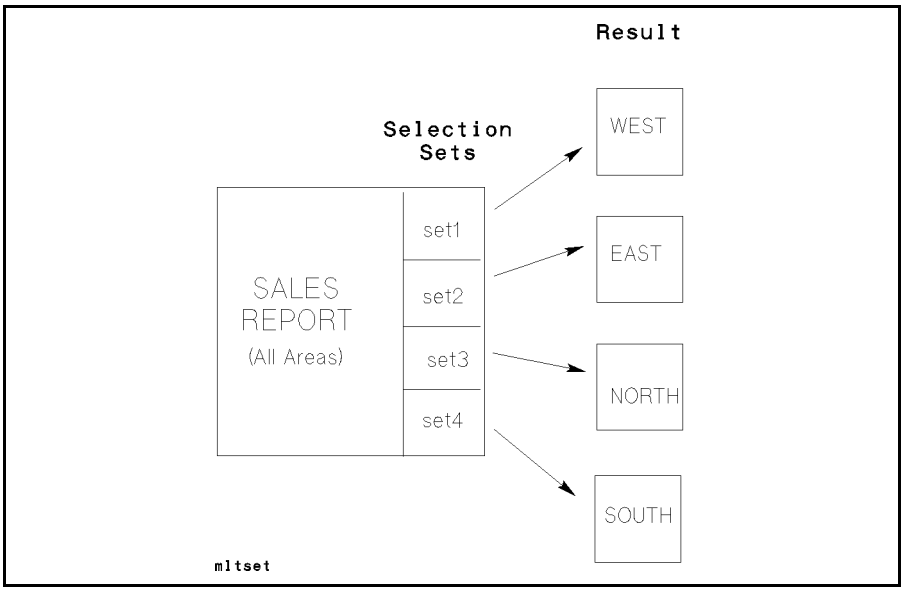
- execution parameters
- default values for parameters
- frozen and required defaults
- multiple selection sets
- calculation language for complex data selection formulas
- data-value-dependent security provided by multiple selection sets and passwords

There are two ways to select data:

For security	You can define data selection when you specify the report so that it becomes an integral part of the report.
For flexibility	You can define data selection with parameters whose values can be selected when the report is requested for execution.

Example

Suppose you have one report that contains all the information about company sales for every sales area. You would like a separate report for each area. By creating separate selection sets, you can produce a sales report for each sales area.



After you create one or more selection sets, you can use the Calculation Language, discussed in Chapter 15, to specify a condition for data selection. The selection condition determines which records

(that is, details) from the final access table are included in the report.

This chapter covers

- selecting data at specification time
- selecting data at execution time
- using selection conditions
- defining parameters

Selecting Data at Specification Time

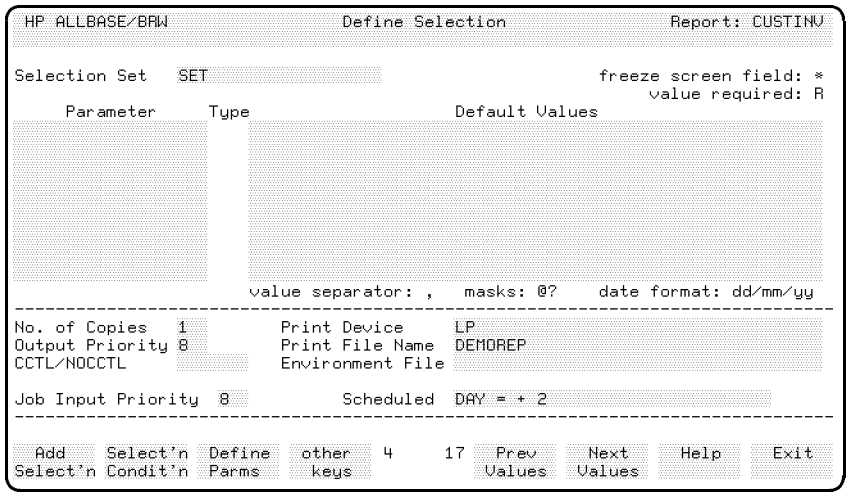
The simplest method of data selection is to define a complete **selection set** when you define the report specifications. A complete selection set includes all the information about run-time parameters and printer setups. When the report is executed, no additional information is needed. You can create multiple selection sets for the same report so that you can use one report as a basis for producing several subsets of the same report. At compile time, the person requesting the report names the selection set that applies to the report he or she wants to run.

Note

If you redefine a selection set after you have compiled the report, you need to re-compile the report.

To define a secure data set

1. At the Define Report screen, choose `Define Selection`.



2. Type a name for the Selection Set.
3. In the Parameter field, type the name of any parameters you have defined and type the pre-determined value of that parameter in the Default Values field.

   If you want to prevent a user from overriding the parameter value at runtime, you can type an asterisk (*) in the last column of the default value field. See the "Execution Defaults" section of this chapter for more information.
4. Type the print specifications (such as number of copies) in the appropriate fields.
5. Choose `Add Select'n`.
6. Choose `Select'n Condit'n` to define selection conditions (described in a following section of this chapter).

A selection condition specifies selection rules that determine which records from the final access table are included in the report.

Parameters [put a definition here].

Selecting Data at Execution Time

For flexibility in the selection of data for a report, you can define the data set so that the person requesting the report can enter the values and operators for previously defined parameters.

Note

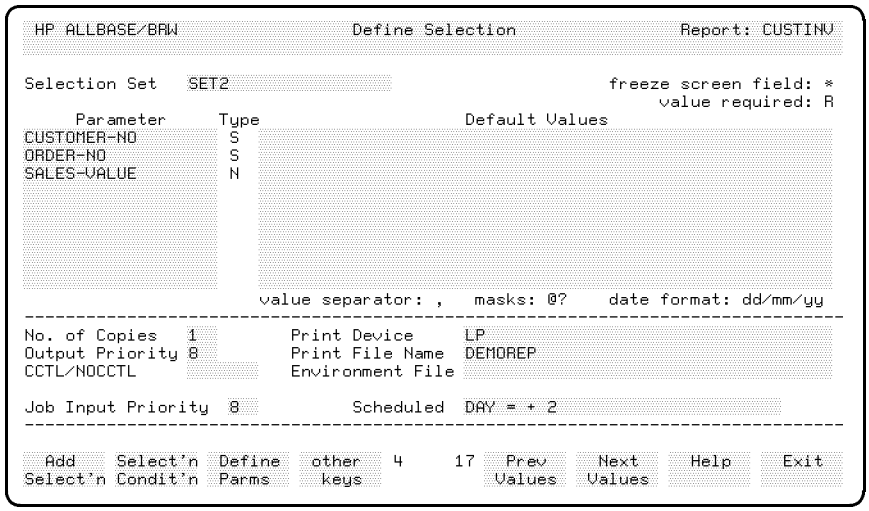
Changes to parameter operators and values at execution time do not require you to re-compile the report.

To define a flexible data set

1. At the Define Report screen, choose `Define Selection`.

```
 HP ALLBASE/BRW                Define Selection              Report: CUSTINV

Selection Set   SET                                        freeze screen field: *
                                                              value required: R
     Parameter      Type                   Default Values




                                     value separator: ,    masks: @?    date format: dd/mm/yy
--------------------------------------------------------------------------------
No. of Copies    1             Print Device      LP
Output Priority  8             Print File Name   DEMOREP
CCTL/NOCCTL                    Environment File

Job Input Priority  8             Scheduled      DAY = + 2
--------------------------------------------------------------------------------
  Add     Select'n  Define   other   4     17  Prev     Next     Help     Exit
Select'n  Condit'n  Parms    keys              Values   Values
```

2. Type a name for the Selection Set.
3. Type the parameter name and parameter type in the corresponding fields.
4. You can type default values or leave the `Default Values` field blank. For example:



You can enter multiple parameters. If you run out of lines available, use `Next Values` to access more lines.

5. Type the print specifications (such as number of copies) in the appropriate fields.
6. Press `Add Selection` when you are satisfied with your definitions.
7. Press `Select'n Condit'n` to go to the Selection Condition screen if you are defining selection conditions.

You define the selection conditions that determine which records from the final access table are included in the report. See the Define Selection Conditions section in this chapter.

You can also go to the Parameter screen from the Define Selection screen. The Parameter screen lets you define parameters you want to use at run time if you did not define the parameters at some other point in the specification process.

Parameters

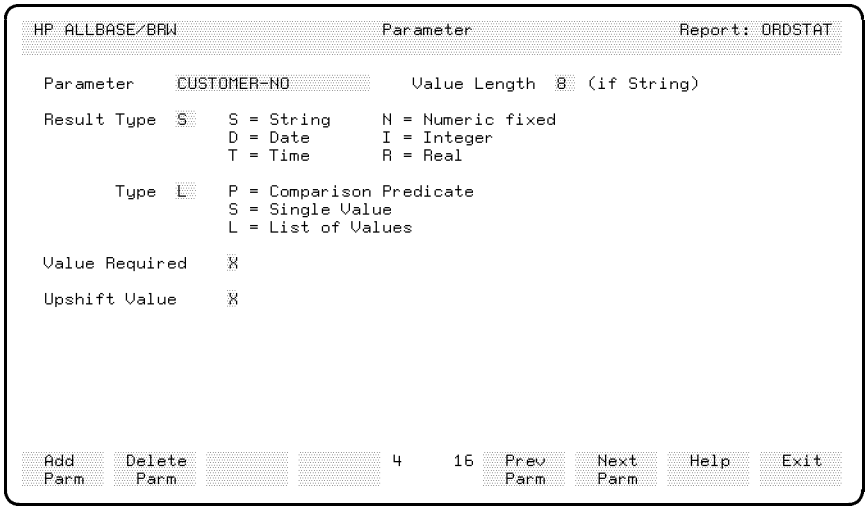
There are three times when you can define parameters:

- When you define layout calculated items
- When you define table calculated items
- When you create a selection set

You access the Parameter screen from either the Layout Calc Item screen, the Table Calc Item screen, or the Define Selection screen.

To define parameters:

1. At the Table Calc Item or Layout Calc Item screen, press `Parms/Funct'ns` and type `P` (for Parameters) at the prompt. Or, at the Define Selection screen, just choose `Define Parms`.



2. Type the name of the parameter in the `Parameter` field.
3. Type other parameter values in the appropriate fields.

Selection Conditions

After you have the parameters and the selection set defined, you can set up some selection rules at the Selection Condition screen. Remember that you must define the parameters and enter the name in the `Parameter` field in Define Selection before you can use the parameter in either a selection condition or in a relation condition.

To define Selection Conditions

1. At the Define Selection screen, choose `Select'n Condit'n`.
2. Type the selection conditions using the Calculation Language as described in Chapter 15.

   This is an example of selection conditions using the Calculation Language:

```
HP ALLBASE/BRW              Selection Condition              Report: DEMOREP1
Selection Set: SET2

customer-no IN ?customer-no

AND order-no SATISFIES ?order-no

AND sales-value SATISFIES ?sales-value
```

3. Press Enter when you are satisfied with the calculation.
4. Choose `Exit` to return to the previous screen.

Note

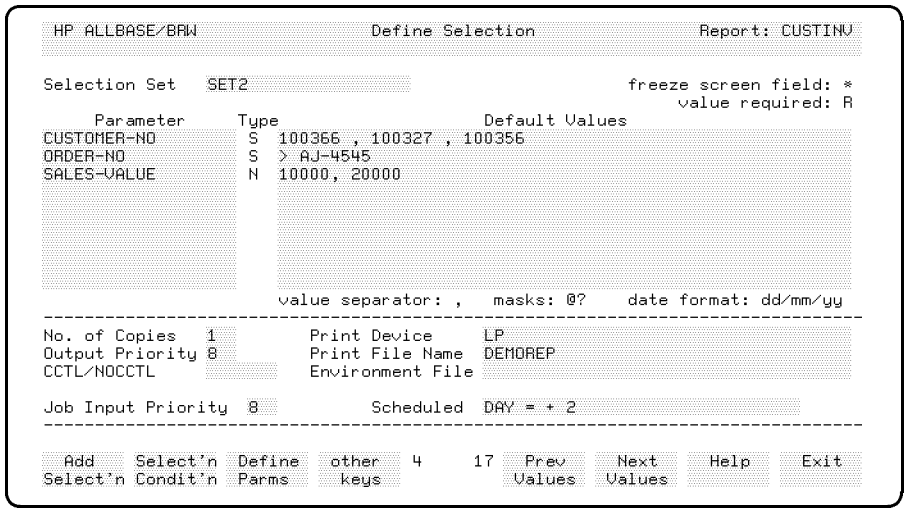
Instead of using a selection condition, you can define the same criteria in a relation condition for the final access table. The only difference is that the relation condition applies to the *whole* report, regardless of the selection set with which the report is compiled, while the selection condition applies to a specific selection set. See Chapter 2 for more about relation conditions.

Execution Defaults

For convenience when you execute a report, you can specify report execution defaults on the Define Selection screen. You can create a default for:

- parameter values
- printer characteristics, (that is, number of copies, output device, output file name, whether to have CCTL codes in printfiles, and the environment file)
- job input priority
- schedule for the report job

This section of the Define Selection screen illustrates execution defaults:



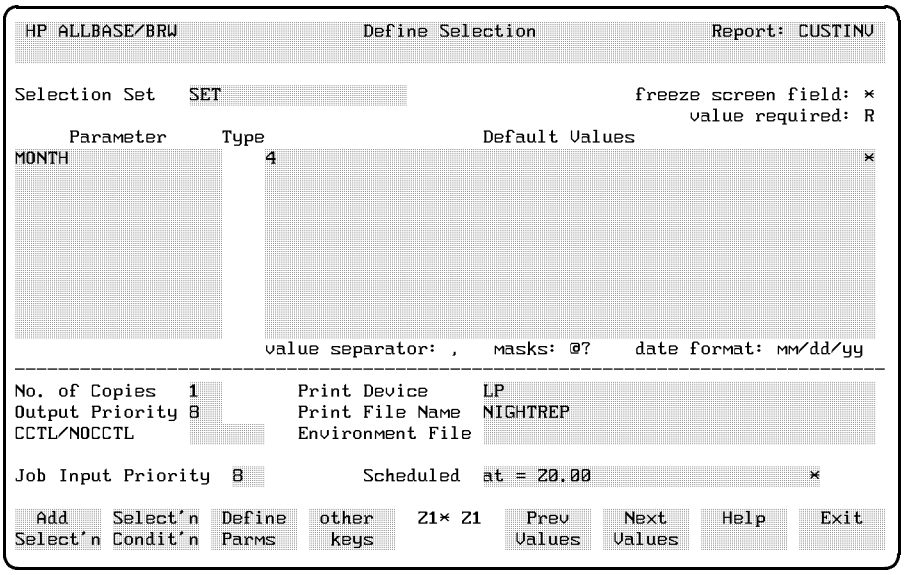
Any defaults you define at the Define Selection screen are also displayed on the Request Report screen for a specific execution file. You can change the defaults on the Request Report screen before you execute the report. If you change the defaults, you don't have to recompile the report. The exception to this rule is if "freeze screen field" or "Value required" are used. See the examples for "Frozen Defaults" and "Required Parameters" in this chapter.

In addition, all defaults, except for schedule details, are used if you execute the report directly using the BRWEXEC program *without* specifying parameter values. See *Requesting Reports Stand-alone* in Chapter 11 for details.

Example: Frozen defaults

You can freeze the schedule, printer characteristics and parameter values for security purposes by adding an asterisk (*) in the right corner of each of these fields on the Define Selection screen. This prevents values from being modified at report execution time.

The schedule and parameter values in the following selection set cannot be changed at run time because there is an asterisk for the parameter in the Default Values field and for the time value in the Scheduled field.



Required parameters can be specified if a formula in the report specification requires a parameter value. When an `R` is in the right hand corner of the Default Values field, the report requester is required to enter a value at report execution time.

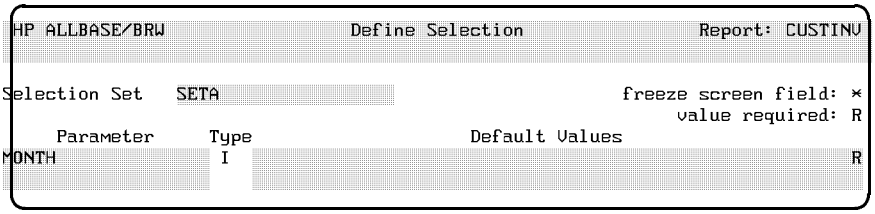
To specify a required parameter:

1. At the Parameters screen, type `X` in the `Value Required` field.
2. Choose `Add Parm` and `Exit`.

Or ...

1. At the Define Selection screen, space over to the far right of the `Default Values` field and type `R`.
2. Press Enter or choose `Add Selection`.

In the example below, the parameter *month* must have a value entered at execution time.



The difference between these two methods is that when you mark `Value Required` at the Parameters screen, that requirment affects any and all reports executed, regardless of the selection set used. If you type the `R` on the Define Selection screen, that parameter is only required when you run a report using that selection set.

Example: Specifying a required parameter

Suppose your Sales Report for all areas contains a COUNTRY item. You know the WEST, NORTH, and SOUTH areas are all within the United States, so you don't need the COUNTRY field. But the EAST area includes England, Germany, France, Spain, and Italy. You want the EAST report to be subdivided by country. You can create a selection set for the EAST that requires a parameter for COUNTRY. When you request the EAST report, you use the selection set that requires a value for country.

Security Dependent on Data Value

For greater control of the data available to different groups of users or departments, HP ALLBASE/BRW provides data-value-dependent security. You can define one report and several selection sets, one for each user group, and secure each department's data with a password. This restricts the data that can be selected by a specific group. The complete data selection, including a selection set for each group of users, is defined when the report is specified and compiled. When the report is executed, the same selection set is always used by a particular group.

Example:

The following final access table has data for four departments:

DEPT	ACCT	MONTH	EXPENSE
ADMIN	13	X	X
ADMIN	14	X	X
ADMIN	15	X	X
MKTG	16	X	X
MKTG	17	X	X
MKTG	18	X	X
PROD	22	X	X
PROD	22	X	X
PROD	22	X	X
SALES	44	X	X
SALES	45	X	X
SALES	46	X	X

The shaded areas on the table are records from the Administrative Department and from the Production Department.

To secure the resulting report so that users in the Administrative Department and the Production Department can access only their respective department's data, data selection can be defined as follows.

- At the Parameter screen, define the parameters listed below:

Parameter	Result Type	Type
department	string	single value
account	string	comparison predicate
month	integer	single value

- Go to the Define Selection screen and create Selection Set named ADMIN and one named PROD. (Do one first, and copy it for the other one.)
- Define a default value of `ADMIN` or `PROD` for the Parameter DEPARTMENT (depending, of course, on which department selection set you are creating).
- Type an asterisk (*) in the far right of the Default Value field for DEPARTMENT. This means that the value is frozen and cannot be overridden when the report is requested.
- Type an `R` in the far right of the Default Value field for MONTH. This means that a value for MONTH must be entered when the report is requested.
- Type the appropriate printer information. Remember, you can send the printing to the print device nearest each department. You don't need to specify the same printer. Also, you could need a different number of copies for each department. That, too, could be defined here.

The selection set ADMIN, defined for the Administrative Department, is shown here:

```
HP ALLBASE/BRW                 Define Selection                Report: DEMOREP1

Selection Set   ADMIN                                       freeze screen field: *
                                                                value required: R
     Parameter      Type                        Default Values
DEPARTMENT           S  ADMIN                                                    *
ACCOUNT              S
MONTH                I                                                           R





                        value separator: ,  masks:  07    date format: dd/mm/yy
--------------------------------------------------------------------------------
No. of Copies   1         Print Device     LP
Output Priority 8         Print File Name NIGHTREP
CCTL/NOCCTL               Environment File

Job Input Priority  8           Scheduled AT - 20.00
------------------------------------------------------------------------------
```

where *department* is frozen, *account* is optional, and *month* is required (since the report cannot find data without it).

- For these selection sets, type this calculation for the defined parameters at the Selection Condition screen:

```
HP ALLBASE/BRW              Selection Condition              Report: ACCOUNTS
Selection Set: ADMIN

department = ?department AND

account SATISFIES ?account AND

month = ?month
```

When compiled with the ADMIN selection set, the report is restricted to records for the Administration department. When compiled with the PROD selection set, the report contains only information for the Production Department.

When the report is compiled for the Administrative Department, the entries on the Compile Report screen are:

```
HP ALLBASE/BRW                  Compile Report

Report Specification File ACCOUNTS.GLSPEC                Password
Report Selection Set      ADMIN
Report Execution File     ACCOUNTS.GLADMIN               Password PAYROLL
```

When the report is compiled for the Production Department, the entries on the Compile Report screen are:

```
HP ALLBASE/BRW                  Compile Report

Report Specification File ACCOUNTS.GLSPEC                Password
Report Selection Set      PROD
Report Execution File     ACCOUNTS.GLPROD                Password WIDGETS
```

What Next?

The final tasks for your report are saving the report and compiling the report. Chapter 10 shows the steps necessary for saving and compiling.

When the report is saved and compiled, you can run it ("request" it). Chapter 11 describes the steps needed to request your report.

If you need help with multi-pass reporting, see Chapter 12.

10

Saving and Compiling a Report

It isn't over until you save and compile the report

At a Glance

When you finish specifying the report, you must save the report. If you exit HP ALLBASE/BRW, a prompt reminds you that you have made changes that need to be saved, and if you respond positively to the prompt, HP ALLBASE/BRW does the saving for you. On the other hand, if you are still working in HP ALLBASE/BRW and you decide to compile the report without saving your latest changes, you may end up with a report that is using a previous version of the specifications.

This chapter contains

- Saving a Report
- Compiling a Report

Saving a Report

When you have finished entering all the report specifications, you must save the report.

To save a report:

1. Go to the Define Report screen.
2. Choose `other keys` until you see `Save Report`.
3. Choose `Save Report`.

The specifications for the report are saved in a **specification file**.

If you don't press `Save Report` before you exit HP ALLBASE/BRW or before you define another report, a message prompts you that the specifications for the current report have been modified and asks you if you want to save them.

Compiling a Report

When you are sure your report is saved the way you want it, or when you want to create a test version of the report, you **compile** the report. Compiling the report collects the records you want in your report, performs any calculations you specified, and organizes the report according to the layout you defined.

To compile a report from the Select Report screen

1. At the Select Report screen, type the name of the report you want to compile in the `Report` field.
2. Type a selection set and a report execution filename for the report.
3. Press `Compile Report`.

HP ALLBASE/BRW creates an **execution file** that contains the resulting report.

To compile a report with BRWCOMP

- Type `RUN BRWCOMP.PUB.SYS`
- Respond to these prompts:

```
Specification File:
Selection Set:
Execution File:
```

To include a password for either the specification file or for the execution file, use this syntax:

filename/*password*.*groupname*

What Next?

After a report is saved and compiled, you can request to execute it.

See Chapter 11 for step-by-step procedures on requesting reports.

11

Requesting Reports

Other terms for requesting a report include running the report or executing the report

At a Glance

When you are ready to print the report or write it to a disk, you **request** the report.

When you request the report, you must have the Report Execution file (created when you *compiled* the report).

You can request the report interactively, through HP ALLBASE/BRW screens, or in batch (in the background). You can use the standard job file, and amended standard job file, or a report job file.

This chapter covers

- requesting a report interactively
- requesting a report in batch
- using job files

Requesting Reports Interactively

Requesting a report interactively means that you start HP ALLBASE/BRW and request the report using the Request Report screen. If your installation does not include the developer version, you can still run reports interactively. (See the section "Requesting Reports Using BRWSTART.")

To Request a Report Interactively

1. Define and compile your report as described in the previous chapters of this manual.
2. From the Select Report screen, choose `Request Report`.

   The name of the current report appears in the `Report` field of the Request Report screen.

```
HP ALLBASE/BRW                    Request Report

Report  SET

Parameter          Type                        Values




                       value separator: ,   masks: @?    date format: dd/mm/yy
------------------------------------------------------------------------------
No. of Copies    1        Print Device      LP
Output Priority  8        Print File Name   DEMOREP2
CCTL/NOCCTL               Environment File

Job Input Priority  8          Scheduled    DAY = +2

 Start          Online              4      9   Prev     Next     Help     Exit
 Report         Review                         Values   Values
```

3. Type the parameter type and values for the report in the `Parameter`, `Type`, and `Values` fields. Choose `Prev Values` or `Next Values` if you want to enter more than 9 parameters. See the "Request Report Screen" section of Chapter 16 for details on each field of the Request Report screen.
4. Type the printer details in the lower part of the screen. You can send the report to a printer or to a disk file. (If you want to view the report online, first request the report to a disk file and then you can view it online.)
5. If you want to schedule the job, type schedule instructions in the `Scheduled` field.
6. Choose `Start Report`.
7. At the prompt, type the user, account, and group names for this report (if they don't appear by default).
8. Press (Enter) or choose `Start Report` again.

9. If there is a password for the user, account, or group, type it in the appropriate field and press Enter.

The job number for the stream job appears at the bottom of the screen.

For more information about each field on this screen and how to use this screen, see the Request Report Screen section in Chapter 16.

Note

HP ALLBASE/BRW is supplied with a special job file, called the standard job file, that is used to stream the job that runs the report. The standard job file is held in the file BRWJ000.PUB.SYS. The standard job file can be used to run any HP ALLBASE/BRW reports. You can modify the standard job file to your own requirements. The standard job file is described in "Using Job Files" later in this chapter.

Requesting Reports Using BRWSTART

BRWSTART allows you to request a report in interactive mode, accessing the Request Report screen from outside HP ALLBASE/BRW. You can use this program with either the full developer system, or with the run-time only version since the BRWSTART program is included with the MPE XL operating system.

BRWSTART calls the Select Report screen. You can then call the Request Report and Online Review screens from the Select Report screen. When using BRWSTART, you can see what parameters are required on-screen before having to enter the parameter values.

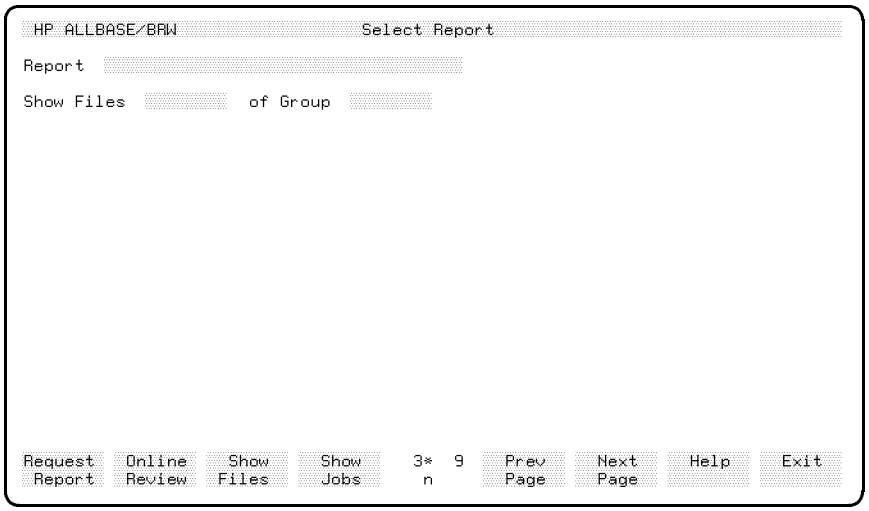
After you start the report, it runs as an independent batch job. The Request Report screen does not wait for the batch job to terminate.

To request a report with BRWSTART

1. At the MPE prompt, type:

```
:RUN BRWSTART.PUB.SYS
```

   The Select Report screen appears.



You can then use the Select Report, Online Review, and Request Report screens the same way described in the earlier section, "Requesting Reports from the Request Report screen."

Note

You cannot use BRWSTART to access those features that concern the specification or compilation of a report. Therefore the Select Report screen displayed using BRWSTART will only display function keys that allow access to the Request Report and Online Review screens, or that allow you to Show Files or Show Jobs. Functions normally available from the Select Report screen (such as Define Table) are not available using BRWSTART.

Requesting Reports with BRWEXEC

With BRWEXEC, you can run a report successfully outside of HP ALLBASE/BRW, even if the HP 3000 you use does not have HP ALLBASE/BRW installed. The BRWEXE program is part of the MPE XL operating system and is all you need to run a compiled HP ALLBASE/BRW report.

Note

The version of the operating system on the HP 3000 on which the report is to be run must be the same as, or later than, the version on the HP 3000 on which the report was developed.

To request a report with BRWEXEC

1. Type this RUN command at the MPE prompt:

   ```
   :RUN BRWEXEC.PUB.SYS
   ```

2. When you are prompted for an execution file name, type the execution filename as shown in the underlined response below:

   ```
   . . .
   Configuration File opened is BRWC000.pub.sys
   Execution File   : filename
   ```

3. When the system prompts for parameters, enter the parameters in this format.

   item operator value

   For example:

   ```
   . . .
   Configuration File opened is BRWC000.pub.sys
   Execution File   : filename
   Parameters       : CUSTOMER-NO = C7
   ```

   You can enter multiple parameters. The parameters can be in any order.

4. Press Return after entering each parameter.

5. To end parameter input, type `//` and press Return. For example:

   ```
   . . .
   Configuration File opened is BRWC000.pub.sys
   Execution File   : filename
   Parameters       : CUSTOMER-NO = C7
   Parameters       : SALES-AREA = WEST
   //
   ```

You can specify different values for the report and override the default values listed by entering new values. BRWEXEC makes sure that the value you enter for each parameter is valid.

BRWEXEC does not prompt for individual parameters, unless the parameter is defined on the Define Selection screen as `R` (required).

You must know which parameters are used in the report, and the type of each parameter (single-value, list-of-values, or comparison predicate). The type determines which operator is valid for that parameter.

The report executes immediately. All other details, for example the output file name, number of copies, and destination, are taken from the execution file's Selection Set. For more information about selection definitions (Parameters and Selection Sets), see Chapter 9.

Using Job Files

There are three types of job files used with HP ALLBASE/BRW to start a background job (in batch mode) to execute the report:

- The standard job file (BRWJ000.PUB.SYS)
- Your own amended version of the standard job file (BRWJOB.*yourgroup.youracct* or BRWJOB.PUB.SYS)
- A report job file (any name of your choice)

When you request a report using HP ALLBASE/BRW or BRWSTART and you type the name of the report's **execution file** in the Report field on the Request Report screen, HP ALLBASE/BRW uses the standard job file or your amended standard job file.

All the parameters used in the report are taken from the execution file and displayed on the Request Report screen, so you can assign values to them. HP ALLBASE/BRW also displays the prompts and default values (if any) for user, group, and account passwords, job input priority, schedule, and so on. When you press `Start Report` on the Request Report screen, these values are assigned to the parameters in the standard job file and used to execute the report. The values for any user-defined parameters used in the report are assigned to the report execution file.

Note

The following values do not appear on the Request Report screen, but HP ALLBASE/BRW prompts you for these values when you choose Start Report.

- $USER
- $GROUP
- $ACCT
- $USERPASS
- $GROUPPASS
- $ACCTPASS

The defaults for $USER, $GROUP, and $ACCT come from your MPE logon. There are no defaults for passwords.

When you request a report using HP ALLBASE/BRW or BRWSTART and you type the name of a **report job file** in the Report field on the Request Report screen, HP ALLBASE/BRW uses that report job file to execute the report. (See the section "Report Job Files" later in this chapter.)

The original standard job file for running reports is held in the file BRWJ000.PUB.SYS: a listing of this file is shown on the next page.

Note that the standard job file calls BRWEXEC.PUB.SYS (line 23 of the job file). The BRWEXEC program actually executes the report.

```
1   DEFINE  JOB  TEMPLATE
2   COMMENT
3   COMMENT  This  is the Standard Report Job for HP ALLBASE/BRW
4   COMMENT reports.
5   COMMENT
6   $USER
7   $USERPASS
8   $GROUP
9   $GROUPPASS
10  $ACCT
11  $ACCTPASS
12  !JOB ^$REPORT,^$USER/^$USERPASS.^$ACCT/^$ACCTPASS,^$GROUP/^$GROUPPASS&
13  !;INPRI=$INPRI
14  !TELL ^$USER.^$ACCT; Report ^$TELLREPORT started
15  !SETJCW CIERROR  = 0
16  !CONTINUE
17  ^$PRINTFILE
18  !IF  CIERROR  <> 0  THEN
19  ! TELL ^$USER.^$ACCT; >>>Invalid Printfile for Report ^$TELLREPORT<<<
20  !  EOJ
21  ! ENDIF
22  !CONTINUE
23  !RUN  BRWEXEC.PUB.SYS
24  ^$REPORT
25  ^$PARMS
26  !EOD
27  !IF JCW >= FATAL THEN
28  !  TELL ^$USER.^$ACCT;  >>>Report  ^$TELLREPORT failed <<<
29  ! EOJ
30  !  ENDIF
31  !IF JCW = WARN  THEN
32  !  TELL  ^$USER.^$ACCT;  >>>No  Data found for Report ^$TELLREPORT <<<
33  !  EOJ
34  !  ENDIF
35  !TELL  ^$USER.^$ACCT; Report ^$TELLREPORT finished
36  !EOJ
```

Amending the Standard Job File

You can substitute an amended job file for the standard job file. That is, you can write a job file to your own specifications (or edit a copy of the standard job file) and *substitute* it for the standard job file. You give it the name BRWJOB. Then, whenever you type an execution file name on the Report field in the Request Report screen, HP ALLBASE/BRW will use your amended job file in place of the standard job file.

Which Standard Job File?

It is possible for you to create and maintain several amended versions of the standard job file on your system. You can have a different standard job file for each group on your HP 3000. When requesting a report, HP ALLBASE/BRW looks for the standard job file in this order:

- BRWJOB.*logon-group.logon-account*;
- BRWJOB.PUB.*logon-account*;
- BRWJOB.PUB.SYS;
- BRWJxxx.PUB.SYS (where xxx is the Native Language Support number returned by the Native Language Support intrinsics);
- BRWJ000.PUB.SYS.

To substitute an amended standard job file

1. Make a copy of BRWJ000.PUB.SYS and put it in the group and directory where it will be used by your HP ALLBASE/BRW report. Call the copy

   `BRWJOB.`*logon-group.logon-account*

2. Edit the copy of the file, substituting the parameters that are correct for your report, and save the file.

After you have created BRWJOB for your report, when you give the name of an execution file in the Report field of the Request Report screen, HP ALLBASE/BRW uses that amended version.

Using Report Job Files

In addition to the standard job file (or an amended standard job file), you can create your own job files called report job files. You use the job files to start and schedule multiple reports (multi-pass reports) and application batch programs.

To avoid starting each report and batch program of a multi-pass report separately, you can set up a report job file with parameters. You can schedule and start the multi-pass report like a single report. (See Chapter 12 for more information about multi-pass reporting.)

Using a report job file also means that you have to enter the parameter values only once for all reports in that job, including the application batch programs.

A report job file has these two parts:

- **A header**, containing parameter definitions and optional default values for parameters, job schedule and print file characteristics.
- **A body**, which is an MPE job, containing parameters anywhere that actual parameter values are to be inserted.

To create a report job file

1. Use an editor program to create a report job file as you would an MPE job file.
2. Define user-defined parameters in the report job header between the BEGINPARMS clause and the ENDPARMS clause.
3. Define standard parameters, such as $PRINTFILE, in the report job header *outside* the BEGINPARMS clause and the ENDPARMS clause.
4. Create the body of the job file, which is an MPE job. Use the parameters in the body anywhere that actual parameter values are to be inserted.

To run a report with a report job file

1. When you are ready to run the report, type the name of the report job file in the `Report` field on the Request Report screen.

Parameters in Report Job Files

There are two types of parameter in report job files:

- **User-defined parameters**. These are parameters that you have defined in the report job file header, in the BEGINPARMS/ENDPARMS clause. Once defined, these parameters can be used anywhere in the report job file body. Note that if you use a report job file, HP ALLBASE/BRW will only display those parameters defined in the BEGINPARMS/ENDPARMS clause in the Parameters field on the Request Report screen. That is, if you want to assign a value to a parameter in a report at run-time, you must define the parameter in the BEGINPARMS/ ENDPARMS clause.

- **HP ALLBASE/BRW standard parameters**. These are parameters supplied with HP ALLBASE/BRW. For example, the user, group, and account from which the job will be streamed, the printfile characteristics, report name, and so on. You define these in the report job file header, but outside the BEGINPARMS/ENDPARMS clause. Only if you define these parameters will you be able to assign a value for them on the Request Report screen.

Parameters (both user-defined and standard parameters) can be used anywhere in the report job file body by specifying a control (^) character followed by the parameter name, as shown here:

```
^sales-area
```

Example: Using parameters in the report job file

In the following example of a report job file, first one report (salesan1.rspec) executes and creates an output file. When the second report (salesan2.rspec) executes, it reads this output file and prints the report. Note the user-defined parameters in the BEGINPARMS/ENDPARMS clause; the HP ALLBASE/BRW standard parameter $PRINTFILE; and that the execution files to be used in the job are explicitly defined within the report job file, on lines 14 and 24.

If this report job file is requested from a Request Report screen, the three user-defined parameters sales-area, sales-%, and product-line will be displayed in the Request Report screen `Parameter` field, with the default value for sales-%. Default values for the Request Report screen fields `Print File Name`, `Print Device`, `Output Priority`, and `No of Copies` are taken from the definition of the standard parameter $PRINTFILE on line 8.

```
1 DEFINE JOB TEMPLATE
2  BEGINPARMS
3           sales-area      (P, SU16)
4           sales-%         (P, I) >=10
4           product-line    (P, SU2)
5  comment  sales-% is percent share of product sales per area
6  ENDPARMS
7  $PRINTFILE salesan;  DEV=LP,12,1
8  comment end of report job file header
9  !JOB salesan, mgr.rdemo,pub
10 !TELL mgr.rdemo; *** Sales  Analysis  Report  started  ***
11 !CONTINUE
12 !RUN  BRWEXEC.PUB.SYS
13 salesan1.rspec
14 sales-area  ^sales-area
15 !EOD
16 !IF JCW >=  FATAL  THEN
17 !   TELL  mgr.rsdemo;  >>> Sales Analysis (Pass 1) failed <<<
18 !   EOJ
19 !ENDIF
20 ^$printfile
21 !CONTINUE
22 !RUN BRWEXEC.PUB.SYS
23 salesan2.rspec
24 percent-sales      ^sales-%
25 product-line       ^product-line
26 !EOD
27 !IF JCW >= FATAL THEN
28 !   TELL mgr.rsdemo;  >>> Sales Analysis (Pass 2) failed <<<
29 !   EOJ
30 !ENDIF
31 !TELL mgr.rsdemo; *** Sales Analysis Report finished ***
32 !EOJ
```

Report Job File Header

The report job file header is defined with declaration commands. Except in the first line in the file, you can enter each command with upper- and lower- case, leading and separating blanks.

A report job file header always begins with:

```
DEFINE JOB TEMPLATE
```

This line is always left-justified in the first line of the file.

The header always ends with the command:

```
!JOB
```

The !JOB command is always the first line of the report job file body.

In the report job file header you define parameters which are used for the job.

Characteristics of parameters

Characteristics of parameters in a report job header are as follows:

- Once defined, the parameters can be used anywhere within a job.
- You can define default values for parameters, for the job schedule, and for the print file.
- If you want to assign a value to a parameter used in a report, you must define that parameter in the report job file header. Only parameters defined in the report job file header are displayed on the Request Report screen.

User-Defined Parameters

You specify user-defined parameters in the BEGINPARMS/ENDPARMS clause of the report job header. BEGINPARMS and ENDPARMS are required at the beginning and end of the parameter definition, respectively.

You define user-defined parameters between the two statements, one parameter per line, using the syntax shown below.

```
BEGINPARMS
parmname(ptype,rtype[,mode])[default value]
ENDPARMS
```

The following table further describes the syntax for user-defined parameters.

Sytax for User-Defined Parameters

Statement Element	Description
parmname	The parameter name (&*parmname*) must follow these rules: ■ It can have up to 19 characters. ■ It must begin with a letter, $, @ or #. ■ It *cannot* contain ^ characters, blanks or parentheses. ■ It must not be the name of a standard parameter.
ptype	One of the following parameter types: P = comparison predicate S = single value L = list of values
rtype	One of these result types for the parameter: S*x* = a string in which *x* must be the string length SU*x* = a string in which all lower-case letters in parameter value will be shifted up. N = fixed point numeric I = integer R = floating point Real D = date T = time
mode	This is optional and can be either `F` for `R`: F = freeze value R = value required
default value	This is optional and has these characteristics: ■ It can be up to 55 characters ■ It can be defined in an extra line using the continuation character ampersand (&).

Examples

Examples of parameters are:

```
sales-area  (P, SU16)
sales-%  (P, I)  >=10
product-line  (P,SU2)
```

Standard Parameters

HP ALLBASE/BRW standard parameters are parameters supplied with HP ALLBASE/BRW that can be used in any report job file. They refer to values used to stream a job, and are associated with fields on the Request Report screen or to values assigned in response to a prompt at the Request Report screen.

Standard parameters are defined in the report file header, but *outside* the BEGINPARMS/ENDPARMS clause. That is, the standard parameters come after the ENDPARMS clause and before the !JOB command. Once defined, a standard parameter can be used in the report job file body.

The following table shows the HP ALLBASE/BRW standard parameters.

Parameter:	Prompt:	Prompt or Field
$USER	user name	Prompts for value
$ACCT	account name	Prompts for value
$GROUP	group name	Prompts for value
$USERPASS	user password	Prompts for value
$ACCTPASS	account password	Prompts for value
$GROUPPASS	group password	Prompts for value
$REPORT	report name and password	Takes value from field on the Request Report screen.
$TELLREPORT	report name and password. The password is not printed by MPE TELL command.	Takes value from field on the Request Report screen.
$PRINTFILE	printfile characteristics	Takes value from field on the Request Report screen.
$INPRI	job input priority	Takes value from field on the Request Report screen.

Note

If you omit any standard parameters, a user requesting the report cannot assign a value to them at run-time. For example, if you do not include the standard parameter $USERPASS in the report job file header, the 'User Password' prompt will not appear on the Request Report screen when you choose `Start Report`.

Note

The $PRINTFILE parameter is made up of values from various fields on the Request Report screen, for example, Output Priority, No of Copies, Print Device etc.. When using HP ALLBASE/BRW Intrinsics, the parameter $PRINTFILE is itself split into many smaller parameters. See Chapter 20, HP ALLBASE/BRW Intrinsics.

Example: Using combinations of parameters

You can use a combination of some or all standard parameters in a report job file header. For example, to allow a report job file to log-on with a variable user name and password but with a fixed account, specify:

```
DEFINE JOB TEMPLATE
BEGINPARMS

    . . .
ENDPARMS
$user
$userpass
!JOB ^$user/^$userpass.MYACCT,PUB
!TELL ^user.MYACCT; *** report job file starts***

    .  .  .
```

In the above example, only the prompts for User and User Password will be displayed on the Request Report screen.

The standard parameters are used in the JOB command, but they could also be used anywhere else in the report job body.

Default Job Schedule

A default job schedule can be defined for a report job file in this format:

```
SCHEDULED schedule
```

Where *schedule* can be like one of these examples:

DATE =	08/31/86
DAY =	MON
DAY =	15 (that is, on the 15th day of the next month)
DAY =	-2 (that is, 1 day *before* month-end)
IN =	2, 12 (that is, after 2 days, 12 hours)
AT =	13:30 (that is, at 1:30 pm)
DATE =	02/08/87; AT = 14:45 (that is, on 8th February 1987, at 2:45 pm)
DAY =	15 ; AT = 10:15 (that is, on the 15th day of the next month, at 10:15 am)

See the description of the STREAM command in the *MPE Commands Reference Manual* for more about job schedules.

Print File Characteristics

Print file characteristics can be changed when you start the report job file if the $PRINTFILE parameter is specified.

You can optionally define defaults for print device, output priority, number of copies, CCTL-codes, and environment file.

When you use the $PRINTFILE parameter in the report job file body, HP ALLBASE/BRW inserts a FILE command with the actual print file characteristics. Place the $PRINTFILE parameter in the first column of the line in the body of the report job file where you want the FILE command to be.

Note

The $PRINTFILE parameter must be defined in the job header, but *outside* of the BEGINPARMS/ENDPARMS clause before you can use the parameter in the file body.

The $PRINTFILE parameter is defined in the header as follows:

```
$PRINTFILE formal file  [;DEV=device,[outpri,copies]]
;CCTL/NOCCTL[;ENV = environment file]
```

formal file	the print file name specified in the report (on the Define Selection screen);
device	the optional, default print device, and if specified, then also:
outpri	an optional, default output priority, used with DEV
copies	an optional, default number of copies, used with DEV
CCTL/NOCCTL	either CCTL, to include carriage-control characters in the printfile (for example, to skip lines), or NOCCTL, not to include such characters. If you select NOCCTL, HP ALLBASE/BRW inserts a blank line in the printfile for every line that must be skipped.
environment file	the optional name for an environment file for the printfile.

When more than one report is run in a job, more than one default printfile may occur. To print each report to $PRINTFILE, insert ^$PRINTFILE *default printfile* in the report body before the BRWEXEC command for each report.

Example

The following is an example of defining the $PRINTFILE parameter in the header of the report job file and using it in the body of the report job file.

```
DEFINE JOB TEMPLATE
BEGINPARMS

 .  .  .

ENDPARMS
$PRINTFILE expenses;  DEV=,,3;CCTL;ENV = ENV1.ENV.SYS
$INPRI 5
!JOB expenses,MGR.MYACCT;inpri=^$INPRI

  .  .  .

^$PRINTFILE
!CONTINUE
!RUN BRWEXEC.PUB.SYS
expense2
 .  .  .
```

The above example sets the Job Input Priority to 5 using the standard $INPRI parameter. Note that the job command line is modified to accept the value of $INPRI.

Comments

Comments can be included anywhere in the report job file header, except in the first line and after a parameter definition which ends with the continuation character, ampersand (&).

For example:

```
COMMENT   This is a comment
```

Report Job File Body

The report job file body is an MPE job containing parameters for which actual parameter values are inserted. Each parameter name in the body must be preceded by a ^ character and must have been defined in the report job file Header.

Logon Command

The report job file body must begin with the logon command for the job, !JOB. The logon command can be defined for a fixed user, account and group, for example:

```
!JOB Expenses,MGR.MYACCT,PUB
```

In addition, you can define a logon command that includes standard parameters, for example:

```
!JOB Expenses,$user/^$userpass.^$ACCT/^$ACCTPASS,PUB
```

When you start the report job file, HP ALLBASE/BRW prompts for user name, account name and the passwords on the Request Report screen.

You can use multiple JOB commands in a report job file body.

Parameters

Standard parameters must be defined *inside* the report job file header, but outside the BEGINPARMS/ ENDPARMS clause. Both user-defined and standard parameters can be used anywhere in the job file body. You must specify a ^ character followed by the parameter name, as shown here.

```
^sales-area
```

HP ALLBASE/BRW replaces the parameter with an actual value when the report job file is started.

You can restrict the length of a user-defined parameter with a result of type String using replacement options, for example:

```
^sales-area[=16]
```

In this example, the length of the parameter value is 16 characters. Longer values are truncated on the right; shorter values are padded with blanks for this case.

```
^sales-area[<=16]
```

In the above example, parameter values longer than 16 characters are truncated on the right.

```
^sales-area[>=4]
```

In above example parameter values shorter than 4 characters are padded with blanks.

A replacement option *only* effects the replacement where it is specified.

To supply parameters for the single reports executed in the report job file, you must specify the parameter name defined in the report followed by a ^ character and the parameter name defined in the report job file header, as shown here:

```
!RUN BRWEXEC.PUB.SYS
salesan2.rf
percent-sales ^sales-%
```

In this example, *percent-sales* is the name in the report and *^sales-%* is the name in the report job file.

This allows you to define names for the parameters in the report job file other than those defined in the report.

For more about running BRWEXEC standalone, see "Requesting Reports with BRWEXEC" earlier in this chapter.

$Printfile

The $Printfile parameter must be specified in the first column of the line where the FILE command for the print file characteristic is needed. For example:

```
. . .

^$PRINTFILE
!CONTINUE
!RUN BRWEXEC.PUB.SYS
salesan2

. . .
```

$Parms This command is used when you need all parameters in a report, and the parameter names in the report and in the report job file are the same. It causes all parameter names and their values to be inserted at this place, for example:

```
DEFINE JOB TEMPLATE
BEGINPARMS
   sales-area     (P, SU16)
   sales-%        (P,  I)  >=  10
   product-line   (P,  SU2)
ENDPARMS
!JOB   salesan, mgr.rsdemo,pub

 .  .  .

!RUN BRWEXEC.PUB.SYS
salesan2.rspec
^$PARMS
!EOD

.  .  .
```

The !EOD command takes the place of the // you enter when you run BRWEXEC interactively.

With actual parameter values, the program looks like this:

```
!RUN BRWEXEC.PUB.SYS
salesan2.rspec
sales-area         SOUTH
sales-%            >= 10
product-line
!EOD
```

Note 

BRWEXEC ignores parameters that are specified in the job but not in the report.

What Next?

Now you should have a copy of your report. If it isn't what you need or what you expected, you can return to any of the tasks chapters and redo steps to perfect the report. Remember to save the changes you make before you request the updated report.

For help with specific screens or with calculations, see the Reference section, Part III.

If you need help with multi-pass reporting, see Chapter 12.

12

Multi-Pass Reporting and Output Files

Writing results into a variety of output files

At a Glance

You can write results from one report into a variety of output files. You can use output files as source tables when creating other reports or as data for use with other application programs (not necessarily HP ALLBASE/BRW). Using one or more output files from one or more reports in another report is called **multi-pass reporting**.

The four types of output files include:

- Intermediate report files (IRF)
- Dictionary output files (DF)
- Quoted BASIC (QB) files
- Data Interchage Format (DIF) files

Each of these output types is discussed in this chapter under output files.

This chapter covers:

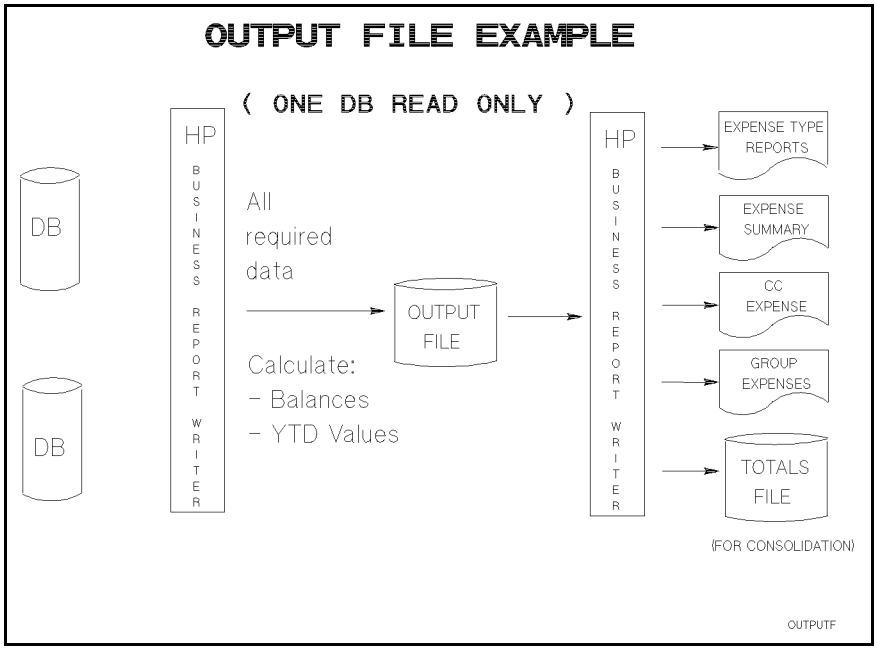
- Multi-pass reporting
- Output files
- Multiple executions of HP ALLBASE/BRW

Multi-pass Reporting

Multi-pass reporting means that you create multiple reports, each one used as a source file for the next report. In some cases, this process improves performance because the database is not re-read for every interim report.

You can use multi-pass reporting:

- To create exception reports; that is, you can execute an HP ALLBASE/BRW report and write records that exceed certain values to an output file, and then use the file holding these exceptions in another HP ALLBASE/BRW report.
- To create database extraction files for later fast access to a subset of data.
- To bundle a series of reports for better performance.
- To integrate reports with application batch programs.



Multi-pass Reporting Flowchart

This figure shows an example of multi-pass reporting using an output file.

Output Files

An output file does not contain the entire report (including headings, details, and footings). Instead, an output file contains only item values, not printer formatting information. One output file can be defined for each lineset:

- Detail lineset
- Break footing lineset
- Break heading lineset
- Page heading lineset
- Page footing lineset
- Report heading lineset
- Report footing lineset

Every time the lineset occurs in the report, a record is written to the output file. For example, the output file of a detail lineset would contain a record for each detail, while the output file of a break footing lineset would contain a record for each occurrence of the break footing.

Note

An output file used by a lineset must be unique; you cannot write data from more than one lineset to the same output file. For instance, you can't write break heading and footing linesets to the same file.

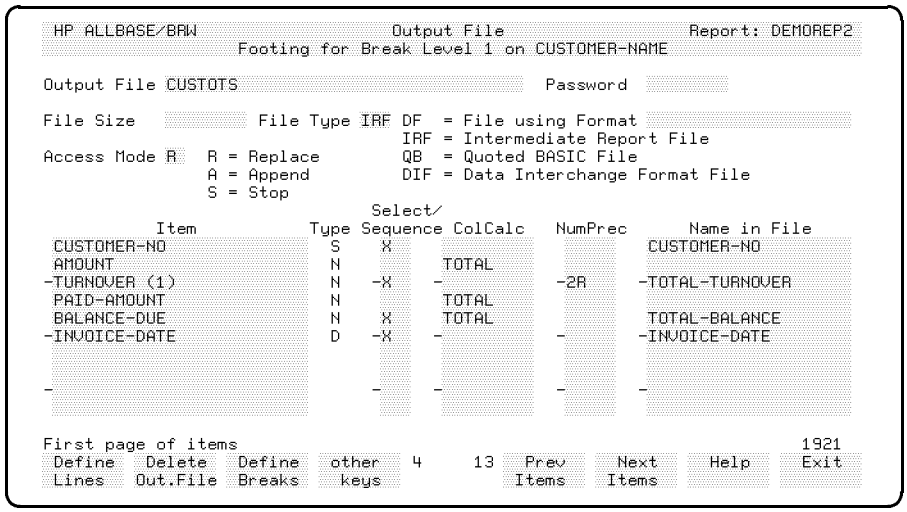
Column calculations and numeric precision can be defined for items written to the output file. See Chapter 15 for details about column calculations and numeric precision.

Types of Output Files

- **Intermediate report files** (IRF). These self-describing files can only be used with HP ALLBASE/BRW reports. They are called self-describing files because they need not be described in a data dictionary or BRWDIC since all data descriptions that HP ALLBASE/BRW needs are held in the file itself.
- **Dictionary output files** (DF). These are ordinary files that must be created in accordance with a file format description from a data dictionary. Data in dictionary output files can be used by *ALLBASE/BRW* or any other application.
- **Quoted BASIC files** (QB) and **data interchange format files** (DIF). These two types are used with PC applications and they are not described in a data dictionary. When you write the results of a report to one of these formats, you cannot use the resulting files as HP ALLBASE/BRW source tables.

Defining an Intermediate Report File

You use the Output File screen to define an intermediate report file and to specify which items are to be written to it.



To define an intermediate report file

1. From the Define Report screen, choose `Output File`.
2. At the prompt, type the lineset information for the lineset you want to use. Press Enter.

   The Output File screen appears. Every item that appears on the Define Lines screen for the specified lineset is displayed in the Item field on the Output File screen.
3. Type the name of the file, password (if any), file size (if known), access mode and file type in the approriate fields.
4. Define the items for the output file. You must define at least one item to be written to the output file. See the next procedure for details on defining the items.
5. Press Enter.

To define items for an output file

1. Type the name of each item that you want written to the output file in the Name in File field or type an X (or a number for QB and DIF files) in the Select/Sequence field.

   If the output file is an intermediate report file, you can choose any name for an item's Name in File. However, if the output file is a dictionary output file, the name in the Name in File field must be a name of an element of the file, that is, the name of an element as specified in the file format in the data dictionary.

2. You can, optionally, specify whether the item is to be subject to a column calculation, and the numeric precision of an item in the output file.

3. Press Enter.

Items positioned on the Define Lines screen *cannot* be deleted or re-arranged here.

If there are more than 9 items, use the Next/Prev Items keys to access the next page and continue your selection. Press Enter for each page.

Note

The items and calculations displayed in the output file screen match what will be displayed on the Define Lines screen. If you add a column calculation to a displayed item, that item reference on the Define Line screen will pick up the column calculation. If you want to write column calculations to an output file and not have them appear on the printed lineset, type the item name a second time on the Output File screen.

To remove the use of an output file

1. Type the name of the output file.
2. Choose `Delete Out.File`
3. Press Enter or choose `Delete Out.File` again to confirm the deletion.

This deletes only the use of the output file for this lineset. It doesn't delete any MPE file.

Using Intermediate Report Files

You can create an intermediate report file if you want to use data from one HP ALLBASE/BRW report with data from other HP ALLBASE/BRW reports. An intermediate report file is a self-describing file since it contains a report's format definition (name, type, length, and position) in the user label. You do not define it in the data dictionary or BRWDIC.

Intermediate report files can only be used as source tables for HP ALLBASE/BRW reports or with HP Information Access.

Note

HP ALLBASE/BRW version A.00.10 or later can use HP Information Access files as source tables. Intermediate report files created with HP ALLBASE/BRW version A.00.10 or later can be used by HP Information Access.

To use an intermediate report file in HP ALLBASE/BRW

1. Assign a unique name to the intermediate report file by typing that name in the `Source Table` field on the Define Table screen.
2. Type a > character and the intermediate report file name in the `Location (DB/DB Env/File)` field.
3. Choose `Add Table`.

```
HP ALLBASE/BRW                  Define Table                  Report: CUSTORD

   Table  CUSTANL               Type  J   J = Join  M = Merge

 Source Table(for SQL preceded by'Owner.')  Location (DB/DB Env/File)  Password
 CUSTOMERS                                  TOYDB.PUB
-PRODSTAB                                  ->PRODS                    -
```

Note

For better performance: It is more efficient to use intermediate report files than dictionary output files if you only want to access the data from other HP ALLBASE/BRW reports. This is because all items in an intermediate report file are held in HP ALLBASE/BRW format and no conversion need take place.

Other Information About Intermediate Report Files

You can specify whether an item is to be subject to a column calculation, and an item's numeric precision in the intermediate report file.

Intermediate report files are built when the report is compiled (*not* when it is executed. Values are only *written* to the files when the report is executed.) If an intermediate report file exists with the same name as that specified on an Output File screen, and you compile the report, the file will be purged as it is rebuilt.

If you want to use an item in an output file that does not appear in the specified lineset, type its name in the Item field. It will then appear on the lineset, but will not be positioned and will not print.

If any other type of file exists with the same name as that specified on an Output File screen, and you compile the report, the compile will fail and the file will not be purged.

When you delete an intermediate report file with the PURGE command, you can only create it again by re-compiling the report. So, to free disc space when data in an intermediate report file is no longer needed, you can run the program REMPTY to clear the file, as shown here:

```
:RUN REMPTY.PUB.SYS

     . . .
Intermediate report file:  ORDERS.PUB

END OF PROGRAM
```

REMPTY deletes all records from an intermediate report file but leaves the file and the data descriptions intact. REMPTY *cannot* be used for other types of file.

Example

The example below shows single items, single components of array items and whole arrays that will be written to an intermediate report file. If you do not specify an index for an array, the whole array is written to the intermediate report file.

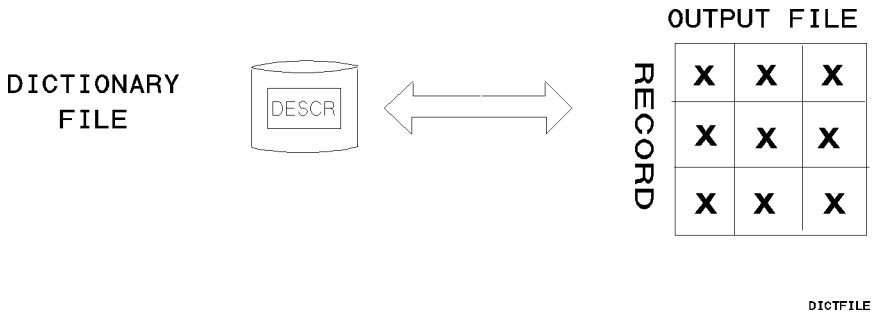
```
 HP ALLBASE/BRW                    Output File                    Report: DEMOREP2
                  Footing for Break Level 1 on CUSTOMER-NAME

Output File CUSTOTS                                  Password

File Size          File Type IRF DF  = File using Format
                                 IRF = Intermediate Report File
Access Mode R   R = Replace      QB  = Quoted BASIC File
                A = Append       DIF = Data Interchange Format File
                S = Stop
                                 Select/
          Item                Type Sequence ColCalc   NumPrec      Name in File
 CUSTOMER-NO                   S    X                              CUSTOMER-NO
 AMOUNT                        N         TOTAL
-TURNOVER (1)                  N   -X    -              -2R      -TOTAL-TURNOVER
 PAID-AMOUNT                   N         TOTAL
 BALANCE-DUE                   N    X    TOTAL                     TOTAL-BALANCE
-INVOICE-DATE                  D   -X    -              -        -INVOICE-DATE


-                                  -     -              -        -

First page of items                                                        1921
 Define   Delete   Define   other   4     13   Prev     Next     Help     Exit
 Lines    Out.File  Breaks   keys              Items    Items
```

Note

Every item in the specified lineset, (that is, the items positioned on the Define Lines screen of the specified lineset), will be displayed in the `Item` field. However, only those items that you specify in the `Select/Sequence` field are written to the intermediate report file. You can choose any name you want for an item in an intermediate report file. Type the new name in the `Name in File` field.

Dictionary Output Files

Dictionary output files can be defined if you want to use the data from a report with other programs. Data in a dictionary output file can be used from either HP ALLBASE/BRW reports or other application programs. The format of the file must be defined in the data dictionary before it is specified in the report. See the following diagram:



The items written to the dictionary output file must match the items defined in the format of that file:

- An item of any type must match an item in the file of the same type. That is, a string in your report can only be written to a string item in the file. However, number types can match another item of any number type. If you assign an item of one number type (for example, Real) to an item in a dictionary output file of a different number type (for example, Numeric), the item will be automatically converted to the type specified in the dictionary output file (in this example, Numeric).
- Arrays must have the same number of components.
- Single array components match single items.
- The entry in the Name in File field must be the same as the element name in the data dictionary file description.

Note

HP ALLBASE/BRW will convert a Real number written to a dictionary output file to either HP Real or IEEE Real format depending on the item's description in the data dictionary. The default is IEEE Real. See Chapter 18 for more information.

String items can have different lengths, in which case values are truncated on the right or padded with blanks. For example, these items are specified on the Output File screen below.

As with intermediate report files, you use the Output File screen to define a dictionary output file and to specify which items are to be written to it.

```
 HP ALLBASE/BRW                    Output File                    Report: CUSTSTAT
                                     Detail

Output File CUSTANL                                  Password

File Size          File Type DF  DF  = File using Format CUSTREC
                                 IRF = Intermediate Report File
Access Mode R   R = Replace      QB  = Quoted BASIC File
                A = Append       DIF = Data Interchange Format File
                S = Stop
                                      Select/
           Item                 Type Sequence ColCalc   NumPrec      Name in File
 CUSTOMER-NO                     S    X                              CUSTOMER-NO
 ADDRESS                         S    X                              ADDRESS-TYPE
-CITY                            S   -X    -            -           -TOWN
 STATE                           S
-                                    -     -            -           -
-                                    -     -            -           -

First page of items                                                        1921
 Define   Delete   Define   other   4     13   Prev     Next     Help     Exit
 Lines    Out.File Breaks    keys                Items    Items
```

Items are only written to the dictionary output file if you enter their names in the Name in File field, or put an X in the Select Sequence field. The name in the Name in File field must be a name of an element of the file as specified in the file description in the data dictionary. Any item in a dictionary output file that is specified in the dictionary but not in the Name in File field will be set to the appropriate Exception Value. See Chapter 17, "Defining Configuration."

You can, optionally, specify whether the item is to be subject to a column calculation, and the numeric precision of an item in the output file.

See Chapter 16, for details about the fields on the Output File screen.

Creation of Dictionary Output Files

Dictionary output files are created when the report is executed, not when it is compiled. You can free the disk space by deleting the file with the MPE command PURGE when data in a dictionary output file is no longer needed.

Quoted BASIC (QB) and Data Interchange Format (DIF) Files

These files are in formats used by PC applications. You can therefore use HP ALLBASE/BRW to process data and create output files which can then be accessed by PC applications.

QB and DIF files cannot be used as HP ALLBASE/BRW source tables. QB and DIF files are not described in a data dictionary.

Creating QB and DIF Files

QB and DIF files are created when the report is executed, not when it is compiled. You can free the disc space by deleting the file with the MPE command PURGE when data in a QB or DIF file is no longer needed.

```
HP ALLBASE/BRW                    Output File                Report: PCREP.RSPEC
                  Footing for Break Level 1 on SALES-AREA

Output File INVREC.PUB.MISACCT                    Password

File Size          File Type QB  DF  = File using Format
                                 IRF = Intermediate Report File
Access Mode R   R = Replace      QB  = Quoted BASIC File
                A = Append       DIF = Data Interchange Format File
                S = Stop
                                    Select/
          Item                Type Sequence ColCalc   NumPrec      Name in File
 CUSTOMER-NO                   S    1
 AMOUNT                        N         TOTAL
-TURNOVER (1)                  N   -3    -            -2R       -
 PAID-AMOUNT                   N         TOTAL
 BALANCE-DUE                   N    2    TOTAL
-INVOICE-DATE                  D   -4    -            -         -


-                                  -     -            -         -

First page of items                                                      1921
 Define   Delete   Define   other   4     13   Prev    Next     Help    Exit
 Lines   Out.File  Breaks    keys              Items   Items
```

Note that with QB and DIF files you can specify a number on the Select/Sequence field to show the order in which the items will appear on each output file record.

Quoted BASIC Format

Quoted BASIC files are used as data transfer files by some standard programs, including Lotus 1-2-3, RBASE 5000, DBASE, and BASIC programs. DSG/3000 can also read Quoted BASIC files as free-formatted data files. In BASIC, quotes are used to delimit strings.

File Layout

The items in the file are separated by commas; strings are enclosed in quotes; and numeric values can have decimals.

Each record in a QB file has the same layout. An example QB file record is shown below.

```
"15 Church Street", 1.02,"10:12:30", 1000, 0.5E+2
```

Between the comma and the value there can be any number of blanks. The file can contain strings (date, time, and ordinary strings) and numeric values (in teger, fixed-point numeric and floating-point real values). No other information is contained in the file.

Data Interchange Format

A DIF (Data Interchange Format) file is a standard file format used by different applications to exchange data. Visicalc, Lotus 1-2-3 and Charting Gallery can read DIF files.

File Layout

A DIF file consists of two parts, a "header" part which describes the layout of data, and a "data" part which contains the actual data values.

The header starts with the keyword TABLE, ends with the keyword DATA and contains the number of vectors per tuple and the number of tuples contained in the file. The data part starts immediately after the keyword DATA and ends with the keyword EOD. The data part is divided into tuples, and each tuple is then divided into vectors. Each tuple contained in the data part starts with the keyword BOT (Beginning Of Tuple).

An example of a DIF file is shown on the next page.

DIF File Layout Example

```
TABLE           Beginning of the Header part
0,1
""
VECTORS
0,6             # vectors = 6
""
TUPLES
0,2             # tuples = 2
""
DATA            end of Header part, start of Data part
0,0
""
-1,0
BOT             Beginning of first tuple
1,0
" January"      vector 1 of type string
1,0
"01/23/1987"    vector 2 of type date
1,0
"12:30:10"      vector 3 of type time
0,235.78        vector 4 of type numeric
V
0,75            vector 5 of type integer
V
0,-.3E+2        vector 6 of type real
V
-1,0            end of first tuple
BOT             Beginning of second tuple
1,0
"  February"
1,0
"02/15/1987"
1,0
"12:10:01"
0,478.25
V
0,874
V
0,.23E-45
V
-1,0            end of second tuple
EOD             end of data
```

Using Output Files

There are several ways that you can use the different types of output files described in this chapter. The following sections describe some suggestions and examples.

Database Extract Files

A database extract file contains a subset of data from a database.

Database extract files allow faster access to:

- a subset of the data in databases
- a summary of the data in databases
- data calculated from data in databases
- a combination of these for reports and other programs

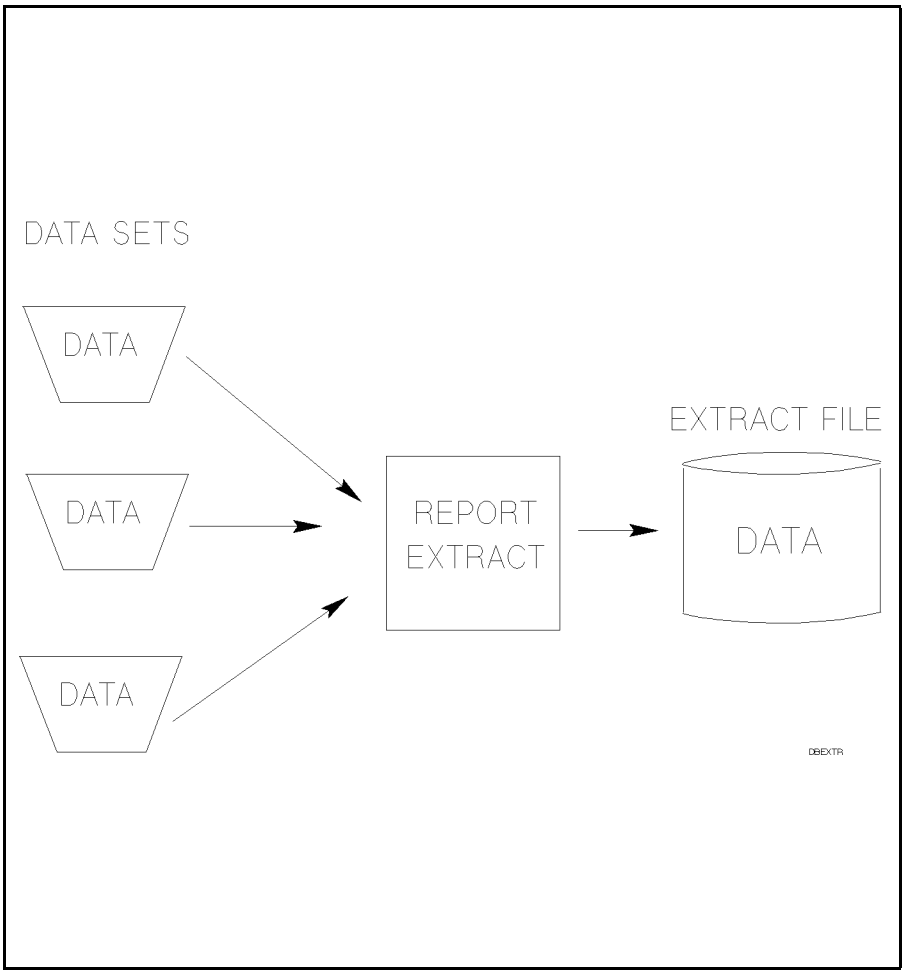
Database extract files can be:

- intermediate report files, if only HP ALLBASE/BRW reports access the file
- dictionary output files if other programs also access the file

Example

Suppose you have a large database of order information, and want to report on orders for certain products only. You can define an HP ALLBASE/BRW report that reads the database and creates an output file containing only data for the certain products required. This output file contains an extract of the data in the database, hence the name database extract file. You can then create reports and use the data from the output file as the source table for the reports. In this way, only one report needs to read the large database and performance is improved, especially if the report that produces the extract file is run overnight.

The following diagram shows how database extract files work:



Bundling Reports

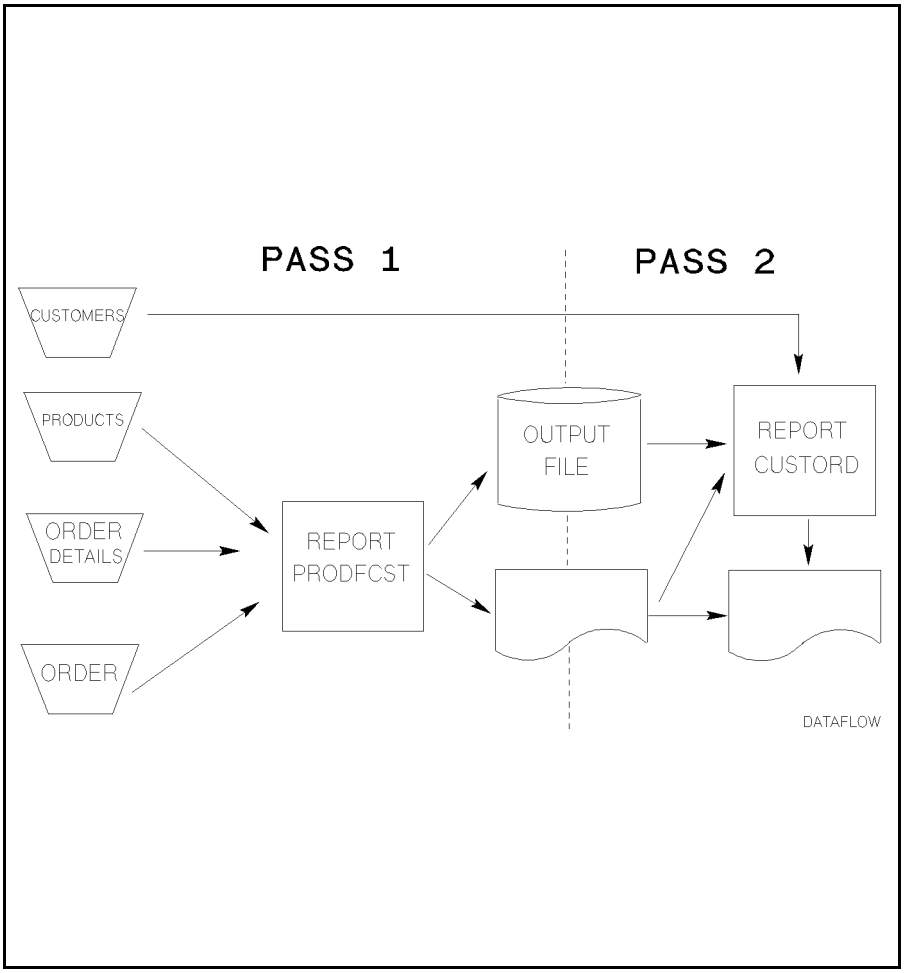
In addition to tuning single reports for best run-time performance, you can improve performance of a whole set of reports by using database extract files.

For example, the Production Forecast report, PRODFCST, and the Customer Statement report, CUSTORD, both access datasets PRODUCTS, ORDERS and ORDER-DETAILS. The report CUSTORD also accesses the dataset CUSTOMERS.

You can improve the overall performance for *both* reports by creating a database extract file for the data from PRODUCTS, ORDERS and ORDER-DETAILS which is used by the PRODFCST report. The report CUSTORD then accesses only this file and the dataset CUSTOMERS.

Performance improvements can be considerable, especially when you report only monthly data and the database contains data for the whole year.

The diagram below shows how data flows from the datasets to the final reports:



The source tables for the PRODFCST report are defined as shown here on the Define Table screen:

```
HP ALLBASE/BRW                    Define Table                    Report: PRODFCST

   Table PRODFORECAST          Type J   J = Join  M = Merge

 Source Table(for SQL preceded by'Owner.')  Location (DB/DB Env/File)  Password
 PRODUCTS                                   TOYDB.PUB
-ORDERS                                    -TOYDB.PUB                 -
 ORDER-DETAILS                              TOYDB.PUB
-                                          -                          -
-                                          -                          -
-                                          -                          -
-                                          -                          -

                Open Join on Source Table

  Add     Define  Relation  other   4     12  Table    Define    Help     Exit
 Table   Relation Condit'n   keys              Calcitem  Lines
```

On the Output File screen an output file is created for PRODFCST, as shown here:

```
HP ALLBASE/BRW                    Output File                    Report: PRODFCST
                                    Detail

Output File PRODS                                  Password

File Size          File Type IRF DF  = File using Format
                                 IRF = Intermediate Report File
Access Mode R  R = Replace       QB  = Quoted BASIC File
               A = Append        DIF = Data Interchange Format File
               S = Stop
                                 Select/
          Item              Type Sequence ColCalc   NumPrec      Name in File
 PRODUCT-NO                  S    X                             PRODUCT-NO
 PRODUCT-NAME                S    X                             PRODUCT-NAME
-PRICE                       N   -X    -           -           -PRICE
 PRODUCT-LINE                S    X                             PRODUCT-LINE
 QUANTITY                    S    X                             QUANTITY
-ORDER-NO                    S   -X    -           -           -ORDER-NO
 CUSTOMER-NO                 S    X                             CUSTOMER-NO
 ORDER-DATE                  D    X                             ORDER-DATE
-                                 -     -           -           -

First page of items                                                      1921
 Define   Delete   Define   other   4     13   Prev     Next     Help    Exit
 Lines   Out.File  Breaks    keys              Items    Items
```

The following section of the Define Table screen shows how the output file for PRODFCST is joined to the source table for CUSTORD:

```
HP ALLBASE/BRW                  Define Table                  Report: CUSTORD

   Table  CUSTANL                 Type  J   J = Join  M = Merge

 Source Table(for SQL preceded by'Owner.')  Location (DB/DB Env/File)  Password
 CUSTOMERS                                  TOYDB.PUB
-PRODSTAB                                  ->PRODS                              -
```

Note

You tell HP ALLBASE/BRW that the source table is an intermediate report file by entering an character before the file name/location in the Location field. Whenever you specify an intermediate report file on the Define Tables screen, enter an arbitrary name in the Source Table field, and an > character, followed by the file name/location, in the Location field. See the above example of >PRODS.

You can create a report job file to produce both reports with one request. See the sample report job file in the section *About Report Job Files* in Chapter 11.

Integrating Reports with Batch Processing

HP ALLBASE/BRW reports can be integrated with other (non-*HP BRW/XL*) batch programs.

For example:

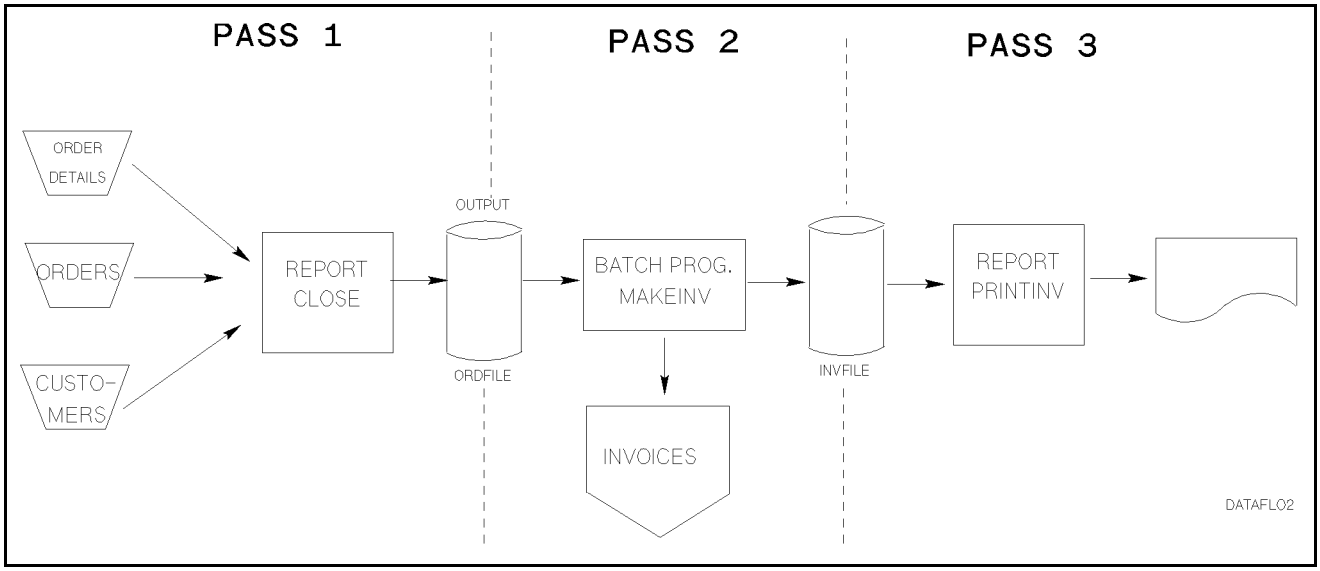
- Batch programs can produce files that are used by HP ALLBASE/BRW reports.
- HP ALLBASE/BRW reports can produce dictionary output files that are used by batch programs.

Example

In the following example, a job generates invoices for all orders that have been shipped and then prints the invoices to be sent to the customers. This job uses the following:

The report CLOSE	Retrieves all orders with shipment dates for all order details, and writes them to the file ORDFILE.PUB.
The batch program MAKEINV	Creates an invoice record for each order and writes the record to the Invoices dataset and to the file INVFILE.PUB.
The report PRINTINV	Prints the invoices to be sent to the customers.

The diagram below shows the flow of data to produce the final report for this example:



You can create a report job file to stream this sequence. See the "Using Report Job Files" section in Chapter 11 for more information about creating report job files.

What Next?

The final tasks for your report are saving the report and compiling the report. Chapter 10 shows the steps necessary for saving and compiling.

When the report is saved and compiled, you can run it ("request" it). Chapter 11 describes the steps needed to request your report.

13

HP BRW-Desk/XL

Sending HP ALLBASE/BRW reports through HP Desk

At a Glance

HP BRW-Desk/XL is a product you purchase separately from HP BRW/XL. You use HP BRW-Desk/XL to send reports through HP DeskManager or through HP DeskManagerPlus.

Note

In this manual, HPDesk is a short term for either HP DeskManager or HP DeskManagerPlus.

HP DeskManager	This is the original HP Desk product. It has simple messaging capability for small to medium electronic mail users. Add-on products such as DeskIntrinsics, DeskMON, and Schedule may have been purchased separately if you already own this original product.
HP DeskManagerPlus	This new version includes intrinsics, DeskMON, and Schedule. You don't have to purchase any extra products. With HP DeskManagerPlus and HP BRW-Desk/XL, you can use HP BRW/XL to send reports through HP Desk. (See the "Requirements" section in this chapter for more details.)

When you run a report, HP ALLBASE/BRW creates message files from the reports or output files and sends them to HPDesk. HPDesk then sends the messages to the specified addressees. With HP BRW-Desk/XL, you can:

- Deliver whole reports and/or output files via HPDesk (and ADVANCEMAIL).

  You can send reports and output files to between 1 and 200 HPDesk addresses. (200 is the HPDesk limit). You can either specify the addresses explicitly on the Define HPDesk Delivery screen, or use an item from a database or file; or calculate the item from within HP ALLBASE/BRW. HP ALLBASE/BRW will take the value of the address from the item on the last detail of the report.

- Send reports or output files to fixed HPDesk destinations; destinations stored in a database or file; or calculated destinations.

- Split reports and/or output files into separate files at specific break levels.

- The recipient can read the *HP ALLBASE/BRW* report in his or her In Tray.

This chapter covers:

- requirements
- identify senders
- outline of steps
- define HPDesk delivery for the report
- request the report
- security

Requirements

To send reports via HP Desk, you need the following:

- HP BRW-Desk/XL on the developer system
- HP DeskManager, version B.03.00 or later, plus the add-on product HPDesk Intrinsics, on the run-time sytem
- **Or,** HP DeskManagerPlus, version xxxx or later, which includes the intrinsics, on the run-time system

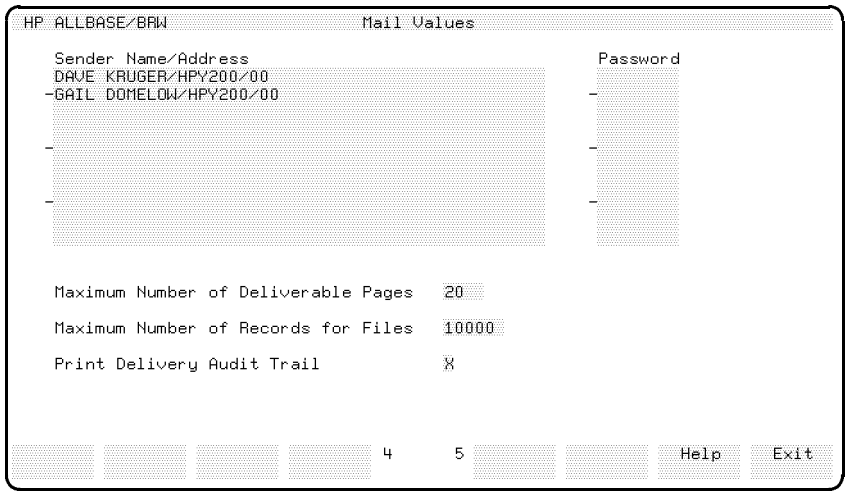
Note

Because HP DeskManagerPlus now **includes** the intrinsics, the intrinsics are no longer available as an add-on product. If you have HP DeskManager without the add-on intrinsics, you need to upgrade to HP DeskManagerPlus.

Identify Senders

Use the program BRWSETUP.PUB.SYS to identify the sender(s) on the Mail Values screen of the appropriate BRWCONF configuration file. That is, set one or more names/addresses under which HP BRW-Desk/XL can log on to HPDesk and send report(s). You only need to identify the sender once for all your reports. Thereafter, HP ALLBASE/BRW looks to the BRWCONF file to find an available sender.

Here is an example of what a completed Mail Values screen looks like:



HP ALLBASE/BRW uses the first sender name and password to log on to HP Desk. If the first sender listed is already logged on to HP Desk, then the next sender name is used. A number of retries are made until either the signon is successful or all the senders in the list have been tried.

If you do not specify a `FROM` name and address when you define the delivery information, the sender used from the BRWCONF file is listed as the sender when the report is delivered by HP Desk.

Using various RCONF files, senders can be selected at group, account, and system level. See Chapter 17, "Defining Configuration", for information on how HP ALLBASE/BRW looks for the RCONF file.

See Chapter 17 for complete details on how to use BRWSETUP and the BRWCONF file.

How It Works

The steps performed by HP ALLBASE/BRW at report execution (when HP BRW-Desk/XL has been selected to deliver a report or output file to users on the HPDesk network) are listed and explained in detail below.

- HP ALLBASE/BRW signs on to HPDesk using the sender name and password defined in the appropriate HP ALLBASE/BRW BRWCONF configuration file. A number of retries will be made if this sender is already signed on to HP Desk. The HP ALLBASE/BRW execution program (BRWEXEC.PUB.SYS) aborts if no sender(s) can sign on to HP Desk.
- The HP ALLBASE/BRW execution program collects the data to be distributed by HP Desk. When a message is completed, HP ALLBASE/BRW closes the files and gives each a unique name. The unique names are in the form R*sssssss*, where:

R	standard prefix
sssssss	milliseconds since midnight.

- HP ALLBASE/BRW writes the message file (which includes the information necessary to distribute the message) and sends the messages to HP Desk.
- HPDesk either accepts the message (if the name address information is correct) or rejects it.
- HP ALLBASE/BRW writes the delivery audit trail messages.
- HP ALLBASE/BRW deletes the message file if it has been accepted by HP Desk.

HPDesk will accept a message only if it is adequately addressed and not too big for the capacity of the HPDesk database.

Define HPDesk Delivery for the Report

To define HPDesk delivery for the report

1. On the Define Report screen, choose `Define Delivery`.

   HP ALLBASE/BRW displays the Define HPDesk Delivery screen for the entire report.

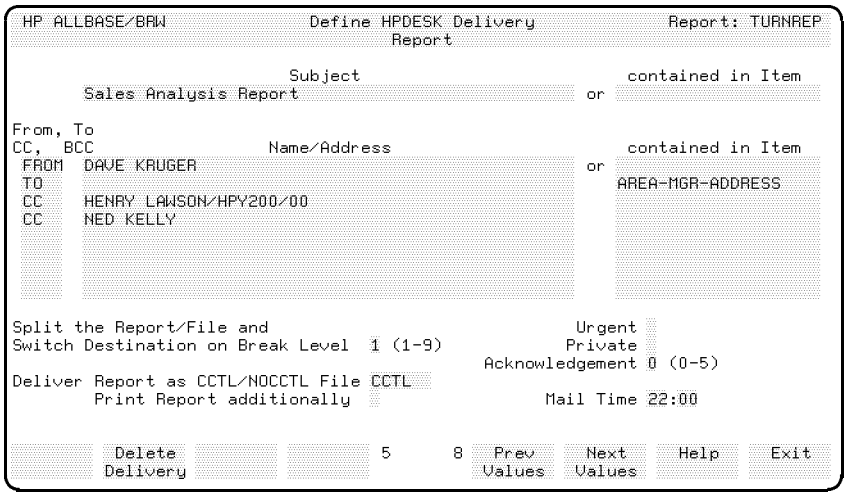
2. Type an optional description of the report in the `Subject` field. This description will be the subject of the HP Desk message. As an alternative, you can type an item name in the `contained in Item` field. The subject for the HPDesk message will be taken from the item.

3. Type `FROM` in the `FROM, TO, CC, BCC` field and type your name and HPDesk address in the `Name/Address` field.

4. For each recipient, type the designator (`TO`, `CC`, or `BCC`) in the `FROM, TO, CC, BCC` field.

5. If you are not getting the name and address information from an item in the report, type the literal name and HPDesk address of each recipient in `Name/Address` field.

6. If you are getting the name and address information from an item in the report, type the name of the item in the `contained in Item` field.



7. If you are finished with the delivery definition, press Enter. If you want to define additional delivery specifications, go on to the other tasks defined in the following procedure steps.

To split the report and switch destinations at a break

1. Define the delivery as described in the previous procedure steps.

2. In the `Split the Report/File ...` field, type the break level (1-9) where you want the split/switch to occur.

3. Go on to the next field, or if you are finished with the delivery definition, press Enter.

If you select to switch the destination of a report on a break level, any higher break levels will not be delivered.

To set the delivery to URGENT and/or PRIVATE

1. Type X in the `Urgent` or `Private` fields.

To require an acknowledgment of the report (optional)

1. In the `Acknowledgment` field, type one of the following numbers for the acknowledgment level values:

```
0 - No Acknowledgment requested
1 - Transmitted Acknowledgment requested
2 - Receipt Acknowledgment requested
3 - Delivered Acknowledgment requested
4 - Read Acknowledgment requested
5 - Reply Acknowledgment requested
```

2. Go on to the next field, or if you are finished with the delivery definition, press Enter.

To send without the carriage control code (CCTL)

1. Enter `NOCCTL` in the `Deliver Report as ...` field.
2. Go on to the next field, or if you are finished with the delivery definition, press Enter.

Sending without the carriage control code is recommended if the recipients of the report use HP AdvanceMail or HP NewWave Mail. Reports sent with the carriage control code will not display properly on the PC.

To print the report in addition to the electronic delivery (optional)

1. Type `X` in the `Print Report additionally` field.
2. Go on to the next field, or if you are finished with the delivery definition, press Enter.

The default is to suppress printing of the report unless you select `Print Report additionally`. Splitting the report or switching destinations does not affect the printed report. If you select to print the report in addition to the HP Desk delivery, the entire report is printed.

To define a mail time (optional)

1. Type the time in 24-hour format in the `Mail Time` fields.

   For example, type the numbers shown in grey:

```
Mail Time 22:00
```

The report will be mailed at 10:00 p.m.

2. If you are finished with the delivery definition, press Enter.

To send an output file

1. Create an output file for the break level you want to send. For example, the report footing. (Use the task instructions in Chapter 12.)
2. At the Output File screen for the break, choose `other keys` until the `Define Delivery` softkey appears.
3. Choose `Define Delivery`. The Define HPDesk Delivery screen that appears is for delivery of the output file only.
4. Define the delivery specifications as described in the procedure steps for a report. Press Enter when you are finished with the definition.

You can send any type of output file, (intermediate report file, dictionary output file, and Quoted BASIC and Data Interchange Format (DIF) files). The Quoted BASIC and DIF files are used in PC applications such as Lotus 1-2-3, Charting Gallery, and others.

Request the Report

You request the report just as you would request any other report, as described in Chapter 11.

Delivery Audit Report

When HPDesk delivery is used, the HP ALLBASE/BRW execution program (BRWEXEC.PUB.SYS) generates delivery audit trail messages in the execution statistics that print to your screen. These messages show:

- the delivery service
- the files generated
- the address where the files were sent
- any messages rejected by HP Desk

The example below shows a sample execution statistics record:

```
HP lBRW/V    <A.02.00>  HP36070 (c) Copyright Hewlett-Packard GmbH 1986, 1987
                                                                        1988
  KSAM    (A.05.01)
  PASCAL  (A.01.11)

Execution File:
Parameters:


report delivered thru HPDESK
- destination switch on level 1
- data file      R1234567
  - subject: Expense Report 8/87
  - sender : Helmut Pohl
  - from   : Helmut Pohl
  - to     : Willi Klenk
  - cc     : Norbert Gronau

output file of level 3 footing delivered thru HPDESK
- data file      R1234568
  - subject: Expense Report 8/87
  - sender : Helmut Pohl
  - from   : Helmut Pohl
  - to     : Willi Klenk
  - cc     : Norbert Gronau
the following addresses have been rejected by HPDESK
  - to     : HP8200/A2 Thomas Heinrich
  - cc     : Acdsdgss

Report Info  :
                                        cpu-sec       elapsed-sec
 time totals for report              ============    =============
                                          3.734            9.413
 (1     segment loads for report)
 (1     segments in report      )
 (1     segments in memory      )

END OF PROGRAM
```

In addition to the execution statistics generated by HP ALLBASE/BRW, usual HPDesk reporting of a successful signon is done to the console and to the HPDesk Statistics file.

Examples

This section gives the details of some example reports.

Example: Switching destinations

Suppose you have a report that lists the turnover for each sales area. The customer records are sorted on SALES AREA as the first-level break, and the total for each area is calculated on the first break footing. You want to send the area turnover totals to the appropriate area managers. So you specify a calculated item, AREA-MGR-ADDRESS, with this formula:

```
IF SALES-AREA = "EASTERN" THEN "Peter SMITH / HPB200/90"
   ELSE IF SALES-AREA = "NORTHERN" THEN "Jane GREY / HPC340/00"
      ELSE IF ...
```

This will return the address of the sales area manager. On the Define HPDesk Selection screen, you type the item name `AREA-MGR-ADDRESS` in the `contained in Item` field instead of typing a literal address for the recipient of the report. Select to switch destination on break level 1 (for the sales area). Then, HP BRW-Desk/XL will do the following:

- Split the report at each level 1 break.
- Send the details and break linesets within that break, as a separate report, to the address taken from the value of AREA-MGR-ADDRESS on the break footing (that is, the last detail in the break).

In this way your report will be split and automatically sent to the appropriate area managers!

When you switch destinations, HP ALLBASE/BRW always takes the value of the address from the item on the break footing (that is, the last detail in the break).

The reports shown next represent the results of sending a report with switched destinations.

Note that for each report, the page number is reset to 1.

```
Page:    1          Customer Details Report:  1        07/11/89
-----    -          --------------------------        --------

Customer Details for Sales Area:  EASTERN
-----------------------------------------

Customer Number     Customer Name                      Month's Sales
---------------     -------------                      -------------


100213              BLACK & WHITE                            2451.91
100432              SUR                                      9100.25
100498              C. WINSTON F & S                         3654.15

Total Sales for Sales Area:   EASTERN            is:       15206.31
---------------------------                      ---
```

```
Page:    1          Customer Details Report:  1        07/11/89
-----    -          --------------------------        --------

Customer Details for Sales Area:  NORTHERN
------------------------------------------

Customer Number     Customer Name                      Month's Sales
---------------     -------------                      -------------


100311              TOOLS EQUIPMENT CORP.                     961.87
100580              STANDARD BUILDING MAINT.                  545.43

Total Sales for Sales Area:   NORTHERN           is:        1507.30
---------------------------                      ---
```

```
Page:    1          Customer Details Report:  1        07/11/89
-----    -          --------------------------        --------

Customer Details for Sales Area:  SOUTHERN
------------------------------------------

Customer Number     Customer Name                      Month's Sales
---------------     -------------                      -------------


100365              LEMWAY INC.                              236.54
100888              REGULAR OIL INDUSTRIES                  8925.00
100946              BROWN BROTHERS                         26455.10

Total Sales for Sales Area:   SOUTHERN             is:      35616.64
---------------------------                        ---
```

```
Page:    1          Customer Details Report:  1        07/11/89
-----    -          --------------------------        --------

Customer Details for Sales Area:  WESTERN
-----------------------------------------

Customer Number     Customer Name                      Month's Sales
---------------     -------------                      -------------


100114              ACME INDUSTRIES                         9580.35
100321              MILLER'S CARS                           5147.89
100327              DAILY POST                             18400.12
100537              MACHINERY CORP.                        31420.75
100568              HIGGIN'S COMPANY                           0.00
100827              TU CONSOLIDATED                        28600.00
100953              BEVERLY'S SHOP                         25423.37

Total Sales for Sales Area:   WESTERN              is:      15206.31
---------------------------                        ---
```

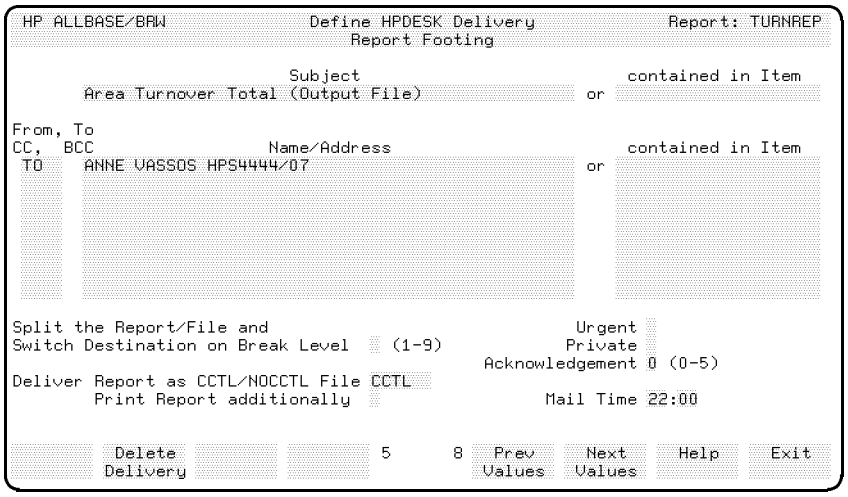
Example: Sending an output file

Suppose you want to send the MIS Department a file through HPDesk that they can download and use in year-end reporting.

To make sure that the MIS department receives the total turnover for all sales areas, you can send them an output file containing the report footing (a summary of the total turnover for the year).

- Create an output file for the report footing.
- From the Output File screen, choose `other keys` and then `Define Delivery`.

- Specify the MIS department on the Define HPDesk Delivery screen for the output file.
- Type Area Turnover Total (Output File) in the `Subject` field.
- Type `TO` in the `FROM`, `TO`, `CC`, or `BCC` field.
- Type the address of the MIS department in the Name/Address field.
- Press Enter.



Note that this will send the output file, not a report, to the MIS department. The output file cannot be read in the In Tray, however, HPDesk will automatically deliver the output file.

The output file will have one record, the total turnover:

```
170902.73
```

If you wanted the formatted report footing (with headings and so on), you could print the report in addition and send the paper copy of the report to the MIS department.

Security

To prevent someone from sending messages without authority, it is recommended that you require users to specify a password before signing on to HP Desk. Authorized sender names and passwords can be entered in the BRWCONF file as described earlier in this chapter. Passwords are encrypted in the BRWCONF file to ensure that security is not compromised.

Note

When you store the passwords in the BRWCONF file, you can maintain or change them regularly without having to alter or recompile the HP ALLBASE/BRW reports.

Security When You Switch Destinations

If you switch the destination on a break level, only the headings and footings for that break and for any lower-level breaks are delivered. This is to secure the data on these linesets. For example, if you switched the destination on break level 3, only break levels 3 to 9 would be delivered. Information in breaks 1 or 2 and the report breaks would not be delivered. In the example report the report footing would not be sent as it contains the total turnover for all areas.

Security When You Print Reports

If you deliver a report via HP Desk, the report will be delivered and not printed (for example, on the lineprinter) unless you select to print the report in addition on the Define HPDesk Delivery screen.

If you select to switch the destination of a report on a break level, any higher break levels will not be delivered. However, this does not affect your *printed report*. That is, if you select to print the report for yourself, in addition to sending the report, the entire report is printed for you.

Error Handling

Address Errors

HPDesk will accept all messages, even if they contain bad or incomplete names/addresses (mailnodes). HP BRW-DESK/XL's action are shown in the following table.

Name present?	Address(MailNode) present?	Effect:
No	No	HPDesk rejects address
Yes (unrecognized or malformed)	No	HPDesk rejects address
Yes (unrecognized or malformed)	Yes (unrecognized or malformed)	HPDesk rejects address
Yes (malformed)	Yes (recognized)	HPDesk rejects address
Yes (unrecognized or recognized but a duplicate)	No	HPDesk rejects address
Yes (unrecognized and unique HPDesk user)	No	Sent to specified name at configured address
Yes (matches a configured foreign alias for a specified gateway)	Yes or No	Sent to HPDesk name and address of the configured foreign alias.
Yes (unrecognized)	Yes (remote node)	Specified name at specified address
Yes (unrecognized)	Yes (local node)	Specified name at specified address. Message will be sent to General Delivery
Yes (recognized)	Yes (recognized)	Specified name at specified address

HPDesk will only send a message to correct addresses. If some of the addresses in a message are incorrect, HPDesk will only mail the message to the correct addresses. HP BRW-Desk/XL will print the addresses to which the message was sent, and the incorrect addresses (to which the message was not sent) on the delivery audit trail.

14

The HP ALLBASE/SQL Interface

Retrieving and combining HP ALLBASE/SQL data

At a Glance

HP ALLBASE/BRW let you access all non-relational data (IMAGE datasets, KSAM files and MPE files) as if the data were stored in a relational database. HP ALLBASE/SQL uses relational data that also can be accessed by HP ALLBASE/BRW. It is possible to retrieve and combine data from IMAGE, HP ALLBASE/SQL, KSAM and MPE within the same report. It is also possible to retrieve and combine data from multiple SQL DBEnvironments.

HP ALLBASE/BRW has features that are not normally available through HP ALLBASE/SQL. When you access HP ALLBASE/SQL data through HP ALLBASE/BRW, however, these features can be used.

The following HP ALLBASE/BRW features increase the functionality of HP ALLBASE/SQL:

- You can access multiple DBEnvironments to retrieve and combine data in the same report.
- Joins, open joins and merges can use HP ALLBASE/SQL tables and combinations of HP ALLBASE/SQL tables with other data sources.
- You can pass user-defined parameters to make data selections.

Note

A dictionary file is not required for an HP ALLBASE/SQL DBEnvironment.

This chapter covers:

- creating, compiling and executing HP ALLBASE/SQL reports
- characteristics
- HP ALLBASE/SQL and HP ALLBASE/BRW
- joining
- native language support

Using HP ALLBASE/SQL

To create, compile, and execute an HP ALLBASE/SQL report

1. Specify the report in the same way that you specify any other report, except that you give the Source Table name using the SQL pathname (including the owner name).
2. On the Define Report screen you can enter an SQL Transaction Isolation Level. This entry is optional. See the "Transaction Handling and Locking" section later in this chapter.
3. Define the relations and layout just as you would any other report.
4. Use the Tune Access screen to control how the join will be done for multiple tables. You cannot use Keys for HP ALLBASE/SQL tables, but you can use Access Sequence to adjust performance and you can use Lock Mode to quarantee that no one changes the table while HP ALLBASE/BRW is executing the query.
5. Use selection sets and selection conditions to define how you want to customize the report at run time.
6. Compile the report as you would any other report.
7. Execute the report as you would any other report.

Example

Suppose you are using the SQL database EXDBE (diagrammed in Chapter 1) and want to create a report that lists user-specified VENDORNAMES and their corresponding PARTNUMBERS. In other words, the report will show all part numbers of items sold by a particular vendor.

Specification

You define the table or tables for an HP ALLBASE/SQL report in much the same way as you define any other report. The information on this Define Table screen tells HP ALLBASE/BRW which HP ALLBASE/SQL tables or views to use and where they are located.

```
HP ALLBASE/BRW                  Define Table                  Report: VENDORS

   Table  ALL-VENDORS          Type  J   J = Join  M = Merge

 Source Table(for SQL preceded by'Owner.')  Location (DB/DB Env/File)  Password
 SUPPLYDB.VENDORS                          EXDBE.PUB
-SUPPLYDB.SUPPLYPRICE                     -EXDBE.PUB                  -

               Open Join on Source Table
```

In this case, the source tables are SUPPLYDB.VENDORS and SUPPLYDB.SUPPLYPRICE. They are both located in EXDBE.PUB for this example.

Notice that the HP ALLBASE/SQL table name is preceded by the owner name (SUPPLYDB). Note also that no password has to be specified; refer to the "Access Authorization" section later in this chapter.

You define the relation between the tables the same way that you define other relations. The information on this Define Relation screen tells HP ALLBASE/BRW the item name (VENDORNUMBER) to use to join the tables shown on the Define Table screen.

```
HP ALLBASE/BRW                Define Relation               Report: VENDORS
Table: ALL-VENDORS            Relation 1 of 1



Relate via          Common Item           in Source Tables
                   VENDORNUMBER           VENDORS
                  -VENDORNUMBER          -SUPPLYPRICE
```

The information on this Tune Access screen controls how the joining of the two tables will be done—either by HP ALLBASE/SQL or HP ALLBASE/BRW.

```
HP ALLBASE/BRW                  Tune Access                 Report: VENDORS
Table: ALL-VENDORS

                                                               IMAGE
                                                              -------
                      Table Access                         Lock Open Item Level
    Source Table      Type Sequence        Key             Mode Mode Security
 VENDORS              SQL     1
-SUPPLYPRICE         -SQL    -1  -                         -    -    -
```

If you use equal access sequence numbers, as shown in this example, the join is done by HP ALLBASE/SQL. You can also use different access sequence numbers to have the join done by HP ALLBASE/BRW. See the "Performance Considerations" section in this chapter for additional information about which method may be best for your environment.

Keyed access cannot be specified for SQL tables, so the `Key` field on this screen is blank.

If you want to be able to decide on a specific vendor at run time, use a parameter for the VENDORNAME. You define a parameter as shown in the example Define Parameter screen shown next.

```
HP ALLBASE/BRW                    Parameter                    Report: VENDORS

  Parameter    VENDORPARM             Value Length  26 (if String)

  Result Type  S   S = String     N = Numeric fixed
                   D = Date       I = Integer
                   T = Time       R = Real

         Type  S   P = Comparison Predicate
                   S = Single Value
                   L = List of Values

  Value Required

  Upshift Value



Parameter added                                                         1540
  Add    Delete                      4* 16   Prev    Next    Help    Exit
  Parm   Parm                                Parm    Parm
```

In this example, ?VENDORPARM is defined as a single value parameter. In addition to defining the parameter, you need to define a selection condition to equate the VENDORNAME to the ?VENDORPARM parameter.

```
HP ALLBASE/BRW              Selection Condition              Report: VENDORS
Selection Set: SETX

VENDORNAME = ?VENDORPARM
```

For more information on using BRW parameters, refer to Chapter 6.

When you run HP ALLBASE/BRW you can enter the name of the specific vendor you want.

Layout

The items to be printed (VEDORNAME and PARTNUMBER) are specified on the layout screens for this example.

Compilation

When this example is compiled, BRWCOMP produces a BRW execution file and generates the following HP ALLBASE/SQL query:

```
SELECT T1.VENDORNAME
   T2.PARTNUMBER
FROM  SUPPLYDB.VENDORS T1
   SUPPLYDB.SUPPLYPRICE T2
```

```
WHERE (T1.VENDORNUMBER=T2.VENDORNUMBER)
AND   (T1.VENDORNAME=?VENDORPARM)
```

Notice that the query above consists of one SELECT statement. This is because the same access sequence number was specified for both tables on the Tune Access screen. This is the default if the HP ALLBASE/SQL tables are from the same DBEnvironment and are listed con secutively on the Define Table screen.

If the access sequence numbers were changed to 1 and 2 respectively, the following two SELECT statements would be generated:

```
SELECT T1.VENDORNAME
   T1.VENDORNUMBER
FROM  SUPPLYDB.VENDORS T1
WHERE VENDORNAME=?VENDORPARM

SELECT T1.VENDORNUMBER
   T1.PARTNUMBER
FROM  SUPPLYDB.SUPPLYPRICE T1
```

The two query results are then combined by BRW.

It is important to point out that the above query is not yet the final query. It becomes the final query at the time of execution, when the value of ?VENDORPARM is supplied.

Execution

The final step in this example is to execute the report. When the report is executed it prompts for parameters to be typed in. The VENDORNAME supplied is then substituted for ?VENDORPARM, thus creating the final query and generating the report.

The final query is passed to HP ALLBASE/SQL. Each query is executed by HP ALLBASE/SQL as a single HP ALLBASE/SQL transaction. Sorting is always done by BRW.

HP ALLBASE/SQL Characteristics

System Catalog

A data dictionary does not have to be set up in HP ALLBASE/BRW for HP ALLBASE/SQL tables. Therefore, there is no HP ALLBASE/BRW dictionary (BRWDIC file) to maintain.

HP ALLBASE/BRW takes advantage of HP ALLBASE/SQL's System Catalog by retrieving information on data definitions directly from the System Catalog. This provides an active link to data definitions. If changes are made in data structures, the changes are reflected in the System Catalog.

The information stored in the System Catalog of a DBEnvironment includes the following:

- the existence of HP ALLBASE/SQL tables and views
- the column names belonging to a table
- the datatype of a column with length, precision and scale
- whether a NULL value is allowed for a column

To define a report (programs BRWXL and BRWCOMP), you need the HP ALLBASE/SQL production database or an HP ALLBASE/SQL test database with the same table structures, owner names and access rights as the production database.

Table Owner

The Define Table screen identifies the source tables used in the report. For an HP ALLBASE/SQL table or view, the ownername.tablename is specified in the Source Table field, and the DBEnvironment is specified in the Location field.

The Define Table screen is the only screen in HP ALLBASE/BRW that requires both the ownername and the tablename to be specified. Within HP ALLBASE/BRW an HP ALLBASE/SQL table or view is identified by its table name only.

Access Authorization

HP ALLBASE/SQL does not have passwords for DBEnvironments, tables, views or columns. Access to an SQL database is controlled via authorities granted to the users of a DBE. This information is kept in the SQL System Catalog and checked against the MPE logon user running an SQL application. A user is identified by HP ALLBASE/SQL as logon user@logon account. Users defining and users running a report on an HP ALLBASE/SQL DBEnvironment must have CONNECT authority to this DBEnvironment, and SELECT authority on all tables and views they want to read.

The HP ALLBASE/BRW programs always open an HP ALLBASE/SQL DBEnvironment with the CONNECT command. This requires the DBEnvironment parameter `Autostart Mode` to be set to `on`. The `on` setting allows a DBEnvironment to be opened with

the CONNECT command instead of the STARTDBE command. The DBEnvironment parameter "User Mode" should be set to `multi` to allow multiple access to the DBEnvironment.

Transaction Handling and Locking

There are two ways of specifying locking within SQL: table lock mode and transaction isolation level. Locking controls concurrency of different applications running in the same DBEnvironment.

In HP ALLBASE/BRW, table lock mode is specified on the Tune Access screen. Transaction isolation level is specified in the Configuration file and may be overwritten for a single report on the Define Report screen.

The default for table lock mode is none, and the transaction isolation level default is RC (Read Committed). Refer to the HP ALLBASE/SQL manual for a description of locking and concurrency.

Listed below is general information pertaining to Table Lock Mode and Transaction Isolation Levels.

Table Lock Mode

SHARE

- Locks the whole table for the duration of the query.
- Allows concurrent reads by other applications, but prohibits concurrency for data manipulation.
- Guarantees that nobody changes the table while BRW is executing the query.
- Deadlock possibility only in combination with locks on other tables.

Transaction Isolation Levels

RR

- Locks the data accessed by BRW for the duration of the query.
- Allows concurrent reads, but restricts concur rency for data manipulation. Parts of the table not accessed by BRW might be manipulated by other applications.
- Guarantees that the data accessed by BRW remains unchanged while BRW is executing the query.
- Likelihood of deadlocks.

CS,RC

- Locks the data currently being accessed by BRW while reading. Locks are not kept until the end of the query.
- Allows most concurrency together with committed data.

- Guarantees consistency for single data but not for the query as a whole.
- Possibility of deadlocks.

RU

- No locks while reading the data.
- Allows most concurrency, but there is a likelihood for reading uncommitted data. Is recommended if there is no need for an exact report.
- No deadlocks with concurrent data manipulations.

The locking strategy should be chosen in accordance with the DBA depending on the DBEnvironment and the applications running on it.

HP ALLBASE/SQL and HP ALLBASE/BRW

Modifying the DBEnvironment

The BRW programs BRWCOMP and BRWEXEC work on the current data definitions in the SQL System Catalog. This is also true when a new report is specified with BRWXL. If the specifications of an existing report are modified by the user, the specifications already done are not automatically checked against the current System Catalog entries. So, if the DBE has been modified, answer 'yes' to the prompt: "Do you want to check the SQL DBEnvironment for changes?". The results of the check done during this upgrade are listed on the Select Report screen. The messages can be used to establish whether changes must be made manually to the report specifications.

Example

Suppose BRWCOMP aborts with an error because an item type changed from numeric to string. In order to correct the situation, you have to enter BRWXL and answer 'yes' to the prompt: "Do you want to check the SQL DBEnvironment for chanages?". In this instance, you must manually change the calculation formula (or formulae) in which the item occurs, and recompile the report.

Performance Considerations

If HP ALLBASE/SQL tables have different access sequence numbers, a query is sent to HP ALLBASE/SQL for each table. HP ALLBASE/SQL tables with the same access sequence number are combined to one HP ALLBASE/SQL query.

In general, it is more efficient to combine all HP ALLBASE/SQL tables from a single DBEnvironment into a single query by giving them all the same access sequence number in the Tune Access screen. This allows HP ALLBASE/SQL to apply internal optimization and internal SQL indexes to join the tables in the most efficient way.

If separate queries are given for each HP ALLBASE/SQL table, HP ALLBASE/BRW will have to save the results of these queries into temporary files and sort these temporary files on the common relation items before combining the tables.

Names and Datatypes

The following table compares the HP ALLBASE/SQL names and datatypes with the HP ALLBASE/BRW names and data types. It also identifies any limits HP ALLBASE/BRW places on HP ALLBASE/SQL.

HP ALLBASE/SQL	HP ALLBASE/BRW
Names	Names in double quotation marks are not supported Names containing native language characters are not supported.
NULL value	A NULL value is sorted as the highest value. Two NULL items are compared as unequal.
NO_VALUE	NO_VALUE is sorted as the highest value. Two NO_VALUE items are compared as equal.
CHAR (1 to 3996) String	Accepts any length, but truncates to 255 characters.
VARCHAR (1 to 3996) String	Accepts any length, but truncates to 255 characters.
DECIMAL	Precision 1 to 15 (number of digits) Scale from 0 to 15 (number of digits to the right of the decimal)
NUMERIC	Allows up to 13 digits to the left of the decimal point, up to 6 digits to the right of the decimal point. If values to the left of the decimal are greater than 13 digits the exception condition OVERFLOW is set. Scales greater than 6 are truncated. Note that this could cause a loss of accurracy.
FLOAT	Real
INTEGER	Integer
SMALLINT	Integer
DATE	DATE
TIME	TIME
DATETIME	(Not supported)
INTERVAL	(Not supported in first release)
NULL	NO_VALUE

Joining

HP ALLBASE/BRW sets NULL values retrieved from HP ALLBASE/SQL to BRW's exception condition, NO_VALUE. Both HP ALLBASE/SQL and HP ALLBASE/BRW sort NULLs/NO_VALUEs highest.

HP ALLBASE/SQL does not retrieve rows joined on two NULL item values. HP ALLBASE/BRW joins NO_VALUE = NO_VALUE, HP SQL evaluates NULL = NULL as false. If you want HP SQL's method, define the access such that HP ALLBASE/SQL joins the data; if you want HP ALLBASE/BRW's method, define the access such that HP ALLBASE/BRW performs the join.

SATISFIES Operator

There is a specific operator SATISFIES in BRW to be used with comparison predicate parameters (see Chapter 6). For example the condition:

```
VENDORNAME SATISFIES ?VENDORNAMEPARM
```

allows one or more values plus a comparison operator to be specified for the parameter value.

Assuming the parameter value "< 5" is passed to BRW at execution time, this condition retrieves those rows with a value smaller than 5 for VENDORNAME. The condition does not return those rows with a NULL value for the VENDORNAME. To have the NULL values for SQL items too, you have to expand your condition to

```
VENDORNAME SATISFIES ?VENDORNAMEPARM OR
VENDORNAME = NO_VALUE
```

Not supplying a paramater value causes all rows to be printed, including the NULLS.

Combining HP ALLBASE/SQL data with IMAGE/3000 or KSAM

When combining HP ALLBASE/SQL data with HP IMAGE/3000 or KSAM data, it is usually best to use the access sequence numbers in the Tune Access screen to load the HP ALLBASE/SQL tables first.

The Tune Access screen does not accept key items for HP ALLBASE/SQL tables, so putting them first allows keyed joins by giving key items for the following HP IMAGE/3000 or KSAM tables.

Native Language Support

HP ALLBASE/SQL supports native languages. This includes sorting and comparison of data stored as 8-bit or 16-bit native language characters in a DBEnvironment. HP ALLBASE/SQL allows the data language of an SQL database - specified with the START DBE NEW command - to be overwritten for specific tables and columns. This means HP ALLBASE/SQL permits a combination of one local language and the default language NATIVE-3000 for storing data within one DBEnvironment.

HP ALLBASE/BRW supports native languages for data stored as 8-bit characters. This includes sorting but not data comparison. The data language to be used for sorting by HP ALLBASE/BRW is defined by the user via the JCW (job control word) NLDATALANG or via the BRWSETUP program in the Data Language field of the Global Values screen.

If the language specified with a DBEnvironment in HP ALLBASE/SQL differs from the one specified in BRW, HP ALLBASE/BRW's execution program generates a warning, but executes the report. The warning message states: "Language of DBE <dbe name> differs from BRW's data language."

HP ALLBASE/SQL also provides support of native language for SQL object names, for example table or column names. This is not supported by HP ALLBASE/BRW, therefore HP SQL tables and columns that are to be accessed by HP ALLBASE/BRW cannot contain native language characters in their names.

15

Calculations

Define calculations for numbers, strings, dates and time

At a Glance

If you need to review some calculation terms, the following table may be of help.

Some Calculation Terms

Arguments	Arguments are the values used in a calculation. Arguments include: ■ items, standard items, and array items ■ functions and standard functions ■ function arguments ■ constants ■ parameters ■ column calculation results
Calculated Items	These are items created from a formula that you define. Once created, calculated items can be used in a report just like an item projected from a Final Access Table or an HP ALLBASE/BRW standard item. Calculated items are split into two categories: ■ Table calculated items. These are calculated items that you define from the Define Table Screen. They are calculated when the table is built. ■ Layout calculated items. These are calculated items that you define from the Define Lines screen. They are calculated whenever a lineset on which they appear is printed.
Calculations	A calculation uses arguments and operators to produce a result. Operators can perform mathematical or logical operations on arguments. Arguments, operators, and results can be of several types. Calculations can be either formulas or conditions.
Column Calculations	HP ALLBASE/BRW allows column calculations. These let you specify, for all the occurrences of an item: the total, the average, the number of occurrences, or the largest/ smallest value. Column calculations do not use formulas, and therefore do not use the Calculation Language. You select a column calculation using the ColCalc field on the Define Lines screen. Column calculations are explained in Chapter 8.

Some Calculation Terms (continued)

Conditions	A condition is a calculation that produces a Boolean result, that is, a result that can be either TRUE or FALSE. For example:
	
	`(CUSTOMER-NO = 000004)`
	
	or
	
	`ORDER-NO < "012"`
	
	There are four types of condition in HP ALLBASE/BRW:
	
	■ Relation conditions. You define relation conditions from the Define Table screen. They determine whether or not to write a record to a table.
	
	■ Suppress line conditions. These are used to determine whether or not to print a line.
	
	■ Suppress lineset conditions. These are used to determine whether or not to print a lineset.
	
	■ Selection conditions. These are used in Selection Sets to determine whether or not to select data for the report when the report is run.
	
	Conditions are described in the section Formulas and Conditions.

Expressions	Expressions are calculations inside parentheses. The HP ALLBASE/BRW Calculation Language lets you use **expressions** anywhere a single argument can be specified.
Formulas	A formula is any calculation where the resulting type is Integer, Real, String, Numeric, Date, or Time. A formula is always specified as an expression, similar to the right side of an assignment statement in PASCAL or BASIC. There are two types of formulas in HP ALLBASE/BRW: ■ Calculated Items ■ Functions

Some Calculation Terms (continued)

Functions	A function is like a program procedure or subroutine. You can use a function to avoid typing complicated formulas. Functions can be used in formulas or conditions; however, functions cannot be used as items. Each function must have at least one argument, and can have up to nine. A function argument is a value that can be passed to the function when it is run. There are three categories of functions: ■ **Table functions**. These are functions that you define from the Define Table screen. A table function can only be used in calculations for the table from which it is defined. ■ **Layout functions**. These are functions that you define from the Define Lines screen. A layout function can only be used in calculations defined in the report layout, that is, in formulas created from the Define Lines screen, not in table calculations. ■ **Global functions**. These are functions that you define with the BRWSETUP program. A global function can be used in any calculation in a report. See Chapter 17, for more information on how to define global functions. See "How Calculated Items Work" in Chapter 19, for a more technical description of how calculated items and functions work and how they affect performance.
Operators and Operations	The term operator is used in the normal mathematical sense, that is, an operator is a mathematical or logical symbol that specifies an operation to be performed on an argument or arguments. For example, the * symbol in a calculation is an operator that tells HP ALLBASE/BRW to multiply the PRICE by 10. Operations can also be performed by functions.
Types	Arguments, operations, and calculation results can be one of the following 6 types. Numeric — Fixed-point decimal numbers Integer — Numbers with no decimal part Real — Floating-point decimal numbers String — Strings of characters Time — Values of hours, minutes, and seconds Date — Values of day, month, and year HP ALLBASE/BRW includes standard functions that affect one or more of the above types, for example, the function YEAR_OF will extract the year from any date value.

Aggregate/Array Operations

HP ALLBASE/BRW includes many aggregate/array operations that act on arrays. One example is an operation that sums all the values of an array.

Aggregate/array operations are calculations very similar to column calculations except that they affect array items.

HP ALLBASE/BRW includes the following aggregate operations:

Function	Definition
SUM	Accumulates the values
AVG	Calculates the average of the values
MIN	Gives the minimum value
MAX	Gives the maximum value
INDEX_MIN	Gives the index of the minimum value
INDEX_MAX	Gives the index of the maximum value
COUNT	Counts the values; ignores NO_VALUE

A typical example of the use of an aggregate operation is the creation of cumulative values, shown here:

```
SUM i = 1 TO 12 OF customers.turnover (i)
```

where *i* is the aggregate index, *1* the start index, *12* the stop index, and *customers.turnover (i)* is the aggregate expression, and in this case, an array item.

In this next example, the customer's turnover values are accumulated from January up to the month specified by the parameter *?month*. The start index, stop index and the aggregate expression can be any numeric expressions.

```
SUM i = 1 TO !!?month!! OF  TURNOVER(i)
```

Arguments

Arguments are the values used in a calculation. Arguments include the following:

- items, standard items, and array items
- functions and standard functions
- function arguments
- constants
- parameters
- column calculation results
- expressions

Priority of Arguments

Items, array items, standard items, user-defined table or layout functions, user-defined global functions, function arguments and standard functions can have the same names. If the same names occur in a report, they are ranked in this order of priority:

Rank	Name, i.e., customer-no
1	Aggregate Index (see Aggregate/Array Operations)
2	Function Argument
3	Calculated Item
4	Item
5	Standard Item

If the same names occur and are followed by an open parenthesis, they are ranked in this order:

Rank	Name (expression), i.e., turnover(4)
1	Function Argument
2	Table or Layout Function
3	Global Functions
4	Item
5	Standard Function

For example, if DATE is the name of both a calculated item and a standard item, it is used as a calculated item, which has the higher priority. Or, if AMOUNT is the name of a function argument and a calculated item, any reference to AMOUNT in the function's formula will be considered a reference to the function argument AMOUNT.

Standard items and standard functions can be explicitly specified as shown in this example:

```
STANDARD.DATE
```

which is the standard item *date*.

Related Sections

- Arrays and Array Items
- Aggregate/Array Operations
- Column Calculation Results
- Constants
- Functions
- Items
- Parameters
- Standard Items

Arrays and Array Items

An array is a combination of one or more items of the same type. Each item in an array (also called a component of an array) is treated exactly like a normal item. Arrays are defined in a data dictionary; you can manipulate but not create arrays in HP ALLBASE/BRW.

Characteristics

Arrays have these characteristics:

- Each array item is referenced by the array name followed by an index in parentheses, as shown here:

  ```
  TURNOVER(4)
  ```

  This will reference the fourth item (or component) of the array TURNOVER.

  ```
  TURNOVER ((MONTH + 4) / 2)
  ```

  This will reference the item whose index is the result of the expression MONTH + 4 / 2.

- As with items, if the array name is *not* unique, it can be preceded by the source table name followed by a period; for example:

  ```
  CUSTOMERS.TURNOVER(?MONTH)
  ```

  This will reference the item in the array TURNOVER (in source table CUSTOMERS) whose index is equal to the value of the parameter ?MONTH.

- The first array component has an index of 1.
- If no index is specified, the first array component is selected. That is, if you specify TURNOVER (in a calculation or in a lineset) without an index, the value of `TURNOVER (1)`is used in the calculation (or is printed).
- If the index points to an array component that does not exist (for example, an array TURNOVER has 12 components, and the array item is TURN OVER(13)), an INDEXVIOLATION occurs. See "Exception Handling" later in this chapter.
- The array index must be of the type Integer.
- Array components can be of any HP ALLBASE/BRW type (so long as all the components of an array are the same type) that is; Numeric, Real, Integer, String, Date, or Time. They behave exactly as normal items of their type. For example, they have a single value per record, and so on.

- Arrays are usually referenced by component; only on the Project Items and Output File screens do you refer to an entire array.

  HP ALLBASE/BRW makes no distinction between normal items and array items. (To HP ALLBASE/BRW, a normal item is an array with one component.)

Related Sections

- Arguments
- Aggregate/Array Operations
- Calculation Language
- Naming Conventions

Calculation Language

The Calculation Language uses arguments, expressions, and operators. Arguments and expressions are used in calculations: items, standard items, arrays, functions, constants, and column calculation results. Operators can perform mathematical or logical operations on arguments.

Related Sections

- Arguments
- Arrays and Array Items
- Aggregate/Array Operations
- Column Calculation Results
- Constants
- Expressions
- Functions
- Operators/Operations
- Parameters

Column Calculation Results

Column calculations do not use the Calculation Language, and are described in Chapter 8, "Defining Report Layout." However, column calculation *results* can be used as arguments within layout calculations or layout functions. This is necessary, for example, to calculate the percentage of variance between two column total values, as shown in the following example.

The layout calculated item and its formula are:

Item	Formula
VARIANCE-%	(100 * ACTUAL - BUDGET) / BUDGET

On the Define Lines screen, VARIANCE-% for the break footing lineset on ACCOUNT is specified as shown here:

```
HP ALLBASE/BRW                    Define Lines                    Report: ACCOUNTS
                    Footing for Break Level 1 on ACCOUNTS-NO
    ....|....10...|....20...|....30...|....40...|....50...|....60...|....70...
    ....|....10...|....20...|....30...|....40...|....50...|....60...|....70...

              Item            Type  Pos   Line  Len    ColCalc   NumPrec SuppRep
    ACTUAL                                              TOTAL
    BUDGET                                              TOTAL
   -VARIANCE-%                      -     -     -     -CALC       -         -
   -                                -     -     -     -           -         -

First page of items                                                      1035
 Insert   Delete   Scroll   other   12     1   Prev     Next     Help     Exit
  Line     Lines    --->     keys                Line     Line
```

The Define Lines screen is fully described in Chapter 16.

VARIANCE-% is calculated each time the value of the item ACCOUNT changes. Because CALC is specified in the ColCalc field for this item, the total values of arguments ACTUAL and BUDGET are used to find the percentage of variance of these totals. If you had not specified CALC, arguments ACTUAL and BUDGET would have the value of whatever was on the last detail line when the lineset was printed.

On the Define Lines screen, HP ALLBASE/BRW uses the definition of the argument items found in the ColCalc and NumPrec field. For example, if a numeric precision for items ACTUAL and BUDGET had been specified in the above example, the variance percentage would then have been calculated from the totals with this precision.

Related Sections

- Arguments
- Arrays and Array Items
- Aggregate/Array Operations
- Calculations
- Constants
- Expressions
- Functions
- Numbers (Defining Numeric Precision)
- Operators/Operations

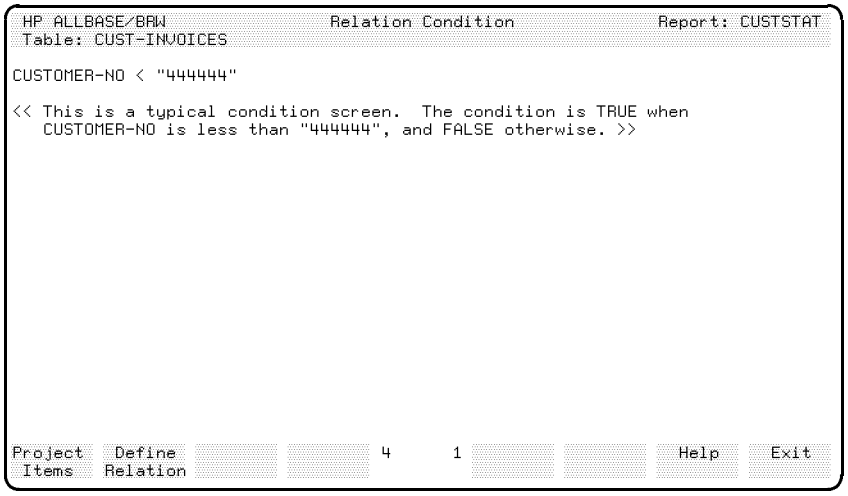
Conditions

A condition is any form of calculation that produces a Boolean result. That is, a result that can be either true or false.

Conditions include:

- selection conditions
- relation conditions
- suppress line conditions
- suppress lineset conditions

The screen for each condition is described in Chapter 16 in alphabetical order. The screens are also pictured in sections of the manual describing tasks that are done using that screen. All the condition screens are similar, and all use the Calculation Language to produce a Boolean (condition) result. A general example of a condition screen is shown below:



This condition is true if CUSTOMER-NO is less than “444444”. If the CUSTOMER-NO is greater than or equal to “444444”, this condition is false. Conditions can use any Calculation Language arguments.

Selection Conditions

Selection conditions are used to define selection criteria for a subset of data. The Selection Condition screen is used to define them. See Chapter 16 for the screen description.

Characteristics

Selection Conditions have the following characteristics:

- They select a subset of data from the final access table.
- They are calculated from items, parameters or constants.
- They can use table calculated items, since a table calculated item is used like any other item after it is projected from its table.

Example

Suppose you want to produce a report that would include only products from Product Line 60 that have been ordered during December. You could use this selection condition:

```
PRODUCT-LINE = "60" AND MONTH_OF (ORDER-DATE) = 12
```

This condition produces a report including only products from Product Line 60 which have been ordered in month 12. That is, when the condition is true the data is selected.

Relation Conditions

Relation conditions are similar to selection conditions but are defined within a table. Relation conditions are defined on the Define Relation Screen. See Chapter 16 for the screen description.

Characteristics

Relation conditions have these characteristics:

- They are used to select a subset of the records to be included in a table.
- They are calculated from items, table calculated items or table functions defined in the same table, parameters, or constants.

Example

If you want a report with only the records for the WESTERN sales area, you could use this Relation Condition:

```
SALES-AREA = "WESTERN"
```

This condition allows only those records whose sales areas are WESTERN to be written to the table. When the condition is true, the data is written to the table.

Note

For better performance: Although relation conditions are defined to select data from the final access table, they are optimized so they are evaluated as early as possible. The same is true for selection conditions. See the "Using the Compile Listing" section in Chapter 19 for more information.

Suppress Line Conditions

Suppress line conditions are used to define criteria under which the printing of a line is suppressed. It is especially useful when there is only one condition you want to exclude from the report. Suppressing the line for that condition is simpler than selecting lines based on multiple conditions.

Suppressed lines are defined on the Suppress Line screen. See Chapter 16 for the screen description.

Characteristics

Suppress line conditions have the following characteristics:

- They are calculated from items, layout calculated items, or layout functions, parameters or constants.
- They can use column calculation results as arguments.

Example

Suppose you have an invoice report where there are multiple entries under one dollar (1.00). You don't want the report cluttered with these amounts. If you want your report to print only lines where the INVOICE-AMOUNT is over 1 dollar, you can suppress the line based on this calculation:

```
INVOICE-AMOUNT < 1.00
```

All lines where the invoice amount is less than 1.00 are suppressed. The report will only contain those lines whose INVOICE-AMOUNT is 1 dollar or over. When the condition is true, the line is suppressed. However, a TOTAL or AVG column calculation will still include these values. (This is not true for suppress lineset conditions.)

Suppress Lineset Conditions

Suppress lineset conditions are similar to suppress line conditions, except that they suppress *complete* lineset instead of specific lines within a lineset. They are defined on the Suppress Lineset screen. See Chapter 16 for a description of the screen.

Characteristics

Suppress lineset conditions have these characteristics:

- They are calculated from items, layout calculated items, or layout functions, parameters, or constants (the same as suppress line conditions).
- They can use column calculation results as arguments (the same as suppress line conditions).
- They are used to suppress *complete* linesets.
- They suppress all of the following:
  - all lines in the lineset
  - pagination defined for the lineset

- any records to be written to the lineset's output file
- column calculations to be calculated, if the lineset is a detail or a break footing

Example

Suppose you are creating a Customer Order report and you do not want any details printed when a customer has no orders. You can use this Suppress Lineset condition:

```
ORDER-NO = NO_VALUE
```

With this condition, no details are printed for a customer who has no orders. When the condition is true, the lineset is suppressed.

Related Sections

- Arguments
- Arrays and Array Items
- Aggregate/Array Operations
- Calculations
- Column Calculation Results
- Conditions/Boolean Operations
- Constants
- Expressions
- Functions
- Operators/Operations
- Parameters

Conditions/Boolean Operations

The HP ALLBASE/BRW Calculation Language can be used to express conditions in IF THEN ELSE operations or as formulas in selection, relation or suppress line/lineset conditions. Conditions use comparisons and Boolean operators, and the results are always true or false.

Comparison Operators

Comparison operators used in condition operations are:

Operator	Definition
=	Equal to
<>	Not equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to
IN	Equal to one of the values in a list of expressions or one of the values of a list parameter
MATCHES	Matches a pattern or a list of patterns
SATISFIES	Satisfies the actual condition of the comparison predicate parameter

Examples of comparison operations include:

```
city = "New York"

paid-amount + discount-amount < invoice-amount

country IN ("Europe", "North America", "South America")

color IN ?color-list
```

For the `color IN ?color-list` example, an actual color can be specified at report execution time in place of the `?color-list` parameter.

Note that the following example:

```
product-no SATISFIES ?product-no
```

allows flexible selection on PRODUCT-NO because the selection criteria (that is, value and operator) are supplied when the user requests the report.

String Comparisons

String comparisons have these characteristics:

- When strings of different lengths are compared, the shorter one is padded with trailing blanks.

Patterns

Patterns have these characteristics:

- They are constants with optional mask characters.
- They are always enclosed in double quote marks.
- They use ? as the mask for one character and @ as the mask for any number of characters, as shown in these examples:

```
customer-no MATCHES  ("ZJ@E", "@B")
```

This comparison selects all customer numbers beginning with ZJ and ending with E, or those ending in B.

```
city     MATCHES  ("?????@")
```

This comparison selects all cities with names of 5 or more characters.

Boolean Operators

Boolean operators allow the combination of single comparisons, as shown here:

Operator	Definition
NOT	True, if comparison is false
AND	True, if both comparisons are true
OR	True, if one or both comparisons are true

Examples of Boolean operators include:

```
product-no  SATISFIES ?product-no  AND
order-no    SATISFIES ?order-no    AND
order-date  SATISFIES ?order-date
```

Constants as Arguments

Constants are arguments that have fixed values in calculations. Constants can be of any HP ALLBASE/BRW type (Real, Numeric, Integer, String, Date, or Time).

Consider the following examples:

- **Numeric constants** such as 5.68, -3.14;
- **Integer constants** such as 0, 45, -2134;
- **Real constants** such as 3.45E2. But see the explanation of Real syntax under Real numbers in the "Numbers" section in this chapter.
- **String constants**, which consist of text enclosed in double quote marks:

```
"I like HP Business Report Writer"
```

- **Date constants**, which require single quote marks and use these conventions, assuming the input format is month-day-year:

'4/16/1992'	<<April 16, 1992>>
'4.16.1992'	<<same>>
'8/24/93'	<<year is 1993>>
'10/18/03'	<<year is 1903>>
'12/6'	<<year is current year>>
'12/92'	<<invalid>>

  Date input format is defined in BRWSETUP. The alternative formats are day-month-year or year-month-day. See also the "Dates" section of this chapter.

- **Time constants**, which require colon(s) and use these conventions:

8:24:37	<<8 o'clock, 24 minutes, 37 seconds>>
10:4	<<4 minutes past 10 o'clock>>
0:14:50	<<14 minutes and 50 seconds>>
00:00:00 or 0:0	<<midnight >>
23:59:59	<<one second before midnight>>
24:00	<<invalid>>

Related Sections

- Arguments
- Calculations
- Dates
- Numbers
- Strings
- Time

Conversion Functions

Four built-in standard conversion functions are available for HP ALLBASE/BRW's calculation language. They are described in the following table.

Standard Conversion Functions

TO_INTEGER(*expr*)	The TO_INTEGER function converts the argument, *expr* to an integer. This function is valid for real, numeric, and string values of *expr.* If the number is out of range, or if the expression is a string, the string does not convert, the function returns an `ILLEGAL_VALUE` message. In all cases except for a real value of *expr*, the value is truncated. In the case of a real value, it is rounded. Digits following a decimal point in a string are ignored.
TO_REAL(*expr*)	The TO_REAL function converts the argument, *expr* to a real value. This function is valid for integer, numeric, and string values of *expr.* Out-of-range values of *expr* are truncated. String values that do not convert cause the function to return an `ILLEGAL_VALUE` message.
TO_NUMERIC(*expr*)	The TO_NUMERIC function converts the argument, *expr* to a numeric value. This function is valid for integer, real, and string values of *expr.* Strings that contain more than 13 digits before the decimal cause the function to return an `OVERFLOW` message. Only 6 digits after the decimal are significant. Any subsequent digits are ignored. String values that do not convert cause the function to return an `ILLEGAL_VALUE` message. If a real value of *expr* is specified that is outside the range of valid numeric values, the function returns an `OVERFLOW` message.
TO_STRING(*expr*)	The TO_STRING function converts the argument, *expr*, to a string value. This function is valid for integer, numeric, and real values of *expr*. A default format is used when converting the argument. This format varies, depending on the source data type. The default format for integers consists of 10 digits with a leading sign that is blank for positive integers. The number is right justified. The default of numeric values of *expr* is 22 characters, right justified, with a leading sign, an embedded decimal point, and 6 digits after the decimal. The default format for real values is the standard G22.16 format, right justified, with a leading sign.

Related Sections

- Arguments
- Functions
- Numbers, real
- Standard Functions

Dates

You can manipulate dates using HP ALLBASE/BRW Date types. Items of type Date are defined in the data dictionary. (See Chapter 18 for the data dictionary definitions of Date types.) Internally, a date always consists of a month, a day, and a 4-digit year (1992, not 92). However, Date items can be output in different ways, depending on the edit-mask (for output) and the data dictionary definition (for input). HP ALLBASE/BRW converts all dates to the form YYYYMMDD and handles them internally in this form. Date types require single quotes and use the following conventions, assuming the input format is month-day-year:

'4/16/1992'	<<April 16, 1992>>
'4.16.1992'	<<same>>
'8/24/91'	<<year is 1991>>
'10/18/03'	<<year is 1903>>
'12/6'	<<year is current year>>
'12/91'	<<invalid>>

Date input format is defined in BRWSETUP. The alternative formats are day-month-year or year-month-day. See also the "Dates" section of this chapter.

Date Extraction Functions

Date extraction functions extract parts of a date value, as shown in these examples,

Function	Date Value	Result
`DAY_OF_WEEK`	('9/28/1992')	<<6>> (<term noglossWhere 1 = Monday,. . . . . 7 = Sunday|)
`DAY_OF_MONTH`	('9/28/1992')	<<28>>
`DAY_OF_YEAR`	('9/28/1992')	<<271>>
`WEEK_OF`	('9/28/1992')	<<39>>
`MONTH_OF`	('9/28/1992')	<<9>>
`YEAR_OF`	('9/28/1992')	<<1992>> (result is always 4-digit year>

Note

The default for the day of the week is that the first day of the week is MONDAY, not SUNDAY.

The result of a date extraction function is an integer-type item.

Date Duration Functions

Date duration functions calculate the difference between two date values as positive numbers. Date duration is specified as shown here:

```
DAYS_DIFF          (date1, date2)
WEEKS_DIFF         (date1, date2)
MONTHS_DIFF        (date1, date2)
YEARS_DIFF         (date1, date2)
```

The result of a date duration function is an integer-type item.

The date duration functions calculate:

```
WEEKS_DIFF      as      DAYS_DIFF / 7
MONTHS_DIFF     as      DAYS_DIFF / 30.4385
YEARS_DIFF      as      DAYS_DIFF / 365.2422
```

and round the result to the nearest integer number.

Date Addition Functions

Date addition functions calculate a date value by adding days, weeks, months or years to a date value, as shown here:

```
ADD_DAYS           (date, days)
ADD_WEEKS          (date, weeks)
ADD_MONTHS         (date, months)
ADD_YEARS          (date, years)
```

The result of a date addition function is a date-type item.

Note

Out-of-range dates are corrected automatically, for example:

```
ADD_MONTHS   (1/31/1992,1)    2/28/1992
```

Date Subtraction Functions

Date subtraction functions calculate a date value by subtracting days, weeks, months or years from a date value, as shown here:

```
SUB_DAYS           (date, days)
SUB_WEEKS          (date, weeks)
SUB_MONTHS         (date, months)
SUB_YEARS          (date, years)
```

The result of a date subtraction function is a date-type item.

Date Operations

HP ALLBASE/BRW supplies many standard functions to manipulate dates.

See the "Standard Functions" section of this chapter for a complete list.

Exception Handling

Exception values indicate whether the value of an item, function, parameter or any expression is invalid. HP ALLBASE/BRW distinguishes between NO_VALUEs and ERROR_VALUEs:

- NO_VALUE signifies that an item or a result has no value. Note that NO_VALUE means "no value." It does not, for example, mean that a numeric item has a value of 0. NO_VALUEs can occur, for example, when you specify an open join on a source table. NO_VALUE is set for an SQL item having a NULL value.

  You can make a particular item value return NO_VALUE by using a calculated item with a formula liek:

  ```
  IF ANSWER = "NONE" THEN NO_VALUE ELSE ANSWER
  ```

- ERROR_VALUE is a name that includes the following:

Value	Definition
OVERFLOW	Numeric overflow occurred.
DIVIDE_BY_ZERO	Divide by zero occurred.
INDEXVIOLATION	Out of range index occurred.
INVALID_VALUE	For example, database contains letters in a display numeric field.

You can use the BRWSETUP program to modify a BRWCONF file to print a value when an exception condition occurs. See Chapter 17 "Configuration" for a description of the BRWSETUP program. Note that non-printable characters in strings do not cause an INVALID_VALUE.

Whenever a new exception value occurs, an execution warning message is issued. All execution warning messages are printed at the end of the report.

This is an example of an exception report:

```
The following exceptions occured in this report:

Field  overflow occurred with item SALES-VALUE       (54*)
Divide by zero occurred with item ORDER/STOCK-QTY    (12*)
```

The number to the right of the message indicates the number of times the exception occurred in the report.

Automatic Exception Value Handling

Automatic exception value handling means that a calculation produces an exception value if one of the arguments has an exception value. For example, if one of the arguments in an addition has no value, the result of the addition will also have no value.

HP ALLBASE/BRW uses an exception algorithm for calculations. If one of the arguments has an exception value, then an exception value results. Note that this *does not* apply to operators =, <>, AND, OR. The table below shows how the system handles arguments with exception values:

Argument 1	Argument 2	Result
ok	ok	ok
ok	no value	no value
ok	error	error
no value	error	error

Comparisons for equal or not equal are always true or false. Same exception values will be treated as equal. Comparisons for less or greater cannot be true or false if one argument has an exception value, so the comparison result will be an exception value (which is interpreted as false if it is used for a selection).

Boolean operators AND and OR use the following logic:

Argument 1	Argument 2	AND	OR
false	false	false	false
true	false	false	true
true	true	true	true
false	no value	false	no value
true	no value	no value	true
false	error	false	error
true	error	error	true

Note that indexes or conditions which have exception values produce exception values as calculation results.

Testing For and Assigning Exception Values

You can test for exception values or assign the result of a calculation to an exception value.

Example

Suppose you might want to have the report automatically insert the current month as the value for the parameter ?MONTH if the parameter has no value. You could use a condition such as the following:

```
IF ?month = NO_VALUE
THEN
   month_of (date)          <<current month>>
ELSE
   ?month
```

Suppose you have a report where the calculated item BUDGET selects the current year, previous year or next year budget depending on the parameter ?YEAR. If ?YEAR has any other value, the budget value produces an error.

```
IF ?year = "CURRENT" THEN cy-budget
ELSE IF ?year = "PREVIOUS" THEN py-budget
  ELSE IF ?year = "NEXT" THEN ny-budget
    ELSE ERROR_VALUE
```

Avoiding Exceptions

In some cases, you want to avoid exceptions in your report.

Example

If the divisor operand in a division is potentially zero and you do *not* want an exception value as the division result, use the following condition:

```
IF b <> 0 THEN a / b ELSE 0
```

Related Sections

- Arguments
- Conditions/Boolean Operators
- Column Calculation Results

Expressions

Expressions are calculations inside parentheses. The Calculation Language lets you use expressions anywhere a single argument can be specified.

Example

In the calculation below, two arguments are used, the item PRICE and the constant 10. The result will be the value of PRICE multiplied by 10:

```
PRICE * 10
```

You could substitute an expression for the constant, as follows:

```
PRICE * (1 + 9)
```

This, of course, produces the same result. You could also substitute a calculated item or function that returns the number 10, for the expression and you would obtain the same result. Full Syntax Diagrams are supplied the "Syntax Diagrams" section in this chapter.

Related Sections

- Calculation Language
- Constants
- Functions
- Syntax

Formulas

Formulas include:

- table calculated items
- layout calculated items
- table functions
- layout functions

Calculated Items

Calculated items are items that are created from a formula that you define. Once created, calculated items can be used in a report just like an item projected from a final access table, or an HP ALLBASE/BRW standard item. Calculated items are split into two categories, table calculated items and layout calculated items, depending upon whether they are to be created when the tables are built or when the report is formatted.

Table Calculated Items

Table calculated items are defined *within* a table . They are created when the table is constructed. The Table Calc Item screen and the Table Calc Item Formula screen are used to define the item and its formula.

Characteristics

Table calculated items have these characteristics:

- Outside the table in which they have been defined, they can be used like any other item projected from the table. (See Chapter 6 for details about projecting items). For example, they can be:

  printed
  written to an output file
  used for selection
  used for sorting
  used in another calculation formula

- They can be any one of the HP ALLBASE/BRW item types:

  N - fixed-point numeric
  R - floating-point real
  I - integer
  S - string
  D - date
  T - time

- They can be calculated from source table items, from other calculated items or functions defined in the same table, or from parameters or constants.

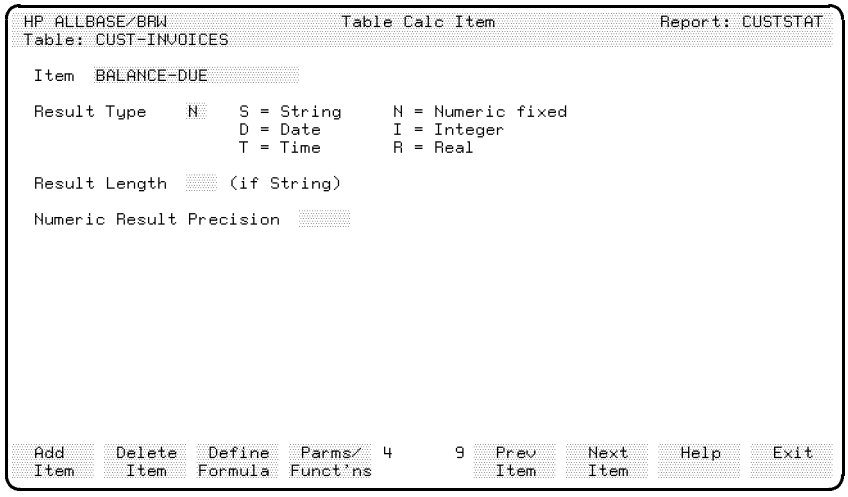
Restrictions

Restrictions on table calculated items are:

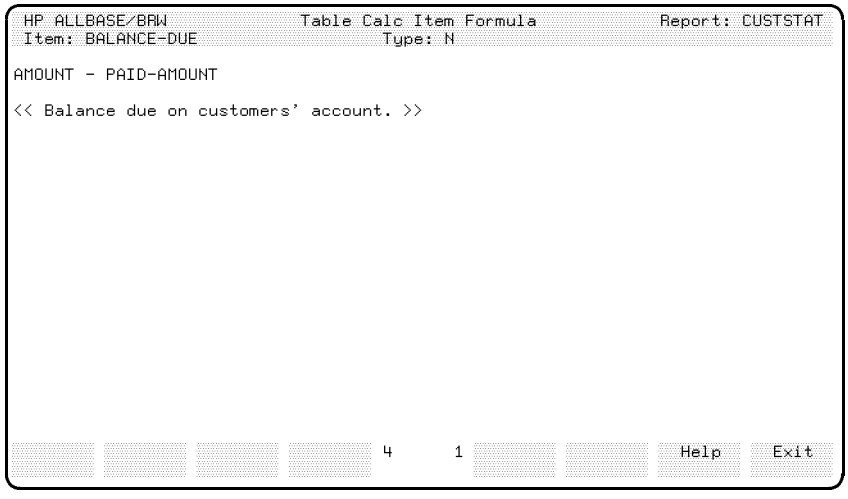
- They cannot be used on the Define Relation screen for the same table in which they are defined.
- They cannot use column calculation results (or certain standard items, such as PAGE) as arguments. See "Standard Items" later in this chapter for a list of available standard items.

Example

Here is an example of a table calculated item, BALANCE-DUE:



This is the Table Calc Item screen for BALANCE-DUE. It is a numeric item.



You define the formula for a table calculated item on the Table Calc Item Formula screen. This formula can use any of the arguments and operators allowed in the Calculation Language, and must obey

its syntax rules. These are described in the "Arguments and Syntax" section of this chapter.

When the table is constructed, every row of the table will have an item called BALANCE-DUE that you can use in the report. This item will behave exactly like any other Numeric item that you project from the table. Each value of BALANCE-DUE will be equal to the value of AMOUNT, minus the value of PAID-AMOUNT, on that row of the table.

Layout Calculated Items

Layout calculated items are very similar to table calculated items, but they are defined within the report layout. Layout calculated items are calculated when the report is formatted. The Layout Calc Item screen and the Layout Calc Item Formula screen are used to define the item and its formula.

Characteristics

The characteristics of layout calculated items are exactly the same as for table calculated items, except that:

- Layout calculated items cannot be used in formulas for table calculated items, relation conditions, or table functions.
- Layout calculated items can use column calculation results in their calculations; for example, a column total or an average value can be used in the calculation for a layout calculated item.
- Layout calculated items can be calculated for each detail lineset, or, you can specify that they be calculated only for certain linesets.

Restrictions

Restrictions on layout calculated items are:

- They cannot be used for selection or sorting.

Note

For better performance:

- Table calculated items are calculated as early as possible in the data access for a report. This means that they are only calculated when their arguments change.

  Table calculated items generally can improve performance if they replace the items from which they are calculated, so reducing the width of the table.

- Layout calculated items are calculated, by default, once for each detail. They can improve performance if they *do not* replace (that is, are additional to) the items from which they are calculated, since layout calculated items are not part of any table.

- If you use a layout calculated item only on a specific break, you can further improve performance if you specify that it is to be calculated *only* if that break occurs.

- Calculated Item performance is further discussed in Chapter 19, "Performance Considerations."

Examples

Suppose you are defining BALANCE-DUE as a table calculated item. Doing that would be efficient if you did not want items AMOUNT and PAID-AMOUNT in your report. That is, you would calculate and project BALANCE-DUE, and not project AMOUNT and PAID-AMOUNT. This would save writing AMOUNT and PAID-AMOUNT on every row of the table, when you did not in fact need them. For more information, see the "Compile Listing" section in Chapter 19.

On the other hand, if you wanted items PAID-AMOUNT and AMOUNT in your report, as well as BALANCE-DUE, you should define BALANCE-DUE as a layout calculated item. It would be calculated once for each detail line, but would not add to the space on each row of the table.

Suppose, however, that you wanted the BALANCE-DUE only for the break on CUSTOMER-NO. You would add item BALANCE DUE on the Define Lines screen for that lineset, and specify CALC in the ColCalc field for BALANCE-DUE. Then the item would only be calculated on that lineset.

Functions

A function is like a program procedure or subroutine. You can use a function to avoid typing complicated formulas. Functions can be used in formulas or conditions; however, functions cannot be used as items. Each function must have at least one argument, and can have up to nine. An argument is a value that can be passed to the function when it is run. Function arguments can (optionally) be used in the function formula.

When you call a function, HP ALLBASE/BRW assigns the values you supply in place of the arguments in the function calculation, performs the calculation, and returns a value as the result of the function. This result is then used in your calculation in place of the function name. The value returned can be of any HP ALLBASE/BRW type (Real, Integer, Numeric, String, Date, or Time).

Table Functions

Table functions are functions that you define for use in table calculated items, relation conditions, and other table functions. The Table Function screen and the Table Function Formula screen are used to define table functions.

Characteristics

Table functions have these characteristics:

- They are only used in other calculation formulas in the *same* table (that is, table calculated items, other table function, and relation conditions defined in the same table).
- They can be calculated from function arguments, source table items, calculated items or other functions defined in the same table, parameters, or constants.
- They can return results of any one of the HP ALLBASE/BRW types:

  N - fixed-point numeric

  R - floating-point real

  I - integer

  S - string

  D - date

  T - time

Restrictions

Table Functions have these restrictions:

- They cannot be used in any formulas except for table calculated items, relation conditions, and other table functions defined for the same table.

Example

Here is an example of a table function, AMOUNT-IN-WORDS:

```
HP ALLBASE/BRW                    Table Function                  Report: CUSTSTAT
Table: CUST-INVOICES

 Function  AMOUNT-IN-WORDS

 Result Type  S    S = String      N = Numeric fixed
                   D = Date        I = Integer
                   T = Time        R = Real

 Arguments:
                   Name                Type
       1   AMOUNT                       I
       2
       3
       4
       5
       6
       7
       8
       9

  Add      Delete   Define              4      13   Prev     Next      Help     Exit
Function Function Formula                          Function Function
```

This is the Table Function screen for item AMOUNT-IN-WORDS. It has one Numeric argument, AMOUNT, and returns a String-type result. It has been defined from the table FINAL. Therefore it can only be used in table calculated items, relation conditions, and other table functions for table FINAL.

```
HP ALLBASE/BRW                Table Function Formula              Report: CUSTSTAT
Function: AMOUNT-IN-WORDS              Type: S
Arguments:
1 AMOUNT  I                2                          3
4                          5                          6
7                          8                          9

IF AMOUNT = 0 THEN " "
ELSE AMOUNT OF
("ONE", "TWO", "THREE", "FOUR", "FIVE", "SIX", "SEVEN", "EIGHT", "NINE")

<< Returns AMOUNT in words.  Note that AMOUNT must be between 0 and 9,
   otherwise an INDEXVIOLATION will occur. >>

                                        8      1                     Help     Exit
```

In this example, the formula for AMOUNT-IN-WORDS is defined on the Table Function Formula screen. This formula can use any of the arguments and operators allowed in the Calculation Language, and must obey its syntax rules. See the Syntax Diagram and Syntax Rules sections for more details.

With this formula, you can call AMOUNT-IN-WORDS from any formula defined for the table FINAL. You must supply an Integer for

AMOUNT, and HP ALLBASE/BRW returns the string depending on the value of AMOUNT.

Layout Functions

Layout functions are similar to table functions but are defined for the report layout. The Layout Function screen and the Layout Function Formula screen are used to define layout functions.

Characteristics

Layout functions have these characteristics:

- They can only be used in the report layout *inside* other calculation formulas for layout calculated items, other layout functions, suppress lineset, and suppress line conditions.
- They are the same types as table functions.
- They are calculated from function arguments, items, layout calculated items, other layout functions, parameters, or constants.

Global Functions

Global functions are similar to table and layout functions but are defined for more than one report. The Global Function screen and the Global Function Formula screen are used to define Global Functions in the BRWSETUP program.

Characteristics

Global functions have these characteristics:

- They may be used in any calculation formula or condition in a report.
- They can return results of any one of the HP ALLBASE/BRW types.
- They are calculated from function arguments, other global functions, or constants.

Note

For better performance: You can improve calculation performance by using calculated items instead of functions if the functions are always used with the same arguments. Consider these points:

- Calculated items have a single value per record, and so are calculated once per record regardless of how many times they are used.
- Functions are calculated each time the function is used because results depend on the values of the function arguments.

Functions as Arguments

You can use a function as an argument in a calculation. Such functions can be user-defined table functions, layout functions, global functions, or HP ALLBASE/BRW standard functions. User-defined functions are described in the "Formulas and Conditions" section of this chapter, and Standard Functions are described in the "Standard Functions" section of this chapter.

You reference (or call) a function in a calculation by the function name, followed by a value for each function argument. The values must be in parentheses and separated by commas. When calling a function, you must supply a value for every function argument. In the example below, VARIANCE-% has two arguments and AMOUNT-IN-WORDS one.

```
VARIANCE-% (ACTUAL-AMOUNT, BUDGET-AMOUNT)

AMOUNT-IN-000'S (AMOUNT)
```

HP ALLBASE/BRW calls the function, assigns the value(s) to the function argument(s), performs the function's formula, and returns the result of the formula to your calculation. This resulting value is then used in your calculation in place of the function name. The value returned can be of any HP ALLBASE/BRW type (Real, Integer, Numeric, String, Date, or Time).

Function Arguments in Calculations

Every function has at least one argument. Function arguments can be used in the function formula.

Example

In this example, the function AMOUNT-IN-000'S is defined:

```
AMOUNT-IN-000'S (AMOUNT)
```

(AMOUNT) is the function argument and is of the Numeric type. You substitute a value for it when you call the function. Within the function, the value is referred to by the function argument name. That is, the formula for AMOUNT-IN-000'S can be something like this example:

```
If  AMOUNT > 1000  Then  "OVER  ONE  THOUSAND"
                   Else  "LESS  THAN  ONE THOUSAND"
```

You can call AMOUNT-IN-000'S in many ways, as shown here:

```
AMOUNT-IN-000'S (2500)

AMOUNT-IN-000'S (PRICE * QUANTITY)
```

This example works as long as you substitute a value for AMOUNT. The value must be a number. A value of another number type such

as Real or Integer is automatically coverted to Numeric. (See "Type Conversion" under the "Types" section of this chapter.) The function argument AMOUNT can be used in the formula for the function where it is defined.

Related Sections

- Arguments
- Numbers
- Standard Functions
- Types

IF THEN ELSE Operations

You can use the construction IF THEN ELSE in a calculation formula to allow different calculations to be used, depending on whether a condition is true or false.

Example

Suppose in your report you want to show the discounted amount for an invoice. The discount for amounts under $1000 is 3 percent, the discount for amounts over $1000 but under $10,000 is 5 percent, and the discount for all other amounts is 10 percent. Here is an example of how to use an IF THEN ELSE operation to calculate the DISCOUNT-AMOUNT:

```
IF amount < 1000
THEN amount * 0.03        <<3% discount>>>
ELSE IF amount < 10000
  THEN amount * 0.05      <<5% discount>>>
  ELSE amount * 0.10      <<10% discount>>>
```

Or, in a shorter form:

```
amount * (IF amount < 1000 THEN 0.03
   ELSE IF amount < 10000 THEN 0.05 ELSE 0.10)
```

Note that the form is always IF THEN ELSE, and ELSE is required. You can nest IF THEN ELSE constructions, as shown in the first example.

Related Sections

- Arguments
- List Operations
- Types

Items

Items provide a single value each time a calculation is performed. Most items, of course, are projected from the final access table. You can also define your own items, for example, a calculated item is an item that you define. You can also use any HP ALLBASE/BRW standard items.

Characteristics

All items have these characteristics:

- Items have a single value per record. That is, on every record, or detail line, an item can have only one value. The value of an item can, of course, change with the next record.
- Items are referenced by the item name, for example:

  ```
  CUSTOMER-NAME
  ```

- If an item name is *not* unique, for example, if an item with the same name appears in more than one source table, it can be preceded by the source table name (without owner name if it is an HP ALLBASE/SQL table), followed by a period. The following example shows item DUE-DATE from source table INVOICES.

  ```
  INVOICES.DUE-DATE
  ```

- Standard items can be explicitly specified as shown in this example:

  ```
  STANDARD.DATE
  ```

  which is the standard item DATE.

Related Sections

- Standard Items
- Arrays and Array Items

List Operations

List operations select from a list of values.

A calculated item, for example, which translates the weekday number into a three-character string, could be specified using nested IF THEN ELSE constructions, like this:

```
IF weekday = 1 THEN  "MON"
ELSE  IF weekday = 2 THEN  "TUE"
  ELSE  IF weekday = 3 THEN  "WED"
    ELSE  IF weekday = 4 THEN  "THU"
      ELSE  IF weekday = 5 THEN  "FRI"
        ELSE  IF weekday = 6 THEN  "SAT"
          ELSE  "SUN"
```

However, a more efficient method is to select from a list of values, as shown here:

```
weekday OF
("MON", "TUE", "WED", "THU", "FRI", "SAT", "SUN")
```

where *weekday* is the index, "MON" is 1 . . . , and "SUN" is 7. Each item in the list must be of the same type and separated by commas.

Note

If the index points outside of the list, the resulting value of the list selection will be an index violation (see "Exception Values" in this chapter).

Related Sections

- IF THEN ELSE

Literals and System Values

Some data dictionaries, in particular the HP Application Dictionary support the definition of pre-defined constant values refered to as **System values** or **Literals**. In either case, a symbolic name is associated with a constant value stored in the data dictionary. Literals are always string constants, and System Values can be either a string constant or an integer. A Literal has a maximum length of 16, and a System Value can be up to 255 characters long.

You can reference a Literal or System Value as a Condition expression or a calculated item formula with the syntax LITERAL(*litname*) or SYSVAL(*sysvalname*), where LITERAL and SYSVAL are keywords and *litname* and *sysvalname* are the symbolic names stored in the data dictionary.

If you want to print out the value of a Literal or System Value in a report, you will need to create a calculated item whose formula references the LITERAL or SYSVAL keywords.

You can find out what Literals of System Values are available for your reports by using the BRWDLIST utility to return a listing of your data dictionary. Section 8 of the listing holds any available Literal values, and Section 9 holds any available System Values.

Because the data dictionary is not referenced at runtime (except to check the timestamp), the Literal and System Value values in a report are those available when the report was compiled. If you change the values, you will need to recompile the related reports.

Naming Conventions

Formula and condition names must follow HP ALLBASE/BRW item name rules. That is:

- They must start with a letter, $, # or @.
- The first letter can be followed by up to 19 characters.
- The name cannot contain points, parentheses, spaces or commas (letters, digits, and dashes are allowed).
- Items of the same type must have unique names.
- Items of different types (for example a layout calculated item and a Layout Function) can have identical names, but note:
  - Follow the rules for priority of arguments (see Arguments earlier in this section).
- No item having the same name as a reserved word can be used in a calculation. A list of reserved words is given later in this chapter.

Related Sections

- Arguments
- Conditions
- Formulas
- Reserved Words
- Standard Functions
- Types

Numbers

There are three number types in HP ALLBASE/BRW:

- Integer
- Fixed-point Numeric
- Floating-point Real

For each type you can:

- use all the arithmetic operators
- use the standard function DIGIT_OF

Additionally, you can use functions that only apply to Real numbers (square root, etc). Numeric and Integer types are covered first, and then Real numbers. It is more efficient to use Numeric types than Real types.

The range of each type of number is shown below.

Type	Range and Description
Fixed-point Numeric (N)	**+/- 9999999999999.999999** That is, 13 digits before the decimal point and 6 decimal places. The number of decimal places of a Numeric item is taken, as far as possible, from the number of places in the data dictionary definition.
Integer (I)	**+/-999999999** No decimals are allowed for Integer types.
Floating-point Real (R)	**Between 10 to the power -77 and 10 to the power 76** Floating-point Real is accurate to 16 decimal places.

Operations on Numbers

Operations on numbers can include arithmetic operations and functions. You can control calculation precision by selecting the appropriate item type (Numeric, Real, or Integer) and by changing the precision of the calculation result. You can change precision at several stages in a report.

Arithmetic Operators

Arithmetic operators include:

Operator	Description	Example
+	Addition	5 + 4 = 9
-	Subtraction	5 - 4 = 1
*	Multiplication	5 * 4 = 20
/	Division	5.0 / 4 = 1.25
**	Exponentiation	5 ** 4 = 625
MOD	Modulo	5 MOD 4 = 1

The following are some notes on arithmetic in HP ALLBASE/BRW:

- *Integer Division.* If both arguments are integers HP ALLBASE/BRW uses Integer division and the result is an Integer. Using Integer division, the result of the division (5 / 4) would be 1.
- *Real or Integer Exponentiation (**).* If you use the exponentiation function (**), HP ALLBASE/BRW chooses whether to use Real or Integer exponentiation depending on the values of the mantissa and exponent. If the mantissa is Real or if the exponent is not an Integer, Real computation is used. Otherwise Integer computation is used.

  For example:

2 ** 3 (=8)	Exponent and mantissa are both Integer, therefore Integer computation is used.
2 ** 3.5 (=11.3137)	The exponent not an Integer, therefore Real computation is used.
3.5 ** 2 (=12.25)	The mantissa is Real, therefore Real computation is used.

- *Modulo (MOD) Calculation.* The modulo (MOD) function is calculated according to the following formula:

  ```
  X mod Y = X - (Y * INT(X / Y))
  ```

  where INT(Z) returns the largest Integer less than or equal to Z.

  For example:

  ```
   13 MOD -2    = 13 - (-2 * INT(13 / -2))
                = 13 - (-2 * -7) = -1
  -13 MOD -2    = -13 - (-2 * INT(-13 / -2))
                = -13 - (-2 * 6) = -1
   13 MOD .5E2  = 13 - (.5E2 * INT(13 / .5E2))
                = 13 - (.5E2 * 0) = 13
  ```

```
13 MOD 2      = 13 - 2 * INT ((3 / 2 ))
              = 13 - 2 * 6 = 1
```

Precision of Numbers

Calculation precision in a calculation can be controlled by any of these options:

- selecting the appropriate item type (Numeric, Real, or Integer)
- optionally changing the precision of the calculation result
- changing the precision of arguments used in the calculation formula

You can change the calculation precision when:

- there is a result of a table calculated item, defined on the Table Calc Item screen
- an item is projected from a table via the Project Item screen
- there is a result of a layout calculated item, defined on the Layout Calc Item screen
- an item is used in a lineset, defined on the Define Lines screen
- there is a result of a column calculation, defined on the Define Lines screen
- an item is to be printed and is defined using scaling/decimals

All parts of the report performed after you set a precision will use the item with that precision.

Note

The screens that control calculation precision are listed alphabetically in Chapter 16.

Defining Precision

You define precision for an item by specifying a number and method code. The number is a power of 10, or 10th exponent, that indicates the precision. The method code indicates how superfluous digits are to be treated.

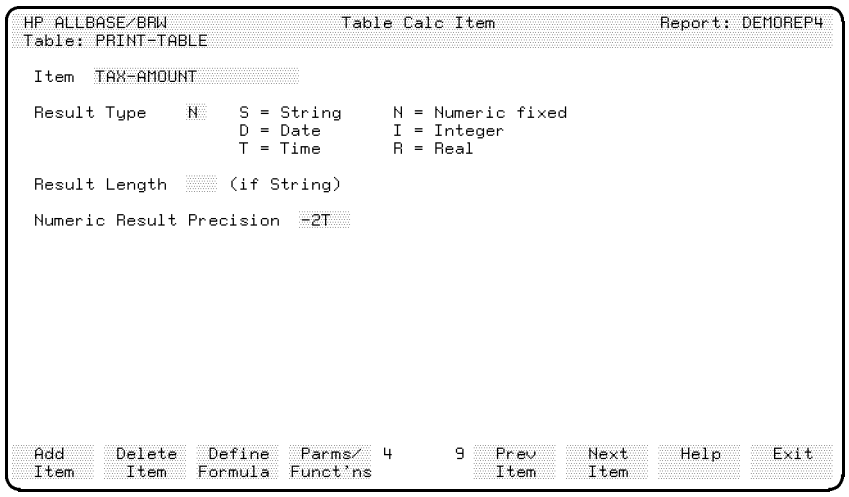
Valid range for Numeric (type N):	-6 through 12;
for Integer (type I):	0 through 8
for Real (type R):	-90 through 76.

Method Codes:	Digits are:
R	Rounded
T	Truncated
C	Ceiled

If you round a number, it is rounded up if the first digit outside your precision is a 5 or greater, and rounded down if the first digit outside your precision is less than 5. (You can choose, on the BRWSETUP program, whether you want 5 to round up or down.) If you ceil a number, it will always be rounded up. See the "Real Numbers" description in the "Numbers" section of this chapter for information on real numbers and precision.

Example

On the Table Calc Item screen below, the table calculated item TAX-AMOUNT is specified with a numeric precision of -2T.



Then, on the Table Calc Item Formula screen for TAX- AMOUNT, the following formula is entered:

```
AMOUNT * TAX-RATE
```

In this example, if the value of AMOUNT is $416.95, and the value of TAX-RATE is 0.07, then the calculation result $29.1865 is truncated to $29.18. Note that the precision applies only to the result. The computation is made at full precision, and the truncation performed AFTER the computation. Note also that, with a precision of -2T, item TAX-AMOUNT can never have more than two decimals.

The following table shows the value of TAX-AMOUNT (29.1865) with various precisions to go with the above example.

Had you specified a precision of:	the result would be:
-5C	29.18650
-4T	29.1865
-3T	29.186
-1T	29.1
-1R	29.2
0C	30
1R	30
1T	20
2T	0
2R	0
3R	0
3C	1000
6C	1000000

Digit Extraction Function

This function is used to extract single digits from a number. The function uses a digit index to determine which digit to extract. The index looks like this:

```
. . . 9  8  6  5  4  3  2  1  0     -1   -2  -3  -4 . . .
```

The blank between 0 and -1 corresponds to the decimal point. When you use this function in a calculation, you specify the number and the digit index like this:

```
DIGIT_OF (1024.38, 1)
```

where 1 is the digit index.

In this example, the index is applied to the number as shown here:

```
                1  0  2  4  .   3   8
                ↓  ↓  ↓  ↓  ↓   ↓   ↓
. . .6  5  4  3  2  1  0     −1   −2  −3  −4 . . .
```

The digit index specifies the 10th exponent (in this case, 2) for the number 1024.38.

The digit extraction function results in an integer-type item.

Related Sections

- Calculation Language
- Operators/Operations
- Types

Real Numbers

This section describes features, functions, and operations that apply to Real numbers only. You can use Real numbers the same way as other number types. For example, you can define file or database items, HP ALLBASE/BRW parameters, or calculated items of type R (Real).

When to Use Real Numbers

Unless a Real number is required, it is more efficient to use a Numeric number.

Take care when using Real numbers. Real numbers are externally represented in powers of 10 and must be converted to an internal format where they are represented in powers of 2. This conversion can cause inconsistencies, for example, when comparing an external Real number such as a parameter with an internal value read from a file. Two ostensibly identical values may not be equal.

Real Items and Numeric Items

The HP ALLBASE/BRW type N (Numeric) allows a maximum of 19 digits. Numeric items are commonly used for numbers with 13 digits left of the decimal and 6 digits right of the decimal. Real number processing requires more CPU time than Numeric. Therefore, unless Real values or Real functions are required, it is better to use a Numeric item.

How HP ALLBASE/BRW Holds Real Numbers

HP Dictionary/V floating point types E and R, and *HP System Dictionary* elements with attribute element-type E or R, convert to HP ALLBASE/BRW type R. HP ALLBASE/BRW uses the Long Real format internally so Real values are converted to Long Real. HP ALLBASE/BRW can convert Real numbers whether or not they are held in IEEE or HP 3000 format.

How to Write Real Numbers in HP ALLBASE/BRW Formulas

In HP ALLBASE/BRW formulas, such as for calculated items or functions, you can express Real values in Integer, fixed point, or scientific notation formats. The following are legal Real values in HP ALLBASE/BRW formulas:

```
247              Integer format
12.36            fixed point format
3.45E2           scientific notation
-.56E27          scientific notation
23.45E-6         scientific notation
```

Values expressed using scientific notation are subject to the following conditions:

- A value must all be on one line.

- There must be no embedded spaces in the value.
- Signs can optionally occur at the beginning of the value and/or following the exponent (E).
- The exponent must not have more than 2 digits.

How to Write Real Numbers in HP ALLBASE/BRW Parameters

You must use a set syntax for parameter values so that the SATISFIES selection condition will work with comparison predicate parameters. Default and run-time parameter values obey the following syntax rules:

- They can have an optional sign.
- They must have a decimal point.
- They must have either a non-zero digit optionally followed by up to 15 additional digits, or, for a zero value, a zero followed by up to 16 zero digits. That is, if the first digit is zero, the Real number must also be zero.
- They must contain the letter E.
- They must have an exponent sign.
- They must have a 1 or 2 digit exponent.

Correct Real Parameters

The following are correct Real parameters:

```
.12345E+3
-.45678E-55
+.654E+0
.0E+0
```

The following are incorrect Real parameters:

2.34E+3 :	A digit must not appear before the decimal point.
.34 E-3 :	Embedded blanks are not allowed.
.6789E :	The exponent must have a value.
.478E3 :	The exponent must have a sign.
.789E+109 :	The exponent cannot have more than two digits.
.123456789012345678E+2 :	The mantissa cannot have more than 16 digits.
.01234E-4 :	The leading digit is a zero and the value is not zero.

Real Numeric Precision

You can set the Numeric precision for Real items in the same way as you can for Numeric and Integer items. Set Numeric precision for Real items by entering a number from -90 to 76 and a letter T, R, or C. The number specifies the degree of precision and the letter specifies whether to truncate, round, or ceil the value. The degree of precision is based on how the Real number looks in Fixed Point form. For example:

Real Value	Fixed Point	NumPrec	New Real Value
.12347E2	12.347	-2R	.1235E2
.76543E2	76.543	2R	.1E3
.12347E4	1234.7	0T	.1234E4

Real Item Functions and Operations

You can use Real items in all the operations and functions currently available for Numeric and Integer items. Some new functions are also available. These functions are available for use with Real numbers ONLY. The functions that are only available with Real numbers are:

SIN, COS, TAN, ARCTAN, ARCSIN, ARCCOS, LN (natural logarithm), LOG (base 10 logarithm), E, SQR, SQRT, and ABS.

Calculated items and functions using these Real functions must have a result of type Real.

Real Item Edits

As for other item types, you can set up default item edits for Reals using the program BRWSETUP.PUB.SYS, and you can modify the item edit for each item in a report. For more information about BRWSETUP, see Chapter 17, "Configuration and Security."

Related Sections

- Formulas
- Parameters
- Standard Deviation
- Types

Operators/ Operations

The term *operator* is used in the normal mathematical sense, that is, an operator is a mathematical or logical symbol that specifies an operation to be performed on an argument or arguments. For instance, in the example below:

```
PRICE * 10
```

The asterisk (*) is an operator that tells HP ALLBASE/BRW to multiply the PRICE by 10.

Operations which you can use with HP ALLBASE/BRW can affect all number types (Integer, Real, and Numeric), real numbers only, strings, dates, and times. You can use IF THEN ELSE constructions, Boolean operations (conditions), list operations (selecting from a list of values), and aggregate (array) operations.

The description of these operators and their precedence in calculations is shown in the following table:

Operators Table

Precedence	Operator	Description
1	(*nnn*)	Anything in parenthesis
2	**	Exponentiation
3	* , / , MOD	Multiplication, division, or modulo
4	+ , - , &	Addition or subtraction
5	= , <> , <= , > , >= , IN , MATCHES , SATISFIES , OF	Comparison operators
6	NOT	Boolean operator
7	AND	Boolean operator
8	OR	Boolean operator

Related Sections

- Condition/Boolean Operations
- IF THEN ELSE Operations
- Numbers
- Types

Parameters

Parameters are arguments which get their values when the report is run (for example, you can supply values for parameters on the Request Report screen).

Characteristics

Parameters have these characteristics:

- They can be of any HP ALLBASE/BRW type: Real, Integer, Numeric, String, Date, or Time.
- You supply a value or values of that type when you run the report. See the "Request Report screen" section of Chapter 16 for details about assigning parameter values when the report is run.
- Parameters must be specified on the Define Selection screen so that the user can enter values when requesting the report. See Chapter 16, for the screen description.
- Parameters can be used in a Selection Condition, Relation Condition, or Relation to select data when a report is run.
- Parameters are referenced by the parameter name preceded by a question mark, for example:

```
?month
```

- Parameter names cannot exceed 19 characters, may start with a letter, $, # or @, and can then be followed by up to 18 letters or numbers, and cannot be the same as a reserved word. A list of reserved words is given later in this chapter.
- Parameters can be used anywhere within the report.
- You can set default values for parameters on the Define Selection screen.
- You can specify that a value MUST be given for a parameter, and that strings will be shifted to upper-case.
- To use a parameter, you must compile the report with a Selection Set.
- Parameters can be one of three different categories: Comparison Predicate, Single Value, and List of Values.

Note 

Real number parameters have special syntax, described under Real number syntax in the "Numbers" section.

Example

You can use a parameter to obtain different results from the same compiled report. For example, you could define a calculated item that calculates the amount of sales tax to be added to an order. The calculation might appear:

```
((PRICE * QUANTITY) * 10) / 100)
```

for a 10 percent sales tax. However, if you define a parameter TAX-RATE, and change the formula to:

```
((PRICE * QUANTITY) * ?TAX-RATE) / 100
```

You could run the report many times, supplying a different value for TAX-RATE each time. Parameters can be used in HP ALLBASE/BRW to do the following:

- Perform calculations.
- Control report format, for example, using suppress line conditions.
- Control reported data, using a parameter to restrict data written to a table.
- Optimize data selection, using a parameter to supply values for keyed access to a table.

Single Value Parameters

Single value parameters are parameters that can be substituted by a single value only. For example:

```
FOREIGN-AMOUNT * ?EXCHANGE-RATE
```

At run-time, the single-value parameter EXCHANGE-RATE will be substituted by whatever value you supply for the parameter when running the report. EXCHANGE-RATE can have only one value.

Characteristics

Single value parameters have these characteristics:

- They have only one value.
- They can be used in calculations with these operators:

  (<, >, <>, =, <=, >=), and with all logical operators.
- You can use a single-value parameter like a normal item in calculations or printing. See Chapter 8, "Report Layout" for details about printing items.

Note

You should specify single value parameters as *required* parameters when you define them. Otherwise, a missing value will set the parameter to NO_VALUE. See Exception Handling later in this chapter.

List-of-Values Parameters

List-of-values parameters can be substituted by a list of one or more items of the same type.

Characteristics

List-of-values parameters have these characteristics:

- The values you substitute for the parameter must be of the same type. You cannot have both Real and String values in the same list.
- A list of values can have any number of values.
- You cannot use values in a list-of-values parameter as items.
- Each value in the list must be stated correctly (strings in double quote marks, and so on) and separated by commas. See the description of each type in the "Constants as Arguments" and the "Numbers" sections of this chapter for a description of how to state each type.
- A list-of-values parameter can only be used with the operator IN. This operator returns a Boolean value depending upon whether one item is in a list of other items.

Example

Suppose you want to use a list-of-values parameter to return a value for CODE when that value is included in your list-of-values for the ?CODE-LIST parameter. You define the operation like this:

```
CODE IN ?CODE-LIST
```

The value returned is true if CODE is equal to any of the list of values in the parameter ?CODE-LIST. Note that if you do not supply a value for ?CODE-LIST, CODE can never be equal to it.

Comparison Predicate

Comparison predicate parameters can be substituted by a value and an operator, for example > 50.

Characteristics

Comparison predicate parameters have these characteristics

- They must be used only with the operator SATISFIES.
- They allow the operator and value to be specified at execution time providing selection of the following:
  - one or more explicit values
  - all except one or more explicit values
  - values which match specific patterns
  - all values out of a range of values
  - values less than a specific value

- □ values less than or equal to a specific value
- □ values greater than a specific value
- □ values greater than or equal to a specific value
- □ all values (when parameter value is missing)

■ They allow the use of the following masks in parameter values:

@	zero or more occurrences of any character
?	one occurrence of any character

The following table shows how comparison predicate parameters work with operators.

Comparison Predicate Execution Time Values

Operator	Value	Items Selected
(Blank)	(Blank)	Unrestricted
(Blank)	List of Values	Item must match values/masks
=	List of Values	Items must match values/masks
< >	List of Values	Item may not match values/masks
>	Single Value	Item must be greater than value
<	Single Value	Item must be less than value
>=	Single Value	Item may not be less than value
<=	Single Value	Item may not be greater than value
R	Two Single Values	Item in the range of values (inclusive)

Examples

Comparison Predicate Value	Results
`COMPANY =0010`	Reports only Company # 0010
`ACCOUNT =41@`	Reports all accounts starting with 41
`DUE-DATE <03/22/82`	Reports all Due-Dates earlier than 03/22/82

Operator SATISFIES

Comparison Predicate parameters must be used with the operator SATISFIES. A comparison predicate returns a Boolean result if the left-hand argument satisfies the comparison predicate parameter. The following example shows a comparison predicate parameter used in a report selection condition:

```
CUSTOMER-NO SATISFIES ?CUSTOMER
```

CUSTOMER-NO is an item (the left-hand argument in the condition); SATISFIES is the operator, and ?CUSTOMER is the comparison predicate parameter. If you substitute > 1000 for parameter ?CUSTOMER when you run the report, the report would only select those customers whose CUSTOMER-NO was greater than 1000. That is, the result of the condition would be true whenever CUSTOMER-NO was less than 1000.

Special Cases When Using Parameters

You can use parameters in a selection condition to select data for a report.

You can also use single-value and list-of-values parameters to allow keyed access to the first source table. See Chapter 6, Defining Data Access, under the section Keyed Access to First Source Table for an example of using parameters in this way.

Chapter 19, "Performance Considerations", includes further examples of access methods and tuning access to source tables.

Default Values For Parameters

You can create a calculated item which provides a default value if a single-value parameter has no value. For example, the calculated item MONTH uses the current month as default value if the parameter ?MONTH has no value.

```
IF ?month = NO_VALUE
THEN
  month_of (date)        <<current month>>
ELSE
  ?month
```

Printing Values As Bar Graphs

You can print numeric values as horizontal bar graphs with the calculated item shown in this example:

```
SUBSTRING ("**************************",1,amount)
```

Reserved Words

Reserved words are words that have a specific meaning in HP ALLBASE/BRW and cannot be used as item names and parameter names.

The following are reserved words and cannot be used as item names:

AND	MAX
AVG	MIN
COUNT	MOD
DIVIDE_BY_ZERO	NOT
ELSE	NO_VALUE
ERROR_VALUE	OF
IF	OR
IN	OVERFLOW
INDEX_MAX	SATISFIES
INDEX_MIN	SUM
INDEXVIOLATION	SYSVAL
INVALID_VALUE	THEN
LITERAL	TO
MATCHES	

The following names are reserved words and cannot be used as parameter names:

$REPORT	$TELLREPORT
$USER	$USERPASS
$ACCT	$ACCTPASS
$GROUP	$GROUPPASS
$SCHEDULE	$ACTUAL-PRINTFILE
$FORMAL-PRINTFILE	$DEVICE
$OUTPRI	$NBR-COPIES
$CCTL	$INPRI
$ENVIRONMENT	

Standard Deviation

A common computation in a report is to compute a Standard Deviation. You can compute a Standard Deviation for any HP ALLBASE/BRW numeric type.

Example

For example, the Standard Deviation for a Real item X that occurs in a report detail can be computed as described in the following steps.

To compute the Standard Deviation of a Real item X

1. Define a Real layout calculated item XSQUARE with this formula:

   ```
   sqr(X)
   ```

2. Define an Integer layout calculated item XCOUNTER which will be used to count the number of details in a break. The formula for this item is unimportant, as it is only to be used with a ColCalc value COUNT, to count the number of details.

   Use a 'dummy' formula such as:

   ```
   1
   ```

3. Define a Real layout calculated item STDEV with the formula:

   ```
   sqrt((XSQUARE - sqr(X) / XCOUNTER) / (XCOUNTER - 1))
   ```

4. Define these items on the break footer lineset:

   ```
   X             TOTAL
   XSQUARE       TOTAL
   XCOUNTER      COUNT
   STDEV         CALC
   ```

   This will compute the Standard Deviation for the Real item X within the break.

Related Sections

- Calculations
- Formulas
- Numbers

Standard Functions

HP ALLBASE/BRW includes many built-in functions. These are listed in the Standard Functions table, and each is described on the following pages.

Standard Functions Table

Type	Function	Format	Explanation
DATE	ADD_DAYS ADD_WEEKS ADD_MONTHS ADD_YEARS	(date1, int)	Returns date which is date1 plus int days, weeks, months or years.
	SUB_DAYS SUB_WEEKS SUB_MONTHS SUB_YEARS	(date1, int)	Returns date which is date1 minus int days, weeks, months, or years.
TIME	ADD_HOURS ADD_MINUTES ADD_SECONDS	(time1, int)	Returns time1 plus int hours, minutes or seconds.
	SUB_HOURS SUB_MINUTES SUB_SECONDS	(time1, int)	Returns time1 minus int hours, minutes or seconds.
I	DAY_OF_WEEK DAY_OF_MONTH DAY_OF_YEAR	(date)	Returns integer value of day (1 ... 6, 1 ... 31, 1 ... 366).
I	WEEK_OF	(date)	Returns integer value of week.
	MONTH_OF	(date)	Returns integer value of month.
	YEAR_OF	(date)	Returns 4 digit year value (0000 ... 9999).
I	HOUR_OF MINUTE_OF SECOND_OF	(time)	Returns integer value of hour, minutes, or seconds (0 ... 23, 0 ... 59).
I	DAYS_DIFF WEEKS_DIFF MONTHS_DIFF YEARS_DIFF	(date1, date2)	Returns number of days, weeks, months, or years between date1 and date2.
I	HOURS_DIFF MINUTES_DIFF SECONDS_DIFF	(time1, time2)	Returns integer value of difference between time1 and time2 in hours, minutes, or seconds.
I	TO_INTEGER	(number) (string)	Returns an integer equivalent to number Returns an integer equivalent to string (string must be a numeric string)

Standard Functions Table (continued)

Type	Function	Format	Explanation
REAL	TO_REAL	(number) (string)	Returns a real number equivalent to number Returns a real number equivalent to string (string must be a numeric string)
REAL	ABS	(number)	Returns absolute value of number
	ARCTAN		Returns arc tangent of number
	ARCCOS		Returns arc cosine of number
	COS		Returns cosine of number
	EXP		Returns exponent to the power of number
	LN		Returns natural logarithm of number
	LOG		Returns base 10 logarithm of number
	SIN		Returns sine of number
	SQR		Returns square of number
	SQRT		Returns square root of number
	TAN		Returns tangent of number
NUMERIC	TO_NUMERIC	(number)	Returns a Numeric number equivalent to number
		(string)	Returns a Numeric number equivalent to string (string must be a numeric string)
STRING	TO_STRING	(number)	Returns a string equivalent to number

Related Sections

- Arguments
- Conversion Functions
- Functions
- Types

Standard Items

Standard items are items supplied with HP ALLBASE/BRW. They are listed in the following table.

Standard Items Table

Name	Type[1]	Length	Definition
DATE	D		Current date
TIME	T		Current time, when execution starts
PAGE[2]	I		Current page
LINE[2]	I		Current line on page
USER	S	8	User executing the report
REPORT-DESCRIPT	S	50	Report description
REPORT-EXEC-FILE	S	35	Report execution file
REPORT-SPEC-FILE	S	35	Specification file name
SELECTION-SET[2]	S	16	Selection set file name
REQUEST-DATA1[2] ... REQUEST-DATA50[2]	S	80	Parameter name and actual values used to execute the report. REQUEST-DATA1 represents the first parameter on the Request Report screen, REQUEST-DATA2 the second, and so on. If less than *n* parameters are specified, REQUEST-DATA*n* will show blanks.
The following standard items are for advanced formatting control. For example, they can be used to indicate which linesets are printed before and after the page heading or footing.			
BREAK-LEVEL[2]	I		Level of last break (1-9); zero before the first break footings are printed.
LINESET-LEVEL[2]	I		Level of current lineset: 0 = report level; 1-9 = break levels; 10 = detail. Remains unchanged if page heading or footing is printed.
PREV-LSET-LEVEL[2]	I		Same as above, but shows level of previous lineset.
LINESET-TYPE[2]	I		Type of current lineset: 0 = detail; 1 = heading; 2 = footing. Remains unchanged if page heading or footing is printed.
PREV-LSET-TYPE[2]	I		Same as above, but shows type of previous lineset.

1 Types are as follows: D = date; T = time; I = integer; S = string.

2 These are not available for table calculated items.

Strings

A string is a series of characters enclosed in double quote marks, For example:

```
"This is a string"
```

Strings can be up to 255 characters long.

String Operations

String operations are used to concatenate string values or to extract a subset out of a string value. During a calculation, string length is unlimited; however, you must specify a string length for the result of the calculated item (maximum length 255 characters). Trailing blanks in strings are ignored when they are used in string calculations. Trailing blanks in constants are not ignored.

String Concatenations

String concatenations look like this example:

```
customer-name & " in " & city & "," & state
```

Where *customer-name* is Regular Oil Industries, *city* is Dallas and *state* is Texas, the result is:

```
"Regular Oil Industries in Dallas, Texas"
```

Note that blank characters must be specified explicitly. (HP ALLBASE/BRW ignores trailing blanks in string items.)

String Extraction Function

This function extracts a subset out of a string value, as shown in the following three examples.

Example 1

```
SUBSTRING ("Whiskey", 2, 3)
```

In this example, “Whiskey” is the source string, 2 the position from which the extraction starts, and 3 the length of the extracted string. The result of the extraction is:

```
"his"
```

Example 2

```
SUBSTRING ("Whiskey", 10, 2)
```

In this example, “Whiskey” is the source string, 10 the position from which the extraction starts, and 2 the length of the extracted string. The result is that blanks are returned, as shown here:

```
" "
```

Blanks are returned if the start position is greater than the string length.

Example 3

```
SUBSTRING ("Whiskey", -3, 4)
```

In this example, “Whiskey” is the source string, -3 the position from which the extraction starts, and 4 the length of the extracted string. The result is:

```
"Whis"
```

Note that if a position less than 1 is specified, position 1 is used.

Using double quote marks (”) in Strings

You can use double quote marks in strings with HP ALLBASE/BRW version A.00.10 and later. If you want to specify the string ABC”D, define a string calculated item ABCD, with formula:

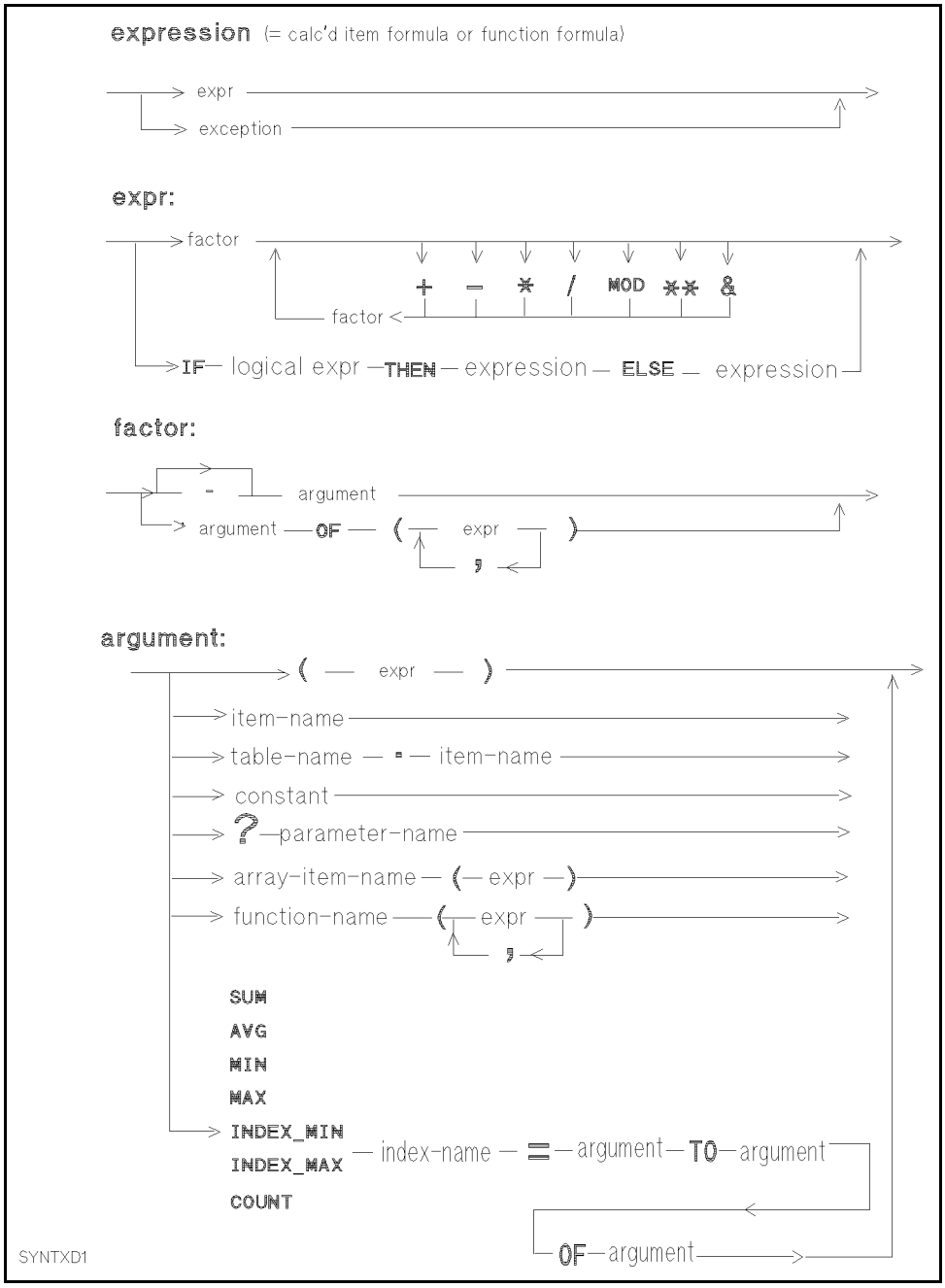
```
"ABC""D"
```

Then specify item ABCD when you want to print ABC”D. Using two double quote marks in a string calculated item will print one set of double quote marks.

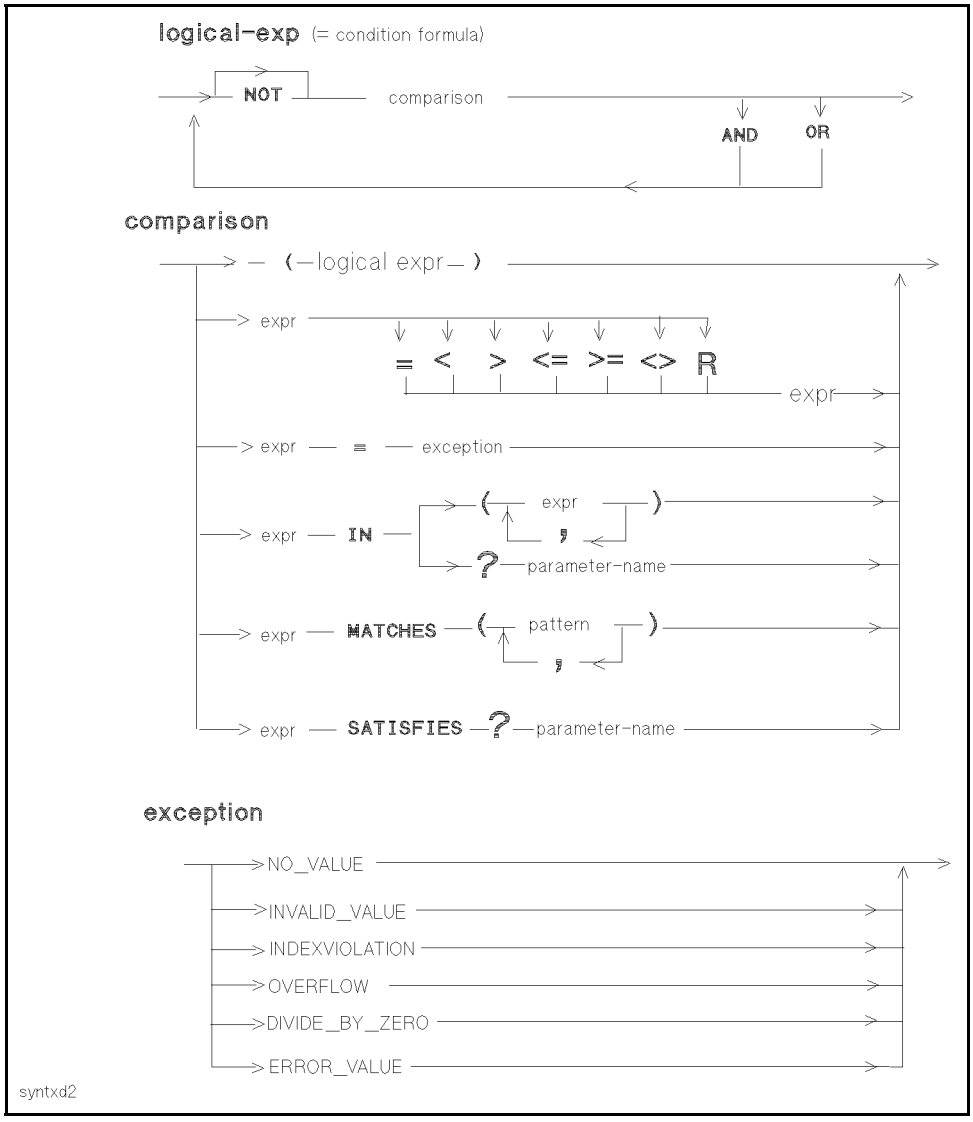
Syntax Diagrams

Syntax diagrams provide you with the complete logic, elements, and structure of the Calculation Language. The diagrams are made up of three related parts:

- **Diagram 1** shows expressions, including calculated items and functions.
- **Diagram 2** shows the conditions (or **logical-expressions**).
- **Diagram 3** shows the primaries of the first two diagrams, that is, names and constants.



Syntax Diagram 1: Expressions



Syntax Diagram 2: Conditions/Logical Expressions

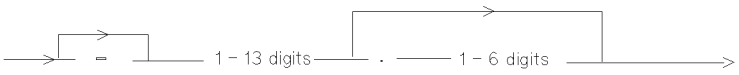
Note that Diagram 2 uses expressions defined in Diagram 1, and IF expressions in Diagram 1 use conditions in Diagram 2.

name (any)

- can consist of up to 20 characters
- must begin with a letter, $, #, or @
- can contain all characters except , . ( ) and blank

Examples: CUSTOMER
SALES-PER-AREA
VARIANCE-%

numeric constant

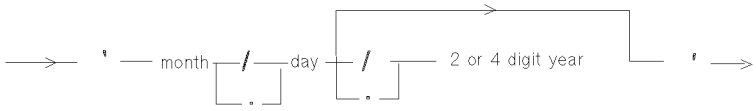


Examples: 49483.61
0.2443
-109

string constant

Example: "I like HP business Report Writer"

date constant



Examples: '4/16/1986' or '4.16.1986'
'12/31/85' '12.31.85'
'7/20' '7.20'

Note: Date formats are controlled by BRWSETUP Alternate formats can be dmy or ymd

time constant

Examples: 21:06:37
4:16

pattern

- is a constant with optional mask characters
- is always enclosed in double quotes
- ? is mask for 1 character
- @ is mask for 0 - n characters

Examples: "ab@cd??" string pattern
"14.@" numeric pattern
"1/@/85" date pattern
"14:@" time pattern

Syntax Diagram 3: Primaries for Diagrams 1 and 2

Syntax Rules For Calculations

The following general rules apply when defining calculations for all types of HP ALLBASE/BRW formulas and conditions. Note that all formulas and conditions are examined by HP ALLBASE/BRW after you enter them, so that misspellings, type conflicts, and other errors will be immediately found.

Rules For Formulas and Conditions

- No item with the same name as an HP ALLBASE/BRW reserved word can be used in a calculation. A list of HP ALLBASE/BRW reserved words is supplied later in this chapter.
- An item, calculated item, and function can all have the same name, but if so HP ALLBASE/BRW uses a priority to determine to which item the name refers. (See "Arguments" section in this chapter. However, no two items, or no two functions can have the same name.
- Items, calculated items, functions, and arrays must all follow the HP ALLBASE/BRW rules for names. That is, they cannot exceed 20 characters; must start with a letter, $, # or @; they can then be followed by up to 19 characters; and they cannot contain points, parentheses, spaces or commas. (Letters, digits, and dashes are allowed).
- A calculation, whether a formula or condition, can be up to two screen pages long. You can have calculations that spread over more than two pages by defining part of the calculation as a function (or by putting part of the calculation in a separate calculated item) and then calling it from within the calculation.
- You can break calculations over more than one line. You can insert new lines or comments anywhere on a calculation screen except in the middle of names and in strings.

- Argument names and reserved words can be written in any combination of uppercase and lowercase characters. That is, except in strings, HP ALLBASE/BRW will not differentiate between and uppercase and lowercase characters.
- You must leave at least one blank space between operators and their arguments. (HP ALLBASE/BRW lets you include characters such as −, /, and & in the middle of names).
- You can add comments between double angle-brackets: for example: <<comments>>.
- You can use all the terminal editing keys to edit a calculation.
- If you cannnot immediately correct an error in a formula, press the function key `Exit  Save`, which is displayed on all Formula and Condition screens if the system detects an error in the formula. HP ALLBASE/BRW saves the formula unchanged for later correction.

Time

You can manipulate times using HP ALLBASE/BRW Time types. Items of type Time are defined in the data dictionary. (See Chapter 18 for the data dictionary definitions of Time types). A time always consists of hours, minutes, and seconds. Time items can be output in many different ways, depending on the edit-mask (for output) and the data dictionary definition (for input). HP ALLBASE/BRW converts all times into packed decimal numbers with 19 digits, including 6 decimal places. You specify times using colons, as below:

```
8:24:37            <<8 o'clock, 24 minutes, 37 seconds>>
10:4               <<4 minutes past 10 o'clock>>
0:14:50            <<14 minutes and 50 seconds>>
00:00:00 or 0:0    <<midnight >>
23:59:59           <<one second before midnight>>
24:00              <<invalid>>
```

Time Operations

You can manipulate times using standard functions.

Time Extraction Functions

Time extraction functions extract parts of a time value and are specified as shown here:

```
HOUR_OF        (time)
MINUTES_OF     (time)
SECOND_OF      (time)
```

They result in integer-type items.

Time Duration Functions

Time duration functions calculate the difference between two time values. These are specified as:

```
HOURS_DIFF     (time1, time2)
MINUTES_DIFF   (time1, time2)
SECONDS_DIFF   (time1, time2)
```

They result in integer-type items.

If the duration is not an integral number of hours or minutes, the difference is rounded to the nearest integral number.

Time Addition Functions

Time addition functions calculate a time value by adding hours, minutes or seconds to a time value and are specified as:

```
ADD_HOURS      (time1, time2)
ADD_MINUTES    (time1, time2)
ADD_SECONDS    (time1, time2)
```

They result in time-type items.

Time Subtraction Functions

Time subtraction functions calculate a time value by subtracting hours, minutes, or seconds from a time value and are specified as:

```
SUB_HOURS      (time1, time2)
SUB_MINUTES    (time1, time2)
SUB_SECONDS    (time1, time2)
```

They result in time-type items.

Note that out-of-range times are corrected automatically, for example:

```
ADD_MINUTES (23:55,10)     results in 0:05
SUB_MINUTES (0:05,10)      results in 23:55
```

Types

This section describes each type of argument in HP ALLBASE/BRW, (Real, Integer, Numeric, String, Date, or Time), and the operations that can be performed upon each type. It describes the standard functions that are supplied with HP ALLBASE/BRW, and also covers exception handling and exception conditions.

Summary of Types

HP ALLBASE/BRW arguments and results can be one of the following types:

N - Fixed-point Numeric	*all numbers*
R - Floating-point Real I - Integer S - String	*characters*
D - Date	*day, month, and year*
T - Time	*hours, minutes, and seconds*

Also, you can manipulate arrays of these types.

Type Mixing

You can use items of all three number types in the same calculation. That is, you can multiply Reals by Integers, Numerics by Reals, etc. The type of arithmetic performed, and thus the type of the result, follows the rules below.

- If both arguments are Integers, integer arithmetic is used and the result is an Integer
- If one argument is an Integer, and the other Numeric, or both arguments are Numeric, Numeric (fixed-point) arithmetic is used and the result is Numeric.
- If either or both arguments are Real, Real computation is used and the result is Real.

  For example:

```
PRICE * QUANTITY
```

  where PRICE is a fixed-point Numeric item and QUANTITY is an Integer.

  The calculation is done with fixed point computation, as one of the arguments is a fixed-point (Numeric) value. The result is automatically converted to a fixed-point Numeric value. Note that the type of item QUANTITY is not altered.

Type Conversion

HP ALLBASE/BRW automatically converts numbers from type to type. That is, if you assign an Integer value to a Numeric or Real item, the Integer is converted to the approriate type.

For example, the index of an array item and the modulo divisor must be specified as integers. If you assign a Real or Numeric value to the modulo divisor or array index, HP ALLBASE/BRW automatically does a type conversion, and truncates superfluous decimals. (That is, no error is returned if you assign a value of a number type to an item or a different number type.)

Type Conversion with Output Files

If an item of type Real, Integer, or Numeric is written to a field in a dictionary output file that is defined to be of a different type of number, HP ALLBASE/BRW converts the value to the type defined in field of the dictionary output file. This can result in an overflow, in which case an output exception value, as defined by the BRWSETUP program, is written to the dictionary output file instead.

Automatic type conversion operates only on number types. For example, you cannot assign a Date type to a String item!

Related Sections

- Arguments
- Exception Handling and Exception Conditions
- Items
- Standard Functions

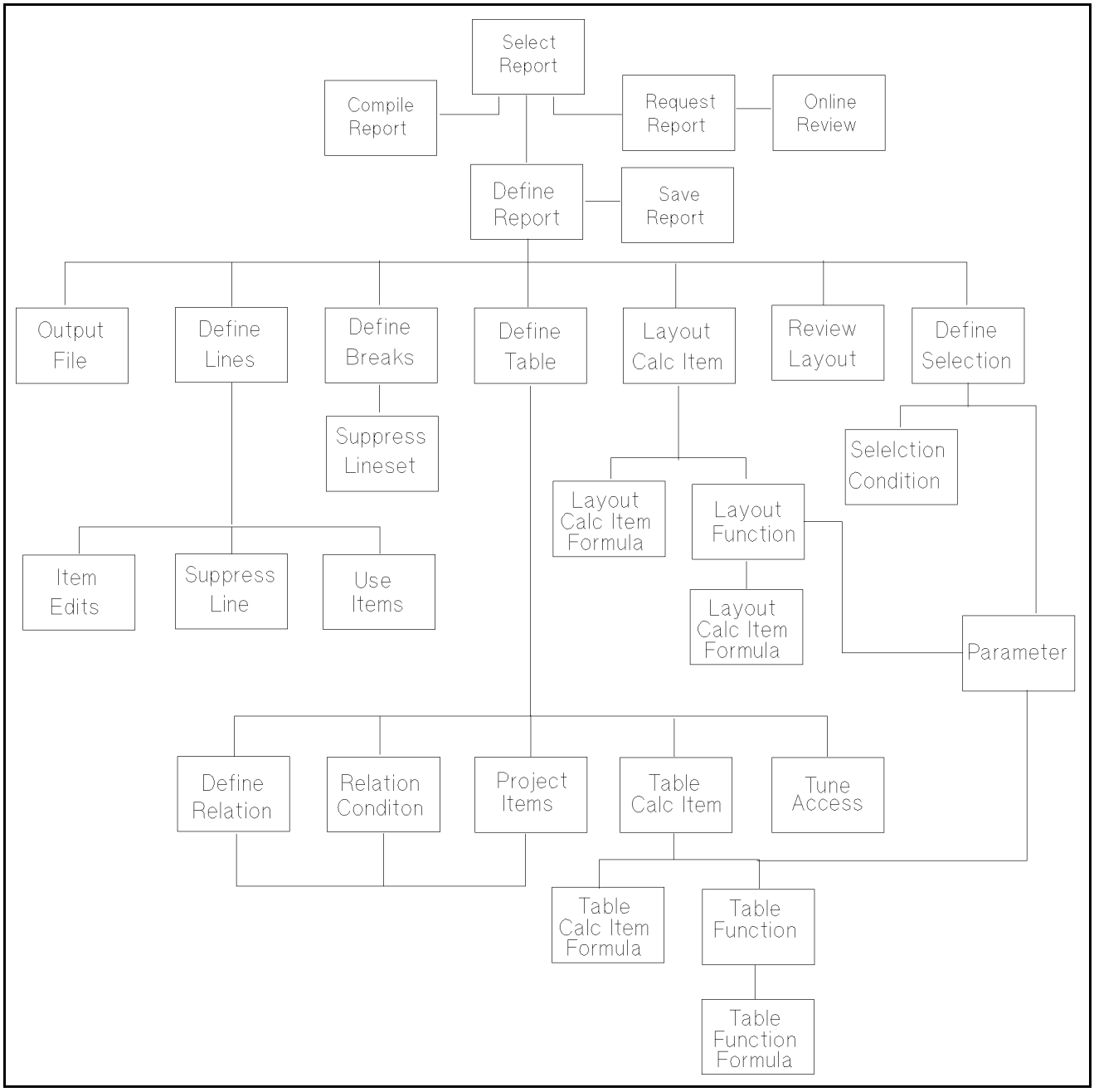
16

Using HP ALLBASE/BRW Screens

Alphabetical reference for HP ALLBASE/BRW screens

At a Glance

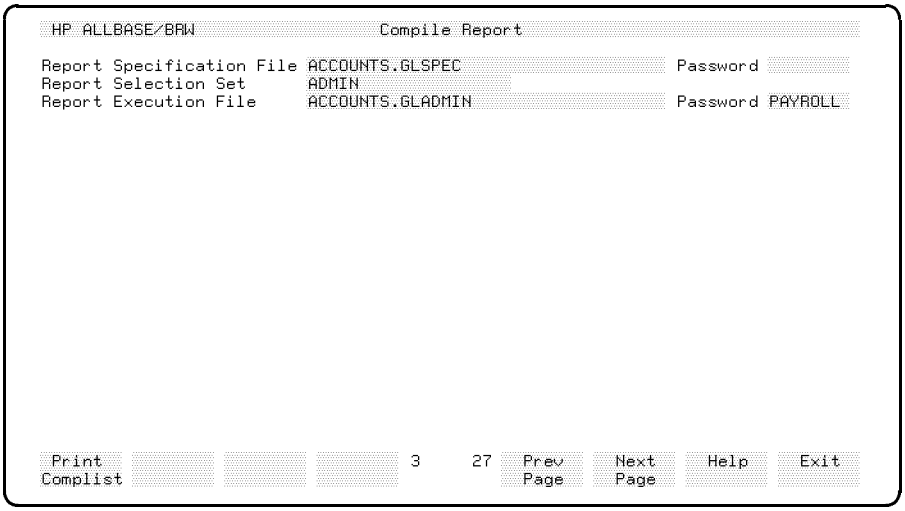
Use this screen hierarchy to see how you get to a particular screen.



Screen Hierarchy

Compile Report Screen

Use this screen to compile a report.



HP ALLBASE/BRW checks that specifications have been correctly entered and displays them on this screen. After the compilation, HP ALLBASE/BRW displays the compilation listing on this screen.

To compile a report

1. Make sure that you have saved the latest specifications for the report you want to compile.
2. Choose `Compile Report` from the Select Report screen to get to the Compile Report screen. The name of the report specification file for the report you selected appears in the `Report Specification File` field.
3. Type the name of the selection set, the execution file, and passwords (if required) for the report. Use the information in the Field Description section if you need more details about each field.
4. Press Enter.

After the report has compiled, the first page of the compile listing is displayed on the screen.

Note

If you attempt to compile a report that is currently loaded, but has never been saved, HP ALLBASE/BRW automatically saves the report specification file before compiling the report. However, the compile operates on the last saved copy of the report specification. So, if you have a report loaded that has been *previously* saved and you select to compile the report *without* saving it again (using `Save Report`), you will be compiling the previously saved version. The compilation will not include current, unsaved changes.

To review the compile listing

1. Choose `Next/Prev Page` to review the compile listing online.

To print the compiled listing

1. Choose `Print Listing` to print a hard copy of the compile listing.
2. At the prompt, type the printer ID and press Enter or choose `Print Listing` again.

Field Descriptions

`Report Specification File` The report specification file, including the group name and account name if the report is not located in the log on group.

`Password` **Optional**. The file lockword for this specification file.

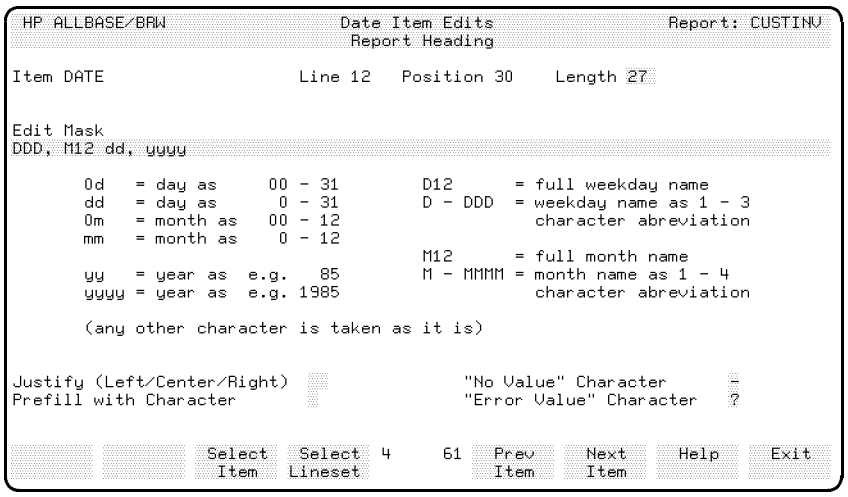
`Selection Set` **Optional**. The name of the selection set that contains print specifications and other customized information for this compilation. If this is left blank, the default print specifications are used.

`Report Execution File` The name of the report execution file that will be created when the report is compiled. The name can include the group name and account name if the report is not located in the logon group.

`Password` **Optional**. The file lockword for this execution file.

Date Item Edits Screens

Use this screen to specify how date items are to be printed in the report.



Anytime you specify a DATE item on the Define Lines screen, you can go to the Date Item Edits screen to edit the way the date appears in the report.

To edit a DATE item

1. Define a line in the report that contains a DATE item.
2. Choose `Item Edits` from the Define Lines screen where the DATE item is defined.
3. At the `Edit-mask for item` prompt, type the name of the DATE item and press Enter.
4. Type the specifications for the edit mask in the `Edit Mask` field.
5. Type other editing specifications in the field provided.
6. When the specifications are completed, press Enter.

Use the descriptions in the Field Descriptions section to understand the edit mask codes and the other fields you can use. See the Softkeys section for information on what the softkeys do.

To select a different item

1. Choose `Select Item` to select a different item to edit.
2. At the `Edit-mask for item` prompt, type the name of the item you want to edit and press Enter.

   If you don't know the name of the item, you can scroll though all the editable item names in the lineset by choosing `Prev Item` or `Next Item`.

To select a different lineset

1. Choose `Select Lineset` to select a different lineset.
2. At the prompt, type the lineset type (D for detail, P for page, R for report, or a break level number) and type H for heading or F for footing if applicable.
3. Press Enter.

Banner Line

The banner line (at the top of the screen) shows in which lineset the current item is positioned.

Field Descriptions

`Item` **Display**. The current item.

`Line/position` **Display**. The line and print field position of the item.

`Length` **Optional**. The print field length defined for this item on the Define Lines screen. You can optionally change the length for this item.

`Edit Mask` Enter the edit mask using the codes displayed on the screen for the date variables. Note the following:

- Code characters must be entered in upper/lower case as indicated on the screen.
- D12 and M12 (see screen) can have up to 12 characters each.
- M - MMMM and D - DDD let you specify the number of letters in the abbreviations.

`Justify` **Optional**. Use one of these codes:

L	left justify
C	center
R	right justify

Default is `L` for left-justify.

`Prefill With Character` **Optional**. The character to be printed instead of leading blanks before the item value. The default is leading blanks.

"No Value" Character

Optional. The print field is filled with this character if a resulting item has no value. For example, for a date field with no value, you could specify that a hypen (-) be used (as shown on the sample screen).

"Error Value" Character

Optional. The print field is filled with this character if an error value occurs.

Softkeys

Select Item

Prompts for the name of the item to be accessed.

Select Lineset

Prompts for the level and type of lineset to be accessed. You respond with:

P (for page)	plus H (page heading) or F (page footing)
R (for report)	plus H (report heading) or F (report footing)
1 - 9 (break level)	plus H (break heading) or F (break footing)
D (for detail line)	

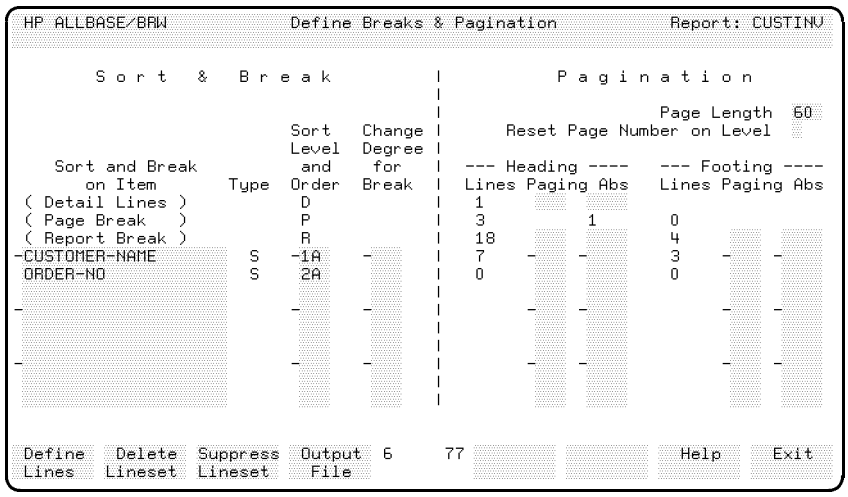
The Item Edits screen for the selected lineset and the current item appears. The current item is the item that appears in the Item field.

Prev/Next Item

Displays the previous or next item in the lineset.

Define Breaks Screen

Use this screen to set the sort sequence and specify pagination requirements for a report layout.



To define a sort and break

1. Type the name of the Sort Item, Level and Order, and Change Degree for Break.
2. Press Enter.

To delete a sort and break

1. After associated linesets have been deleted, blank out the name of the item.
2. Press Enter.

   Other items are rearranged in sort order.

To change sort levels

1. Type in new sort levels.
2. Press (Enter).

The system automatically rearranges items according to the new sort levels.

Optional settings (RESET PAGE on LEVEL, SUPPRESS REPITITION, SWITCH DESTINATION LEVEL and COLCALC LEVEL) which are related to break items are changed accordingly.

To delete a lineset

1. Choose `Delete Lineset`.
2. At the prompt, type the level and type of the lineset you want to delete.
3. Press (Enter) or choose `Delete Lineset` to confirm the deletion.

To define pagination

1. Type Paging and Absolute Position for this lineset. Also, change Page Length and Reset Page Number on Level, if required.
2. Press (Enter).

Field Descriptions for Sort and Break

`Sort and Break on Item` The name of the item for which you are defining sorts and breaks.

The breaks for detail line headings and footings (`Detail Lines`), page headings and footings (`Page Break`), and report headings and footings (`Report Break`) are already shown for your information.

`Type` **Display**. The type of this item.

`S` = String

`N` = Numeric

`R` = Real

`I` = Integer

`T` = Time

`D` = Date

`Sort Level` Sort level for this item, from 1 (highest) to 9 (lowest). These can be non-sequential numbers, for example, 4, 8, 9. Non-sequential numbers are reduced to sequential when you press Enter. For example, if you selected sort levels 4, 8, and 9, they would become 1, 2, and 3. Or, if you had selected 1, 2, and 3 but had deleted 2 before pressing Enter, the sequence would automatically become 1 and 2.

`Order` **Optional**. Type `A` for an ascending sort order for this item (default); `D` for a descending sort order.

`Change Degree for Break` **Optional**. For string items, type the position of the character, counting from the left and including leading blanks, at which a break should occur.

For numeric items, type a number which is the least significant digit for the break as the 10th exponent of this digit.

Valid ranges	for numeric type	N: -6 through 12
	for integer type	I: 0 through 8
	for real type	R: -90 through 76

For time items, type one of the following:

H -	to break when the hour changes
HM -	to break when the hour or minute change
HMS-	to break when the hour, minute or second changes (default)

For date items, type one of the following:

Y -	to break when the year changes
YM -	to break when the year or month change
YMD-	to break when the year, month or day changes (default)

Field Descriptions For Pagination

Pagination for detail linesets is defined in the heading column on the right side of the screen. Pagination of linesets for page break, report break and all other breaks are defined under the respective heading and footing columns.

Page Length — **Optional**. The number of lines on the page. The same value is displayed on the Define Report screen. This can be changed here or on the Define Report screen.

Reset Page Number on Level — **Optional**. The sort level at which the report's page numbering resets to page 1. A new page is started automatically when the page number is reset to 1. If a break which has a level lower than the reset page number level occurs, the page number is reset to 1 between the footing and heading of that break.

Lines — **Display**. The number of lines defined for each lineset.

Paging — **Optional**. One or any combination of the following codes to indicate when a new page should start:

A	= after printing a lineset;
B	= before printing a lineset;
S	= a lineset should be split.

The default is that linesets are not split, and a new page is started before a lineset if it does not fit completely on the rest of the page.

Abs — **Optional**. Absolute position at which a lineset is placed on the page.

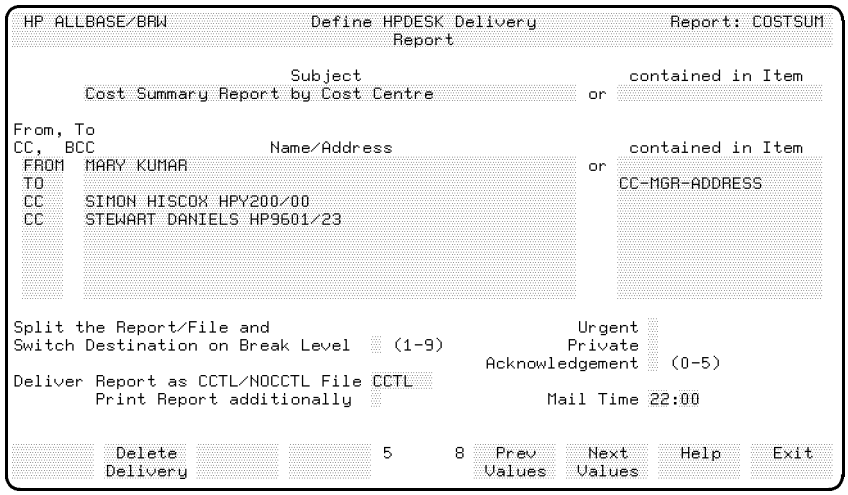
The default is that a lineset is printed directly after the preceding lineset.

To reserve space even if the lineset is not to be printed on that page, type R after the line number. If another lineset is absolute positioned, it can occupy this space, even if it is reserved. Only linesets which are not absolute positioned leave this space free.

If an absolute positioned lineset overlaps page heading or footing, the page heading/footing is suppressed on that page.

Define HPDesk Configuration Screen

Use this screen to define all the information necessary to allow HP ALLBASE/BRW to submit a report or output file to HPDesk.



HPDesk distributes the report according to the information entered here.

Note

You will only be able to access this screen if you have purchased HP BRW-Desk/XL.

Access this screen by choosing `Define Delivery` on the Define Report screen or appropriate Output File screen.

To define HPDesk delivery information

1. Type a literal address in the `Name/Address` field, or type the name of an item which contains or will calculate the address in the `contained in Item` field.
2. Press Enter.
3. A copy of the report or output file is sent to every address on this screen.
4. Choose `Next/Prev Values` to review more names/ addresses.

To Delete the HPDesk Delivery

1. Choose `Delete Delivery` to select the HPDesk delivery of the report or output file.

Field Descriptions

`Subject` **Optional**. Used by HPDesk as the subject of the message. You can use a string constant, a database item, or a calculated item. (The item type must be S(tring).

Name/Address Fields

Use these fields to specify the recipients of a message, using the standard HPDesk conventions for recipient categories. These fields are used to describe the complete distribution list for a message.

`FROM, TO, CC, BCC` The category of the sender and each recipient. At least one recipient is required. You don't need to specify the sender if you want the message to use the name of the sender(s) configured in **BRWCONF**. See Chapter 17 for information about configuring HP ALLBASE/BRW for HPDesk.

`Name/Address` **Optional**. Type a string constant in the `Name/Address` field, or the name of a database item or calculated item in the `contained in Item` field (if you use items, the item type must be S(tring) to specify the name and address of the recipient(s). You can use up to 200 names/addresses with each message; HPDesk will attempt to deliver to all the names.

The string constant or item can specify an HPDesk name/address or a distribution list. With distribution lists, the Public Distribution Lists in the Filing Cabinet are searched first, then the List Area of the sender is searched. Names/ addresses must be in the following HPDesk format.

[*firstname* [*secondname*]]

surname [/*location* [/*sublocation*]]

These HPDesk addresses are not validated by the BRW specification and compile programs.

`Split Report/File and Switch Destination on Break Level` **Optional.** The break level where the report or output file will be split and the destination switched, that is the report or output file will be divided into portions where each of them will be sent to individual destinations. The default is `no Destination Switch` which means that the whole report or output file will be sent to the specified destination.

Urgent

Optional. Use this field to make your message URGENT. If not specified, a default of NORMAL is taken for a message.

Private

Optional. Use this field to define the security of the message. If not specified the message will be given a security level of NORMAL (that is, not PRIVATE).

Acknowledgment

Optional. Use this field to define the level of Acknowledgment required on the message. As per standard HPDesk conventions, it will apply to just the TO recipients. The default will be that no Acknowledgment will be requested.

Acknowledgment Level - values:

```
0 - No Acknowledgment requested
1 - Transmitted Acknowledgment requested
2 - Receipt Acknowledgment requested
3 - Delivered Acknowledgment requested
4 - Read Acknowledgment requested
5 - Reply Acknowledgment requested
```

Print Report with CCTL/NOCCTL

Optional. You can send a report via HPDesk with or without carriage control codes. Files without carriage-control codes will be larger. The default is to send the report to HPDesk with carriage-control codes (CCTL).

Print Report additionally

Optional. Can be used only when defining delivery of report files. Use this field to specify that a report must be printed. The default is to suppress the printing of the report.

Mail Time

Optional. Use this field to specify the time when the message should be mailed. The message will remain in the out-tray of the sender with status: 'deferred' mail until the time specified. If the specified time has already passed (for example, mail time 1.00) and you run the report at 12:00, the message will be deferred until 1:00 the following day.

The default is: mail the message immediately.

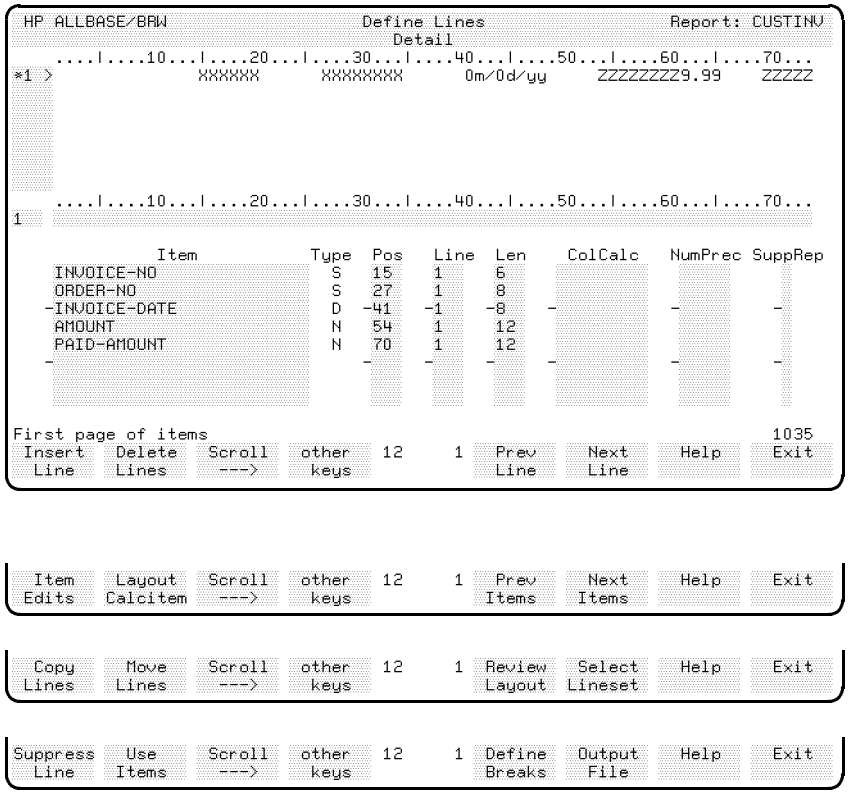
Softkeys

`Delete Delivery` Press this key to delete the delivery specification. At the prompt, press the key again to confirm the deletion.

`Prev/Next Values` Use these keys to add more than 8 recipients to a message. Up to 200 recipients may be defined for a message. This number includes all names specified by FROM, TO, CC, BCC, plus all names appearing in distribution lists. This number is the maximum number that is supported by HPDesk and is not checked by BRW at specification time.

Define Lines Screen

Use this screen to layout text and items in each lineset for a report.



This screen consists of three parts. The top section is called the report window and displays the current report layout. The middle section contains both the current line number field and the line text field in which literal text is entered. The bottom of the screen contains item layout details.

Since your report may be wider than the screen can show at one time, the `Scroll --->` softkey lets you see other portions of the screen and access more space for text.

Adding a new line of text to a lineset

To enter literal text

1. Type the new line number in the small field to the left of the ruler bar.
2. Type text in the wide field under the ruler bar.
3. Press Enter.

   The new line becomes the current line.

   The line number and line are displayed in the top half of the screen.

To continue typing text that is longer than 75 characters

1. Use `Scroll --->` to access additional room for the text.
2. Type the remaining text for this line.
3. Press Enter.

Inserting a line in front of another line in the lineset

1. Type the line number before which the new line will be inserted.
2. Type text for the line, using `Scroll --->` to access more space for the line if necessary.
3. Press `Insert Line`.

   The lines following the inserted line are automatically renumbered. The new line becomes the current line.

Modifying a line of text

To modify a line of text

1. Select the line to be modified by choosing `Prev Line` or `Next Line`. You can also use QUICK BROWSE to access the line. See Chapter 1 for details about the QUICK BROWSE feature.
2. Make changes to the text.
3. Press Enter.

Positioning items

To position items

1. Type the name of the item in the `Item` field.
2. Type the position, line number, length, column calculation (if any), and numeric precision for this item. See the Field Description section for details about these fields.

   To position an item in the next available position on a line, type > in the `Pos` field.
3. Type a number (1-9) in the `SuppRep` field if you want to suppress repetition for this item. The number represents the break level for the item.
4. Press Enter.

The item's type, position, line number and length are displayed in the top half of the screen. An edit mask is used to display the items. For instance, the edit mask `0m/0d/yy` is displayed where a date item using that edit mask is to be placed. `XXXXXX` represents the edit mask for a string item and `ZZZ` represents the edit mask for a numeric item.

Replacing a line of text

To replace a line

1. In the line number field, type the number of the line on which you want to replace text.
2. In the text field under the ruler bar, type the replacement text, overwriting previous text if necessary. Use `Scroll --->` to access more space for the line.
3. Press Enter.

Deleting and Copying

To delete lines in a lineset

1. Choose `Delete Lines`.
2. When you see the prompt for the line number(s) to be deleted, type a single line number, or type *number*/*number*, to indicate a range of numbers. For example, 2/6 means that lines 2 through 6 will be deleted.
3. Press Enter.

   Note that, if items have been positioned on the deleted lines, the items remain on the item layout section of the screen but are not printed. They disappear from the report window.

   The subsequent lines are renumbered.

To copy "as is" lines to the current lineset

1. Choose `Copy Lines`.
2. When you see the prompt, type:
   - a single line number or *number*/*number*, to indicate a range of line numbers
   - the lineset level and type from which you are copying
   - the destination line number
3. Press Enter.

   The lines and items are copied, and the line numbers are renumbered within the current lineset.

To copy a sample layout of lines to the current lineset

1. Choose `Copy Lines`.
2. When you see the prompt, type the following:
   - a single line number or *number*/*number*, to indicate a range of line numbers
   - the lineset level and type
   - the destination line number

- S for sample layout

3. Press Enter.

   Only the layout (from the report window) is copied; items are not copied. This feature provides you with a working layout in the current lineset.

Moving lines and items

To move lines and their positioned items

1. Press `Move Lines`.
2. When you see the prompt, type:
   - a single line number or number/number, to indicate a range of numbers that you want to move;
   - the destination line number.
3. Press Enter.
   - The following lines are renumbered.

To delete an item from a lineset

1. Blank out the name of the item.
2. Press Enter.

To define calculations/precision without printing the item

1. Type the item in the `Item` field.
2. Do not type a position or line number.
3. Press Enter.

Banner Line

The banner line (at the top of the screen) shows which lineset is current; for example, Page Heading in the banner indicates that this is the page heading lineset.

Report Window

Display. Line numbers and text of the current lineset. Uses these symbols:

*	indicates the current line
<	indicates text or items to the left of the screen display
>	indicates text or items to right of the screen display

The defined edit mask is displayed at the position of the item.

Literal text is displayed as it was typed in the line text field.

At the top and bottom of the report window, a ruler indicates column positions along the line. For any number along the ruler, the first digit corresponds to the exact column position. In this example,

```
....|....10...
```

the digit 1 indicates position 10 on the line.

Field Descriptions

Line Number Number of the current line being edited. This field is not labelled on the screen. It is the small field on the left side of the screen.

Line Text Text for the current line can be positioned at any point along this line; this field is also not labelled on the screen. It is located just below the ruler line near the center of the screen.

Item Name of each item in this lineset; can be an alias name if one has been defined on the Project Items screen.

Specify array components as shown:

```
turnover (1)
turnover (2)
```

Specify parameters in this way:

```
?month
```

Type **Display**. The type of the item:

S	string
N	numeric fixed point
I	integer
R	floating-point Real
D	date
T	time

Pos **Optional**. The column position at which the print field for this item begins. Use > to put the item in the next free position after two blanks. If you do not put a column position in this field, the item will not print on the report. There can be cases where this is what you want to do. For instance, if you want to use the item for a calculation, but do not want it to print in the report.

`Line` **Optional**. The line number on which this item is positioned in this lineset. Default is the current line.

`Len` **Optional**. The print field length for this item. The default is determined by the item length and edit mask defined on the Item Edits screen.

`ColCalc` **Optional**. The type of column calculation to be performed on this item and the level at which the calculation is to be reset. For item types N, R, and I, this can be one of the following:

COUNT
TOTAL
AVG
MIN
MAX
CALC

Optionally, you can add a running level and/or a source level to the name of the column calculation, for example TOTAL1,2. These are described earlier in this chapter under "Column Calculation Reset Levels" and "Column Calculation Source Levels."

`NumPrec` **Optional**. Numeric precision of this item if you want precision changed at this stage in the report. Type a number and method code. The number indicates the power of 10 or 10th exponent that indcates the precison. See Chapter 15, Defining Calculations, for an explanation of precision.

Valid number range	for item type N:	-6 through 12;
	for item type I:	0 through 8;
	for item type R:	-90 through 76.

The method code indicates how superfluous digits are to be treated:

Method Code	Digits are:
R	rounded
T	truncated
C	ceiled

Note the following examples:

If the value is:	with this numeric precision:	the result with changed precision:
1024.3849	-2R	1024.38
864186.46	3C	865000.00

`SuppRep` **Optional**. A number (from 1 to 9) corresponding to the sort level to suppress repeated printing of an item value, that is, an item is only printed on the first detail within this break.

Soft Keys

`Insert Line` After you have typed text for a line, choose this softkey to insert the line. If you have not typed a line number in the line number field, or have typed a new line number in the line number field, the new line is added *after* the last line number. If you typed an existing line number in the line number field, the line is added *before* the line number you used.

`Delete Lines` Choose this key when you want to delete a line. You are prompted to type a line number or a range of numbers to be deleted. The syntax for the range is: *number*/*number*. For example, 2/6 deletes lines 2 through 6.

`Prev Line` Changes the current line number to the previous line number.

`Next Line` changes the current line number to the next line number.

`Item Edits` Takes you to the Item Edits screen for the lineset you are on. At the prompt: `Edit-mask for item`, typing the name of the item you want to edit.

`Layout Calcitem` Takes you to the Layout Calc Item screen.

`Prev Items` Shows the previous group of items if the list of items is too large for one screen.

`Next Items` Shows the next group of items if the list of items is too large for one screen.

`Copy Lines` Choose this softkey to copy a line or a group of lines. Answer the prompt with:

- a single line number or a range of numbers. Use this syntax for a range of numbers: *number/number*.
- the lineset level and type for the lineset you are copying
- the line number where you want the copy

`Move Lines` Choose this softkey to move a line or a group of lines. Answer the prompt with:

- a single line number or a range of numbers. Use this syntax for a range of numbers: *number/number*.
- the lineset level and type for the lineset you are copying
- the line number where you want the copy

`Review Layout` Takes you to the Review Layout screen.

`Select Lineset` Choose this softkey to select a different lineset for the Define Lines screen. For instance, if you are on the Define Lines screen for the Report Heading, you can choose `Select Lineset` and answer the prompt with another lineset type (such as Report Footing) and level.

This choice prompts for the level and type of lineset to be accessed. You respond with:

P (for page)	plus H (page heading) or F (page footing)
R (for report)	plus H (report heading) or F (report footing)
1 - 9 (break level)	plus H (break heading) or F (break footing)
D (for detail line)	

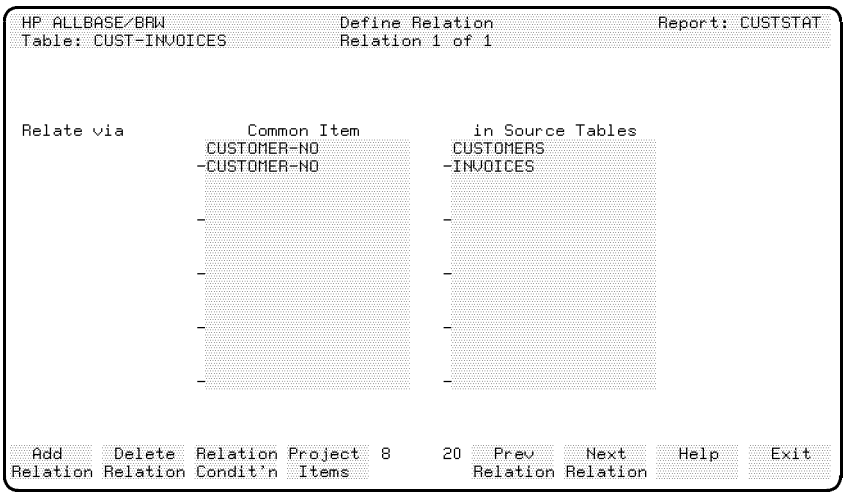
The Item Edits screen for the selected lineset and the current item appears. The current item is the item that appears in the `Item` field.

`Suppress Line` Takes you to the Suppress Line screen. Respond to the prompt `Suppress condition for line` by typing the number of the line you want to suppress.

`Use Items` Takes you to the Use Items screen.

Define Relation Screen

Use the Define Relation screen to specify common items to be used for a join.



To specify a new relation

1. Type the name of the Common Item and its Source Table.

   If the common item has the same name in more than one source table, you can, if you want, enter the one name common to both items (and do not specify a source table). HP ALLBASE/BRW looks up all occurrences of the item in all the source tables and uses all of them as the relation definition. The names of the items and their source tables will appear in the fields.

   If the common item has different names in its source tables, enter item name and source table name for all occurrences of that item.

2. Choose `Add Relation`.

To modify a relation

1. Use `Prev Relation` or `Next Relation` to find the relation you want to modify.
2. Make the required changes to the relations.
3. Press Enter.

To delete a relation

1. Use `Prev Relation` or `Next Relation` to find the relation you want to delete.
2. Choose `Delete Relation`.
3. Press Enter or choose `Delete Relation` again to confirm the deletion.

Banner Line

The banner line displays the name of the joined table for which you are defining this relation and the number of relations defined. For example, "Relation 3 of 3" means you are now defining relation number three from a total of three that exist.

Field Descriptions

`Common Item` The name of the item for which you are defining a relation. Single-value and list-of-values parameters can also be specified in a relation by preceding the name with a question mark, for example, `?MONTH`.

`In Source Tables` **Required**. The name of the data set, HP ALLBASE/SQL table, file or HP ALLBASE/BRW table in which this item resides. This must be the same name as specified on the Define Table screen. This field can be blank if the common item has the same name in each source table.

Note 

One relation stands for one common item. Define different relations if you have more than one common item.

Softkeys

`Add Relation` Choose this to add the defined relation to the report. You can add multiple relations to a report. The banner line of the screen shows how many relations have been defined and which relation you are working with. For instance: `Relation 1 of 1` or `Relation 2 of 4`.

`Delete Relation` Choose this to delete a relation from the report.

`Relation Condit'n` This takes you to the Relation Condition screen.

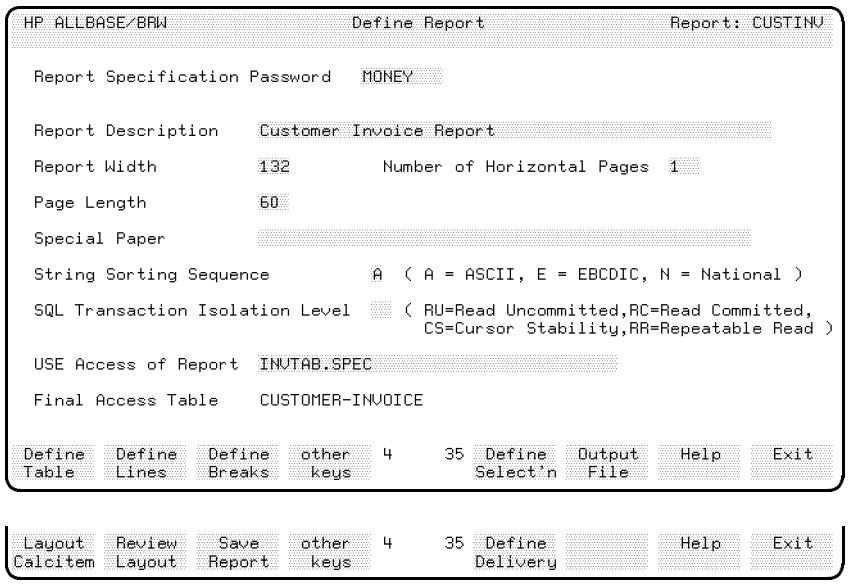
`Project Items` This takes you to the Project Items screen.

`Prev Relation` This shows the previous relation.

`Next Relation` This shows the next relation.

Define Report Screen

Use this screen to specify global characteristics for your printed report.



To define global characteristics for a report

1. Type global characteristic definitions in the appropriate fields.
2. Press Enter.

See the following Field Descriptions for details on each field.

Field Descriptions

Report Specification Password

Optional. Password restricting access to the specifications for this report.

Report Description

Optional. Description of the report which is displayed on the Define Report screen. Can also print this Report Description using the standard item REPORT-DESCRIPT.

Report Width

Maximum width is 300 characters. Default width is 132.

Number Of Horizontal Pages

The maximum number of horizontal pages, governed by the following equation:

```
(Number of Horizontal Pages * Page Width) <= 300
```

HP ALLBASE/BRW tells you if you attempt to have too many horizontal pages. The default is 1.

Page Length

Required. Maximum number of lines on a page is 999. The default is 60.

Special Paper

Optional. User-defined message requesting that special paper be mounted on the printer each time the report is printed. This message appears on the console at print time. After mounting and aligning the special paper, the operator replies that printing can start.

String Sorting Sequence

The sort-character-set for strings. The possible entries are:

A	Sort according to the ASCII character set.
E	Sort according to the EBCDIC character set.
N	Sort according to the Native Language Rules for the default Native Language. The default method of sorting and the default Native Language are set in the configuration file (See Chapter 17, under Specification Values and Global Values).

The default is `A` for ASCII.

`SQL Transaction Isolation Level`

Optional. The locking strategy for SQL transactions. For read only applications, RC (read committed) is the best suited locking strategy. Consult the HP ALLBASE/SQL Reference Manual for a description of the other isolation levels (RU = read uncommitted, CS = cursor stability, RR = repeatable reads). If this field is left blank, the execution value in the BRW Configuration File is used.

`Use Access of Report`

Optional. The name of a Report Specification File if you want to use the access definition of another report. This lets you generate reports with a common access definition.

`Final Access Table`

Display. The name of the final access table defined for this report. If this field is blank, you cannot define the layout or selection for this report, because either there is no data access defined, or there is no unique final access table (that is, the defined tables did not build a hierarchy.) To define the tables, use `Define Table` to get to the Define Table screen.

Softkeys

`Define Table`

Takes you to the Define Table screen.

`Define Lines`

Takes you to the Define Lines screen. If you select `Define Lines`, you are prompted for the lineset you want to define. Type the lineset level and type as follows:

- The lineset level must be R (Report), P (Page), 1..9 (Break Level), or D (Detail). The Break Level refers to the level of a sort item as specified on the Define Breaks screen.
- The lineset type must be H (Header) or F (Footer). If the lineset level is D, leave the type field blank.

`Define Breaks`

Takes you to the Define Breaks screen.

`Define Select'n`

Takes you to the Define Selection screen.

`Output File`

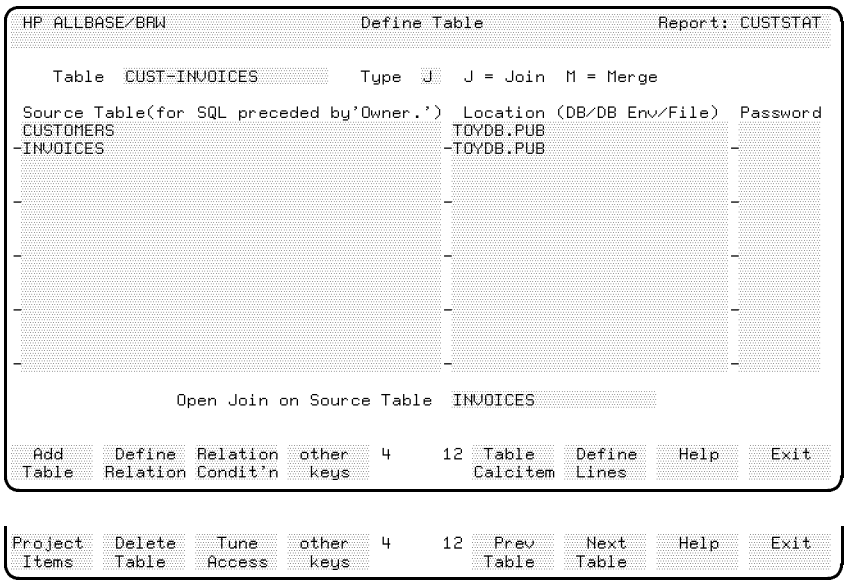
Takes you to the Output File screen.

`Save Report`

Saves your work. HP ALLBASE/BRW saves the report specification file automatically when you exit the program or access another report. If you make frequent changes, it is better to use `Save Report` to save the report in case of system failure.

Define Table Screen

Use this screen to specify the source tables and the type of combination to be performed to create the result table.



To add a new table

1. Type the new Table, Type, Source Table, Location, and Password. For an open join table, also type the name of the source table for the open join.
2. Choose `Add Table`.

To modify a table

1. Use the `Next Table` or `Prev Table` softkeys to access the table you want to modify. You can also use quick browse to access a table. See Chapter 1 for details about the quick browse feature.
2. Type modifications to the table definition.
3. Press Enter.

Note

Changing source tables or open join definitions causes changes in other data access screens. You should check all relation, relation condition, item projection and tune access specifications to ensure they are still correct.

To rename a table

1. Use `Next Table` or `Prev Table` to access the table you want to rename. You can also use quick browse to access a table. See Chapter 1 for details about the quick browse feature.
2. Change the name in the Table field.

3. Press Enter.
4. Press Enter again to confirm the rename.

To delete a table

1. Use `Next Table` or `Prev Table` to access the table you want to delete. You can also use quick browse to access a table. See Chapter 1 for details about the quick browse feature.
2. Choose `Delete Table`.
3. Press Enter or choose `Delete Table` to confirm the deletion.

To copy a table

1. Type the name of the new table and press Clear Display.
2. Choose `Add Table`.
3. When you see the prompt, type the name of the table to be copied and press Enter.

The table and all definitions of item projection, relations, relation conditions, calculated items, functions and data access are copied.

Field Descriptions

`Table` The name of the HP ALLBASE/BRW table.

`Type` The type of combination to be performed on the source tables, either J (join) or M (merge). The default is J.

`Source Table` The name of the source tables that will be used to build the table.

In HP ALLBASE/BRW a source table is the name of an IMAGE data set, the *owner_name.table_name* of an HP ALLBASE/SQL table, the format name of a KSAM or MPE file, or the name of another HP ALLBASE/BRW table defined in the same report specification.

If you are using more than one HP ALLBASE/SQL table from the same DBEnvironment, the tables should be listed consecutively. This is because HP ALLBASE/SQL queries to the same DBEnvironment are then combined to one HP ALLBASE/SQL query by default.

If you are using an intermediate report file (an output file created by another report), you must assign it a name here.

Note 

For HP ALLBASE/SQL tables you must specify both the owner name and the table name in the Source Table field of this screen, however, elsewhere within HP ALLBASE/BRW you only need to specify the table name.

To define a report, you need the HP ALLBASE/SQL production database or an HP ALLBASE/SQL test database with the same table structures, owner names, and access rights as the production database. You must have SELECT authority on all HP ALLBASE/SQL tables and views you intend to specify. In addition, you need CONNECT authority to the HP ALLBASE/SQL DBEnvironment.

`Location` The location of the source tables, either a database, DBEnvironment, or file name. These names must be qualified with group and account names if the report is to be executed from another logon group or account.

If an IMAGE database or file is specified with an account other than the logon account, the dictionary file in that other account is used along with the specified database or file.

An HP ALLBASE/SQL DBEnvironment does not require a dictionary file.

An intermediate report file name must be preceded by the symbol **>**.

`Password` **Optional**. The IMAGE database password or the file lockword, if any. The default for a database password is a semicolon (;)—this requires the user to be the creator of database. If this is a file, the default password is blank. See "File Lockwords/Passwords" in Chapter 22, for more information on how to specify passwords.

If the IMAGE database or file is not described in the data dictionary, HP ALLBASE/BRW does not verify the location and password.

HP ALLBASE/SQL does not know passwords for DBEnvironments, tables, views, or columns. Access to an HP ALLBASE/SQL database is controlled via authorities granted to the users of a DBEnvironment. This information is kept in the SQL System Catalog and checked against the MPE logon user running an SQL application. A user is identified by HP ALLBASE/SQL as *logonuser@logonaccount*.

`Open Join on Source Table`

Optional. The source table for which NO_VALUE is to be used if the other source tables contain records not found in this source table.

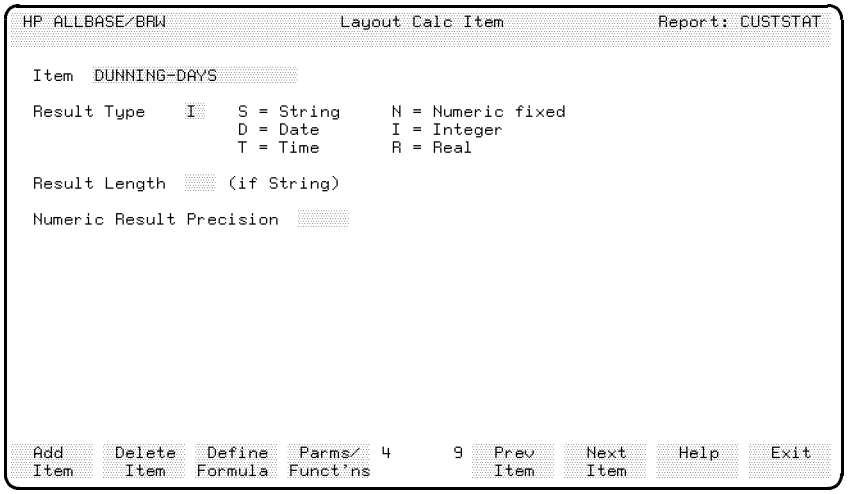
The default is that only records found in all source tables will be used.

Softkeys

There are two sets of softkeys for this screen. `other keys` takes you to the other set of softkeys. The softkeys for this screen take you to the screen named on the key. When you use `Define Lines` you need to respond to the prompt for lineset and type. Type the lineset type (D for detail, P for page, R for report, or a break level number) and type H for heading or F for footing if applicable.

Layout Calc Item Screen

Use this screen to define a layout calculated item.



To add a new item

1. Type the name of the Item, Result Type, Result Length, and Numeric Result Precision.
2. Choose `Add Item`.

To copy an item

1. Type the name of the new item and press (Clear Display).
2. Choose `Add Item`.
3. When you see the prompt, type the name of the item from which you want to copy and optional Table Name if the item is in another table. For a layout item, type >LAYOUT in the Table Name field.
4. Press (Enter).

The item and its formulas are copied.

To modify an item

1. Use `Next/Prev Item` to access the item you want to modify. You can also use QUICK BROWSE to access an item. See Chapter 1 for details about the QUICK BROWSE feature.
2. Make the required changes to the item.
3. Press (Enter).

To delete an item

1. Use `Next/Prev Item` to access the item you want to delete. You can also use QUICK BROWSE to access an item. See Chapter 1 for details about the QUICK BROWSE feature.

2. Choose `Delete Item`.
3. Press Enter or choose `Delete Item` to confirm the deletion.
   a. The item and its formula are deleted.

To rename an item

1. Use `Next/Prev Item` key to access the item you want to rename. You can also use QUICK BROWSE to access an item. Chapter 1 for details about the QUICK BROWSE feature.
2. Type the new name over the old item name.
3. Make any required changes to the item's definition and press Enter.
4. Press Enter to confirm the rename.

Field Descriptions

`Item` The name of the calculated item.

`Result Type` One of the codes listed on the screen for the Result Type.

`Result Length` **Optional**. Type the result length of the string. Maximum is 255; minimum is 1. Required only if this item is a string.

`Numeric Result Precision` **Optional**. Type a number and method code for the numeric precision of the calculated item, if you want precision to be changed at this stage of the report.

Valid range for Numeric (type N): -6 through 12; for Integer (type (I): 0 through 8; for Real (type R): -90 through 76.

Method Codes:	Digits are:
R	rounded
T	truncated
C	ceiled

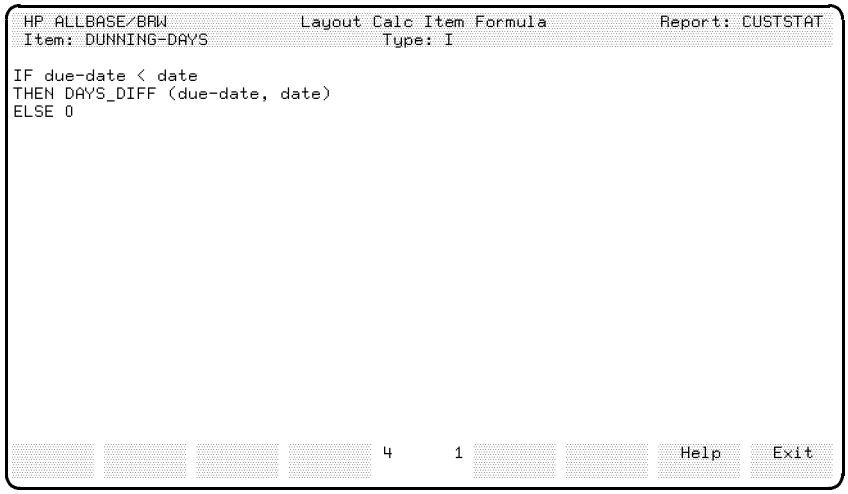
This is illustrated in the following examples:

If calculation results in:	With this numeric precision:	The result with changed precision is:
1024.3849	-2R	1024.38
864186.46	3C	865000.00

See the description of Real Numbers in Chapter 15 for a description of Real output forms and precision.

Layout Calc Item Formula Screen

Use this screen to define a formula for a layout calculated item.



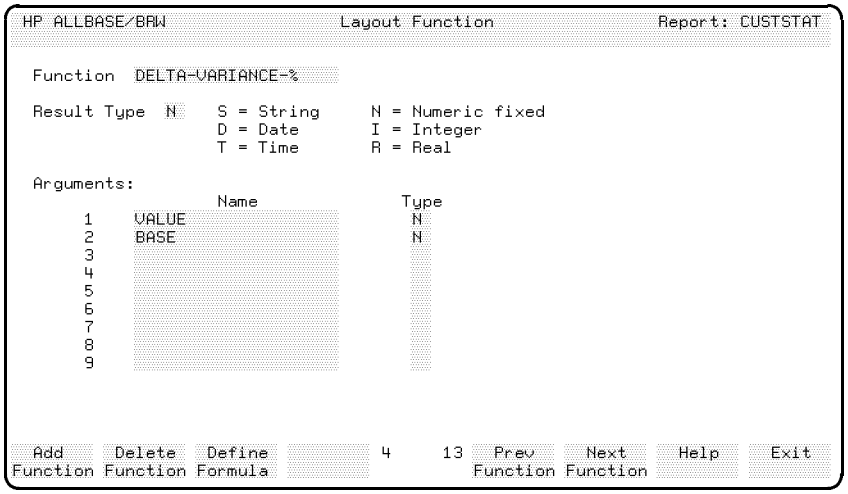
To define a formula

1. Type the calculation formula.
   - Leave blanks around any operators in the formula.
   - You can use all terminal editing and arrow keys when typing and editing a formula.
   - Use (Prev Page) or (Next Page) keys on the terminal to access the next page of the formula.
2. Press (Enter).

The formula is checked for correctness. If an error is detected, the part of the formula with the error is highlighted. Correct the error and press (Enter) again. Or, choose `Exit Save` to save the formula as it is for later correction.

Layout Function Screen

Use this screen to define a layout function.



To add a new function

1. Type the name of the Function, Result Type, the Function Argument and Type.
2. Press `Add Function`.

To copy a function

1. Type the name of the new function and press (Clear Display).
2. Choose `Add Function`.
3. When you see the prompt, type the name of the function from which you want to copy and optional table name if the function is in another table. For a layout function, type >LAYOUT in the Table Name field.
4. Press (Enter). The function and its formulas are copied.

To modify a function

1. Use `Next/Prev Function` key to access the function you want to modify. You can also use QUICK BROWSE to access a function. See Chapter 1 for details about the QUICK BROWSE feature.
2. Make the required changes to the function.
3. Press (Enter).

To delete a function

1. Use `Next/Prev Function` key to access the function you want to delete. You can also use QUICK BROWSE to access a function. See Chapter 1 for details about the QUICK BROWSE feature.
2. Press `Delete Function`.

3. Press Enter or Delete Function to confirm the deletion.

To rename a function

1. Use Next/Prev Function key to access the function you want to rename. You can also use QUICK BROWSE to access a function. See Chapter 1 for details about the QUICK BROWSE feature.
2. Type the new name over the old function name.
3. Make any required changes to the function definition and press Enter.
4. Press Enter to confirm the rename.

Field Descriptions

Function The name of the function.

Result Type One of the codes listed for Result Type.

Arguments The formal names of the function arguments. Use these names within the function formula. Actual values for the arguments must be provided when the function is called.

Type Type of each argument. Use the codes listed on the screen.

Softkeys

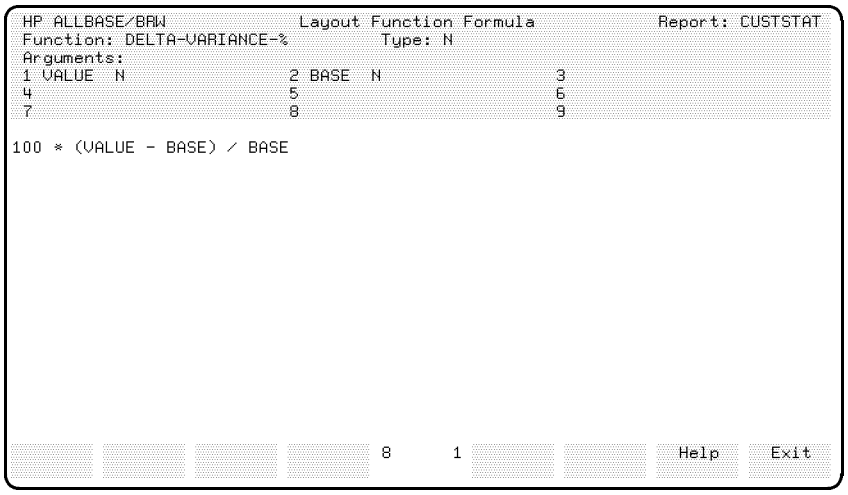
Add Function Choose this key to add a new function.

Delete Function Choose this key to delete a function.

Prev Function
Next Function Use these keys to access other functions.

Layout Function Formula Screen

Use this screen to define a formula for the layout function defined at the Layout Function screen. The top portion of the screen shows the function name and the arguments already defined for the function.



To define a formula

1. Type the calculation formula.

   Leave blanks around any operators in the formula.

   You can use all terminal editing and arrow keys when typing and editing a formula.

2. Use (Prev Page) or (Next Page) keys on the terminal to access additional pages for the formula.

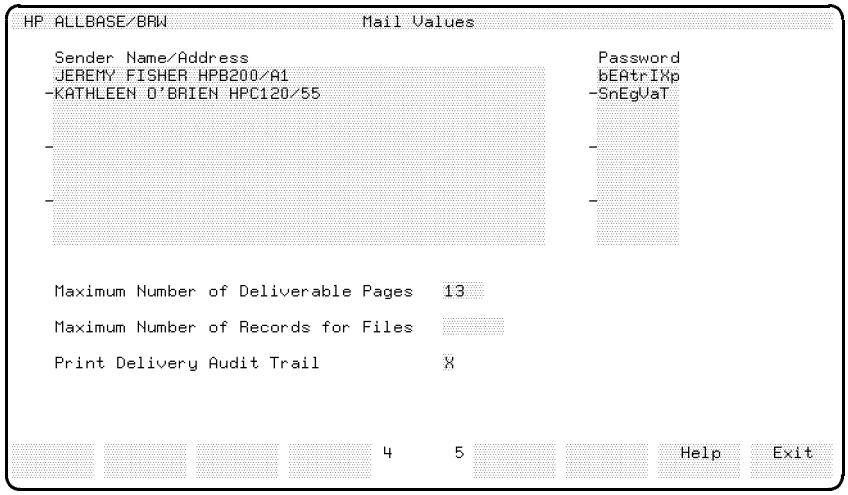
3. Press (Enter).

The formula is checked for correctness.

If an error is detected, the part of the formula with the error is highlighted. Correct the error and press (Enter) again. Or, choose `Exit Save` to keep the formula as it is for later correction.

Mail Values Screen

This screen is part of the BRWSETUP.PUB.SYS program. Use this screen to define default specifications for HP BRW-Desk/XL. This screen is for use only when you have HP BRW-Desk/XL in addition to HP ALLBASE/BRW.



To change the default specifications

1. Type the desired value in the appropriate fields. See the Field Descriptions section for details on each field.
2. Press Enter.

Field Descriptions

Sender Name/Address

Required. The HPDesk user name/address that is used when signing on to HPDesk. This address is not validated at specification time. You can specify up to 9 different senders, and a minimum of 1 sender is required. HP BRW-Desk/XL will use this pool of senders to do the sign on to HPDesk. If it cannot sign on with the first sender, the next sender will be used until the sign on is successful or until the configured number of senders is exceeded.

Password

Optional. The password of the user signing on. Note that HPDesk passwords are case-sensitive; that is, if the password is PeTeR, and you supply PETER in the Password field, HPDesk will not let HP BRW-Desk/XL log on.

These passwords are encrypted in the configuration file.

Maximum Number of Deliverable Pages

Optional. The user-defined limit on the size of the message that can be sent to HPDesk. This limit is counted in report pages. If a message exceeds this limit, the message will not be submitted to HPDesk and the delivery audit trail will show that fact. The default is the maximum size of an item permitted in the HPDesk system.

Maximum Number of Records for Files

Optional. The user-defined maximum number of records for output files that can be generated and sent via HP BRW-Desk/XL. If the number is exceeded,the *HP ALLBASE/BRW* execution program (BRWEXEC.PUB.SYS) will abort. The default is the maximum size of an item permitted in the HPDesk system.

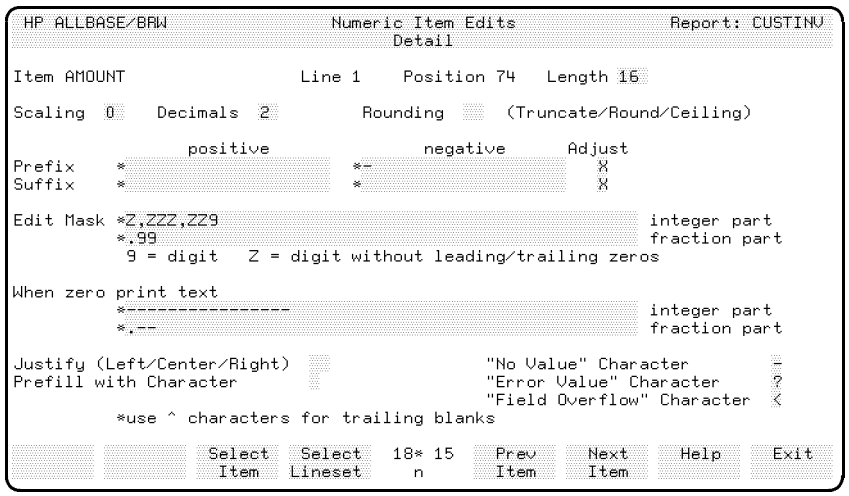
Print Delivery Audit Trail

Optional. Blank out the X in this field if you do not want to print the delivery audit trail messages (a record of the mailing process that is generated whenever a report using HP BRW-Desk/XL is run). The default is an X in this field: that is, to print the delivery audit trail.

Numeric/Integer Item Edits Screen

Use the Numeric Item Edits screen to specify how Numeric and Integer item values are edited before being printed in a report. You get to this screen from the Define Lines screen. Choose `other keys` until you can choose `Item Edits`. At the prompt, you type the name of the item you want to edit.

Although the same screen appears whether you edit a numeric or an integer item, the defaults on the screen are different, as described below the screen.



This screen shows the defaults for a numeric item, that is, 12 digits (Z) without leading or trailing zeros plus one digit (9) with no restriction on leading or trailing zeros; then a decimal point (period), followed by 6 digits (9) in the decimal places.

For an integer item, the default is no decimal point, 8 digits (Z) with no leading or trailing zeros and one digit (9) with no restriction on leading or trailing zeros.

To specify a numeric or integer item edit

1. Use the `Prev/Next Item` or the `Select Item` function key to access the item.
2. Type the item edit definition for this item.
3. Press Enter.

Banner Line

The banner line (at the top of the screen) shows in which lineset the current item is positioned. This example screen shows a Detail lineset.

Field Descriptions

`Item` **Display**. The current item.

`Line/position` **Display**. The line and print field position of the item.

`Length` **Display**. The print field length defined for this item on the Define Lines screen. You can optionally change the length for this item.

`Scaling` **Optional**. The number of places the decimal point moves to the left, or to the right if the scaling is negative.

`Decimals` **Optional**. How many decimal places are to be appended to the scaled number. Note: be sure to include enough 9's or Z's in the fraction part of the edit mask for numeric items to accomodate what is entered in this field.

`Rounding` **Optional**. The scaling operation for this item; one of these three codes:

T	Truncate
R	Round
C	Ceil

Below are examples of scaling the number 3426.5809.

If scaling is:	and the number of decimals is:	The results are:		
		T	R	C
0	2	3426.58	3426.58	3426.59
2	1	34.2	34.3	34.3
-2	0	342658	342658	342659

Prefix/Suffix positive/negative — **Optional**. Type a prefix/suffix for positive and negative occurrences of this item. For example, the prefix or suffix can be a minus sign, Cr, Db, brackets, or so on. Use ^ for indicating trailing blanks. The prefix or suffix takes up space in the field. Be sure to allow for this additional space.

Adjust — **Optional**. To position the prefix/suffix directly next to the number, type an X. Default is that the prefix is left-justified and the suffix is right-justified in the print field.

Edit mask — The edit mask for this item, using these characters:

9	digits
Z	digits without leading/trailing zeros

Use ^ to show trailing blanks.

The sample screen shown on the previous page illustrates the edit mask to print numbers with thousands characters and decimal points.

For an amount such as $69, the superfluous thousands characters are omitted.

Note that you can also use text (THOUSANDS, DOLLARS, etc.) within the edit mask.

When zero print text — Type the text to be printed if the item value is zero. You can do this separately for the integer part and the fraction part, for example:

```
-.--
```

Use ^ to show trailing blanks.

Justify — Type one of the following:

L	left justify
C	centre
R	right justify

Default is to align on the decimal point.

Prefill with Character — **Optional**. Specify the character to be printed instead of leading blanks. See the following examples in which an asterisk is the prefill character:

If you use an adjusted prefix, the result is:

```
*******-49
```

Without an adjusted prefix:

```
-*******49
```

"No Value" Character	**Optional**. The print field is filled with this character if the item has no value.
"Error value" Character	**Optional**. The print field is filled with this character if an error value occurs.
"Field overflow" Character	**Optional**. The print field is filled with this character if the item value is too long for the edit mask.

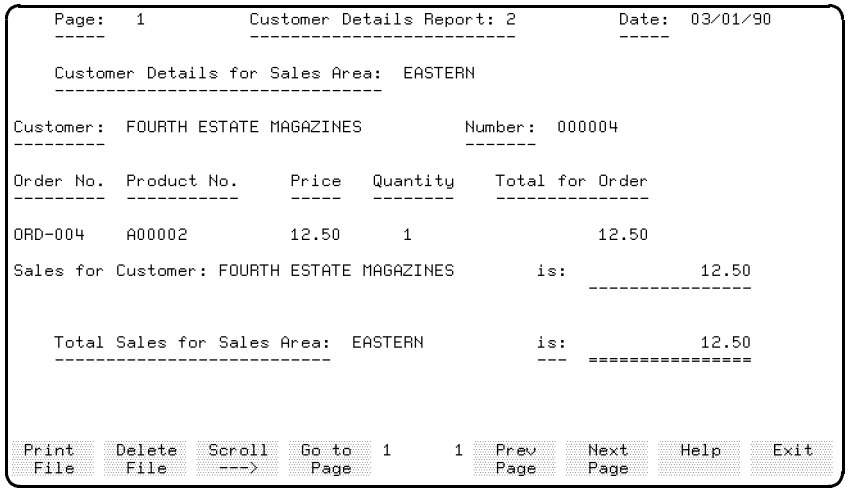
Softkeys

Select Item	Prompts for the name of the item to be accessed.
Select Lineset	This choice prompts for the level and type of lineset to be accessed. You respond with:

P (for page)	plus H (page heading) or F (page footing)
R (for report)	plus H (report heading) or F (report footing)
1 - 9 (break level)	plus H (break heading) or F (break footing)
D (for detail line)	

Online Review Screen

Use this screen to review a report which is stored in a disc file.



To review all parts of a report

1. Use the `Scroll-->`, `Next/Prev Page`, or `Go To Page` keys to review parts of the report that do not appear on the screen.

To print a report

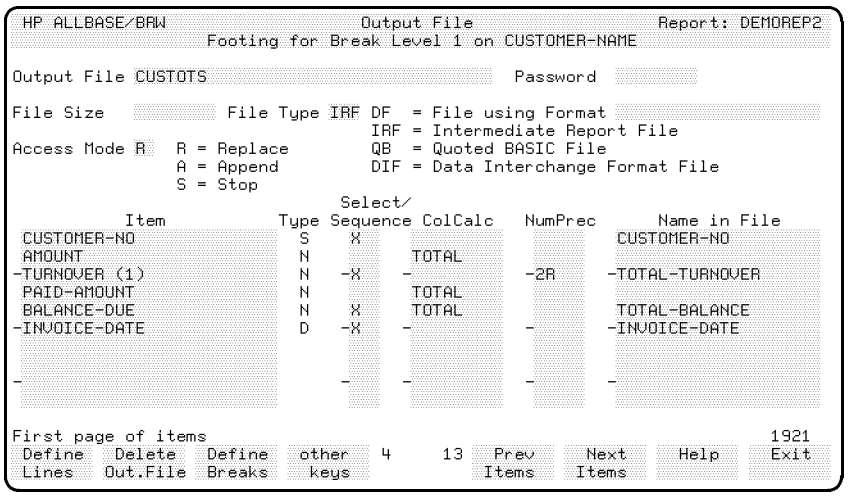
1. Choose `Print File`.
2. When you see the prompt, type the printer ID if it is different from the default.
3. Press Enter or choose `Print File` again.

To delete a report

1. Choose `Delete File`.
2. Press Enter or choose `Delete File` again to confirm the deletion.

Output File Screen

You use the Output File screen to define an intermediate report file and to specify which items are to be written to it.



To define an output file

1. From the Define Report screen, choose `Output File`.
2. At the prompt, type the lineset information for the lineset you want to use. Press Enter.

   The Output File screen appears. Every item that appears on the Define Lines screen for the specified lineset is displayed in the Item field on the Output File screen.
3. Type the name of the file, password (if any), file size (if known), access mode and file type in the approriate fields.
4. Define the items for the output file. You must define at least one item to be written to the output file. See the next procedure for details on defining the items.
5. Press Enter.

To define items for an output file

1. Type the name of each item that you want written to the output file in the Name in File field or type an X (or a number for QB and DIF files) in the Select/Sequence field.

   If the output file is an intermediate report file, you can choose any name for an item's Name in File. However, if the output file is a dictionary output file, the name in the Name in File field must be a name of an element of the file, that is, the name of an element as specified in the file format in the data dictionary.

2. You can, optionally, specify whether the item is to be subject to a column calculation, and the numeric precision of an item in the output file.

3. Press Enter.

Items positioned on the Define Lines screen *cannot* be deleted or re-arranged here.

If there are more than 9 items, use the Next/Prev Items keys to access the next page and continue your selection. Press Enter for each page.

To remove the use of an output file

1. Type the name of the output file.
2. Choose `Delete Out.File`
3. Press Enter or choose `Delete Out.File` again to confirm the deletion.

This deletes only the use of the output file for this lineset. It doesn't delete any MPE file.

Other Information

You can specify whether an item is to be subject to a column calculation, and an item's numeric precision in the intermediate report file.

Intermediate report files are built when the report is compiled (*not* when it is executed. Values are only *written* to the files when the report is executed.) If an intermediate report file exists with the same name as that specified on an Output File screen, and you compile the report, the file will be purged as re-built.

If you want to use an item in an output file that does not appear in the specified lineset, type its name in the Item field. It will then appear on the lineset, but will not be positioned and will not print.

If any other type of file exists with the same name as that specified on an Output File screen, and you compile the report, the compile will fail and the file will not be purged.

When you delete an intermediate report file with the PURGE command, you can only create it again by re-compiling the report.

So, to free disc space when data in an intermediate report file is no longer needed, you can run the program REMPTY to clear the file, as shown here:

```
:RUN REMPTY.PUB.SYS

      . . .
Intermediate report file:  ORDERS.PUB

END OF PROGRAM
```

REMPTY deletes all records from an intermediate report file but leaves the file and the data descriptions intact. REMPTY *cannot* be used for other types of file.

Example

The example below shows single items, single components of array items and whole arrays that will be written to an intermediate report file. If you do not specify an index for an array, the whole array is written to the intermediate report file.

```
HP ALLBASE/BRW                    Output File                    Report: DEMOREP2
                  Footing for Break Level 1 on CUSTOMER-NAME

Output File CUSTOTS                                Password

File Size          File Type IRF DF  = File using Format
                                 IRF = Intermediate Report File
Access Mode R   R = Replace      QB  = Quoted BASIC File
                A = Append       DIF = Data Interchange Format File
                S = Stop
                                  Select/
          Item                Type Sequence ColCalc   NumPrec      Name in File
 CUSTOMER-NO                   S    X                              CUSTOMER-NO
 AMOUNT                        N         TOTAL
-TURNOVER (1)                  N   -X    -            -2R         -TOTAL-TURNOVER
 PAID-AMOUNT                   N         TOTAL
 BALANCE-DUE                   N    X    TOTAL                     TOTAL-BALANCE
-INVOICE-DATE                  D   -X    -            -           -INVOICE-DATE


-                                  -     -            -           -

First page of items                                                        1921
 Define   Delete   Define   other   4     13   Prev     Next     Help     Exit
 Lines    Out.File Breaks    keys              Items    Items
```

Note

Every item in the specified lineset, (that is, the items positioned on the Define Lines screen of the specified lineset), will be displayed in the field labeled Item. However, only those items that you specify in the Select/Sequence field are written to the intermediate report file. You can choose any name you want (use the Name in File field) for an item in an intermediate report file.

Field Descriptions

`Output File` The name of the output file.

`Password` **Optional**. The lockword for this output file.

`File Size` **Optional**. The maximum size of the file in records. If this is blank, the default defined in HP ALLBASE/BRW's configuration file is used.

`Access Mode` The access mode determines what HP ALLBASE/BRW does if another file of the same name is encountered on the system at run time. Type one of the following codes:

`R` =	Replace the records in the file, that is, all records currently in the file will be lost and any new records will be written, starting from the beginning of the file.
`A` =	Append any records to the end of the existing file, that is, all records currently in the file will be kept, and any new records will be written to the file, starting from the last old record.
`S` =	If the output file is a dictionary output file that already exists, or the output file is an intermediate report file that already contains records, and you specify `S`, HP ALLBASE/BRW will not write any records to the output file. That is, all records currently in the file will be kept and no new records will be written.

`File Type` **Optional**. The file type for this output file. Type one of the following:

`IRF` =	An intermediate report file that is a self-describing file for use as a source table in another report.
`DF` =	A file in data dictionary format. The file format must be described in the dictionary file BRWDIC.PUB.
`QB` =	A Quoted Basic file format.
`DIF` =	Data interchange format file.

`Item` **Display**. Displays all items selected for this lineset as specified on the Define Lines screen.

`Type` **Display**. Displays type defined for this item:

`S` =	string
`N` =	numeric fixed point
`D` =	date
`I` =	integer
`R` =	real
`T` =	time

`Select/Sequence` Specify an X or a number to indicate that you want an item in the output file. With QB and DIF files, the number will indicate the position of the items in the record. **If you leave this field blank, the item will NOT be included in the output file.**

`ColCalc` **Display/Optional**. Displays the specifications entered on the Define Lines screen for this item. This can be changed for the output file, as explained below.

The type of column calculation to be performed on this item and the level at which the calculation is to be reset. For items types N, R and I, this can be one of these:

AVG
COUNT
MAX
MIN
TOTAL

Optionally, add a reset level and/or a source level to the name of the column calculation. See the "Column Calculations" section in Chapter 7 for details.

Note

The items and calculations displayed in the output file screen match what will be displayed on the Define Lines screen. If you add a column calculation to a displayed item, that item reference on the Define Lines screen will pick up the column calculation. If you want to write column calculations to an output file and not have them appear on the printed lineset, type the item name a second time on the Output File screen.

`NumPrec` **Optional**. Numeric precision of this item if you want precision to be changed for the output file. Type a number and method code, as described in the following tables.

The number indicates the power of 10, or 10th exponent that indicates the precision.

Valid range for Numeric (type N):	-6 through 12;
for Integer (type I):	0 through 8
for Real (type R):	-90 through 76.

The method code indicates how superflous digits are treated:

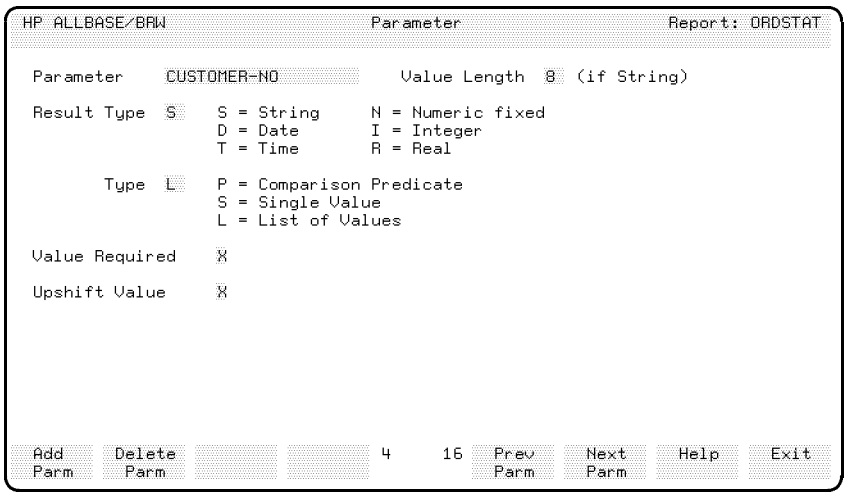
Method Codes:	Digits are:
R	rounded
T	truncated
C	ceiled

See Chapter 13, Calculations, for a full description of precision and the special cases for Real number precision.

`Name in File` The item name to be used in the output file. If the output file is a dictionary output file, the name must refer to the name of an element in the file as defined in the file description in the data dictionary.

Parameter Screen

Use this screen to define parameters for which you supply values at report execution time.



To add a new parameter

1. Type the name of the Parameter, Value Length, Result Type, Parameter Type, Value Required and Upshift Value. See the Field Descriptions section for details on each field.
2. Choose `Add Parm`.

To copy a parameter

1. Type the name of the new parameter and press Clear Display.
2. Choose `Add Parm`.
3. When you see the prompt, type the name of the parameter from which you want to copy.
4. Press Enter. The parameter is copied.

To modify a parameter

1. Use `Next/Prev Parm` key to access the parameter you want to modify. You can also use QUICK BROWSE to access a parameter. See Chapter 1 for details about the QUICK BROWSE feature.
2. Make the required changes to the parameter.
3. Press Enter.

To delete a parameter

1. Use `Next/Prev Parameter` key to access the parameter you want to delete. You can also use QUICK BROWSE to access a parameter. See Chapter 1 for details about the QUICK BROWSE feature.
2. Choose `Delete Parm`.
3. Press Enter or choose `Delete Parm` to confirm the deletion.

To rename a parameter

1. Use `Next/Prev Parm` key to access the parameter you want to rename. You can also use QUICK BROWSE to access a parameter. See Chapter 1 for details about the QUICK BROWSE feature.
2. Type the new name over the old parameter name.
3. Make any required changes to the parameter's definition and press Enter.
4. Press Enter to confirm the rename.

Field Descriptions

`Parameter` The name of the parameter.

`Value Length` The maximum length of a string parameter. (Maximum allowed length must be between 1- 55 characters).

`Result Type` The result type. Use one of the codes shown on the screen.

- P = a relation operator and a list of values, including masks, i.e., B@, WHITE;
- S = single value, i.e., RED;
- L = one or more values, excluding masks, i.e., YELLOW, BLACK, BROWN, BEIGE.

`Value Required` **Optional**. Type X if a value *must* be input at report execution. Recommended if parameter type is S or L.

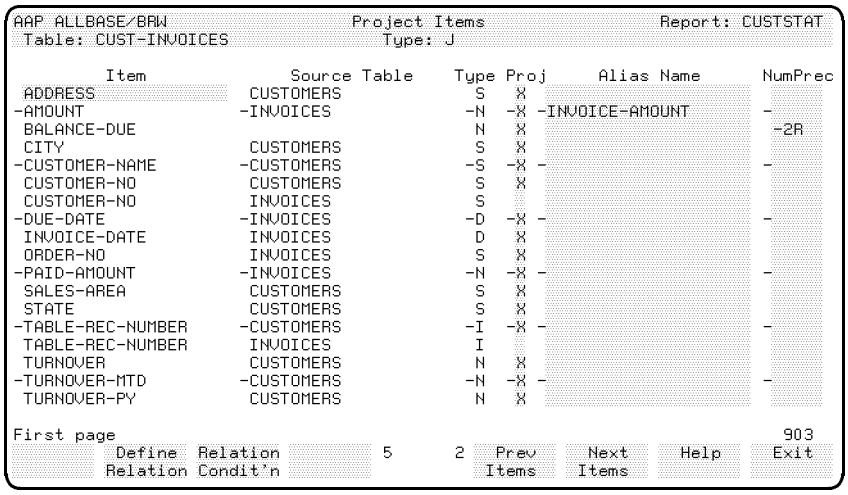
`Upshift Value` **Optional**. Type X to shift automatically to upper-case the value of this parameter. This is useful if the parameter is used to select values of an item containing only upper-case data in the database.

Note 

See the description of Real numbers in Chapter 15 for the special syntax required when supplying a Real paramter when you run the report.

Project Items Screen

Use this screen to edit the list of items in the table. This screen shows all items from the source tables and which items are to be projected from the table.



To review the list of items

1. Use `Prev Item` or `Next Item` to review the list of items or use quick browse. See Chapter 1 for details about quick browse.

To project items from a table

1. Type `X` in the `Project` field to project an item .
2. You can type an alias name for any item in the `Alias Name` field. This alias name will then be used in the report.

   For a joined table, the alias name must be unique within the selected items for this table.

   For a merged table, items with the same alias name appear as the same item in the new table.
3. Press Enter.

To exclude items from projection

1. Blank out the `X` in the `Project` field for any item you want excluded from projection.
2. Press Enter.

Banner Line

Displays the table name and type of combination for this table, either J (join) or M (merge).

Field Descriptions

`Item` **Display**. All items from the source tables in alphabetical order.

`Source Table` **Display**. For items located in the source tables, the source is listed. For a table calculated item, this field is left blank.

`Type` **Display**. The type of this item.

`Project` An `X` for all items to be projected from the table. The defaults are as follows:

- For a joined table, all items must have unique names. If they do not have unique names, specify aliases for them (see below).
- For a merged table, items with identical names are projected as the *same* item.

`Alias Name` **Optional**. The name of the item to be used outside the table.

- For a joined table, if the name of an item in a table is different from its name in the source table, this column contains the name to be used after projection from the table. It must be a unique name within projected items in the table.
- For a merged table, an item can have the same alias name if it is from a different source table; that is, you must give items from different source tables the same alias name if they are to be treated as one item after projection.

`NumPrec` **Optional**. The numeric precision of an item, if you want it changed after the item is projected. This consists of a number and a method code.

Valid number ranges are as follows:

Item Type	Number
N:	-6 to 12
I:	0 to 8
R:	-90 to 76

The method code indicates how superfluous digits are treated:

Method Code	Digits are:
R	rounded
T	truncated
C	ceiled

See the description of setting precision, including the special cases for Real precision, in Chapter 15.

Note

Project, Alias Name, and NumPrec only apply *outside* the table, for example, in a report layout definition, or if the table is used as source table, in another table. Therefore, if items have the same name and any are used in a formula for a table calc item, **the name must be qualified with the source table name**.

For example, suppose you have two items named ORDER-DETAILS, one in the Quanty table and one in the Orders table. You have given one of the items an alias name so the two items are not confused in the final report. However, you want to use one of the items in a formula for a table calc item. **You can't use the alias name.** You must use the original item name, and qualify it with the source table name, like this: ORDER-DETAILS.QUANTITY.

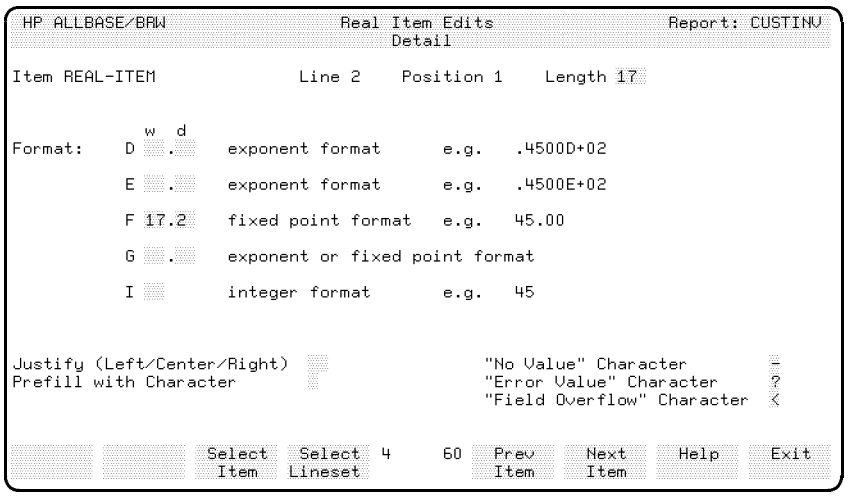
For HP ALLBASE/SQL source tables, the owner name is not necessary; only the table name is used.

Note

Source tables which are IMAGE data sets, KSAM files, or MPE files always have the item TABLE-REC-NUMBER included in the result table, in addition to their other items. HP ALLBASE/SQL tables do not have the item TABLE-REC-NUMBER included. TABLE-REC-NUMBER provides the logical record number in that data set or file.

Real Item Edits Screen

Use the Real Item Edits screen to specify how Real item values are edited before being printed in a report.



To specify a Real item edit

1. Use the `Prev/Next Item` or the `Select Item` function key to access a different item.
2. Type the item edit definition for the item. See the Field Descriptions section for details on each field.
3. Press Enter.

Banner Line

The banner line (at the top of the screen) shows in which lineset the current item is positioned.

Field Descriptions

`Item` **Display**. The current item.

`Line/position` **Display**. The line and print field position of the item.

`Length` **Optional**. The print field length defined for this item on the Define Lines screen. You can optionally change the length for this item.

`Format` **Optional**. Format for Real numbers. Real numbers do not use edit-masks, as do integers and numeric items. However, you can print real numbers in a variety of formats as shown on the next table.

Dw.d

w = field width including signs, decimal point, exponent character D, and d digits.

d = number of digits The value will be printed with a decimal point preceding all digits. For example:

Format D10.4 (width 10, digits 4)

Value	Output
-12.342	-.1234D+02
999.99	.1000D+04

Ew.d

w = field width including signs, decimal point, exponent character E, and d digits.

d = number of digits The value will be printed with a decimal point preceding all digits. For example:

Format E10.4 (width 10, digits 4)

Value	Output
-12.342	-.1234E+02
999.99	.1000E+04

Fw.d

w = field width including sign and decimal point.

d = number of decimals The value will be printed in fixed point format with d decimal places. For example:

Format F6.3 (width 6, decimals 3)

Value	Output
-5.667914	-5.668

Gw.d w = field width

d = number of digits — The value is printed using E format if its absolute value is less than 0.1 or greater than or equal to 10**d (after rounding to d significant digits). Otherwise it is printed using F format with blanks in the 4 spaces reserved for the exponent part. For example:

Format G12.4 (width 12, digits 4. ^ denotes a space)

Value	Output
-12345	^^-.1235E+02
1234	^^^1234.^^^^
-12	^^-12.00^^^^

Explanations for the above examples:

1. -12345 rounds to -12350 which has an absolute value greater than 10**4 so format E12.4 is used.)
2. 1234 rounds to 1234 which is less than 10**4 and greater than 0.1 so format F8.0 is used.
3. -12 rounds to -12.00. Format F8.2 is used. Note that the number of decimal places is calculated from the number of digits given in the G format and the actual value of the data.

Iw w = field width. This is integer format. For example:

Format I5

Value	Output
-52	^^-52
+452.75	^^453

`Justify` **Optional**. Type one of the following:

L	left justify
C	center
R	right justify

Default is to align on the decimal point.

`Prefill with Character` **Optional**. Specify the character to be printed instead of leading blanks. For example, if the length is 20, the value (in E10.4 format) is -1234E+02, and the prefill character is as asterisk, the number will be printed:

```
**********-1234E+02
```

`"No Value" Character` **Optional**. The print field is filled with this character if the item has no value.

`"Error Value" Character` **Optional**. The print field is filled with this character if an error value occurs.

`"Field Overflow" Character` **Optional**. The print field is filled with this character if the item value is too long for the edit mask.

Softkeys

`Select Item` Prompts for the name of the item to be accessed.

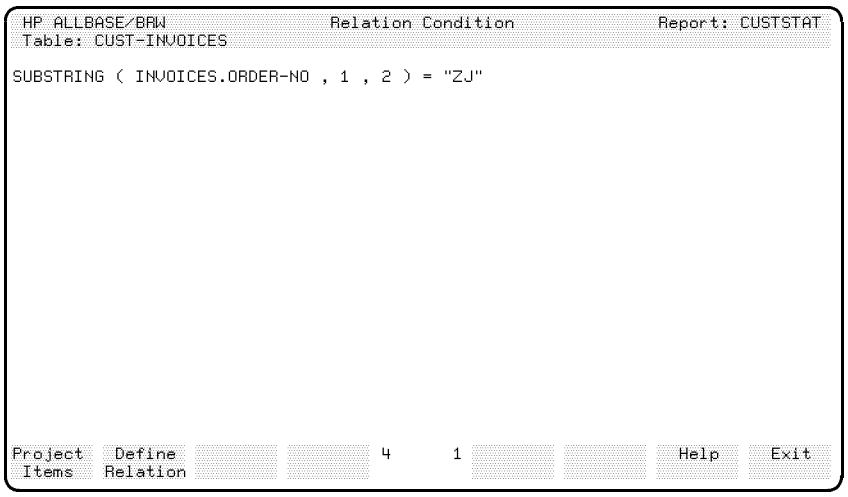
`Select Lineset` Prompts for the level and type of lineset to be accessed. You respond with:

P (for page)	plus H (page heading) or F (page footing)
R (for report)	plus H (report heading) or F (report footing)
1 - 9 (break level)	plus H (break heading) or F (break footing)
D (for detail line)	

The Item Edits screen for the selected lineset and the current item appears. The current item is the item that appears in the `Item` field.

Relation Condition Screen

Use this screen to enter the formula which selects a subset of data for the joined table.



To define a relation condition

1. Type the condition formula.

   Leave blanks around any operators in the formula.

   You can use all terminal editing and arrow keys when typing and editing a formula.

2. Press Enter.

   The formula is checked for correctness.

   If the system detects an error, the part of the formula with the error is highlighted. Correct the error and press Enter again. Or, choose `Exit Save` to save the formula as it is (in spite of the error).

To delete the relation condition

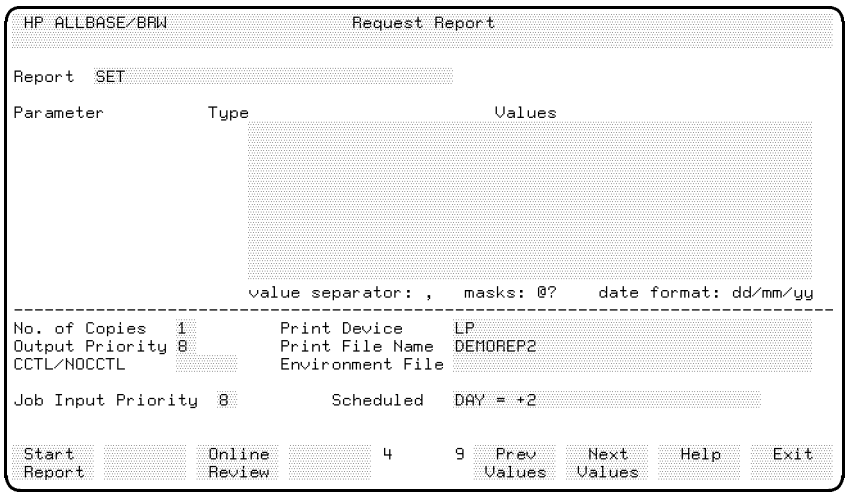
1. Use Clear Display to delete the formula.
2. Press Enter.

Softkeys

Softkeys on this screen take you to the screen named on the softkey label.

Request Report Screen

Use this screen to request execution of a report or reports and programs contained in a report job file.



You also use this screen to access the Online Review screen if you want to display a report on the screen.

To request a report

1. Type schedule, printer details, and parameter values for this report. Use the `Next/Prev Values` if you want to use more than 9 parameters.
2. Choose `Start Report`.

   The values are checked to see if they are valid.
3. When you see the prompt, type the user, account and group name for this report.
4. Press Enter or `Start Report` again.
5. If you are prompted for a password (when one has been specified for the user, account and/or group name), type the password(s) and press Enter.

When the job is streamed. HP ALLBASE/BRW displays the job number at the bottom of the screen.

To review a report online

1. Execute the report (as above), specifying the print device as Disc.
2. Press `Online Review`.

Field Descriptions

`Report` The name of the report you want to request. The default display is the report execution file that is currently loaded. You can use one of the following:

- the name of the execution file if you are requesting a report to be printed;
- the name of a report job file if you are executing reports and programs contained in that file, or
- the name of the print file if you are reviewing the report online.

`Parameter` **Display**. The parameters in the selection set for this report. If the list is longer than 9 parameters, use the `Next/Prev Values` to review the entire list.

If you are requesting a report job file, the parameters defined in that file are displayed here.

`Type` **Display**. The data (or result) type of the parameter: S = string; N = numeric; D = date; T = time; I = integer; R = Real.

`Values` **Optional**. For **comparison predicate** parameters, type a comparison operator and one or more values, as shown:

Operator	Valid Values
=	one or more values separated by
blank	commas. Can include these
<>	mask characters (wild cards):
	? = mask for one character
	@ = mask for zero, one or more
	characters
>	one value
>=	one value
<=	one value
<	one value
R	two values specifying upper and lower
	boundaries of the range

If no value is specified, all values will match this parameter.

For **list-of-values** parameters, type one value or a list separated by commas.

For **single-value** parameters, type one value.

For **string** parameters it is possible to pass a value that contains a double quote ("). For example, to specify the string ABC"D, type the values as "ABC""D". Using two double quotes ("") in a string results in one double quote (").

If you see the message "frozen values" in these fields, the value *cannot* be changed from the default.

Scheduled — Instructions for when the report is to be printed. For example, type one of the following:

```
DATE  = 08/31/86
DAY   = MON
DAY   = 15
        (that is, on the 15th day of the next month)
DAY   = -2
        (that is, 1 day !!before!!  month-end)
IN    = 2, 12
        (that is, after 2 days, 12 hours)
AT    = 13:30
        (that is, at 1:30 pm)
DATE  = 02/08/87; AT = 14:45
        (that is, on 8th  February  1987, at 2:45 pm)
DAY   = 15 ; AT = 10:15
        (that is, on the 15th day of the next month, at 10:15 am)
```

If you see the message "frozen values" in these fields, the value *cannot* be changed from the default.

No. of Copies — The number of copies to be printed. The default is 1.

Print Device — The device name on which the report is to be printed. If you want to review the report online, type DISC in this field.

Output Priority — The output priority for this report. The default is 8.

If you see the message "frozen values" in these fields, the value *cannot* be changed from the default.

`Job Input Priority` The job input priority for the report job. See Chapter 11 for information on the jobs that run HP ALLBASE/BRW reports.

If you see the message "frozen values" in these fields, the value *cannot* be changed from the default.

`Environment File` **Optional**. The name of the environment file for printing this report. [What is the default?]

If you see the message "frozen values" in these fields, the value *cannot* be changed from the default.

`CCTL/NOCCTL` **Optional**. The code for printing with carriage control codes or no carriage control codes. Type CCTL in this field (default) if you want the printfile to be printed with CCTL (carriage-control) codes, NOCCTL if you want the printfile printed with out CCTL codes. If you choose NOCCTL, HP ALLBASE/BRW will insert an empty line in the printfile for every line that must be skipped.

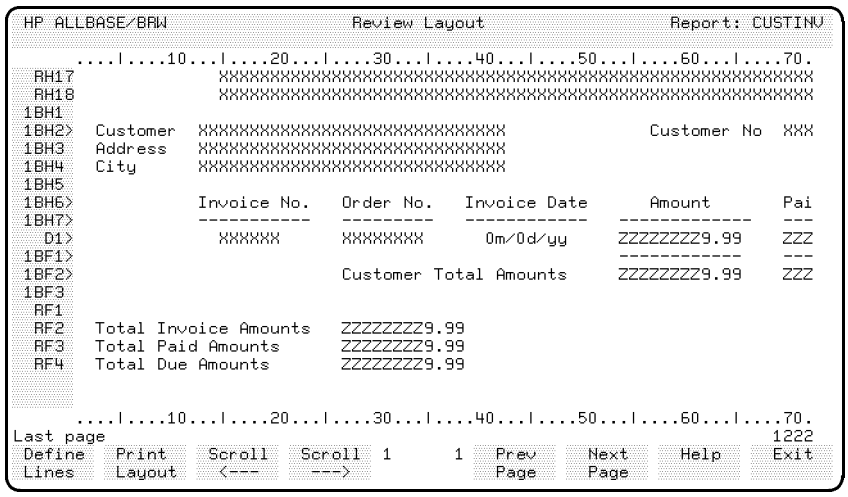
If you see the message "frozen values" in these fields, the value *cannot* be changed from the default.

`Print File Name` **Optional**. This is the spool or disc file name for the report. The default is the report name.

If you see the message "frozen values" in these fields, the value *cannot* be changed from the default.

Review Layout Screen

Use this screen to review a sample of the report layout on-line and to print a copy of the sample.



To review on-line

Use the `Scroll` and `Prev/Next Page` functions keys to review the entire sample layout.

To print a sample

1. Press `Print Layout`.
2. When you see the prompt, type the print device you want to use.
3. Press Enter.

Field Descriptions

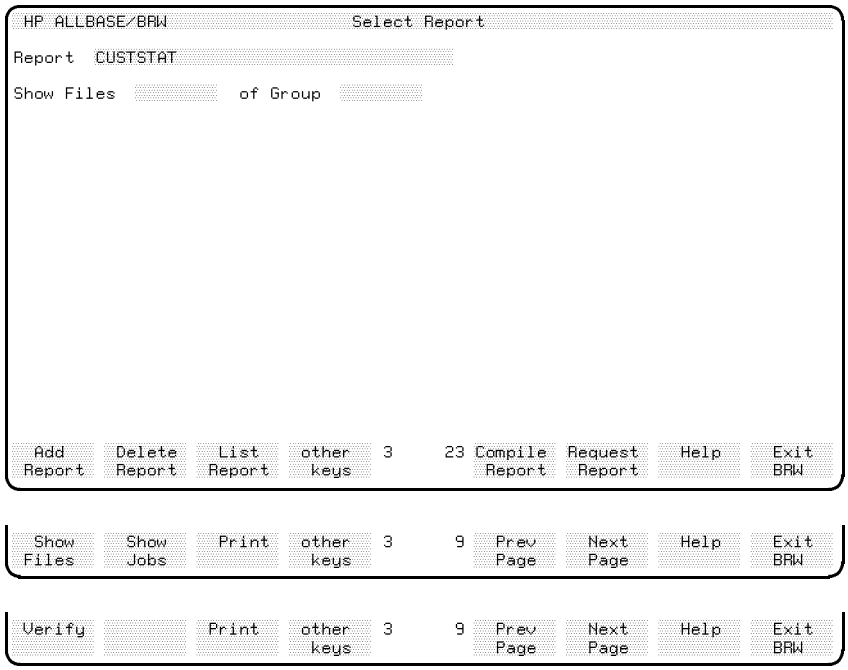
`Line` The description of the line. This unlabeled field is located to the left of the layout area.

The following are examples of the symbols used on the sample layout.

`RH7`	Report Heading, line 7. Use Prev Page to see previous lines in the Report Heading.
`RH8`	Report Heading, line 8.
`1BH1`	First break level heading, line 1.
`1BF1`	First break level footing, line 1.
`RF1`	Report Footing, line 1.
`<`	Text or items are to the left of the current screen for this line. Use `Scroll <---` to see that portion of the line.
`>`	Text or items are to the right of the current screen for this line. Use `Scroll --->` to see that portion of the line.

Select Report Screen

This screen is the entry point to HP ALLBASE/BRW. Use it to select and maintain reports, list report specifications, current files or jobs.



To add a report

1. Type the name of the report specification file in the `Report` field.
2. Choose `Add Report`.
3. When you see the prompt, type the name of the report from which you want to copy, or press Enter to create a new report.

To modify a report

1. Type the name of the report specification file you want to modify.
2. Press Enter.

   HP ALLBASE/BRW displays the following prompt if your report accesses an HP ALLBASE/SQL table:

```
Do you want to check the HP ALLBASE/SQL DBE for Changes (Y/N)?
```

   If you press `Y`, HP ALLBASE/BRW automatically upgrades your specifications according to the changes in the DBEnvironment. The default is `N`.

The changes made during this upgrade are listed on the Select Report screen. You can print out the listing by choosing `Print`. It can be helpful to have this information and a specification file listing to identify what changes (if any) must be made manually.

To delete a report

1. Type the name of the file you want to delete.
2. Choose `Delete Report`.
3. When you see the prompt, choose `Delete Report` again to confirm the deletion.

To list report specifications

1. Type the name of the report specification file.
2. Choose `List Report`.
3. When you see the prompt, type the print device.
4. Press Enter.

To compile a report

1. Type the name of the report specification file.
2. Choose `Compile Report`.

To request a report

1. Type the name of the report execution file.
2. Choose `Request Report`.

To verify MPE/V compatibility

1. Choose `other keys` until the `Verify` label appears.
2. Choose `Verify`.
3. Message appears after report is checked for compatiblity to run on MPE/V.

To exit HP ALLBASE/BRW

1. Choose `Exit BRW`.
2. If you made changes to the report, a prompt asks you if you want to save them.
   a. Type `Y`, then press Enter to save the changes.
   b. Type `N`, then press Enter to preserve the unchanged version.
   c. Choose `Cancel` to cancel the exit command.

   If no changes have been made, confirm the exit command by pressing Enter. Or you can cancel the command by choosing `Cancel`.

To display a particular file

1. Type the name of the file in the `Show Files` field.
2. Type the name of the group in the `of Group` field.
3. Choose `Show Files`.

To display all HP ALLBASE/BRW files in a group

1. Leave the `Show Files` field blank.
2. Type the name of the group in the `of Group` field.
3. Choose `Show Files`.

Note

You can use the ?, #, or @ wildcards in either the `Show Files` field or the `of Group` field.

To display current jobs

1. Choose `Show Jobs`.

To print messages displayed on screen

1. Choose `Print`.

Field Descriptions

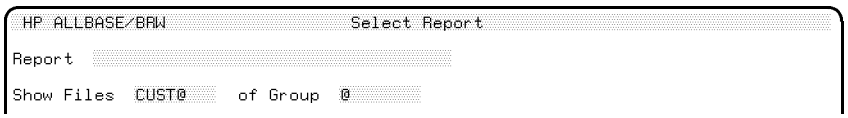
`Report`
The name of the report you want to create or select. This creates or selects an MPE file with the extension RSPEC (Report Specification). The report name must follow rules for MPE file names.

`Show Files ... of Group`
Optional. Use this field to display any of the HP ALLBASE/BRW files in your account. Type a file name (or a fileset name with wild-card characters) or a group name (or groupset name with wild-card characters). Wild-card characters are any of the following MPE wild-card characters:

- `@` matches zero or more alphanumeric characters
- `#` matches one numeric character
- `?` matches one alphanumeric character

If the `Show Files` field is left blank, all HP ALLBASE/BRW files are displayed. If the `of Group` field is left blank, logon group files are displayed.

For example:



This displays all customer reports in this account. That is, all HP ALLBASE/BRW reports that begin with the letters CUST.

Softkeys

`Add Report` Begins the definition of a new report specification file.

`Delete Report` Deletes an existing report specification file.

`List Report` Lets you print a listing of the report specification.

`Compile Report` Compiles the report.

`Request Report` Takes you to the Request Report screen when you can request the execution of a report.

`Show Files` Type the name of a file and/or a group and choose `Show Files`. The file name can be blank to show all files. If the group name is blank, you will see files in the current group and account. You can use MPE wildcards, @, #, or ? in the `Show Files` and `of Group` fields.

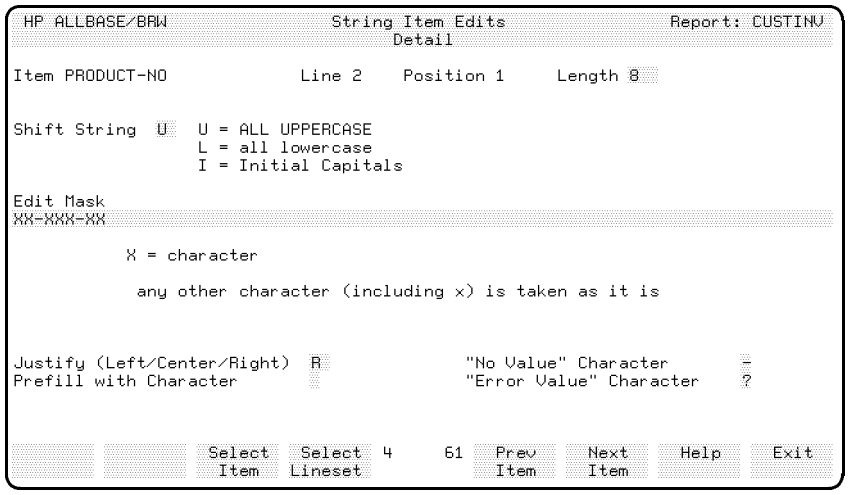
`Show Jobs` Shows the current MPE jobs that are running.

`Print` Prints messages that are displayed on the screen.

`Verify` Verifies the compatibility of the report to run on the version of MPE loaded on the system.

String Item Edits Screen

Use this screen to specify how the values of string items are to be edited before they are printed on a report. String item edits are optional.



To specify a string edit

1. Use `Prev/Next Item` or `Select Item` to access the item.
2. Type item edits for this item.
3. Press Enter.

Note

Unlike other Item Edits, String Item Edits allow a blank Edit Mask field, so that no editing is performed.

Banner Line

The banner line (at the top of the screen) shows in which lineset the current item is positioned.

Field Descriptions

`Item` **Display**. The current item.

`Line/position` **Display**. The line and print field position of the item.

`Length` **Optional**. The print field length defined for this item on the Define Lines screen. You can also change the length for this item.

`Shift String` **Optional**. Type one of the following to specify how text is to be printed:

- U upper case
- L lower case
- I initial capitals on each word

`Edit Mask` **Optional**. The Edit Mask for this item. Use X's to indicate characters in the string and any other characters to indicate masked part of the string. Use blanks in the mask to allow for blank spaces in the string.

Default is that the item is not edited with a mask.

`Justify` Type one of these codes:

- L left justify
- C center
- R right justify

The default is that the value is left-justified.

`Prefill with Character` The character to be printed instead of leading blanks.

`"No Value" Character` **Optional**. The print field is filled with this character if the item has no value.

"Error Value" Character	**Optional**. The print field is filled with this character if an error value occurs.

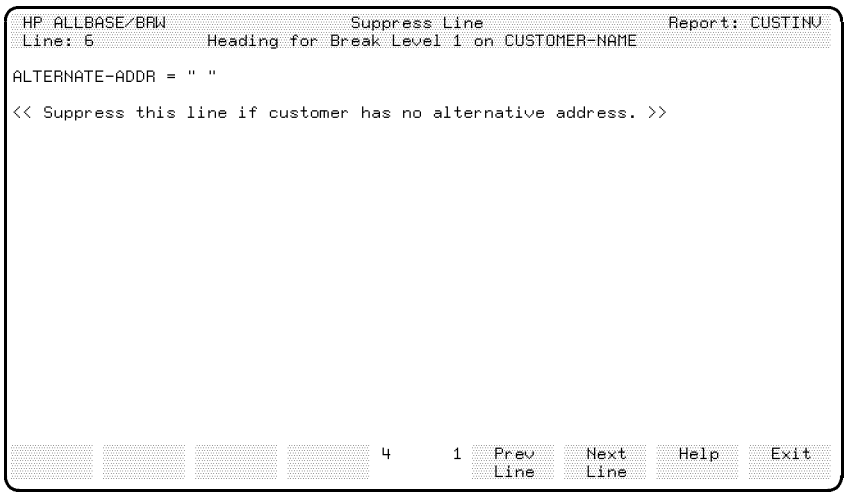
Softkeys

Select Item	Prompts for the name of the item to be accessed.
Select Lineset	Prompts for the level and type of lineset to be accessed. You respond with:

P (for page)	plus H (page heading) or F (page footing)
R (for report)	plus H (report heading) or F (report footing)
1 - 9 (break level)	plus H (break heading) or F (break footing)
D (for detail line)	

Suppress Line Screen

Use the Suppress Line screen to set the conditions under which specific lines are not printed.



Access this screen from the Define Lines screen. Respond to the prompt at the Define Lines screen `Suppress condition for line` by typing the line number you want to suppress.

To specify a suppress line condition

1. Type the suppress line condition.

   Leave blanks around any operators in the condition.

   You can use all terminal editing and arrow keys when typing or editing a condition.

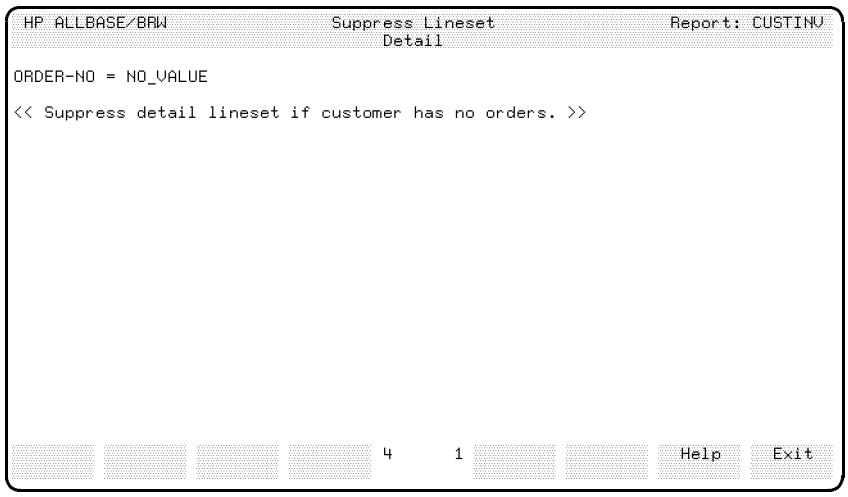
2. Press Enter.

   The condition is checked to make sure it is correct.

   If an error is detected, the part of the formula with the error is highlighted. Correct the error and press Enter again. Or, choose `Exit Save` to save the condition as it is for later correction.

Suppress Lineset Screen

Use this screen to set the conditions under which specific linesets should not be printed or sent to the output file.



Access this screen from the Define Breaks screen.

To specify a suppress lineset condition

1. Type the suppress lineset condition.

   Leave blanks around any operators in the condition.

   You can use all terminal editing and arrow keys when typing or editing a condition.

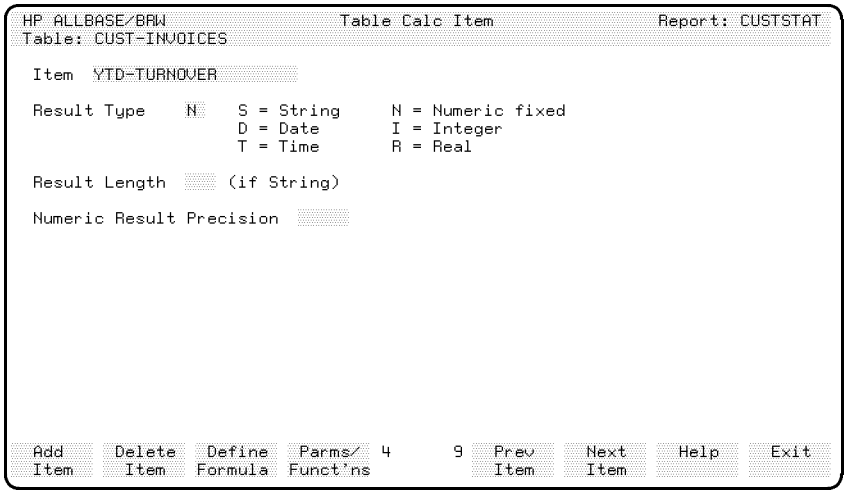
2. Press Enter.

   The condition is checked for correctness.

   If an error is detected, the part of the formula with the error is highlighted. Correct the error and press Enter again. Or, press the function key `Exit Save` to save the condition as it is for later correction.

Table Calc Item Screen

Use this screen to define a table calculated item.



To add a new item

1. Type the name of the Item, Result Type, Result Length, and Numeric Result Precision.
2. Choose `Add Item`.

To copy an item

1. Type the name of the new item and press (Clear Display).
2. Choose `Add Item`.
3. When you see the prompt, type the name of the item from which you want to copy and optional Table Name if the item is in another table. For a layout calculated item, type >LAYOUT in the Table Name field.
4. Press (Enter).

The item and its formulas are copied.

To modify an item

1. Use `Next/Prev Item` key to access the item you want to modify. You can also use QUICK BROWSE to access an item. See Chapter 1 for details about the QUICK BROWSE feature.
2. Make the required changes to the item.
3. Press (Enter).

To delete an item

1. Use `Next/Prev Item` key to access the item you want to delete. You can also use QUICK BROWSE to access an item. See Chapter 1 for details about the QUICK BROWSE feature.

2. Choose `Delete Item`.
3. Press Enter or choose `Delete Item` to confirm the deletion.

The item and its formula are deleted.

To rename an item

1. Use `Next/Prev Item` key to access the it em you want to rename. You can also use QUICK BROWSE to access an item. See Chapter 1 for details about the QUICK BROWSE feature.
2. Type the new name over the old item name.
3. Make any required changes to the item's definition and press Enter.
4. Press Enter to confirm the rename.

Field Descriptions

Item The name of the calculated item.

Result Type One of the codes listed for the Result Type.

Result Length Type the result length of the string. Maximum is 255; minimum is 1. Required only if this item is a string.

Numeric Result Precision Type a number and method code for the numeric precision of the calculated item, if you want precision to be changed at this stage of the report.

Valid range for Numeric (type N): -6 through 12; for Integer (type (I): 0 through 8; for Real (type R): -90 through 76.

Method Codes:	Digits are:
R	rounded
T	truncated
C	ceiled

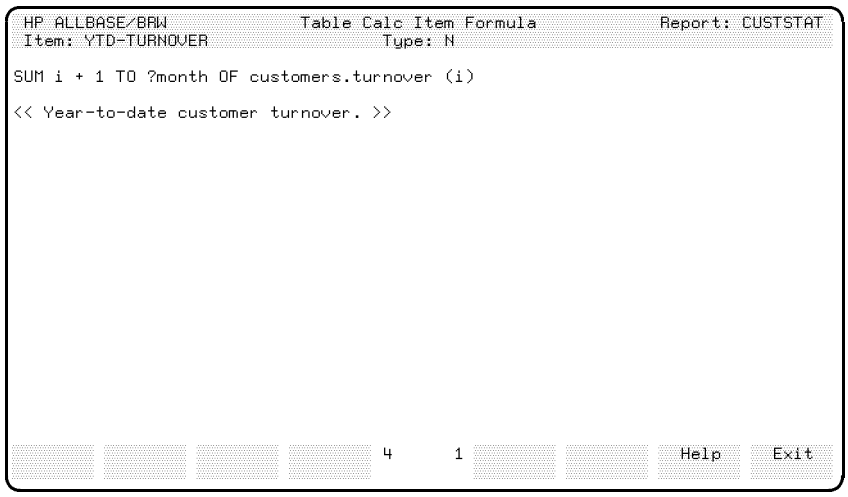
This is illustrated in the following examples:

If calculation results in:	with this numeric precision:	the result with changed precision is:
1024.3849	-2R	1024.38
864186.46	3C	865000.00

See the description of Real Numbers in Chapter 4 for a description of Real output forms and precision.

Table Calc Item Formula Screen

Use this screen to define a formula for a table calculated item. This screen consists of two blank pages on which you can type a formula.



To define a formula

1. Type the calculation formula.

   Leave blanks around any operators in the formula.

   You can use all terminal editing and arrow keys when typing and editing a formula.

   Use the (Prev Page) or (Next Page) keys on the terminal to define the next page of the formula.

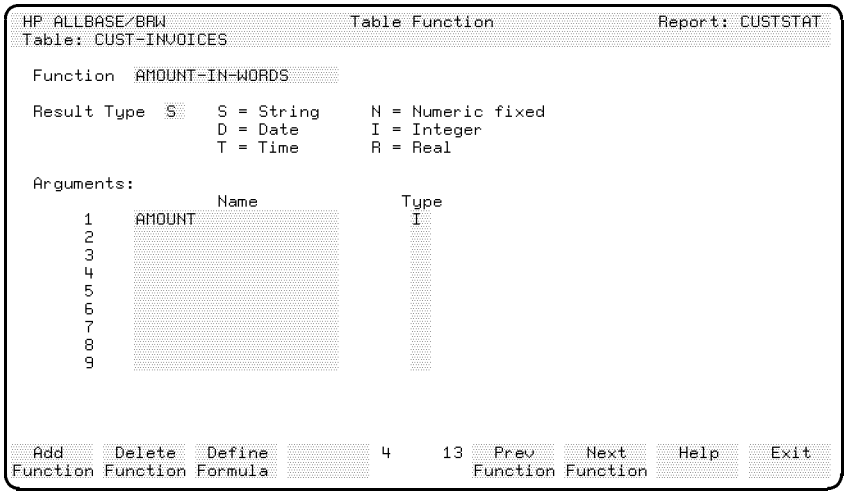
2. Press (Enter).

   The formula is checked for correctness.

   If an error is detected, the part of the formula with the error is highlighted. Correct the error and press (Enter) again. Or, Choose `Exit Save` to keep the formula as it is for later correction.

Table Function Screen

If you use a particular calculation several times in a table, use this screen to define this routine as a table function.



To add a new function

1. Type the name of the Function, Result Type, the Function Argument and Type.
2. Choose `Add Function`.

To copy a function

1. Type the name of the new function and press (Clear Display).
2. Choose `Add Function`.
3. When you see the prompt, type the name of the function from which you want to copy and optional table name if the function is in another table. For a layout function, type >LAYOUT in the Table Name field.
4. Press (Enter). The function and its formulas are copied.

To modify a function

1. Use `Next/Prev Item` key to access the function you want to modify. You can also use QUICK BROWSE to access a function. See Chapter 1 for details about the QUICK BROWSE feature.
2. Make the required changes to the function.
3. Press (Enter).

To delete a function

1. Use `Next/Prev Item` key to access the function you want to delete. You can also use QUICK BROWSE to access a function. See Chapter 1 for details about the QUICK BROWSE feature.

2. Choose `Delete Function`.
3. Press Enter or choose `Delete Function` to confirm the deletion.

To rename a function

1. Use `Next/Prev Function` key to access the function you want to rename. You can also use QUICK BROWSE to access a function. See Chapter 1 for details about the QUICK BROWSE feature.
2. Type the new name over the old function name.
3. Make any required changes to the function definition and press Enter.
4. Press Enter to confirm the rename.

Field Descriptions

`Function` The name of the function.

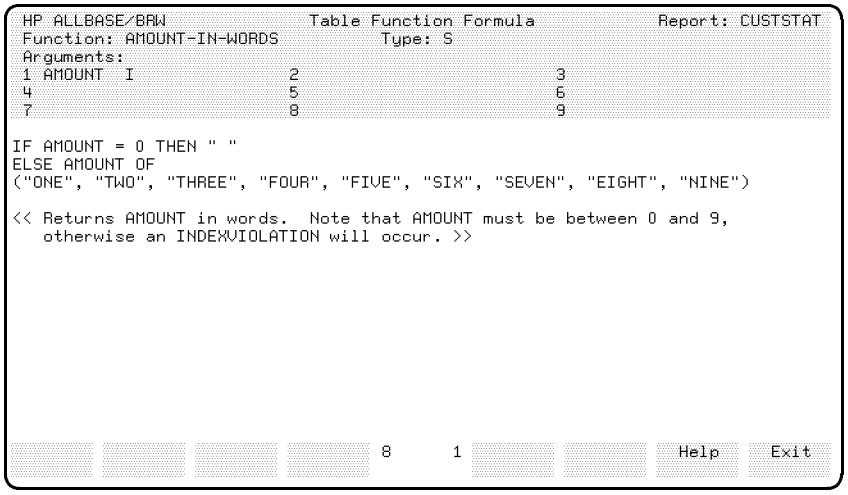
`Result Type` One of the codes listed for Result Type.

`Arguments` The formal names of the function arguments. Use these names within the function formula. Actual values for the arguments must be provided when the function is called.

`Type` Type of each argument. Use the codes listed on the screen.

Table Function Formula Screen

Use this screen to define a formula for a table function.



To define a formula

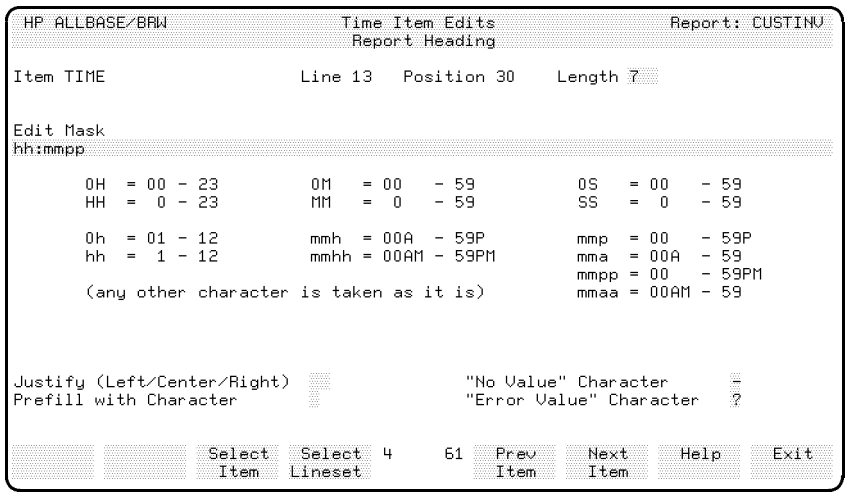
1. Type the calculation formula.
   a. Leave blanks around any operators in the formula.
   b. You can use all terminal editing and arrow keys when typing and editing a formula.
   c. Use (Prev Page) or (Next Page) terminal keys to define the next page of the formula.
2. Press (Enter).

   The formula is checked for correctness.

   If an error is detected, the part of the formula with the error is highlighted. Correct the error and press (Enter) again. Or, choose `Exit Save` to save the formula as it is for later correction.

Time Item Edits Screen

Use this screen to specify how a time item value is to be edited before it is printed in the report.



To specify a time item edit

1. Use the `Prev/Next Item` or the `Select Item` function key to access the item.
2. Type the item edits for this item.
3. Press Enter.

Banner Line

The banner line (at the top of the screen) shows in which lineset the current item is positioned.

Field Descriptions

`Item` **Display**. The current item.

`Line/position` **Display** The line and print field position of the item.

`Length` **Display**. The print field length defined for this item on the Define Lines screen. You can optionally change the length for this item.

`Edit Mask` The edit mask. Use the codes displayed on the screen for the time variables. Upper-case codes are used to indicate a 24-hour clock; lower-case for a 12-hour clock.

`Justify` **Optional**. To change the justification, type one of these codes:

L	left justify
C	center
R	right justify

Default is that the value is left-justified.

`Prefill with Character` **Optional**. The character to be printed instead of leading blanks before the item value.

`"No Value" Character` **Optional**. The print field is filled with this character if a resulting item has no value.

`"Error Value" Character` **Optional**. The print field is filled with this character if an error value occurs.

Softkeys

`Select Item` Prompts for the name of the item to be accessed.

`Select Lineset` Prompts for the level and type of lineset to be accessed. You respond with:

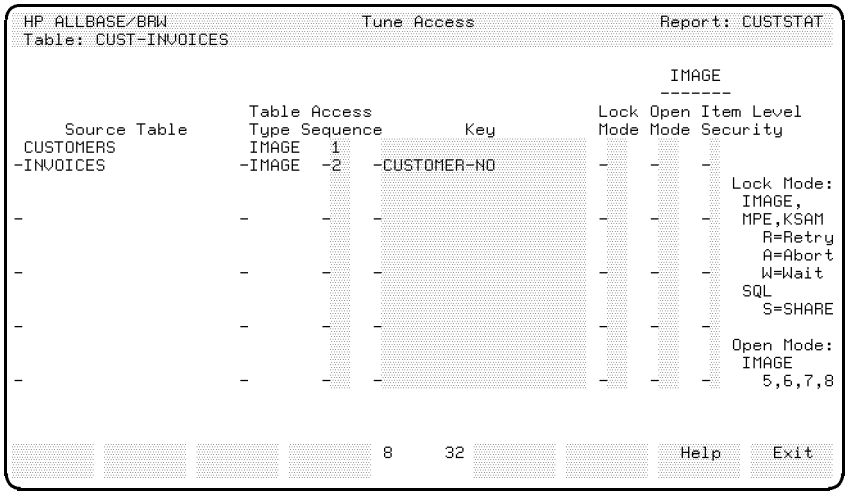
P (for page)	plus H (page heading) or F (page footing)
R (for report)	plus H (report heading) or F (report footing)
1 - 9 (break level)	plus H (break heading) or F (break footing)
D (for detail line)	

The Item Edits screen for the selected lineset and the current item appears. The current item is the item that appears in the `Item` field.

`Prev Item`
`Next Item` To access another item in the lineset. As you access other items, the screen will change to reflect the item type, for instance, if you access a date item, the screen becomes the Date Item Edits screen.

Tune Access Screen

Use this screen to specify how data is to be retrieved from the source tables.



To define how the data is to be retrieved

1. Type the Access Sequence, Key, Lock Mode (Open Mode and Item Level Security for IMAGE data sets) for each source table.
2. Press Enter.

Banner Line

Shows for which table you are tuning access.

Field Descriptions

Source Table

Display. All source tables defined for this table.

Table Type

Display. The table type for each source table (IMAGE, SQL, KSAM, MPE, BRW, IRF = Intermediate Report File).

Access Sequence

The order in which the source tables are to be accessed. Note that the source table on which an open join is made must be accessed last. SQL tables belonging to the same DBEnvironment can be combined into one query by specifying the same access sequence number for each table. This is the default if the tables are listed consecutively on the Define Table screen and no open join is specified.

If different numbers are specified for HP ALLBASE/SQL tables in the access sequence number field, then separate queries will be generated.

Key **Optional**. The name of the key from the following table, if the report requires keyed access. If this field is left blank, files are read serially. There is no keyed access to HP ALLBASE/SQL tables.

Source Table Type	Key	Result
IMAGE detail	search item	chained read
IMAGE master	search item	calculated read
KSAM file	key element	read by value
Anything but HP ALLBASE/SQL	TABLE-REC-NUMBER	directed read

Items defined as relations can be used as keys. If a key is made up of several items, all of them must be defined as relations on the Define Relation screen.

Note

If TABLE-REC-NUMBER is to be used as a key to a source table, a relation must also be defined with TABLE-REC-NUMBER for that source table and an other item which provides the record number.

If keyed access is used for the first source table in the access sequence, a relation must be defined in which a single-value or list-of-values parameter provides the values for that key.

Lock Mode **Optional**. The type of lock desired. Use the following two tables:

Locks on IMAGE data sets, or files

R Retry N number of times every T seconds, where N and T are configuration variables. See Chapter 17 for details about configuration variables.

A Try once, then stop.

W Wait until lock is granted. *Caution:* This may cause deadlocks if more than one lock is specified with keyed access!

Locks on an HP ALLBASE/SQL table

S Allow other users to read but not change the table. A lock on an HP ALLBASE/SQL table is always unconditional, that is, HP ALLBASE/BRW will wait until the lock has been acquired.

`Open Mode`

Optional. (IMAGE only) The mode to be used by HP ALLBASE/BRW when opening an HP IMAGE/3000 database. HP ALLBASE/BRW will use DBOPEN with this mode when opening the database. The possible DBOPEN modes are:

Access Mode	Capabilities
5	Read, allow concurrent modify
6	Read, allow concurrent modify
7	Read, exclusive
8	Read, allow concurrent read

Consult the *HP IMAGE/3000 Data Base Management Reference Manual* for a full description of the DBOPEN modes and capabilities. (The default DBOPEN mode is 5, unless it has been altered on the Execution Values screen when configuring HP ALLBASE/BRW. See Chapter 17.)

It is not possible to have two different database open modes for data sets that are in the same data base, and on the same Define Tables screen, and that have the same password.

Note

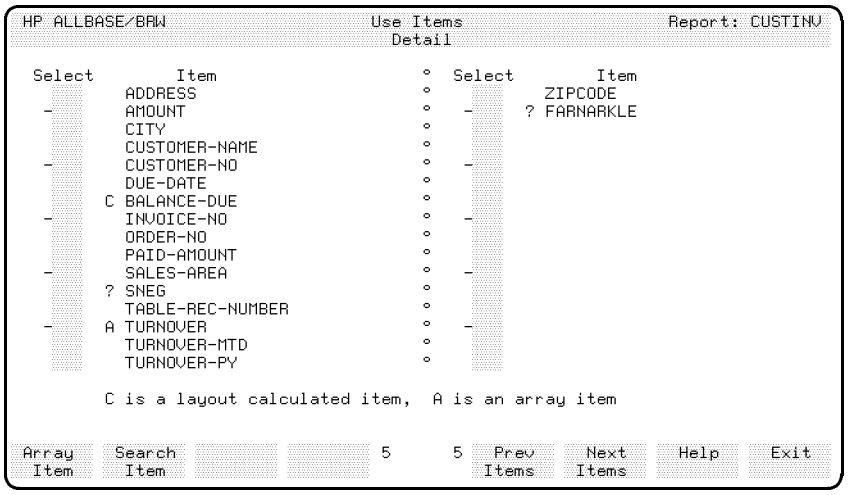
HP ALLBASE/BRW opens an HP ALLBASE/SQL DBEnvironment with the CONNECT command. The DBEnvironment parameters "Autostart Mode" and "User Mode" must be set by the DB Administrator to ON and MULTI, respectively.

`Item Level Security`

Optional. (IMAGE only) By default, HP ALLBASE/BRW reads all items in an IMAGE data set, and the password on the Define Table screen for that data set must give access to all items in the data set. If your data set has security at item level, type an `X` in this field and HP ALLBASE/BRW will only try to read the items needed for that report, and not all the items in the data set. In this case the password on the Define Tables screen need give access only to those items needed in the report.

Use Items Screen

Use this screen to select the items that you want to appear in the current lineset of your report. These items are from the final access table, layout calculated items and parameters.



To include an item in your report

1. Type an 'x' or a number between 1 and 99 in the Select field adjacent to the items that you want to include.

   HP ALLBASE/BRW includes in the lineset those items that you have marked with an 'x', followed by those items that you have marked with numbers.

2. Press Enter.

The items selected will appear on the Define Lines screen and Output File screen as non-positioned items.

To include an array element in your lineset

1. Choose `Array Item`.
2. Type the name of the array item at the prompt.

   You need to type only enough characters to uniquely identify the array name.

   If you leave the prompt blank, HP ALLBASE/BRW displays the elements of the next array item in the final access table.

3. Press Enter.

   When HP ALLBASE/BRW displays the array elements, type an 'X' or a number between 1 and 999 in the Select field adjacent to the element that you want to include.

4. Press Enter.

To search for a particular item or array

1. Choose `Search Item`.
2. Type the name of the item at the prompt.

   You need to type only enough characters to uniquely identify the item name.
3. Press (Enter).

If no item matches the characters you enter, HP ALLBASE/BRW positions the cursor at the next item, in ASCII order, that begins with character following those you entered. If no item name begins with characters following those you entered, HP ALLBASE/BRW positions the cursor at the end of the list of items.

If you leave the prompt blank, HP ALLBASE/BRW positions the cursor at the first item in the final access table.

Banner Line

The banner line (at the top of the screen) shows in which lineset the items will be postitoned.

Field Descriptions

`Select` An 'X' or a number between 1 and 999 beside the item that you want to include.

`Item` The name of the item.

Softkeys

`Array Item` Prompts for the name of the array to display.

`Search Item` Prompts for the name of the item to search for.

`Prev Items` Displays the previous screenful of items (if any).

`Next Items` Displays the next screenful of items (if any).

Array Item Softkeys

When HP ALLBASE/BRW is displaying the elements of an array item, it displays the following Softkeys:

`All Items`	Press this to redisplay all the items in the final access table.
`Search Element`	Prompts for the number of the element that you want to position the cursor at.
`Prev Array`	Displays the elements of the previous array (if any).
`Next Array`	Displays the elements of the next array (if any).
`Prev Elements`	Displays the previous screenful of elements in this array (if any).
`Next Elements`	Displays the next screenful of elements in this array (if any).

17

Configuration and Security

System, account, and individual group level configuration and security

At a Glance

You change the configuration of HP ALLBASE/BRW with the configuration program BRWSETUP. Here are the specific things that you can configure using BRWSETUP:

- Name of the default print device
- The number of pages after which a new spool file is started
- The maximum size of print files on a disk
- The maximum size of execution work files
- Open mode, lock retries, and wait intervals for IMAGE databases
- Transaction isolation levels and transaction priority for HP ALLBASE/SQL databases
- Whether to round .5 up or down
- The execution values used for output files

In addition to basic configuration tasks, the system administrator or system manager must manage and guard the location of source tables and source table passwords for data security. Password capabilities and protections are already built in to HP ALLBASE/BRW and you can also define a program called BRWPASSEXIT that can help you further maintain the security on your data.

This chapter covers

- configuration levels
- defining configuration
- running the BRWSETUP configuration program
- security features
- creating and using BRWPASSEXIT

Configuration Levels

Configuration can be defined on the system, account, or individual group level. There are three types of configuration you can set:

- Configuration defaults that affect all reports when **run** under the configuration
- Configuration defaults that only affect reports **created** under the configuration
- Global functions that can be used by any report when specified or compiled under the configuration

System-Level Configuration

HP ALLBASE/BRW has one system-level configuration file for each user interface language. The file names are:

BRWC000.PUB.SYS for language number 0 (NATIVE-3000)

BRWC001.PUB.SYS for language number 1 (American English)

BRWC*xxx*.PUB.SYS for other languages

Use LISTF BRWC@.PUB.SYS to see which files you have on your system.

If you have more than one configuration file in PUB.SYS, HP ALLBASE/BRW uses the configuration file of the user interface language defined for your system. HP ALLBASE/BRW uses Native Language Support intrinsics when handling messages.

Account and Group-Level Configuration

To establish a configuration for a group or account

1. Copy the system configuration file you use (depending on your user interface language) into the account or group you want to configure. Call the new configuration file BRWCONF in the group or account, for example:

   ```
   FCOPY FROM=BRWC000.PUB.SYS;TO=BRWCONF.PUB;NEW
   ```

2. Use the program BRWSETUP to set up the configuration for that account. (See "Running BRWSETUP" later in this chapter.)

HP ALLBASE/BRW looks for the BRWCONF file in this order:

- BRWCONF.*logon-group.logon-account*
- BRWCONF.PUB.*logon-account*
- BRWCONF.PUB.SYS
- BRWC*xxx*.PUB.SYS (where *xxx* is the Native Language Support number returned by the Native Language Support intrinsics)
- BRWC000.PUB.SYS

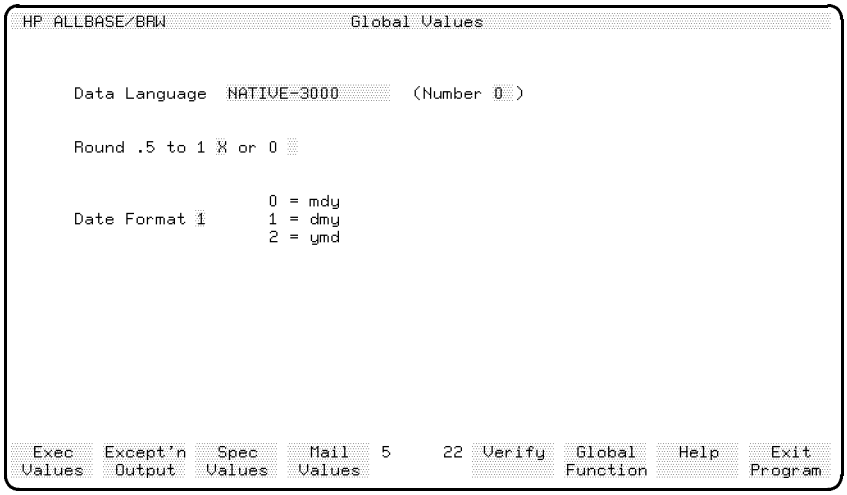
Running BRWSETUP

To run the configuration program for HP ALLBASE/BRW

1. Log on in the group and account of the configuration file you want to change.
2. Run the configuration program, as shown here:

   ```
   RUN BRWSETUP.PUB.SYS
   ```

   The first configuration screen appears.



3. Modify the configuration for data language, rounding, and data format. See the details for the Global Values Screen, later in this chapter.
4. Use the other softkeys on this menu to go to the configuration screen or screens you want to change.
5. Modify the configuration for each screen as necessary. The following sections of this chapter describe all the configuration screens for this program.

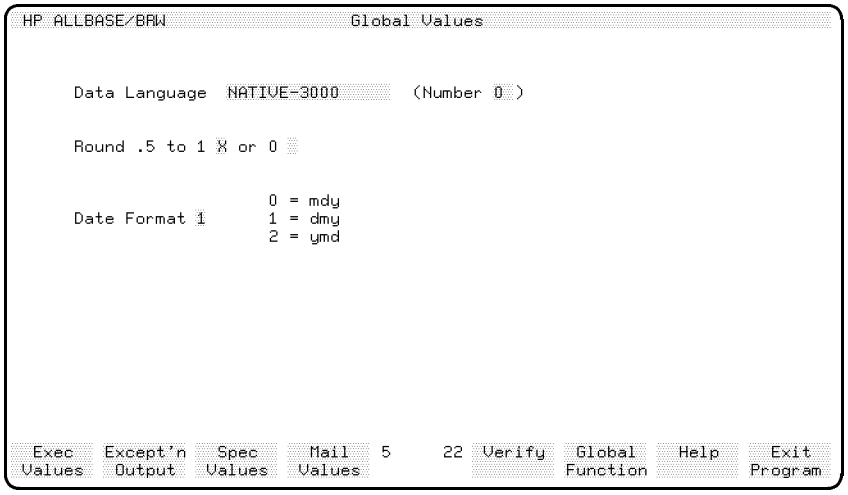
To exit the configuration program

1. Choose `Exit Program`.

The modifications are saved in the configuration file and are active for the BRWEXEC and BRWXL programs. (They are also active for BRWCOMP if you use this program to compile your reports.)

Global Values Screen

Use this screen to specify the data language, rounding rule, and input date format to be used by HP ALLBASE/BRW. This is the first screen you see when you run BRWSETUP.



To modify global values

1. Type the data language or number, the rounding rule, and the date format to be used. See the Field Descriptions section for details on each field.
2. Press Enter

To verify MPE V Compatibility

1. Choose `Verify`.

   The system checks for functions with names greater in length than 16 characters.

Field Descriptions

`Data Language` This field is used to specify the data language to be used. Type the name of the language, or leave the field blank and type the language number only.

`(Number )` The language number. When you press Enter, the system provides the value for Data Language. This entry allows you to specify sorting according to the native language rules for the data language. (Sorting is described on the Specification Values screen in this chapter.)

The data language specified here can be temporarily overwritten for a session or job by specifying the JCW (job control word) NLDATALANG.

`Rounding .5 to 1 or 0`

Type `X` after `.5 to 1` to configure HP ALLBASE/BRW so that 0.5 values are rounded up. Or, enter `X` after `0` so values are rounded down.

For example, if you have specified the numeric precision -2R for an item, the values are rounded as shown in these examples:

Value	Round 0.5 up	Round 0.5 down	Comment
54.865000	54.87	54.86	different result
54.864999	54.86	54.86	same result
54.865001	54.87	54.87	same result

Precision of numbers is described in Chapter 15, "Calculations".

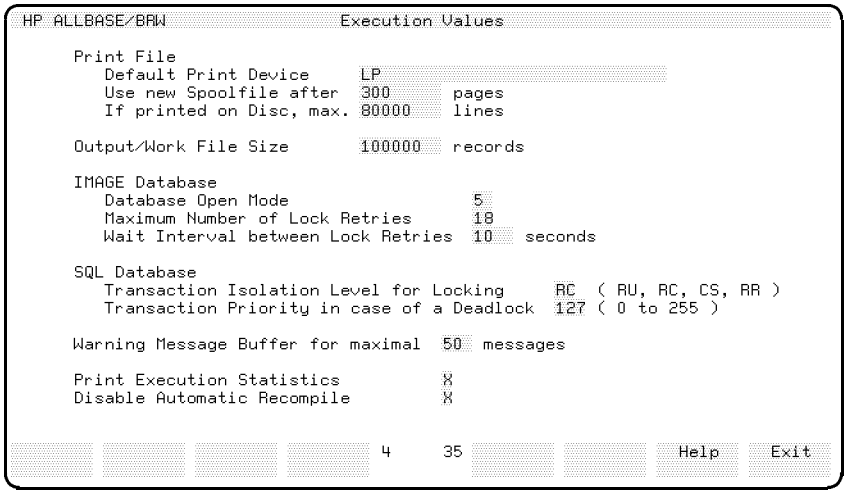
`Date Format`

Type the number of the date format to be used for all date inputs.

```
0 = mdy (month, day, year)
1 = dmy (day, month, year)
2 = ymd (year, month, day)
```

Execution Values Screen

Use this screen to specify the print file configuration and internal configuration for BRWEXEC. Get the this screen by choosing `Exec Values` at the Global Values screen.



1. Type printfile and other BRWEXEC internal configuration details.
2. Press Enter.

Field Descriptions

Print File fields

`Default Print Device`

Type the ID for the device on which reports should usually be printed. (The default is `LP`.)

Once you type the device ID in this field, it becomes the default for all reports. However, users can override this default device by specifying another device on the Define Selection screen or on the Request Report screen.

`Use new Spoolfile after ... pages`

To avoid spool file overflow, enter the maximum number of pages a spool file can have. The default is 300. When the limit is reached, BRWEXEC prints this message:

```
****************************
* reached spool limit      *
* next spool file is #0173 *
****************************
```

and uses a new spool file. At the beginning of the new spool file BRWEXEC prints this message:

```
************************************
* continuation of spool file #0172 *
************************************
```

You can estimate the number of pages which fit into a spool file with the following formula:

$$\#\ \text{pages} = \frac{\text{sectors per spool file extent} * 7500}{\text{report width} * \text{non-blank lines per page}}$$

The spool file size is configured for your system as the number of sectors per spool file extent. A spool file can have up to 32 extents. Other limiting factors are the overall number of sectors for spool files configured for the system and the size of the free space portions on the discs. For each spool file extent, there must be continuous free space on the disc.

If printed on Disc, max ... lines Type the maximum number of lines for reports printed to disk. The default is 80,000.

Output/Work File Size Type the maximum number of records in temporary execution files which are used to store the resulting data of the report's table. The default is 100000.

IMAGE Database fields

The fields described here are for IMAGE data only.

Database Open Mode Type the mode to be used by HP ALLBASE/BRW when opening an IMAGE database. HP ALLBASE/BRW will use DBOPEN with this mode when opening an IMAGE database. (However, you can alter the DBOPEN mode for an individual database by using the Tune Access screen when you define the tables for a report.) The possible modes are:

Access Mode	Capabilities
5	Read, allow concurrent modify
6	Read, allow concurrent modify
7	Read, exclusive
8	Read, allow concurrent read

Consult the *IMAGE Data Base Management Reference* manual for a full description of the DBOPEN modes and capabilities.

Maximum Number of Lock Retries

Type the number of times BRWEXEC should retry if a report locks a dataset or a file with lock mode "retry." BRWEXEC tries the lock until it is granted or until the configured number of retries is exceeded.

Wait Interval between Lock Retries

Type the number of seconds to elapse between retries. The default is 10 seconds. See Chapter 6, Defining Data Access, for details about data retrieval methods.

HP ALLBASE/SQL Database fields

The fields described here are for HP ALLBASE/SQL data only.

Transaction Isolation Level for Locking

Use this field to specify the locking strategy.

- RU = no locks while reading the data.
- RC, CS = locks the data currently being accessed by BRW while reading; locks are not kept until the end of the query.
- RR = locks the data accessed by BRW for the duration of the query.

The default is RC. This field may be overwritten for a single report on the Define Report screen. Refer to Chapter 14, "The HP ALLBASE/SQL Interface" for more information on locking.

Transaction Priority in case of Deadlock

Use this field to specify a priority value . If a deadlock occurs, HP ALLBASE/SQL will roll back a transaction, based on the priority value, to free the lock, and if HP ALLBASE/BRW is the transaction chosen to roll back, it aborts, displaying as error message. The SQL default value of 127 is recommended. Values range from 0 to 255 with smaller numbers having a higher transaction priority. Should your report have precedence over other DBE applications, adjust the Transaction Priority on the configuration screen and run the report again. It is not necessary to recompile the report.

Note

HP ALLBASE/BRW opens the HP ALLBASE/SQL DBEnvironment with the CONNECT command. The DBEnvironment parameters `Autostart Mode` and `User Mode` must be set by the DB Administrator to ON and MULTI, respectively.

Warning Message Buffer for maximal \messages

Type the maximum number of warning messages to be held in the buffer. If error values occur when a report is executed, the BRWEXEC program prints appropriate warning messages at the end of a report. During execution, warning messages are collected in a buffer in memory.

Print Execution Statistics

Optional. Type an X in this field if you want to print the full execution statistics. If this field is blank, only the Report Summary (one line) is printed.

Disable Automatic Recompile

Optional. If this is field blank, HP ALLBASE/BRW compares the time-stamp on the report with the time-stamp on the BRWDIC file and automatically recompiles the report, when the report is executed, if the BRWDIC file was created since the report was last run. The use of this feature is also discussed in Chapter 18, "Preparing a Data Dictionary".

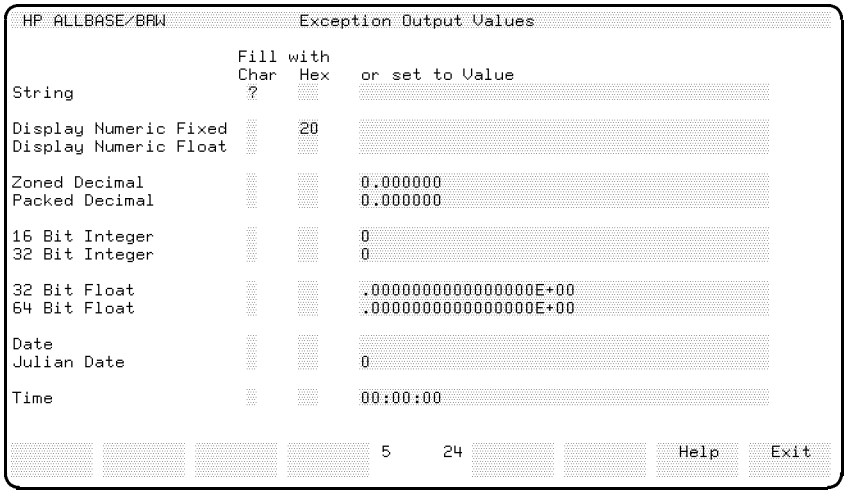
Remember that if the automatic recompile is disabled, which is the default, HP ALLBASE/BRW cannot catch a BRWDIC change that affects the report.

Note

The defaults set in this section are held in the configuration file and so affect all reports that use the configuration file.

Exception Output Values Screen

Use this screen to define values written to a dictionary output file if an item results in NO_VALUE or an error value. To get to this screen, choose `Except'n Output` on the Global Values screen.



To define exception output values

1. Type an exception value for each of the data types displayed on the screen.

   For each data type, specify a `Char` or `Hex` value, or type a `set to Value`. The value you specify is used if the item results in NO_VALUE or an error value. Each data type can have an entry in only one of the three fields.

2. Press Enter.

Field Descriptions

The left column of the screen lists the data types for HP ALLBASE/BRW. Type a `Char` or `Hex` fill value (as shown below) for each data type, or type a specific value.

`Char` Type the character. The field will be filled with the specified character.

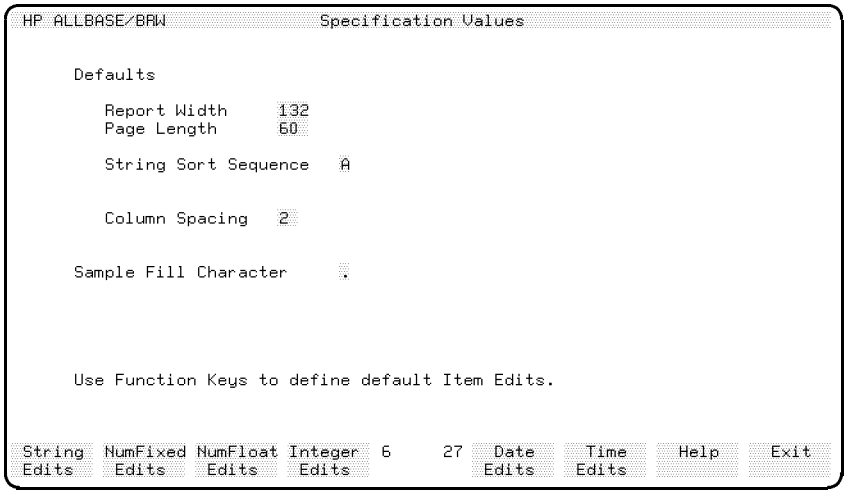
`Hex` Type a Hex digit between 00 and FF. This can be used to fill a field with non-printable ASCII characters.

`or set to Value` Type a value. The table below explains how different data types are converted in HP ALLBASE/BRW.

Data Type	HP ALLBASE/BRW Data Type
String	String in HP ALLBASE/BRW
Display Numeric Fixed	Numeric numbers in HP ALLBASE/BRW
Display Numeric Float	Real numbers in HP ALLBASE/BRW
Zoned Decimal	Numeric numbers in HP ALLBASE/BRW
Packed Decimal	Numeric numbers in HP ALLBASE/BRW
16 Bit Integer	Integer numbers in HP ALLBASE/BRW
32 Bit Integer	Integer numbers in HP ALLBASE/BRW
32 Bit Float	Real numbers in HP ALLBASE/BRW
64 Bit Float	Real numbers in HP ALLBASE/BRW
Date	Date in HP ALLBASE/BRW
Julian Date	Date in HP ALLBASE/BRW
Time	Time in HP ALLBASE/BRW

Specification Values Screen

Use this screen to define default report specifications. Use the function keys to specify default item edits. To get to this screen, choose `Spec Values` at the Global Values screen.



This screen shows the HP ALLBASE/BRW defaults. Print width, page length, and sort method can be changed for individual reports on the Define Report screen for that report. Defaults for item edits can be changed for individual reports on the Define Lines screen for each type of item edit in the report.

Note

If you change the HP ALLBASE/BRW defaults here, they become the defaults for all new reports.

Changing the defaults for report width, page length, sort method, or item edits *does not* affect the settings already stored in the specification files for existing reports.

To change the default specifications

1. Type the desired value in the appropriate fields.
2. Press Enter.

Field Descriptions

`Report Width` Defines the default report width. The default is 132 characters.

`Page Length` Defines the default number of lines per page. The default is 60 lines.

`String Sort\Sequence` Use this field to specify the default sort-character-set for strings. The possible entries are:

A Sort according to the ASCII character set.

E Sort according to the EBCDIC character set.

N Sort according to the Native Language Rules for the Data Language specified on the Global Values screen.

The ASCII sort is faster than the National Language sort, but it sorts characters according to their ASCII values rather than by their alphabetic values. The Data Language can be temporarily overwritten for a session or job by specifying the JCW (job control word) NLDATALANG.

`Column Spacing` Defines the number of blank spaces inserted between items when the specified line position is given as >. The default is 2.

`Sample Fill\Characters` Used on Define Lines to reserve space in a report for printing lengths greater than the defined Edit mask. You can enter a blank. The default is a point (.).

Function Keys

Use function keys to access the configuration screens for each type of item edit. The HP ALLBASE/BRW defaults are shown for each of these screens later in this chapter.

Default Item Edits Screens

There are six Item Edits screens you can configure in BRWSETUP, one for each item type. These configuration screens correspond closely to the Item Edit screens in HP ALLBASE/BRW. If you need additional help in filling out these screens, press f8 to get help, or see the section for that screen in Chapter 16.

The next subsections show and describe the default configuation for each item edit screen.

To change these defaults

1. Type the new default in the field and press Enter.

Default String Edits Screen

This is the screen for string item edits. To get this screen, choose `String Edits` on the Specification Values screen.

```
 HP ALLBASE/BRW            Default String Item Edits

Item SAMPLE-STRING          Line      Position       Length 0

Shift String       U = ALL UPPERCASE
                   L = all lowercase
                   I = Initial Capitals

Edit Mask


           X = character

            any other character (including x) is taken as it is


Justify (Left/Center/Right)                "No Value" Character      -
Prefill with Character                     "Error Value" Character   ?


                                   7     15                 Help     Exit
```

The default for the edit mask is blank. The `Shift String` field is also blank, but the default is that character strings are accepted as they are, that is, no shifting occurs. Justification is to the left. No character is prefilled.

See the "String Item Edits Screen" section in Chapter 16 for more details about these fields.

Default Numeric Item Edits Screen

This is the screen for numeric item edits. To get this screen, choose `NumFixed Edits` on the Specification Values screen.

```
 HP ALLBASE/BRW              Default Numeric Item Edits

Item SAMPLE-FIXED           Line      Position      Length 0

Scaling  0    Decimals  2          Rounding       (Truncate/Round/Ceiling)

                 positive                negative      Adjust
Prefix     *                      *-                     X
Suffix     *                      *                      X

Edit Mask  *ZZZZZZZZZZZZ9                                     integer part
           *.99                                               fraction part
            9 = digit   Z = digit without leading/trailing zeros

When zero print text
           *                                                  integer part
           *                                                  fraction part

Justify (Left/Center/Right)                   "No Value" Character       -
Prefill with Character                        "Error Value" Character    ?
                                              "Field Overflow" Character <
          *use ^ characters for trailing blanks

                                    6     10                 Help     Exit
```

The default for the edit mask is that it has 6 decimal places, a point (period), a 9 in the first place left of the decimal point, and twelve Z's.

```
ZZZZZZZZZZZZ9.999999
```

Keep in mind that the actual item edit-mask for a numeric item is taken, as far as possible, from the data dictionary definition. The rules for building a numeric item edit-mask are:

- If the BRWCONF default (the default on this screen) is blank, the item size and number of decimal place is taken from the dictionary.
- If the BRWCONF default has a value (like the default value shown), the number of decimals is taken from the dictionary as long as the dictionary definition does not *exceed* the BRWCONF default.

  For example, if the default on this screen has six decimal places, but the dictionary definition of the item has three decimal places, then the actual item edit-mask will have three decimal places. But, if the dictionary definition has eight places, the actual edit-mask will have only 6 places (conforming to the maximum number shown here).

Default Real Item Edits Screen

This screen shows the defaults for the Real Item Edits (floating-point real number items). To get this screen, choose `NumFloat Edits` on the Specification Values screen.

```
HP ALLBASE/BRW              Default Real Item Edits

Item SAMPLE-FLOAT           Line      Position      Length 0


             w  d
Format:    D   .     exponent format       e.g.   .4500D+02

           E 22.16   exponent format       e.g.   .4500E+02

           F   .     fixed point format    e.g.   45.00

           G   .     exponent or fixed point format

           I         integer format        e.g.   45


Justify (Left/Center/Right)                 "No Value" Character       -
Prefill with Character                      "Error Value" Character    ?
                                            "Field Overflow" Character <

                                     8     14                 Help     Exit
```

The default is E 22.16. For a full description of what this format means, and for a description of the other fields on this screen, see the "Real Item Edits Screen" section in Chapter 16.

Default Integer Item Edits screen

This screen shows the defaults for the Integer Item Edits. To get to this screen, choose `Integer Edits` on the Specification Values screen.

```
 HP ALLBASE/BRW             Default Integer Item Edits

Item SAMPLE-INT             Line      Position      Length 0

Scaling  0    Decimals  0          Rounding      (Truncate/Round/Ceiling)

                 positive                 negative      Adjust
Prefix    *                       *-                       X
Suffix    *                       *                        X

Edit Mask *ZZZZZZZZ9                                            integer part
          *                                                     fraction part
           9 = digit   Z = digit without leading/trailing zeros

When zero print text
          *                                                     integer part
          *                                                     fraction part

Justify (Left/Center/Right)                  "No Value" Character       -
Prefill with Character                       "Error Value" Character     ?
                                             "Field Overflow" Character  <
          *use ^ characters for trailing blanks

                                    6     10                  Help     Exit
```

For integer item edits, the default is that the mask has no decimals, a 9 in the rightmost place, and Z's fill the remaining places in the field. The 9 indicates one digit. The Z's indicate digits without leading or trailing zeros.

ZZZZZZZZ9

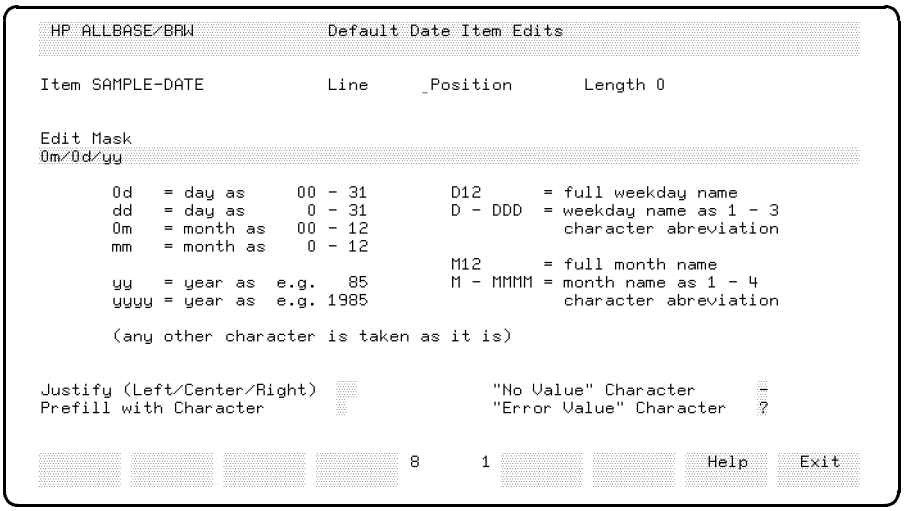
Note

The maximum number of digits for an integer is 10.

For a description of the other fields on this screen, see the "Integer Item Edits Screen" section in Chapter 16.

Default Date Item Edits screen

This screen shows the defaults for the Date Item Edits. To get to this screen, choose `Date Edits` on the Specification Values screen.



The default for date items is 0m/0d/yy. For details on the edit mask format or for other fields on this screen, see the "Date Item Edits Screen" section of Chapter 16.

Default Time Item Edits Screen

This screen shows the defaults for the Time Item Edits. To get to this screen, choose `Time Edits` on the Specification Values screen.

```
 HP ALLBASE/BRW              Default Time Item Edits

Item SAMPLE-TIME            Line      Position       Length 0

Edit Mask
HH:0M

       0H  = 00 - 23        0M   = 00   - 59          0S   = 00   - 59
       HH  =  0 - 23        MM   =  0   - 59          SS   =  0   - 59

       0h  = 01 - 12        mmh  = 00A  - 59P         mmp  = 00   - 59P
       hh  =  1 - 12        mmhh = 00AM - 59PM        mma  = 00A  - 59
                                                      mmpp = 00   - 59PM
       (any other character is taken as it is)        mmaa = 00AM - 59


Justify (Left/Center/Right)               "No Value" Character     -
Prefill with Character                    "Error Value" Character   ?

                                    8      1                    Help     Exit
```

For time items, the default if HH:0M. For details on the edit mask format or for other fields on this screen, see the "Time Item Edits Screen" section of Chapter 16.

Global Functions

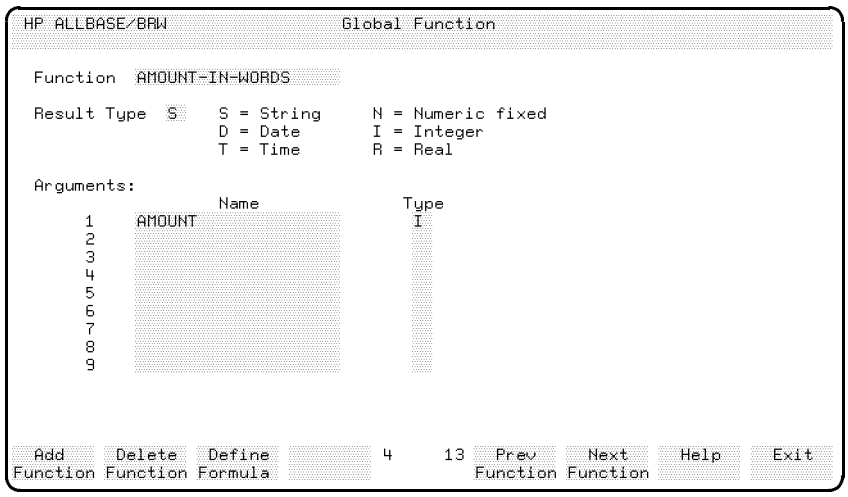
Global functions are very similar to table and layout functions but are defined for use by more than one report.

The Global Function screen and the Global Function Formula screen are used to define Global Functions in the BRWSETUP program.

For information on proper function syntax for the formula, refer to the Syntax Diagrams in Chapter 15.

Global Functions Screen

Use this screen to define a calculation that you want to use in several reports. Get the this screen by choosing `Global Functions` on the Global Values screen.



To add a new function

1. Type the name of the Function, the Result Type, the Function Argument, and the Type.
2. Choose `Add Function`.
3. Choose `Define Formula` to go to the Global Function Formula screen. See the Global Function Formula Screen section of this chapter for details on using this screen.

To copy a function

1. Type the name of the new function and press the CLEAR DSPLY key on the terminal keyboard.
2. Choose `Add Function`.
3. When the system prompts, type the name of the function you want to copy.
4. Press Enter. The function and its formulas are copied.

To modify a function

1. Use `Next/Prev Function` key to access the function you want to modify. You can also use QUICK BROWSE to access a function. See Chapter 1 for details about the QUICK BROWSE feature.
2. Make the required changes to the function.
3. Press Enter.

To delete a function

1. Use `Next/Prev Function` key to access the function you want to delete. You can also use QUICK BROWSE to access a function.
2. Choose `Delete Function`.
3. Press Enter or choose `Delete Function` to confirm the deletion.

To rename a function

1. Use `Next/Prev Function` key to access the function you want to rename. You can also use QUICK BROWSE to access a function.
2. Type the new name over the old function name.
3. Make any required changes to the function definition and press Enter.
4. Press Enter to confirm the rename.

Field Descriptions

`Function` The name of the function.

`Result Type` One of the codes listed for Result Type.

`Arguments` The formal names of the function arguments. Use these names within the function formula. Actual values for the arguments must be provided when the function is called.

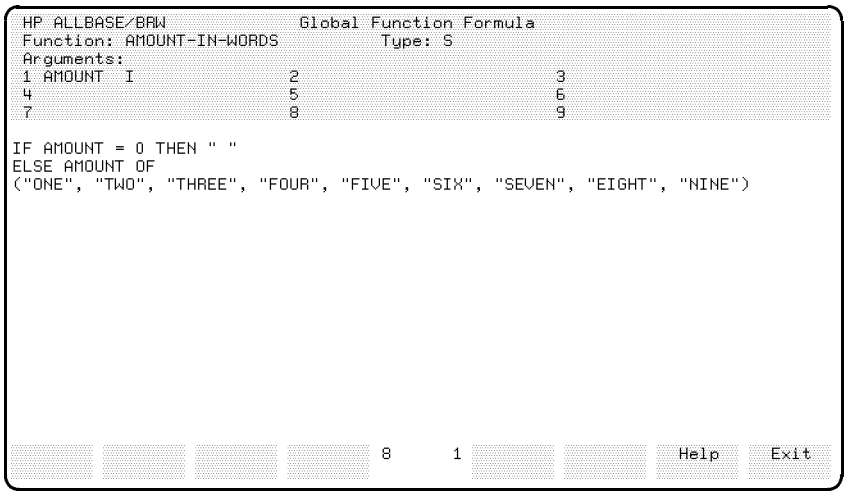
`Type` Type of each argument. Use the codes listed on the screen.

Soft Keys

`Add Function`	Adds a new function.
`Delete Function`	Deletes the current function.
`Define Formula`	Displays the Global Function Formula screen.
`Prev/Next Function`	These keys bring the previous or next function information to the screen.

Global Function Formula Screen

Use this screen to define a global function formula. Get to the screen by chooseing `Define Formula` on the Global Values screen.



Use this screen to define a formula for a global function.

To define a formula

1. Type the calculation formula.

   Leave blanks around any operators in the formula.

   You can use all terminal editing and arrow keys when entering and editing a formula.

   Use (Prev/Next Page) terminal keys to define the next page of the formula.

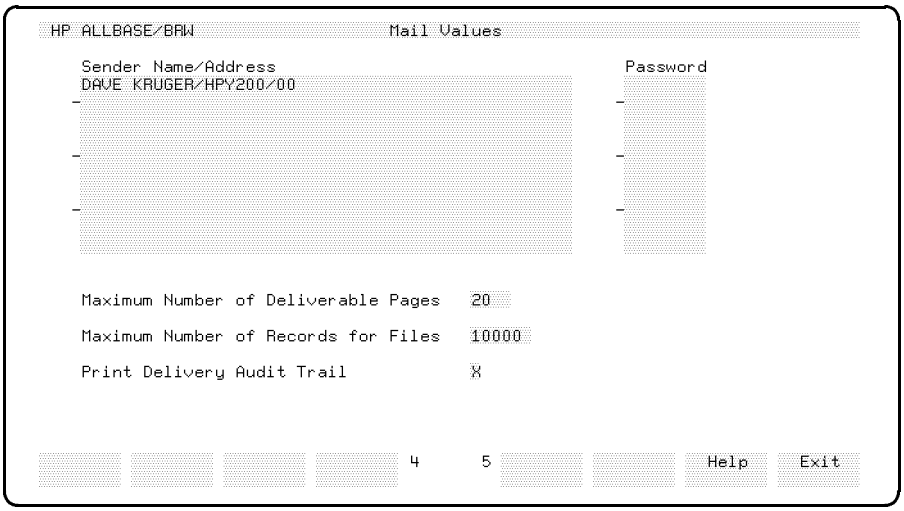
2. Press (Enter).

   The system checks the formula for correctness.

   If the system detects an error, the part of the formula with the error is highlighted. Correct the error and press (Enter) again. Or, press `Exit Save` to save the formula as-is for later correction.

Mail Values Screen

Use this screen to define default specifications for HP BRW-Desk/XL. Get to this screen by choosing `Mail Values` on the Global Values screen.



Note that this is only used if you have BRW-Desk/XL. You must specify a global sender name and desk address. Other global values most likely to be needed are the maximum number of deliverable pages and the maximum number of records for files. You can also decide whether you want to always print a delivery audit trail or not. See Chapter 13 for more information about HP BRW-Desk/XL.

1. Type the sender name and desk address in the `Sender Name/Address` field.
2. Type a password, if necessary, for the sender.
3. To change the default specifications, type the desired value in the appropriate fields. See the field descriptions on the next page.
4. Press Enter.

Field Descriptions

Sender Name/Address

This is the HPDESK user name/address that is used when signing on to HPDESK. This address is not validated at specification time. You can specify up to 9 different senders, and a minimum of 1 sender is required. HP BRW-Desk/XL will use this pool of senders to do the sign on to HPDESK. If it cannot sign on with the first sender, the next sender will be used until the sign on is successful or until the configured number of senders is exceeded.

Password

Optional. This is the password of the user signing on. Note that HPDESK passwords are case-sensitive; that is, if the password is PeTeR, and you supply PETER in the Password field, HPDESK will not let HP BRW-Desk/XL log on.

These passwords are encrypted in the configuration file.

Maximum Number of Deliverable Pages

Optional. This option allows the user to place a limit on the size of the message that can be sent to HPDESK. This limit is counted in report pages. If a message exceeds this limit, the message will not be submitted to HPDESK and the delivery audit trail will show that fact. The default is the maximum size of an item permitted in the HPDESK system.

Maximum Number of Records for Files

Optional. Use this field to set the maximum number of records for output files that can be generated and sent via HP BRW-Desk/XL. If the number is exceeded, the HP ALLBASE/BRW execution program (BRWEXEC) will abort. The default is the maximum size of an item permitted in the HPDESK system.

Print Delivery Audit Trail

Optional. Blank out the X in this field if you do not want to print the delivery audit trail messages (a record of the mailing process that is generated whenever a report using HP BRW-Desk/XL is run). The default is an X in this field: that is, to print the delivery audit trail.

Security Features

Note

The BRWPASSEXIT program is not necessary with SQL tables. The reason for this is that there are no passwords for SQL tables. HP ALLBASE/SQL security is established through access rights set in the HP ALLBASE/SQL System Catalog.

Password Encryption

HP ALLBASE/BRW encrypts passwords in HP ALLBASE/BRW files. For example, it will encrypt the password of a report specification file. You will not notice this feature when using HP ALLBASE/BRW, as it is designed only to prevent passwords being read by people not using HP ALLBASE/BRW.

BRWLIST

BRWLIST does not print report specification passwords.

$TELLREPORT

You can use the parameter $TELLREPORT in your Report Job Files. This is exactly the same as $REPORT, except that it will not display the report lockword/password (if any) for the report. For example, if you displayed the message "Job for report $REPORT started" using the MPE TELL command, both the report and its password/lockword would be displayed. However, if you changed the message to "Job for report $TELLREPORT started", only the report name would be displayed on the terminal screen by the TELL command.

Dictionary Passwords

Some data dictionaries hold passwords for files/datasets on the system. These passwords are not copied into the HP ALLBASE/BRW dictionary file (BRWDIC) by the dictionary conversion programs. You need to enter these passwords for each report in the Define Table screen.

Automatic Password Retrieval BRWPASSEXIT

HP ALLBASE/BRW can use a user-supplied program, called BRWPASSEXIT, to specify both the actual location names for source tables and the source table passwords. The location name for a source table is the full name of the database or file where report data is to be found, with optionally the group and account where the database or file is kept. For example, the short name for a dataset used as a source table could be INVOICES. Its location name is database TOYDB, in group PUB, of account MYACCT.

Why use BRWPASSEXIT?

You may not want all your users to know the location names and passwords for the databases. BRWPASSEXIT can give users access to the databases so they can write or run report, but not allow them to change the database in any way. Using BRWPASSEXIT can enhance your data security and make password maintenance simpler, especially if you change passwords often.

Note

All database passwords must allow read access to the datasets specified, or at least to the items needed in the report if item level security is defined. So if the password does not allow read access, the DBGET will fail. You can, of course, restrict update access. You do not need "@" access if you use item level security.

Creating BRWPASSEXIT

Note

Someone familiar with the database structure, like the database manager, needs to be responsible for writing the BRWPASSEXIT program. BRWPASSEXIT mst be capable of accepting the short name and returning the location and password. Examples of typical BRWPASSEXIT data structures are given later.

To create BRWPASSEXIT

1. Build the BRWPASS executable library (XL) where you

```
:LINKEDIT

  linkEd > BUILDXL BRWPASS
  linkEd > ADDXL FROM=<your passexit object file>;&
         TO=BRWPASS
  linkEd > EXIT
```

   For more information on the Linkeditor, see the HP LinkEditor Reference Manual.

2. Write the BRWPASSEXIT program and include it in the executable library.

   You must write the program BRWPASSEXIT and compile it in native mode. The program must follow these rules:

a. The program must not write over any HP ALLBASE/BRW data or code! (This is very unlikely if you code the program in PASCAL or COBOL, but possible if you use ASSEMBLER)

b. The procedure BRWPASSEXIT takes a record structure with the following fields as a parameter:

Location name:	Packed array of 30 characters (Input/Output)
Report name:	Packed array of 30 characters (Input/Output)
User, Group, Account:	3 packed arrays of 8 characters (Input)
Password:	Packed array of 8 characters (Output)
Common area:	Array of 100 words (Output)
Status:	1 word (Input/Output)

The status parameter ('status' or 'RETURN-STATUS ') can have two values:

0 : The password is being returned successfully.

-1 : The user is not allowed to access the password.

Note  For examples of the data structures for these parameters, see the next sections.

c. BRWPASSEXIT must return status=0 (and blanks for the password) for any locations that are not known to BRWPASSEXIT.

3. Compile BRWPASSEXIT in native mode.

To use BRWPASSEXIT

1. When you specify the report, do not type a password for a source table on the Define Tables screen.
2. When you run the report, HP ALLBASE/BRW looks for the BRWPASSEXIT program in the BRWPASS extended library (XL) in this order:

   a. BRWPASS.*logon group*.*logon account*

   b. BRWPASS.PUB.*logon account*

   c. BRWPASS.PUB.SYS

If BRWPASSEXIT is not found, the call is skipped and the report will run with the assumption that the user is the creator of the source table. That is, HP ALLBASE/BRW tries to open the databases using a database password of ";" and tries to open files using blanks for passwords.

If your BRWPASSEXIT procedure does not recognize the Location Name, or does not want the given user, group, or account to have access to the location, it should return a status=-1 (or any value besides zero). A non-zero status from BRWPASSEXIT will cause report execution to stop immediately.

If BRWPASSEXIT is found, and access to the password and location of the table is allowed, then the report is executed. The return status=0.

Note

HP ALLBASE/BRW will not call BRWPASSEXIT if you supply an incorrect password: BRWPASSEXIT will only be called when no password is supplied. If the password (or location name) is incorrect the report will abort without calling BRWPASSEXIT.

BRWPASSEXIT Examples

PASCAL Example

```
$subprogram 'brwpassexit'$
$OS 'MPE/XL'$

{Compiler directives to compile only BRWPASSEXIT}

shortint  = -32768..32767;
t_pac8    = PACKED ARRAY [1..8] OF char;
t_pac30   = PACKED ARRAY [1..30] OF char;
a_comarea = ARRAY [1..100] OF shortint;

fmt_user_rec = RECORD

location    :  t_pac30;
reportname  :  t_pac30;
user        :  t_pac8;
group       :  t_pac8;
account     :  t_pac8;
password    :  t_pac8;
comarea     :  a_comarea;
status      :  shortint;

END; {RECORD}

procedure BRWPASSEXIT (var passexit_rec : fmt_use_rec);
```

PASCAL

This procedure gets the correct location and password for a source table, accepts the "short" passed location and returns the full location and password (status=0) or indicates that access is not allowed (status=1). This example is shown for simplicity.

You could also obtain an integer by decoding the location and use that as a CASE ordinal, or as a key file of passwords. A sample program is shown on the next page.

```
label 9999;

begin

  with passexit_rec do
  begin

    if location = 'ADB' then
      begin
      location :='ADB.ADBGROUP.ADBACCT';
      password := 'READADB'; {read access only required}
      status :=0; {location recognized:}
      goto 9999;
      end;
{ and so on through the various short locations }

{PAYROLL database only accessible to people logged on to the FINANCE
account.  For anyone else, reject access.}

if location = 'PAYROLL' then
  if account = 'FINANCE' then
    begin
    location := 'PAYROLL.PUB.FINANCE';
    password := 'BUCKS';
    status := 0;
    goto 9999;
    end
  else
    begin
    status := -1;
    goto 9999;
    end;

{ and so on through those short locations, if any, where password access is not permitted.}

   status := 0;

{If the location is NOT RECOGNIZED, set the status to 0, then BRW will
try to read the file using the default location and no password. }
   9999:
  end; {with}
 end;  {procedure brwpassexit}

 {main program outer block}
 begin
 end.
```

COBOL Example

Compile the program:

```
:COB85XL %%progname%%
```

```
 LINKAGE SECTION

01  COM-RECORD.
  05  LOCATION           PIC X(30).
  05  REPORTNAME         PIC X(30).
  05  LOGON-USER         PIC X(8).
  05  LOGON-GROUP        PIC X(8).
  05  LOGON-ACCOUNT      PIC X(8).
  05  PASSWORD           PIC X(8).
  05  COMAREA     OCCURS 100 TIMES PIC S9(4) COMP.
  05  RETURN-STATUS      PIC S9(4) COMP.
```

COBOL Parameter

The figure on the next page shows a COBOL sample program.

```
$CONTROL DYNAMIC
 IDENTIFICATION DIVISION
 PROGRAM-ID. BRWPASSEXIT.
 AUTHOR.
 DATE-WRITTEN.
 DATE-COMPILED.
 ENVIRONMENT DIVISION.
 DATA DIVISION.
 LINKAGE SECTION.
 01 COM-RECORD.
  05 LOCATION            PIC X(30).
  05 REPORTNAME PIC X(30).
  05 LOGON-USER PIC X(8).
  05 LOGON-GROUP PIC X(8).
  05 LOGON-ACCOUNT PIC X(8).
  05 PASSWORD PIC X(8).
  05 COMAREA  OCCURS 100 TIMES PIC S9(4) COMP.
  05 RETURN STATUS PIC S9(4)  COMP.

 PROCEDURE DIVISION USING COM-RECORD.

 P1

  IF LOCATION = "ADB" THEN
   MOVE "ADB.ADBGROUP.ADBACCT" TO LOCATION
   MOVE "READADB" TO PASSWORD
   MOVE 0 TO RETURN-STATUS
   GO TO BRW-END.

  IF LOCATION = "BDB" THEN
   MOVE "TOYDB.PUB.ITF3000" TO LOCATION
   MOVE "doctor" TO PASSWORD
   MOVE 0 TO RETURN-STATUS
   GO TO BRW-END.

  IF LOCATION = "PAYROLL" THEN
    IF LOGON-ACCOUNT = "FINANCE" THEN
      MOVE "PAYROLL.PUB.FINANCE" TO LOCATION
      MOVE "BUCK" TO PASSWORD
      MOVE 0 TO RETURN-STATUS
      GO TO BRW-END
    ELSE
      MOVE -1 TO RETURN-STATUS
      GO TO BRW-END.

 MOVE 0 TO RETURN-STATUS

BRW-END
```

18

Data Dictionaries

Creating an HP ALLBASE/BRW dictionary file (BRWDIC)

At a Glance

HP ALLBASE/BRW requires that the structure of the data (for example, in databases and files) you use when you define reports must be described in a data dictionary. HP ALLBASE/BRW uses the information in the data dictionary to construct its own dictionary file, called BRWDIC. The BRWDIC file contains all the data-description information HP ALLBASE/BRW needs to create, compile, and execute reports.

Note

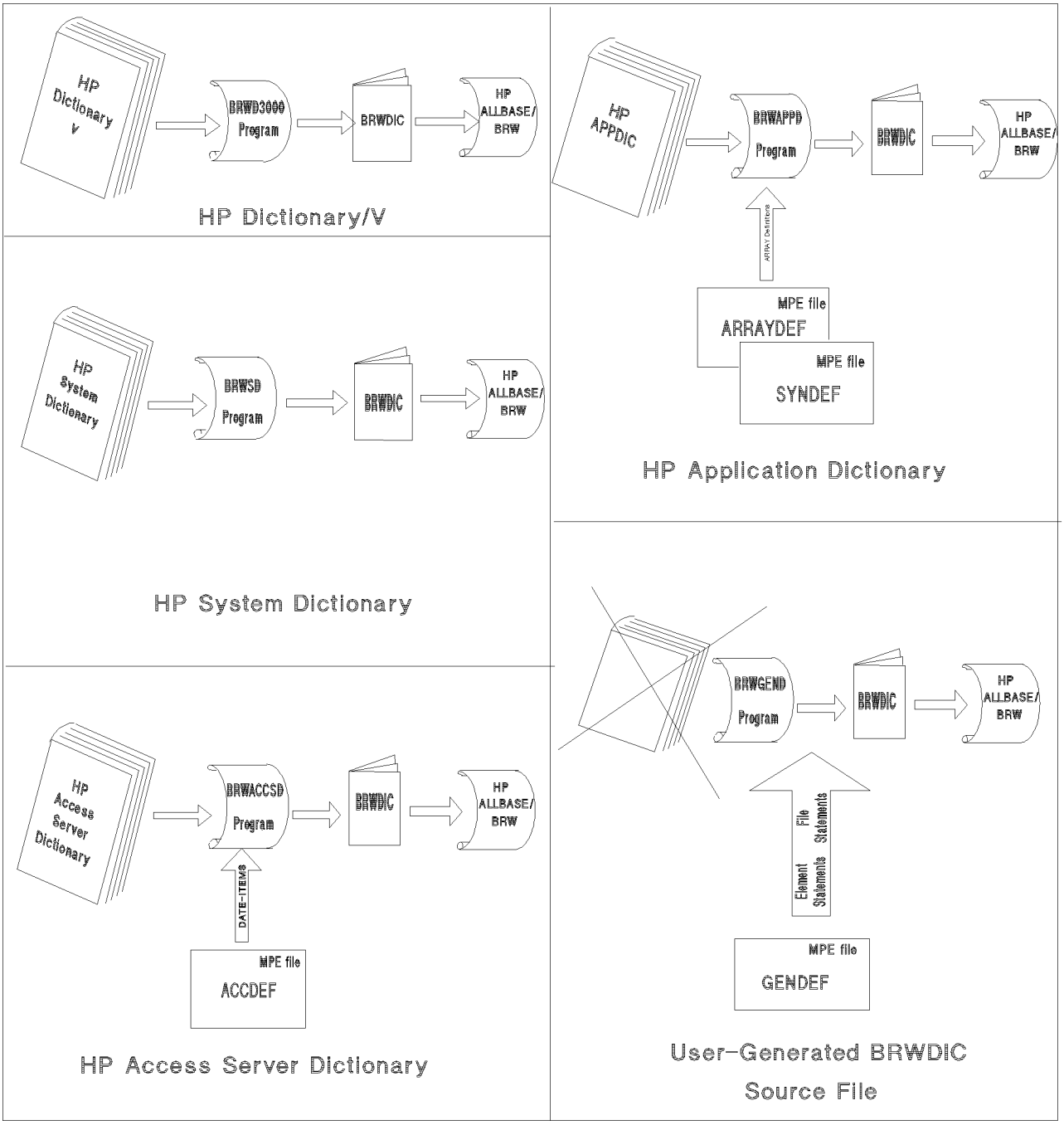
A data dictionary is not required for HP ALLBASE/SQL data. HP ALLBASE/BRW goes directly to the HP ALLBASE/SQL's system catalog for information on data structures. If a dictionary is created, HP ALLBASE/BRW does not use it for HP ALLBASE/SQL data.

The BRWDIC can be created from any of these dictionaries, using the HP ALLBASE/BRW program listed:

Dictionary	Program
HP Dictionary/3000	BRWD3000.PUB.SYS
HP System Dictionary	BRWSD.PUB.SYS
HP Application Dictionary	BRWAPPD.PUB.SYS
HP Information Access Server	BRWACCSD.PUB.SYS

Note

If you don't have one of the above dictionaries, you can put your dictionary into a text file and use BRWGEND.PUB.SYS to generate your own user-defined dictionary.



Using Data Dictionaries

This chapter covers

- HP ALLBASE/BRW dictionary file (BRWDIC)
- HP Dictionary/3000
- HP System Dictionary
- HP Application Dictionary
- HP Information Access data dictionary
- user-generated BRWDIC source file
- changes in dictionary files

HP ALLBASE/BRW Dictionary File (BRWDIC)

These are the major features of BRWDIC:

- It supports HP Dictionary/3000, HP System Dictionary, HP Application Dictionary, HP Information Access data dictionary, and user-generated dictionaries.
- It lets you use multiple data dictionaries in one report. All five data dictionary generators—HP Dictionary/3000, HP System Dictionary, HP Application Dictionary, HP Information Access data dictionary, and user-generated dictionaries—can be used in combination.
- Reports can be automatically recompiled after changes to the BRWDIC.

Removing the Data Dictionary

HP ALLBASE/BRW programs (other than the dictionary generators) use only the BRWDIC files and never access the original dictionaries, so the data dictionaries can be removed from the system after the BRWDIC is created.

Also, if you want to run reports on another HP 3000 that does not have HP ALLBASE/BRW installed, see "Requesting Reports with BRWEXEC" in Chapter 11 for information on running HP ALLBASE/BRW reports on other HP 3000s.

Rules for creating and modifying BRWDIC

- You cannot directly modify a BRWDIC dictionary file. That is, if the data dictionary is altered, you must prepare a new BRWDIC file from the modified data dictionary.
- BRWDIC accepts names up to 20 characters in length (since that is the maximum that HP ALLBASE/BRW can accept).

Using BRWDLIST

You can print out the contents of the BRWDIC using the program BRWDLIST. This will print the information to a temporary file called RPRINT.

To print the contents of BRWDIC

1. Run BRWDLIST:

```
:RUN BRWDLIST.PUB.SYS
```

2. Save the RPRINT file.

```
:SAVE RPRINT
```

Using Multiple Data Dictionaries or a Central Dictionary

You can combine data described in several dictionaries in a single report. This is especially useful if you want to combine data from different applications in one report. Typically, each application uses its own dictionary. Also, it is typical that each application uses a separate account on the system. Therefore, each account that contains databases or files used in reports must have its own BRWDIC dictionary.

Which BRWDIC file does HP ALLBASE/BRW use?

HP ALLBASE/BRW accesses the BRWDIC file in the PUB group of the account for the database or file named on the Define Table screen. If you name multiple databases or files from different accounts, HP ALLBASE/BRW accesses the corresponding BRWDIC file in the account where each database or file resides. In other words, HP ALLBASE/BRW accesses the appropriate BRWDIC file for each of the databases or files, as long as you have one dictionary in each account.

What about a Central Dictionary?

If you have one central BRWDIC dictionary that you want to be used for all databases, regardless of which account they are in, you can use file equations in the logon UDCs of each of the three accounts. Here is a sample of some file equations you could customize for your accounts:

```
FILE BRWDIC.PUB = BRWDIC.PUB.DICTACCT
FILE BRWDIC.PUB.FINANCE = BRWDIC.PUB.DICTACCT
FILE BRWDIC.PUB.PRODUCTN = BRWDIC.PUB.DICTACCT
FILE BRWDIC.PUB.ORDERS = BRWDIC.PUB.DICTACCT
```

With these file equations, you could use the databases and files in the FINANCE, PRODUCTN, and ORDERS accounts and they would all access the central dictionary in the DICTACCT. If you wanted to prevent the PRODUCTN account from having access to files in the FINANCE account, omit the file equation for the FINANCE account when you create the logon UDCs for the PRODUCTN account.

Another solution would be to create a BRWDIC file and copy it to each of the accounts. However, since the BRWDIC file is generally quite large, this may not be the best solution.

Multiple BRWDIC in the same account

HP ALLBASE/BRW always looks in the PUB group of the account where the data is located (given on the Define Table screen) for BRWDIC.

For two or more BRWDICs to exist in the same account, they must be given different names, as shown in the steps below:

To have two BRWDIC files in the same account

1. Create the first BRWDIC file, using any of the dictionary generators:

   ```
   :RUN BRWxxx.PUB.SYS
   ```

   The result is a file named BRWDIC.PUB.*logonaccount*.

2. To create the second BRWDIC file, first use a file equation and then run the dictionary generator:

   ```
   :FILE BRWDIC.PUB = BRWDICx.PUB
   :RUN BRWDICxxxx.PUB.SYS
   ```

   The result is a file named BRWDIC*x*.PUB.*logonaccount*.

3. Remove the file equation for BRWDIC.PUB using the RESET command:

   ```
   :RESET BRWDIC.PUB
   ```

4. Use two file equations to simulate a separate account for one of the data sources and for one of the BRWDIC files:

   ```
   :FILE yourdb.yourgrp.dummyaccount = yourdb.yourgrp
   :FILE BRWDIC.PUB.dummyaccount = BRWDICx.PUB
   ```

   The dummy account does not need to be an existing account.

   You can include this file equation in the logon UDC for the account where the data sources are located. Or, you can include the file equations in the report job file so that the file equation is set when the job logs on. See Chapter 11 for a description of report job files.

Note

When you specify the *yourdb* data source on the Define Table screen, you must use the *dummyaccount* in the pathname. For instance, if your file equation is:

```
FILE TOYSDB.PUB.FINANCEX = TOYSDB.PUB
```

Then you would specify `TOYSDB.PUB.FINANCEX` in the location field on the Define Table screen.

Leave the account blank when specifying data defined in the BRWDIC that is not equated to the dummy account. HP ALLBASE/BRW looks for BRWDIC.PUB and finds BRWDIC.PUB in the logon account.

HP Dictionary/3000

Setting up HP Dictionary/3000

See the HP Dictionary/3000 manual for a full explanation of HP Dictionary/3000.

Use the following programs to set up HP Dictionary/3000:

DICTINIT — For each dictionary you want, run this program to create an empty dictionary.

DICTDBD — This program loads the data dictionary with the item and dataset definitions of an IMAGE database. It defines every item, but does not know about decimal places or date and time items. These must be completed using DICTDBM.

DICTDBM — Since DICTDBD loads only an extract from the database's root file, you must also run DICTDBM to specify the number of decimal places for numeric elements and add date and time item formats. That is, use DICTDBM to refine and complete the dictionary definitions, if needed.

Terminology

The terminology in HP Dictionary/3000 and HP ALLBASE/BRW is not always the same. The table below shows the difference in terminology in these two products:

HP ALLBASE/BRW	HP Dictionary/3000
Item	Element
Format	File/Element relation
File	MPEF, MPER, KSAM file types
Dataset	DETL, AUTO, MAST file types
Database	Parent database BASE file type
Location	File/Location relation

In this section the terminology used is that of HP Dictionary/3000. For example, the term **element** is used instead of **item**. (However, date and time items are HP ALLBASE/BRW constructions only, so they are described as date and time **items**, rather than date and time **elements**.)

Elements/Items

For the element (item) name, follow these rules:

- Use a maximum of 20 characters for the element name, unless the report is to run with BRW/V, in which case you can use a maximum of only 16.
- Use a letter as the first character of the element name.
- Do not include spaces, dots, commas and parentheses in the element name.
- For existing elements whose names conflict with the above rules, define alias names. If an element has an alias name, that name is used by HP ALLBASE/BRW.

HP ALLBASE/BRW supports the following element data types:

Strings	Maximum length of 255 characters
Numeric elements	Maximum of 13 integer digits and 6 decimal places
Packed numeric elements	Maximum of 13 integer digits and 6 decimal places, with Trailing Overpunch or No Sign
Zoned numeric elements	Maximum of 13 integer digits and 6 decimal places, with Trailing Overpunch or No Sign
16-Bit integers	
32-Bit integers	
64-Bit integers	(type N in HP ALLBASE/BRW)
Integers with decimals	(type N in HP ALLBASE/BRW)
32-Bit and 64-Bit Real numbers	Real numbers can be either HP 3000 or IEEE format. You can (optionally) tell HP ALLBASE/BRW which format a Real number element is stored in by adding the long-name !REAL-HP 3000 or !REAL-IEEE to that element. If there is no long name, HP ALLBASE/BRW assumes real numbers in IMAGE databases are HP 3000 real numbers, and real numbers in MPE and KSAM files are IEEE real numbers.
HP Business BASIC floating-point decimal (D)	This type converts to fixed-point numeric in HP ALLBASE/BRW. Note that the range of numeric numbers is + or - 9(13).9(6). If a type D number overflows, it will be set to OVERFLOW; if it underflows it will be set to zero.

This table illustrates the HP Dictionary/3000 item types, and how they are handled by HP ALLBASE/BRW. A number suffix equals the number of words. For example I2 equals a 2-word (4-byte) integer.

Dictionary/3000 Type	Decimals	BRW Type	Remarks
I, I2, J, J2, K, K2	0	I (integer)	
I, I2, J, J2, K, K2	> 0	N (numeric)	
I4, J4, K4		N (numeric)	
P, Z, 9, Z+, D, P+		N (numeric)	Maximum 13 integer digits and 6 decimals
Z (sign ' ' or sign = 'TO')		N (numeric)	Trailing Overpunch assumed
Z (sign = 'TS' 'LS' or 'LO'		Not supported	Not supported
R2, R4, E2, Er		R (real)	
U, X		S (string)	Maximum length 255 bytes
X, 9, I, J, K		D (date)	Long-name begins !DATE
X, 9, I, J, K		T (time)	Long-name begins !TIME
B, S, *		Not supported	Not supported

HP ALLBASE/BRW supports both parent and child elements. You can use either the child elements alone, or the child elements and the parent element. If several parent elements have the same child element, HP ALLBASE/BRW treats all occurrences of that child element in a dataset or file as synonyms, that is, as the same element. Only the parent and child items are taken, not any intermediate child items.

String elements or numbers to be used as date or time items in HP ALLBASE/BRW are defined by specifying the date or time type in the long name of the element.

Dictionary/3000 DATE-ITEM Table (DISPLAY and INTEGER Format)

Dictionary/3000 Long Name	Length Required (1 wd = 16 bits)	Base Type	Example: (Date is 04/28/92)
!DATE-DMY-D	8 Bytes	X	28/04/92
!DATE-DMYY-D	10 Bytes	X	28/04/1992
!DATE-MDY-D	8 Bytes	X	04/28/92
!DATE-MDYY-D	10 Bytes	X	04/28/1992
!DATE-YMD-D	8 Bytes	X	92/04/28
!DATE-YYMD-D	10 Bytes	X	1992/04/28
!DATE-DDY-D	6 Bytes	X	118/92
!DATE-DDYY-D	8 Bytes	X	118/1992
!DATE-YDD-D	6 Bytes	X	92/118
!DATE-YYDD-D	8 Bytes	X	1992/118
!DATE-DM-D	5 Bytes	X	28/04
!DATE-MD-D	5 Bytes	X	04/28
!DATE-MY-D	5 Bytes	X	04/92
!DATE-MYY-D	7 Bytes	X	04/1992
!DATE-YM-D	5 Bytes	X	92/04
!DATE-YYM-D	7 Bytes	X	1992/04
!DATE-DMY	3 Words	I, J, or K	
!DATE-DMYY	3 Words	I, J, or K	
!DATE-MDY	3 Words	I, J, or K	
!DATE-MDYY	3 Words	I, J, or K	
!DATE-YMD	3 Words	I, J, or K	
!DATE-YYMD	3 Words	I, J, or K	
!DATE-DDY	2 Words	I, J, or K	
!DATE-DDYY	2 Words	I, J, or K	
!DATE-YDD	2 Words	I, J, or K	
!DATE-YYDD	2 Words	I, J, or K	
!DATE-DM	2 Words	I, J, or K	
!DATE-MD	2 Words	I, J, or K	
!DATE-DD	1 Word	I, J, or K	
!DATE-MY	2 Words	I, J, or K	
!DATE-MYY	2 Words	I, J, or K	
!DATE-YM	2 Words	I, J, or K	
!DATE-YYM	2 Words	I, J, or K	
!DATE-CALENDAR	2 Bytes	I, J, or K	
!DATE-JULIAN	4 Bytes	I, J, or K	

Dictionary/3000 DATE-ITEM Table (NUMERIC and ALPHANUMERIC Format)

Dictionary/3000 Long Name	Length Required (1 wd = 16 bits)	Base Type	Example: (Date is 04/28/92)
!DATE-DMY	6 Bytes	X or 9	280492
!DATE-DMYY	8 Bytes	X or 9	28041992
!DATE-MDY	6 Bytes	X or 9	042892
!DATE-MDYY	8 Bytes	X or 9	04281992
!DATE-YMD	6 Bytes	X or 9	920428
!DATE-YYMD	8 Bytes	X or 9	19920428
!DATE-DDY	5 Bytes	X or 9	11892
!DATE-DDYY	7 Bytes	X or 9	1181992
!DATE-YDD	5 Bytes	X or 9	92118
!DATE-YYDD	7 Bytes	X or 9	1992118
!DATE-DM	4 Bytes	X or 9	2804
!DATE-MD	4 Bytes	X or 9	0428
!DATE-DD	3 Bytes	X or 9	118
!DATE-MY	4 Bytes	X or 9	0492
!DATE-MYY	6 Bytes	X or 9	041992
!DATE-YM	4 Bytes	X or 9	9204
!DATE-YYM	6 Bytes	X or 9	199204

Dictionary/3000 TIME-ITEM Table (All Formats)

Dictionary/3000 Long Name	Length Required	Base Type	Example: (Time is 15:43:24)
!TIME-HM24	4 Bytes	X or 9	1543
!TIME-HMS24	6 Bytes	X or 9	154324
!TIME-HM12	5 Bytes	X	0343P
!TIME-HM24-D	5 Bytes	X	15:43
!TIME-HMS24-D	8 Bytes	X	15:43:24
!TIME-HM12-D	6 Bytes	X	03:43P
!TIME-HM24	2 Words	I, J, or K	
!TIME-HMS24	2 Words	I, J, or K	
!TIME-CLOCK	4 Bytes	I, J, or K	

Notes:

- The EXAMPLE column shows how the date and time are held. There is no EXAMPLE for integers because they are held as 16-bit words.
- When held as integers, date or time item types have the same Long Names as when held in numeric or alphanumeric form (X or 9). However, the base type must be an integer (I, J, or K).
- Where no year is held, (for example, in date item !DATE-DM), the year is assumed to be the current year. HP ALLBASE/BRW adds the current year when the date is accessed.
- Where only the last two digits of the year are held (for example, in date item !DATE-DMY), the century is assumed to be 1900. This is also the case for item !DATE-CALENDAR.
- Where no day is held (for example, in date item !DATE-MY), the day is assumed to be the first of the month.
- Where the day is held as DD (for example, in date item !DATE-DDY), the date is held as the number of days from the start of the year (January 1st). The example date is 04/28/87, that is the 118th day of 1987.
- You must define every item whose length is expressed in WORDS as an array of 2-byte integers. For example, to enter a 3-word integer Date Item (such as !DATE-YYMD), run DICTDBM and enter:

```
STORAGE LENGTH = 2
COUNT  = 3.
```

  The STORAGE LENGTH is always 2 and the COUNT is 1, 2, or 3 depending on the number of words. (The above item is actually held as 3 * 16-bit integers).
- !DATE-JULIAN date is a 4-byte integer. The Julian date is calculated using the CACM Algorithm 199.
- !DATE-CALENDAR and !TIME-CLOCK are in the format returned by the corresponding HP 3000 Intrinsics. They must be defined as 2- and 4-byte integers respectively.

Datasets

Datasets are defined as files of type DETL, MAST or AUTO. When specifying a file name, follow these rules:

- The file name must be the real name of the IMAGE dataset.
- Do not include spaces, points, commas and parentheses in the file name.

File/element relations describe which elements are in the dataset. Those elements which are search elements can also be defined.

If a search element consists of several components, it is called a **composite search element**. To define a composite search element you must first define an element with the name of the search element. Then use element/element relations to describe the components of the composite search element. The composite search element becomes a **parent element**, and the components become its **child elements**.

If several composite search elements have the same child element, HP ALLBASE/BRW treats all occurrences of that child element as synonyms, in other words, as the same element.

If you use parent elements as keys, you must define relations for either:

- all the child elements associated with the parent element, or,
- if you have included parent elements in the BRWDIC, you need only define a relation for the parent element.

If no long name is supplied, HP ALLBASE/BRW assumes that a real number contained in an IMAGE dataset is in HP 3000 format.

Databases

Databases are defined as files of type BASE, where the file name must be the name of the database with out group and account. File/file relations describe which datasets are in the database.

MPE Files

MPE files are defined as files of type MPEF or MPER, where the file name must be the file format name in the data dictionary without group and account. File/element relations describe which elements are in the MPE file.

HP ALLBASE/BRW only supports MPE files with fixed length records.

If no long name is supplied, HP ALLBASE/BRW assumes that a real number contained in an MPE file is in IEEE format.

KSAM Files

KSAM files are defined as files of type KSAM, where the file name must be the file format name in the data dictionary without group and account. File/element relations describe which elements are in the KSAM file.

HP ALLBASE/BRW only supports KSAM files with fixed length records.

If no long name is supplied, HP ALLBASE/BRW assumes that a real number contained in a KSAM file is in IEEE format.

Locations

Locations are defined as groups in the account of the data dictionary and are optional. File/location relations describe groups for databases, MPE files and KSAM files.

HP ALLBASE/BRW does not support accounts as locations, although HP Dictionary/3000 allows this. A database or a file in another account must also have a data dictionary in that account.

Preparing a BRWDIC Dictionary File With HP Dictionary/3000

BRWD3000 prepares a BRWDIC dictionary file from an HP Dictionary/3000-type data dictionary file.

To create BRWDIC with BRWD3000

1. Run the program.

   ```
   :RUN BRWD3000.PUB.SYS
   ```

2. Type the dictionary password.

   ```
   Dictionary Password:    not displayed
   ```

   If there is no password, type a colon (;) at the prompt. If you do not enter a password and just press Return at the prompt, the program will terminate.

3. Respond YES (Y) or NO (N) to the following prompts:

   ```
   Include Parent Elements (N/Y)?
   Use Names up to 20 Characters (Y/N)?
   ```

If you do not include parent elements in the BRWDIC, you can subsequently only use a parent element in a relation by specifying a separate relation for each of its child elements. The default is N, to NOT include parent elements.

The BRWDIC can support names up to 20 characters long. The default is Y, for names up to 20 characters long. If you do not want to use names that long, you can answer N to the prompt and use only names with a maximum length of 16 characters. In this case, names longer than 16 characters are truncated and checked to ensure that the truncated names are still unique. HP BRW/V only supports names up to 16 characters long. If you want reports to run with BRW/V, answer N. The default is to support names up to 20 characters.

BRWD3000 opens the HP Dictionary/3000 dictionary with the logical name DICT.PUB, and creates the BRWDIC file with the logical name BRWDIC.PUB. If you want to use file equations on either of these files, remember to use the logical names.

If the program encounters an end of file in the dictionary file, you can increase the file size of the dictionary file by entering a FILE command, as shown here:

```
:FILE BRWDIC.PUB;  DISC=5000
```

Then, run the BRWD3000 program again.

Warning Messages

The program sends a warning message if one of these invalid definitions for HP ALLBASE/BRW occurs:

- invalid element type
- invalid syntax in the format name
- more than 18 decimals
- format without elements

The BRWDIC is built anyway, even though these warnings may occur.

HP System Dictionary

The following programs are used to set up HP System Dictionary:

SDINIT	The System Dictionary initialization utility. For each dictionary you want, run this program to create an empty dictionary.
SDDBD	Run this program to load the IMAGE database root file. SDDBD loads the data dictionary with the element and dataset definitions of an IMAGE database.
SDMAIN	The System Dictionary program used to create, maintain, and report entries. Since SDDBD loads only an extract from the database's root file, you must also run the program SDMAIN to define the search element components; define the number of decimal places for numeric elements; define MPE and KSAM files; and add date and time items.
SDCONV	This System Dictionary utility program converts HP Dictionary/3000 definitions to HP System Dictionary.

Refer to the following books for details about how to use the System Dictionary:

- *HP System Dictionary/XL General Reference Manual* volumes 1 and 2
- *HP System Dictionary/XL Utilities Reference Manual*
- *HP System Dictionary/XL SDMAIN Reference Manual*

Extensions to HP System Dictionary

The System Dictionary contains a number of predefined entity types, relationship types, relationship classes, and attributes, known as the **core set**. These structures are created when the dictionary is initialized. HP ALLBASE/BRW supports two extensions to the HP System Dictionary Core Set:

- UNIQUE
- ELEMENT-SUBTYPE

UNIQUE

You can add the extra attribute, UNIQUE, to the following Core Set relationship type.

IMAGE-DATASET

ELEMENT

IMAGE-DATASET

IMAGE-DATABASE

chains

If you assign the attribute UNIQUE to this relationship, it will indicate that all the SEARCH-ELEMENT chains in a detail dataset have length of 1. That is, there is only one element in each chain. This saves you having to read for the end-of-chain marker for SEARCH-ELEMENT chains with only one element.

ELEMENT-SUBTYPE attributes

You can add an extra attribute, ELEMENT-SUBTYPE, wherever you can use the attribute ELEMENT-TYPE. ELEMENT-SUBTYPE is used to identify date and time item types.

To add these extensions to your dictionary

1. Add a scope name and scope password to the DEFINE statement at the beginning of the BRWSDEXT file. The scope must have extent capability (for example, your DASCOPE) so it can add the extensions to your system dictionary when the file is opened.
2. Use the file BRWSDEXT as the command input file when running SDMAIN. For example:

```
:FILE SDIN = BRWSDEXT.PUB.SYS
:RUN SDMAIN.PUB.SYS
```

Note 

If you do not add the scope password to the BRWSDEXT file, then SDMAIN will prompt for it at run time.

Terminology

The terminology in HP System Dictionary and HP ALLBASE/BRW is not always the same. The table below shows the difference in terminology in these two products:

HP ALLBASE/BRW	HP System Dictionary
Item	(Entity-type) ELEMENT
Format	One of the following relationship types: IMAGE-DATASET contains RECORD FILE contains RECORD KSAMFILE contains RECORD (the above can be linked with the relationship type: RECORD contains ELEMENT)
File	Entities of type FILE with attribute file-type = SEQUENTIAL or RELATIVE
KSAM File	Entities of type KSAMFILE with attribute ksamfile-type = DATA
Dataset	Entities of type IMAGE-DATASET with attribute image-dataset-type = DETAIL, MANUAL, or AUTOMATIC
Database	Entities of type IMAGE-DATABASE with attribute image-database-type = IMAGE or TURBO

In this section the terminology used is that of HP System Dictionary. For example, the term *element* is used instead of *item*. However, date and time items are called *items*, since date and time items are HP ALLBASE/BRW constructions.

Items/Elements

Items are defined as entities of type ELEMENT in HP System Dictionary. When defining the element name in HP System Dictionary, follow these rules:

- Use a maximum of 20 characters for the element name; however, only use a maximum of 16 characters if the report is to run with HP BRW/V. (HP System Dictionary allows 32 characters.)
- Use a letter as the first character of the element name.
- Do not include spaces, dots, commas and paren theses in the element name.

Element Alias Names

Define alias names for existing elements whose names conflict with the above rules. Whenever possible, HP ALLBASE/BRW uses alias names for elements. HP ALLBASE/BRW uses the image-alias for elements in image-datasets and the standard-alias for elements in MPE and KSAM files. HP ALLBASE/BRW uses the following logic when looking for alias names:

- The alias is taken from the *RECORD contains ELEMENT* relationship or the *ELEMENT contains ELEMENT* relationship.
- If no alias is found above, HP ALLBASE/BRW takes the alias from the ELEMENT definition.
- If no alias is found above, HP ALLBASE/BRW takes the primary name of the element.

Finding an Element's Attributes

HP ALLBASE/BRW uses the attribute back-reference-flag when looking for an element's attributes. If the back-reference-flag is FALSE, HP ALLBASE/BRW takes the attributes from the *RECORD contains ELEMENT* relationship or the *ELEMENT contains ELEMENT* relationship. If the back-reference-flag is TRUE, HP ALLBASE/BRW takes the attributes from the ELEMENT definition.

HP ALLBASE/BRW supports the following element data types:

HP ALLBASE/BRW supports the following element data types:

Strings	Maximum length of 255 characters
Numeric elements	Maximum of 13 integer digits and 6 decimal places
Packed numeric elements	Maximum of 13 integer digits and 6 decimal places, with Trailing Overpunch or No Sign
Zoned numeric elements	Maximum of 13 in teger digits and 6 decimal places, with Trailing Overpunch or No Sign
16-Bit integers	
32-Bit integers	
64-Bit integers	(type N in HP ALLBASE/BRW)
Integers with decimals	(type N in HP ALLBASE/BRW)
32-Bit and 64-Bit real numbers	Real numbers can be either HP 3000 or IEEE format. You can (optionally) tell HP ALLBASE/BRW which format a Real number element is stored in by adding the ELEMENT-SUBTYPE HP 3000 or IEEE to that element. If there is no ELEMENT-SUBTYPE, HP ALLBASE/BRW assumes Real numbers in TurboIMAGE databases are HP 3000 real numbers, and real numbers in MPE and KSAM files are IEEE real numbers.
HP Business BASIC floating-point decimal (D)	This type converts to fixed-point numeric in HP ALLBASE/BRW. Note that the range of numeric numbers is + or - 9(13).9(6). If a type D number overflows, it will be set to OVERFLOW; if it underflows it will be set to zero.

The next table illustrates the HP System Dictionary element types, and how they are handled by HP ALLBASE/BRW. A number suffix equals the number of words. For example I2 equals a 2-word (4-byte) integer.

System Dictionary Element Types

Dictionary Element Type	Decimals	BRW Type	Remarks
I, I2, J, J2, K, K2	0	I (integer)	
I, I2, J, J2, K, K2	> 0	N (numeric)	
I4, J4, K4		N (numeric)	
P, Z, 9, Z+, D, P+		N (numeric)	Maximum 13 integer digits and 6 decimals
Z (SIGN ' ' or sign = 'TO')		N (numeric)	Trailing Overpunch assumed
Z (sign = 'TS' 'LS' or 'LO')		R (real)	
R2, R4, E2, E4		R (real)	
U, X		S (string)	Maximum length 255 bytes
X, 9, I, J, K		D (date)	ELEMENT-SUBTYPE is a date
X, 9, I, J, K		T (time)	ELEMENT-SUBTYPE is a time
B, S, *		Not Supported	Not Supported

HP ALLBASE/BRW supports both parent and child elements. The relationship *ELEMENT contains ELEMENT* establishes a child element that is part of a parent element. You can use either the child elements alone, or the child elements and the parent element. If several parent elements have the same child element, HP ALLBASE/BRW treats all occurrences of that child element in a dataset or file as synonyms, that is, as the same element.

ELEMENT-SUBTYPE attributes

Elements to be used as date or time items in HP ALLBASE/BRW are defined in HP System Dictionary using the extra attribute ELEMENT-SUBTYPE. You can add an ELEMENT-SUBTYPE attribute to the entity ELEMENT, the relationship *RECORD contains ELEMENT*, and the relationship *ELEMENT contains ELEMENT*. To use ELEMENT-SUBTYPE, you must first extend your system dictionary using the file BRWSDEXT.PUB.SYS, as described earlier. You can only use ELEMENT-SUBTYPEs when the element-type of the element is X, 9, I, J, or K. (But note that you can optionally add an ELEMENT-SUBTYPE to real numbers to tell HP ALLBASE/BRW in which format (HP 3000 or IEEE) the real number is held).

System Dictionary DATE-ITEM Table (DISPLAY and INTEGER Format)

System Dictionary ELEMENT-SUBTYPE	Length Required (1 word = 16 bits)	Element Type	Example: (Date is 04/28/92)
DMY-D	8 Bytes	X	28/04/92
DMYY-D	10 Bytes	X	28/04/1992
MDY-D	8 Bytes	X	04/28/92
MDYY-D	10 Bytes	X	04/28/1992
YMD-D	8 Bytes	X	92/04/28
YYMD-D	10 Bytes	X	1992/04/28
DDY-D	6 Bytes	X	118/92
DDYY-D	8 Bytes	X	118/1992
YDD-D	6 Bytes	X	92/118
YYDD-D	8 Bytes	X	1992/118
DM-D	5 Bytes	X	28/04
MD-D	5 Bytes	X	04/28
MY-D	5 Bytes	X	04/92
MYY-D	7 Bytes	X	04/1992
YM-D	5 Bytes	X	92/04
DMY	3 Words	I, J, or K	
DMYY	3 Words	i, J, or K	
MDY	3 Words	I, J, or K	
MDYY	3 Words	I, J, or K	
YMD	3 Words	I, J, or K	
YYMD	3 Words	I, J, or K	
DDY	2 Words	I, J, or K	
DDYY	2 Words	I, J, or K	
YDD	2 Words	I, J, or K	
YYDD	2 Words	I, J, or K	

System Dictionary DATE-ITEM Table (DISPLAY and INTEGER Format) (continued)

System Dictionary ELEMENT-SUBTYPE	Length Required (1 word = 16 bits)	Element Type	Example: (Date is 04/28/92)
DM	2 Words	I, J, or K	
MD	2 Words	I, J, or K	
DD	1 Word	I, J, or K	
MY	2 Words	I, J, or K	
MYY	2 Words	I, J, or K	
YM	2 Words	I, J, or K	
YYM	2 Words	I, J, or K	
CALENDAR	2 Bytes	I, J, or K	
JULIAN	4 Bytes	I, J, or K	

System Dictionary DATE-ITEM Table (NUMERIC and ALPHANUMERIC Format)

System Dictionary ELEMENT-SUBTYPE	Length Required (1 word = 16 bits)	Element Type	Example: (Date is 04/28/92)
DMY	6 Bytes	X or 9	280492
DMYY	8 Bytes	X or 9	28041992
MDY	6 Bytes	X or 9	042892
MDYY	8 Bytes	X or 9	04281992
YMD	6 Bytes	X or 9	920428
YYMD	8 Bytes	X or 9	19920428
DDY	5 Bytes	X or 9	11892
DDYY	7 Bytes	X or 9	1181992
YDD	5 Bytes	X or 9	92118
YYDD	7 Bytes	X or 9	1992118
DM	4 Bytes	X or 9	2804
MD	4 Bytes	X or 9	0428
DD	3 Bytes	X or 9	118
MY	4 Bytes	X or 9	0492
MYY	6 Bytes	X or 9	041992
YM	4 Bytes	X or 9	9204
YYM	6 Bytes	X or 9	199204

System Dictionary TIME-ITEM Table (All Formats)

System Dictionary ELEMENT-SUBTYPE	Length Required	Element Type	Example: (Time is 15:43:24)
HM24	4 Bytes	X or 9	1543
HMS24	6 Bytes	X or 9	154324
HM12	5 Bytes	X	0343P
HM24-D	5 Bytes	X	15:43
HMS24-D	8 Bytes	X	15:43:24
HM12-D	6 Bytes	X	03:43P
HM24	2 Words	I, J, or K	
HMS24	2 Words	I, J, or K	
CLOCK	4 Bytes	I, J, or K	

Notes:

Notes for the previous tables:

- The EXAMPLE column shows how the date and time are held. There is no EXAMPLE for integers because they are held as 16-bit words.
- When held as integers, date or time item types have the same ELEMENT-SUBTYPE attributes as when held in numeric or alphanumeric form (X or 9). However, the base type must be an integer (I, J, or K).
- Where no year is held, (for example, in date item DM), the year is assumed to be the current year. HP ALLBASE/BRW adds the current year when the date is accessed.
- Where only the last two digits of the year are held (for example, in date item DMY), the century is assumed to be 1900. This is also the case for item CALEND.
- Where no day is held (for example, in date item MY), the day is assumed to be the first of the month.
- Where the day is held as DD (for example, in date item DDY), the date is held as the number of days from the start of the year (January 1st). The example date is 04/28/87, that is the 118th day of 1987.
- There is no hyphen in the ELEMENT-SUBTYPE name for the item HMS24D. (This is because the maximum length of an ELEMENT-SUBTYPE name is 6 characters).
- All items whose length is expressed in WORDS must be defined as an array of 2-byte integers. For example, to enter a 3-word integer Date Item (such as YYMD), run SDMAIN and enter:

  BYTE-LENGTH = 2 COUNT = 3

The BYTE-LENGTH is always 2 and the COUNT is 1, 2, or 3 depending on the number of words. (The above item is actually held as 3 * 16-bit integers).

- A JULIAN date is a 4-byte integer. The Julian date is calculated using the CACM Algorithm 199.
- CALEND and CLOCK are in the format returned by the corresponding HP 3000 Intrinsics. They must be defined as 2- and 4-byte integers respectively.

Keys

HP ALLBASE/BRW finds key information in the following ways:

- For master datasets: HP ALLBASE/BRW takes the key from the relationship IMAGE-DATASET key ELEMENT.
- For detail datasets: HP ALLBASE/BRW takes the key from the relationship:

```
IMAGE-DATASET
ELEMENT
IMAGE-DATASET
IMAGE-DATABASE
chains
```

- For KSAM files: HP ALLBASE/BRW takes the key from the relationship KSAMFILE key ELEMENT.

HP ALLBASE/BRW supports the extra attribute UNIQUE for the last two relationships. Use the attribute UNIQUE to indicate that all search-element chains have length 1 (second relationship), or that duplicate key entries are not allowed (third relationship).

Note

BRWDIC is not capable of holding database-specific key information. For example: a detail data set is a member of two databases and each database has a different search path to its copy of the data set. This is illustrated in the two *chains* relationships below.

```
detailset search-element1 sort-element master1 base1

detailset search-element2 sort-element master2 base2
```

HP ALLBASE/BRW will allow you to specify any of the two search-elements as a key in either database; it will not restrict the use of search-element2 to base2.

Datasets

Datasets are defined as entities of type IMAGE- DATASET with attribute image-dataset-type = DETAIL, AUTOMATIC, or MANUAL. Elements that are related to the first primary record (or, if no primary record exists, elements that are related to the first record) become the elements of the dataset.

When specifying an IMAGE-DATASET name, follow these rules:

- The file name must be the real name of the IMAGE dataset.
- Do not include spaces, points, commas and parentheses in the file name.

The relationship *IMAGE-DATASET contains RECORD* describes which elements are in the dataset. Search elements can also be defined using the relationship IMAGE-DATASET key ELEMENT.

The relationship *ELEMENT contains ELEMENT* describes the components of a composite search element. The composite search element becomes a parent element, and the components become its child elements.

If several composite search elements have the same child element, HP ALLBASE/BRW treats all occurrences of that child element as synonyms, in other words, as the same element.

If no ELEMENT-SUBTYPE is supplied, HP ALLBASE/BRW assumes that a real number contained in an IMAGE dataset is in HP 3000 format.

Databases

Databases are defined as entities of type IMAGE- DATABASE, where the entity name must be the name of the database without group and account. The relationship IMAGE-DATABASE contains IMAGE-DATASET describes which datasets are in the database.

MPE Files

MPE files (sequential or relative) are defined as entities of type FILE, where the filename must be the file format name in the data dictionary without group and account. The relationship *FILE contains RECORD* describes which elements are in the MPE file.

HP ALLBASE/BRW only supports MPE files with fixed length records. Elements that are related to the first primary record (or, if no primary record exists, elements that are related to the first record) become the elements of the file.

If no ELEMENT-SUBTYPE is supplied, HP ALLBASE/BRW assumes that a real number contained in an MPE file is in IEEE format.

KSAM Files

KSAM files are defined as entities of type KSAMFILE, where the file name must be the file format name in the data dictionary without group and account. The relationship *KSAMFILE contains RECORD* describes the record format for the KSAM file.

HP ALLBASE/BRW only supports KSAM files with fixed-length records. Elements that are related to the first primary record (or, if no primary record exists, elements that are related to the first record) become the elements of the KSAM file.

If no ELEMENT-SUBTYPE is supplied, HP ALLBASE/BRW assumes that a real number contained in a KSAM file is in IEEE format.

Using Byte-offsets

RSYSDIC cannot compute the offset of either an element within a record or an element within a group element (that is, within a *RECORD contains ELEMENT* relationship or an *ELEMENT contains ELEMENT* relationship). This is because the logon-scope might not have access to all elements in that relationship. You must therefore always take the offset of any element within such a relationship from the value of the byte-offset attribute associated with the relationship.

For example, a record contains 6 elements, and you want to find the position of the third element in the record. If your logon-scope has access to all the elements in the record, you can find the size of the first two elements, and therefore the start of the third element. If your logon-scope does not have access to either of the first two elements, you will have to find the start of the third element from the byte-offset attribute associated with the *RECORD contains ELEMENT* relationship that links the third element to the record.

(If the byte-offset attribute = 0, (the default byte-offset), the element is placed directly after the last element in the record. Therefore, if you do not assign a byte-offset in SDMAIN, all the elements in a record will take the default byte-offset attribute of 0, and will be placed one after the other.)

Defining Overlays

You can use the byte-offset attribute to define overlays of elements. (HP ALLBASE/BRW does not support the relationship *ELEMENT redefines ELEMENT*). For example: an element part-number consists of 3 letters and 3 digits (ABC123, QWE080 etc), and the 3 digits tell you the weight of the part.

You can set up two relationships:

```
part-record contains part-number; element-type= X;
length= 6; byte-offset= 1

part-record contains part-weight; element-type= 9;
length= 3; byte-offset= 4
```

The element part-weight will overlay the numeric half of part-number, and you can use part-weight in calculations.

Preparing a BRWDIC Dictionary File With HP System Dictionary

BRWSD prepares a BRWDIC dictionary file from an HP System Dictionary-type dictionary file.

To create BRWDIC with the System Dictionary

1. Run BRWSD:

```
:RUN BRWSD.PUB.SYS
```

2. Respond to these prompts. A description of the choices follows these steps.

```
External Scope Name:
Scope Password:          not displayed
Domain:                  default is the common domain
Version:                 if unknown, omit
Version Status:          only appears if version is omitted
                         choose P(roduction), T(est), or A(rchive)
Include Parent Elements (N/Y)?
Use Names up to 20 Characters (Y/N)?
```

If you press (Return) at the External Scope Name prompt, the program will terminate.

Description of choices

The default for `Version Status` is Production.

If you want to include parent elements in the BRWDIC, answer Y to the prompt `Include parent elements (N/Y)?` If you choose to include parent elements, they appear with their associated child elements on the Project Items screen and can be used in relations.

If you do not include parent elements, you can subsequently only use a parent element in a relation by specifying a separate relation for each of its child elements. The default is to NOT include parent elements.

If you want to use names in the BRWDIC, which can be up to 20 characters long, answer Y to the prompt `Use Names up to 20 Characters (Y/N)?` If you want to use only names with maximum length 16 characters, answer N to the prompt. In this case, names longer than 16 characters will be truncated and a check will be made to ensure that the truncated names are still unique. If you want reports to run with BRW/V, answer N. The default is to support names up to 20 characters.

BRWSD opens the HP System Dictionary with the logical name SYSDIC, and creates the BRWDIC file with the logical name BRWDIC.PUB. If you want to use file equations on either of these files, remember to use the logical names.

If the program encounters an end of file in the dictionary file, you can increase the file size of the dictionary file by entering a FILE command, as shown here:

```
:FILE BRWDIC.PUB; DISC=5000
```

Then, run the BRWSD program again.

Warning Messages

The program sends a warning message if one of these invalid definitions for HP ALLBASE/BRW occurs:

- invalid element type
- invalid syntax in the format name
- more than 18 decimals
- format without elements

These messages do not prevent the program from creating the BRWDIC file.

HP Application Dictionary

HP Application Dictionary is the data dictionary used with HP-customizable applications, including HP Financial Accounting, HP Financial Budgeting, Materials Management/3000, Production Management/3000, and Maintenance Management/3000. Refer to the appropriate application reference manual for more details about customizable applications.

Using HP Application Dictionary With HP ALLBASE/BRW

If you have an HP-customizable application on your system, the HP Application Dictionary is automatically included. Therefore no information on constructing the data dictionary is required here.

Caution 

However, if the files ARRAYDEF and SYNDEF do not exist on your system, you need to build them (even if they would be empty). If you have a customized application, you must build these files carefully. The next sections describe these files for you. Therefore, please read this entire section even if you already have an HP Application Dictionary data dictionary on your system.

Terminology

Terminology in HP Application Dictionary and HP ALLBASE/BRW is not always the same. The table below shows the difference in terminology:

HP ALLBASE/BRW	HP Application Dictionary
Item	Field
Format	Format
File	File
Dataset	Dataset
Database	Database
Location	Database or File

In this section the terminology used is that of HP Application Dictionary. For example, the term *field* is used instead of *item*. However, although HP Application Dictionary refers to fields, it also uses the term *Group Item*. Therefore you will also find the term Group Item in this section.

Note

Avoid name changes in the HP Application Dictionary. HP ALLBASE/BRW assumes that a field with a changed name is a new field.

For example, if you change the name of field CUSTOMER-NO to KUNDEN-NUMMER, (to assist your German branch), HP Application Dictionary recognizes that the two fields are identical. However, HP ALLBASE/BRW loads the field into the BRWDIC as the new field name, KUNDEN-NUMMER. This causes problems if you subsequently try to access the field using an HP ALLBASE/BRW standard report that still refers to the field as CUSTOMER-NO!

Single Fields and Group Items

Group items can contain single fields. You can include either just the single fields in HP ALLBASE/BRW, or both the single fields and their associated group items.

If you use group items as keys, you must define relations for either:

- all the single fields associated with the group item, or,
- if you have included group items in the BRWDIC, you need only define a relation for the group item.

Formats

The BRWDIC file will include the following format types:

detail-dataset

manual-master

automatic-master

data exchange

You can also include screen formats. If you do include the screen formats, each screen format is defined in the BRWDIC as a file, and each field of the screen is defined as a field in that file. An additional 16-character field called SCREEN$NAME is the first field in the record structure for the file. The offsets of the other fields of the screen format are updated automatically.

Array Fields (ARRAYDEF)

Since HP Application Dictionary is not capable of defining arrays, they must be defined as single fields. HP ALLBASE/BRW can convert these single fields into arrays for more efficient use in reports.

Array fields are described in a KSAM file called ARRAYDEF, which is read by the BRWAPPD program to identify array fields for use by BRW/XL. An ARRAYDEF file is required, whether or not you have arrays to define.

To build the ARRAYDEF file

- Create a KSAM file in the PUB group of the account holding the data dictionary.

  In the KSAM file, array fields look like this:

DEBIT01	12	DEBIT
DEBIT02	0	DEBIT
DEBIT03	0	DEBIT
...	0	DEBIT
DEBIT12	0	DEBIT
CREDIT01	12	CREDIT
CREDIT02	0	CREDIT
CREDIT03	0	CREDIT
...	0	CREDIT
CREDIT12	0	CREDIT

When BRWDIC is created from the APPDIC, the file ARRAYDEF.PUB.*application_account* is checked to see which array fields are available and what their definitions are. If you define arrays, you must put them in the ARRAYDEF file. If you do not define arrays, you must create an empty ARRAYDEF file (see step 2 in the following procedure).

Building the ARRAYDEF File

To build the ARRAYDEF file (if necessary)

1. If you have arrays to define, create an editor file (ARRAYTMP) with the following format:

   a. Field Name: 16 characters

      These first 16 characters form a KSAM key.

   b. 2 blank spaces

   c. Number of elements in array: unsigned number

   d. 2 blank spaces

      These 2 blank spaces are required to separater the number of elements from the array name.

   e. Array name: 16 characters

   The number of elements in an array is specified in the record containing the first array component and is zero for each subsequent array component.

```
CY-PER1-BUDGET    13  CY-BUD
CY-PER1-CREDIT    14  CY-CRED
CY-PER1-DEBIT     14  CY-DEB
CY-PER1-RATE      14  CY-RATE
CY-PER10-BUDGET    0  CY-BUD
CY-PER10-CREDIT    0  CY-CRED
CY-PER10-DEBIT     0  CY-DEB
CY-PER10-RATE      0  CY-RATE
CY-PER11-BUDGET    0  CY-BUD
CY-PER11-CREDIT    0  CY-CRED
CY-PER11-DEBIT     0  CY-DEB
CY-PER11-RATE      0  CY-RATE
CY-PER12-BUDGET    0  CY-BUD
CY-PER12-CREDIT    0  CY-CRED
NY-PER2-RATE       0  NY-RATE
```

2. Run the KSAM utility.

```
:RUN KSAMUTIL.PUB.SYS
```

3. Create the ARRAYDEF file:

```
> BUILD ARRAYDEF.PUB; REC=-80,2,F,ASCII;&
 KEY=B,1,16,,RDUP;KEYFILE-ARRAYDEK
```

   The & indicates that the command is continued on the next line.

```
> EXIT
```

4. Copy the editor file into the KSAM file:

```
:FCOPY FROM=ARRAYTMP;TO=ARRAYDEF.PUB
```

Note

If you don't have arrays to define, you only need to build the empty files using step 3.

Synonyms (SYNDEF)

Synonyms are fields which appear more than once in a dataset because they are components of more than one composite search field. The HP Application Dictionary requires that all fields in a format, including components of group fields, have unique names. This means that each appearance of a synonym field must have a different name.

HP Application Dictionary does not allow two fields in the same dataset to have the same name. If two fields with the same meaning exist, they are given different names.

HP ALLBASE/BRW allows the definition of fields with different names as synonyms, and treats the synonyms as one field. In a KSAM file called SYNDEF, which is read by the BRWAPPD program, synonyms look like this:

```
COMPANY          COMPANY
COMPANY2         COMPANY
COMPANY4         COMPANY
COST-CENTER1     COST-CENTER
COST-CENTER2     COST-CENTER
```

Note

A SYNDEF file is REQUIRED whether or not synonyms appear in the dictionary. The SYNDEF file must be kept in the PUB group of the account holding the data dictionary.

Building the SYNDEF File

A file called SYNDEF.PUB.*application account* keeps track of these synonyms. This file is checked when BRWDIC is created and only the synonym (or primary field) is used. SYNDEF can be built in the same way as described previously for ARRAYDEF except that the format is:

- Synonym name: 16 characters
- 2 blanks
- Primary name: 16 characters

The following shows a partial listing of SYNDEF:

```
ACCOUNT          ACCOUNT
ACCOUNT1         ACCOUNT
ACCOUNT2         ACCOUNT
ACCOUNT3         ACCOUNT
COMPANY          COMPANY
COMPANY2         COMPANY
COMPANY4         COMPANY
COSTCENTER       COSTCENTER
COSTCENTER1      COSTCENTER
COSTCENTER2      COSTCENTER
COSTCENTER4      COSTCENTER
LEDGER2          LEDGER
LEDGER3          LEDGER
LEDGER4          LEDGER
LEDGER5          LEDGER
```

Preparing a BRWDIC Dictionary File With HP Application Dictionary

BRWAPPD prepares a BRWDIC dictionary file from an HP Application Dictionary data dictionary.

To create the BRWDIC with BRWAPPD

1. Run BRWAPPD:

```
:RUN BRWAPPD.PUB.SYS
```

2. Respond to the following prompts:

```
APPDIC Password:        not displayed
Include Group Items  (N/Y) ?
Include Screen Formats (N/Y) ?
```

   If you press Return for the password prompt, the program will terminate.

If you want to include Group Items in the BRWDIC, answer Y to the prompt `Include Group Items (N/Y)?`. If you choose to include Group Items, they appear with their associated Single Fields on the Project Items screen and can be used in relations.

If you do not include Group Items, you can subsequently only use a Group Item in a relation by specifying a separate relation for each of its Single Fields. The default is to NOT include Group Items.

If you want to include the screen format, answer Y to the prompt `Include Screen Formats (N/ Y) ?` If you do include the screen formats, each screen format is defined in the BRWDIC like a file, with each field of the screen defined as an element in the file. An additional 16-character element called SCREEN$NAME is the first element in the record structure for the file. The default is to NOT include screen formats.

BRWAPPD opens the HP Application Dictionary dictionary with the logical name APPDIC.PUB, and creates the BRWDIC file with the logical name BRWDIC.PUB. If you want to use file equations on either of these files, remember to use the logical names.

If the program encounters an end of file in the dictionary file, you can increase the file size of the dictionary file by entering FILE command, as shown here:

```
:FILE BRWDIC.PUB; DISC=5000
```

Then, run the BRWAPPD program again.

HP Information Access Data Dictionary

HP Information Access Server is the HP 3000 side of HP Information Access, a data-retrieval system that gives users access to PC databases and HP 3000 data on the host HP 3000.

If you have Access Server on your system already, you can use the Information Access data dictionary to create the HP ALLBASE/BRW dictionary file, BRWDIC.

This data dictionary is on the HP 3000 host system and consists of two databases:

- HDPDIC

  This database contains all the information needed to allow access to data.

- HDPENV

  This database contains information about users.

The HP ALLBASE/BRW program BRWACCSD extracts this information from the data dictionary and loads it into the BRWDIC file.

The combination HP ALLBASE/BRW and HP Information Access gives you a powerful reporting system.

HP ALLBASE/BRW is the MIS tool for creating operational data on a production basis.

HP Information Access gives PC-user's a tool for accessing data on the HP 3000 through their PC—on demand.

These tools can share the same dictionary, the HP Information Access dictionary. However, as HP ALLBASE/BRW supports several different dictionaries, you must also construct a BRWDIC file.

There are three possible scenarios when setting up such a reporting system:

a. **You have HP ALLBASE/BRW and are now adding HP Information Access.**

   In this case, you have already defined the BRWDIC file and you need to set up the HP Information Access Data Dictionary. Please refer to the *HP NewWave Office: Information Access Server Database Administration* manual for instructions.

b. **You have HP Information Access and are now adding HP ALLBASE/BRW.**

   You can use the HP Information Access Data Dictionary as a source for constructing the BRWDIC file. However, to take advantage of HP ALLBASE/BRW features for Date- and Time- items, as well as sub-items, you must define some extra information in an MPE file called ACCSDEF before building the BRWDIC file. These steps are described in the next section.

c. **You purchase HP ALLBASE/BRW and HP Information Access together.**

You should first set up the HP Information Access data dictionary as described in the *HP NewWave Office: Information Access Server Database Administration* manual, and then construct the BRWDIC file as descibed in the next section.

Terminology

You should note that not all terms used in the HP Information Access data dictionary mean the same thing in HP ALLBASE/BRW. Here is a list of the terms used and their different meanings:

HP Information Access Data Dictionary	HP ALLBASE/BRW
table	format
data source	location

Note also that the following list of HP Access Server features is not supported by HP ALLBASE/BRW:

- Environment information - access groups, users, security information and so on.
- Information contained in View Tables, except that for:
  - Using IMAGE compound-items (you do not have to break down the compound item into single items).
  - Defining parent/child relations in the ACCSDEF file with the SUB function.
- Controlling the form in which IMAGE-zoned and -packed item types are displayed by HP ALLBASE/BRW.

The following table lists the HP Information Access Data Dictionary element-types and describes how they are handled by HP ALLBASE/BRW.

HP Information Access Type	Length (1 word = 16 bits)	BRW Type	Remarks
L (logical)	1 word	I (integer)	logical
I (integer)	1 word	I (integer)	
I (integer)	2 words	I (integer)	
I (integer)	4 words	N (numeric)	only 13 digits supported
R (real)	2 words	R (real)	
R (real)	4 words	R (real)	
C (character)		S (string)	maximum length 255 bytes
P (packed)		N (numeric)	packed
Z (zoned)		N (numeric)	zoned
L, I, C		D (date)	date specified in ACCSDEF
L, I, C		T (time)	time specified in ACCSDEF

Supported Features

The following HP ALLBASE/BRW item types are not supported by HP Information Access Server:

- The element types DATE and TIME.
- The definition of the sub-items of some types, used to describe parent/child relations.

Defining Date and Time Using an ACCSDEF File

In order to describe the items Date and Time to HP ALLBASE/BRW, you must define an additional MPE file which lists these items. This file must be called ACCSDEF and can be defined with a conventional text editor. If ACCSDEF is in the group in which you are creating the BRWDIC, the BRWACCSD program automatically loads it.

Define the Date and Time items in the ACCSDEF file as shown in the following example:

```
Example:
--------
: Comment   :                                          ;
: The entries of the example are taken from TOYDB  ;

SUBBEGIN

   TABLE order-details
   sh-date-year  = SUB (shipment-date,1,C2)
   sh-date-month = SUB (shipment-date,3,C2)
   sh-date-day   = SUB (shipment-date,5,C2)

   TABLE orders

   order-date-year  = SUB (order-date,1,2)
   order-date-month = SUB (order-date,3,5)
   order-date-day   = SUB (order-date,5,2)

SUBEND

EDITBEGIN

   TABLE orders

     ITEM order-date
       HEADING "string converted to date-ymd"
       DATE DATE-YMD
     ITEM order-date-year
       HEADING "year component"
     ITEM order-date-month
       HEADING "month component"
     ITEM order-date-day
       HEADING "day component"

   TABLE order-details

     ITEM shipment-date
       HEADING "string converted to date-ymd"
       DATE DATE-YMD
     ITEM sh-date-year
       HEADING "year component"
     ITEM sh-date-month
       HEADING "month component"
     ITEM sh-date-day
       HEADING "day component"

   EDITEND
```

Summary of example

This example illustrates how the characteristics of the following items are changed:

SHIPMENT-DATE

in the dataset ORDER-DETAILS, and

ORDER-DATE

in the dataset ORDERS

Here is an outline of the steps used to change the items:

a. Create three sub-items (children), each with a two-digit value for year, month and day.

b. Change the type of the parent item (currently C for character) to date.

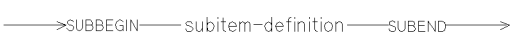
c. A short one-line explanation of what has been changed in the item appears below the name of that item.

Syntax for ACCSDEF file

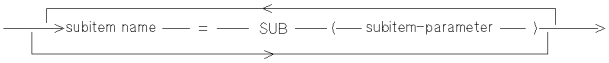
The figure on the next page illustrates the syntax for the ACCSDEF file.

Overall structure

subitem-part:

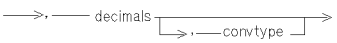


subitem-definition:

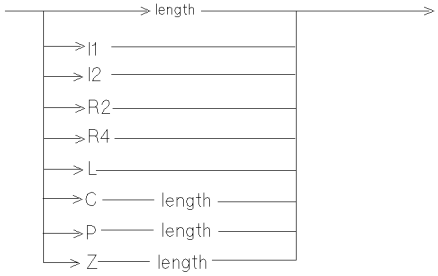


subitem parameter:

optionals:

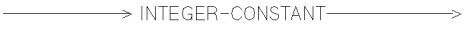


dtype:



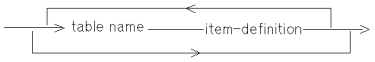
subitem name, parentitem name:

position, decimals, length:

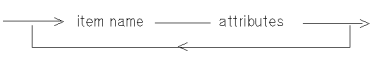


editpart:

edit-definition:



item-definition:



attributes:

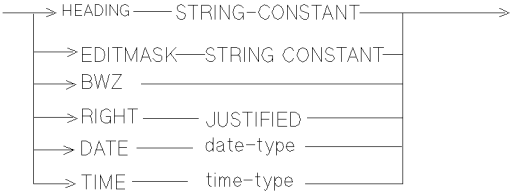


table-name, item-name

ACCSDEF syntax

The figure on the previous page shows the syntax for ACCSDEF. The rules below and also the following sections describe this syntax in greater detail.

Comments

You must enclose any comments in semicolons (;)

```
;COMMENT;
```

and you may not type a comment within a comment.

Sub-item Part

The Sub-item part implements the SUB function used in the View Tables in the HP Information Access data dictionary.

Type in the following:

```
sub-item-name = SUB (parent-item-name; position,
which must be positive; dtype; decimals; convtype)
```

An **Identifier** must:

- Be 1 to 16 characters in length.
- Begin with an alphabetic character (A through Z) followed by:
  - any characters from A to Z (lower or upper case)
  - digits from 0 to 9
  - any one of these special characters: + - * / ? ' # % & @

Example:

Assume you have an item defined in the HDPDIC as:

```
SHIPMENT-DATE X6
```

The value of the item represents a three-part date composed of *year*, *month*, *day*, with two digits for each of the three parts; each part can have a value of more than two digits if required.

Now you define the sub-item by splitting the date into its individual components:

Note that one of the ITEMs must begin with 1, the smallest positon.

A valid shipment date could be:

```
"880309"

SHIP-DATE-YEAR   = SUB (SHIPMENT-DATE,1,C2)
SHIP-DATE-MONTH  = SUB (SHIPMENT-DATE,3,C2)
SHIP-DATE-DAY    = SUB (SHIPMENT-DATE,5,C2)
```

This is then converted to:

```
SHIP-DATE-YEAR  = "88"
SHIP-DATE-MONTH = "03"
SHIP-DATE-DAY   = "09"
```

You can now access these subitems with HP ALLBASE/BRW.

Edit Part

In the Edit part of the definition process you can add the following attributes to an item:

- heading text and edit mask

  These are both strings of a maximum length of 30 characters.

- Date type and Time type

  You can change any item's type to a Date- or Time type with HP ALLBASE/BRW. The original type of the item must, however, be equal to the length of the new type you wish to assign to the item. (This is not supported by HP Access Server.)

ACCSDEF Date and Time-Item Tables

You can use the following tables for reference purposes. They list the various Date- and Time-types you can specify in the supporting file ACCSDEF, together with the required type and length of the corresponding element-entries in the HP Access Server Data Dictionary.

DATE-ITEM Table (DISPLAY and INTEGER Format)

Dictionary/3000 Long Name	Length Required (1 wd = 16 bits)	Base Type	Example: (Date is 04/28/92)
DATE-DMY-D	8 Bytes	C	28/04/92
DATE-DMYY-D	10 Bytes	C	28/04/1992
DATE-MDY-D	8 Bytes	C	04/28/92
DATE-MDYY-D	10 Bytes	C	04/28/1992
DATE-YMD-D	8 Bytes	C	92/04/28
DATE-YYMD-D	10 Bytes	C	1992/04/28
DATE-DDY-D	6 Bytes	C	118/92
DATE-DDYY-D	8 Bytes	C	118/1992
DATE-YDD-D	6 Bytes	C	92/118
DATE-YYDD-D	8 Bytes	C	1992/118
DATE-DM-D	5 Bytes	C	28/04
DATE-MD-D	5 Bytes	C	04/28
DATE-MY-D	5 Bytes	C	04/92
DATE-MYY-D	7 Bytes	C	04/1992
DATE-YM-D	5 Bytes	C	92/04
DATE-YYM-D	7 Bytes	C	1992/04
DATE-DMY-I	3 Words	I, L	
DATE-DMYY-I	3 Words	I, L	
DATE-MDY-I	3 Words	I, L	
DATE-MDYY-I	3 Words	I, L	
DATE-YMD-I	3 Words	I, L	
DATE-YYMD-I	3 Words	I, L	
DATE-DDY-I	2 Words	I, L	
DATE-DDYY-I	2 Words	I, L	
DATE-YDD-I	2 Words	I, L	
DATE-YYDD-I	2 Words	I, L	

DATE-ITEM Table (DISPLAY and INTEGER Format) (continued)

Dictionary/3000 Long Name	Length Required (1 wd = 16 bits)	Base Type	Example: (Date is 04/28/92)
DATE-DM-I	2 Words	I, L	
DATE-MD-I	2 Words	I, L	
DATE-DD-I	1 Word	I, L	
DATE-MY-I	2 Words	I, L	
DATE-MYY-I	2 Words	I, L	
DATE-YM-I	2 Words	I, L	
DATE-YYM-I	2 Words	I, L	
DATE-CALENDAR	2 Bytes	I, L	
DATE-JULIAN	4 Bytes	I, L	

DATE-ITEM Table (NUMERIC and ALPHANUMERIC Format)

Date/Time Keyword	Length Required (1 word = 16 bits)	HDPDIC Type Required	Example: (Date is 04/28/92)
DATE-DMY	6 Bytes	C	280492
DATE-DMYY	8 Bytes	C	28041992
DATE-MDY	6 Bytes	C	042892
DATE-MDYY	8 Bytes	C	04281992
DATE-YMD	6 Bytes	C	920428
DATE-YYMD	8 Bytes	C	19920428
DATE-DDY	5 Bytes	C	11892
DATE-DDYY	7 Bytes	C	1181992
DATE-YDD	5 Bytes	C	92118
DATE-YYDD	7 Bytes	C	1992118
DATE-DM	4 Bytes	C	2804
DATE-MD	4 Bytes	C	0428
DATE-DD	3 Bytes	C	118
DATE-MY	4 Bytes	C	0492
DATE-MYY	6 Bytes	C	041992
DATE-YM	4 Bytes	C	9204
DATE-YYM	6 Bytes	C	199204

TIME-ITEM Table (All Formats)

Date/Time Keyword	Length Required	HDPDIC Type Required	Example: (Time is 15:43:24)
TIME-HM24-I	4 Bytes	C	1543
TIME-HMS24-I	6 Bytes	C	154324
TIME-HM12-I	5 Bytes	C	0343P
TIME-HM24-D	5 Bytes	C	15:43
TIME-HMS24-D	8 Bytes	C	15:43:24
TIME-HM12-D	6 Bytes	C	03:43P
TIME-HM24-I	2 Words	I, L	
TIME-HMS24-I	2 Words	I, L	
TIME-CLOCK	4 Bytes	I, L	

Preparing a BRWDIC with HP Information Access Data Dictionary

After you have specified which information from the HP Information Access data dictionary you require for use with HP ALLBASE/BRW, you must create the BRWDIC using the BRWACCSD program.

The BRWACCSD program extracts the information from the data dictionary's database, HDPDIC, and creates a dictionary file, BRWDIC.

Before you can run the BRWACCSD program, you must use a file equation to ensure that BRWACCSD can open the database root-file HDPDIC and the ACCSDEF file even when they are not in the same group where you create the dictionary file, BRWDIC. Typically, HDPDIC resides in the group PPCDICT of the account HPOFFICE, and ACCSDEF resides in the group where you normally work. In our example, ACCSDEF resides in *yourgroup* on *youraccount*.

To create the BRWDIC with BRWACCSD

1. Type the following file equations:

   ```
   FILE HDPDIC=HDPDIC.ppcdict.hpoffice
   FILE ACCSDEF=ACCSDEF.yourgroup.youraccount
   ```

   Use the groups and accounts appropriate for your data.

2. Run BRWACCSD:

   ```
   :RUN BRWACCSD.PUB.SYS
   ```

   This prompt appears:

   ```
   Dictionary Password    :
   ```

3. Type the dictionary password and press Return. If you press Return without typing in anything, the BRWACCSD program terminates. The password does not appear on the screen when you type it.

4. Respond to the next prompt:

```
Dictionary Password    :  not displayed
Use Parent Items (N/Y): _
```

   a. Type Y (yes) if you wish to include (use) the parent items in the dictionary file, BRWDIC.

   b. Type N (no) or press Return if you do not wish to include them.

   If ACCSDEF is in the group where you are working, the following appears:

```
*** Supporting File opened is ACCSDEF

Print to File (N/Y)  :
```

   a. Type in Y (yes) if you want the list of error messages from the analysis of ACCSDEF to be written to the ACCSPRNT file.

   b. Type N (no) or press Return if you want the list of error messages to be written to the $STDLIST file.

The BRWACCSD program now begins analyzing the ACCSDEF supporting file.

```
* * * Start analysing ACCSDEF

* * * ERRORS 0       Warnings 0
* * * Start analysing HPDIC
END OF PROGRAM
```

After analyzing ACCSDEF, the total number of errors and warnings is displayed on the screen.

If the error total is 0, the BRWACCSD program begins analyzing the HDPDIC. (The total number of warnings is irrelevant.)

Once the analysis is complete and there were no errors, BRWACCSD creates the HP ALLBASE/BRW dictionary file, BRWDIC. (If there are warnings, the BRWDIC file will still be created.)

User-Generated Dictionary Source File

You can generate a dictionary source-file that is a simple MPE file containing a list of "statements" you specify according to your needs. These are later converted into the HP ALLBASE/BRW BRWDIC by the program BRWGEND.

Typically, you include the statements in a text file — an unnumbered MPE file that you create with a text editor. The various kinds of statements, along with their syntax and semantics, are described in later sections of this chapter.

Terminology

Terminology in the user-generated dictionary source file and HP ALLBASE/BRW is not always the same. The table below shows the differences in terminology:

HP ALLBASE/BRW	HP User-Generated Dictionary
Item	Element Item is entity of a record
Format	Record/Item relation

Creating the Dictionary Source-File

The dictionary source-file can contain the following types of statements. You must always specify the statements in this order:

TITLE/SCHEMA statements

FILE statements

ELEMENT statements

RECORD statements

ITEM statements (repeated for each record)

BUILD statements

The next example, shown below and continued on the next page, illustrates a user-generated dictionary:source file, using the sample database TOYDB.

```
TITLE "description of database TOYDB"
FILE products                   TYPE IMAGE MASTERS              OF toydb
FILE order-master               TYPE IMAGE AUTOMATIC MASTER     OF toydb
FILE customers                  TYPE IMAGE MASTER               OF toydb
FILE order-details              TYPE IMAGE DETAIL               OF toydb
FILE orders                     TYPE IMAGE DETAIL               OF toydb
FILE invoices                   TYPE IMAGE DETAIL               OF toydb

ELEMENT product-no              X(6)
ELEMENT product-name            X(16)
ELEMENT price                   9(7)V9(2)
ELEMENT product-line            X(2)
ELEMENT quantity                9(5)
ELEMENT order-no                X(8)
ELEMENT customer-no             X(6)
ELEMENT customer-no-b           X(3)
ELEMENT customer-no-e           X(3)
ELEMENT customer-name           X(30)
ELEMENT city                    X(30)
ELEMENT state                   X(30)
ELEMENT zipcode                 X(6)
ELEMENT address                 X(30)
ELEMENT sales-area              X(16)
ELEMENT turnover                X(15)V9(2)
ELEMENT turnover-py             X(15)V9(2)
ELEMENT turnover-mtd            X(15)V9(2)
ELEMENT shipment-date           DATE
ELEMENT ship-date-month         X(2)
ELEMENT ship-date-day           X(2)
ELEMENT order-date              DATE
ELEMENT order-date-month        X(2)
ELEMENT order-date-day          X(2)
ELEMENT invoice-no              X(6)
ELEMENT invoice-date            DATE
ELEMENT due-date                DATE
ELEMENT amount                  9(11)V9(2)
ELEMENT paid-amount             9(11)V9(2)

RECORD products
   ITEM product-no              KEY
   ITEM product-name
   ITEM price                   PACKED SIZE 4
   ITEM product-line
   ITEM quantity                INTEGER

RECORD order-master
   ITEM order-no                KEY

RECORD customers
   ITEM customer-no             KEY
   ITEM customer-name
   ITEM city
   ITEM state
   ITEM zipcode
   ITEM address
   ITEM sales-area
   ITEM turnover                PACKED SIZE 8 OCCURS 12
   ITEM turnover-py             PACKED SIZE 8
   ITEM turnover-mtd            PACKED SIZE 8
```

(Continued ...)

```
RECORD order-details
   ITEM order-no                KEY
   ITEM product-no              KEY
   ITEM quantity                INTEGER
   ITEM shipment-date           DATE-MD
      REDEFINED BY
        ITEM ship-date-month
        ITEM ship-date-day
      END

RECORD orders
   ITEM customer-no             KEY
   ITEM order-no                KEY
   ITEM order-date              DATE-MD
      REDEFINED BY
        ITEM order-date-month
        ITEM order-date-day
      END

RECORD invoices
   ITEM invoice-no
   ITEM customer-no             KEY
   ITEM order-no                KEY
   ITEM amount                  PACKED SIZE 6
   ITEM paid-amount             PACKED SIZE 6
   ITEM invoice-date            DATE-MD
   ITEM due-date                DATE-MD

BUILD
```

Statements

The statements in the user-generated source file consist of three parts, in the following order:

Statement type	Description
Keyword	A *keyword* is a word defined by the system. These words can be used as file names only if a percent sign (%) in inserted before the word (i.e. %file)
Name	A *name* must: ■ Be 1 to 20 characters long. ■ Begin with a $, @, #, or letter, which must be followed by: □ any alphabetical characters from A through Z □ any digit from 1 through 9 □ any of the following special characters: _ # %
Options	Not all statements must include options - you specify them as required.

Syntax Rules

- If you are typing in a statement and there is not enough space to write everything in that line, type in an ampersand (&) at the end of the line and type in the rest of the statement in the line below; using the ampersand, you can continue to type in a statement on several lines.
- Enclose comments in semi-colons (;). Note that if the end of the comment is the end of the text on that line, the semicolon (;) is not necessary.

```
;comment comment comment;
```

- You cannot write a second comment within a comment.

The TITLE or SCHEMA Statement

The TITLE or SCHEMA statement consists of the key words TITLE or SCHEMA, followed by a string of up to forty characters enclosed in double quotes (”). With the TITLE or SCHEMA statement you do not instruct the BRWGEND program to create anything. This statements simply marks the beginning of the specification process.

[SCHEMA / TITLE] "*characterstring*"

The FILE Statement

With the FILE statement you specify a format and its type. A format can be either one of the following:

- An MPE or a KSAM file.
- A dataset:
  - IMAGE MASTER
  - IMAGE AUTOMATIC MASTER
  - IMAGE DETAIL

The FILE statement consists of the keyword FILE, followed by the name, the type, and the OPEN option (this is optional).

```
FILE mpefile            TYPE MPE
 FILE ksamfile TYPE KSAM
FILE customers TYPE IMAGE MASTER OF toydb
FILE customers-expldb TYPE IMAGE MASTER OF expldb
   & OPEN customers
```

The file name must be unique. If you want to specify two formats (for example from separate databases) which have the same name, you can use the OPEN option to make the name unique. The OPEN name then represents the physical name of the format.

The ELEMENT Statement

With the ELEMENT statement you specify an item, its type, display size, and number of decimal positions.

The ELEMENT statement consists of the keyword ELEMENT, followed by the name, the element type and element options.

```
ELEMENT customer-name  X(30)
ELEMENT quantity 9(5)
ELEMENT turnover 9(15)V9(2)
ELEMENT order-date DATE
ELEMENT actual-time TIME              &
  HEADING "the actual time is displayed" &
  PICTURE "hh:mmpp"
```

The name of an element statement has to be unique. You cannot specify diverse elements with the same name even if they have different characteristics. You can, however, use the same element in multiple record tables.

The element is usually of the type:

- X for character
- 9 for numeric
- DATE or TIME

If the element is of the character- or numeric type, you must specify the size.

In the example above, the element's type is character (X) which is a customer's name thirty characters long, and the quantity is numeric, a number five digits long.

Additionally, if the element's type is numeric, you can also specify the number of decimal positions. In the example above, turnover is a numeric element seventeen digits long, including two digits to the right of the decimal point.

```
ELEMENT turnover 9(15)V9(2)
```

RECORD Statement

With the RECORD statement you describe the relationship between the record and item. With the ITEM statement you assign the item a specific characteristic which is only valid in that particular RECORD statement.

For every file which you specify in a FILE statement, there must be one corresponding RECORD statement.

The RECORD statement consists of the keyword RECORD, followed by the name of the record (which must be a name defined in a FILE statement), and one or more ITEM statements.

```
RECORD products
 ITEM product-no CHARACTER   KEY
 ITEM quantity INTEGER
 ITEM product-name CHARACTER
```

ITEM Statement

The ITEM statement consists of the keyword ITEM, followed by the name of the item, which must be a name defined in an ELEMENT statement. (If you wish, you can define the item type and size and item options. However, the BRWGEND program derives its own default values for type and size.)

Element Type	Size	Default Item Type	Default Item Size (Bytes)
X(n)	n	CHARACTER	n
9(n)	n <=4	INTEGER	2
9(n)	n <=9	DOUBLE	4
9(n)	n > 9	LONG	8
DATE	8	HPDATE	2
TIME	8	HPTIME	2

The following table lists the item types in the user- generated dictionary source-file and shows how they are handled by HP ALLBASE/BRW.

Element Type	Item Type	Length (words) (1 wd - 16 bits)	BRW Type	Remarks
X	CHARACTER		S	Max 255 Bytes
9	LOGICAL	1	I	logical
9	INTEGER	1	I	
9	DOUBLE	2	I	
9	BINARY4	4	I	
9	REAL	2	R	
9	LONG	4	R	
9	PACKED		N	packed
9	ZONED		N	zoned
9	DECIMAL		N	num, display
9	FPDECIMAL2	2	N	short floating point decimal
9	FPDECIMAL4	4	N	HP Business Basic floating point decimal
DATE	date-types		D	listed below
TIME	time-types		T	listed below

For integer, decimal, packed, or zoned data, you can specify a size with the SIZE keyword. Unlike IMAGE or COBOL conventions, the HP ALLBASE/BRW size is always in bytes. For packed and zoned, the default SIZE in bytes is $(n + m)/\mathrm{Z} + 1$, where $n + m$ comes from the ELEMENT defraction:

```
ELEMENT TURNOVER 9(m) V9(m)
```

The next table lists the various date- and time types you can specify with an ITEM statement in the MPE file in which you store all these statements (in our example, the GENDEF file), as well as the required length of the item and the type of the corresponding element definition.

DATE TYPES Table

Date/Time Keyword	Required Length (1wd = 16 bits)	Required Element Type	Example: (Date is 04/28/92)
DATE-DMY-D	8 Bytes	DATE	28/04/92
DATE-DMYY-D	10 Bytes	DATE	28/04/1992
DATE-MDY-D	8 Bytes	DATE	04/28/92
DATE-MDYY-D	10 Bytes	DATE	04/28/1992
DATE-YMD-D	8 Bytes	DATE	92/04/28
DATE-YYMD-D	10 Bytes	DATE	1992/04/28
DATE-DDY-D	6 Bytes	DATE	118/92
DATE-DDYY-D	8 Bytes	DATE	118/1992
DATE-YDD-D	6 Bytes	DATE	92/118
DATE-YYDD-D	8 Bytes	DATE	1992/118
DATE-DM-D	5 Bytes	DATE	28/04
DATE-MD-D	5 Bytes	DATE	04/28
DATE-MY-D	5 Bytes	DATE	04/92
DATE-MYY-D	7 Bytes	DATE	04/1992
DATE-YM-D	5 Bytes	DATE	92/04
DATE-YYM-D	7 Bytes	DATE	1992/04
DATE-DMY-I	3 Words	DATE	
DATE-DMYY-I	3 Words	DATE	
DATE-MDY-I	3 Words	DATE	
DATE-MDYY-I	3 Words	DATE	
DATE-YMD-I	3 Words	DATE	
DATE-YYMD-I	3 Words	DATE	
DATE-DDY-I	2 Words	DATE	
DATE-DDYY-I	2 Words	DATE	
DATE-YDD-I	2 Words	DATE	
DATE-YYDD-I	2 Words	DATE	
DATE-DM-I	2 Words	DATE	
DATE-MD-I	2 Words	DATE	
DATE-DD-I	1 Word	DATE	
DATE-MY-I	2 Words	DATE	
DATE-MYY-I	2 Words	DATE	
DATE-YM-I	2 Words	DATE	
DATE-YYM-I	2 Words	DATE	
HPDATE	2 Bytes	DATE	

ACCSDEF DATE-ITEM Table (NUMERIC and ALPHANUMERIC Format)

Date Keyword	Required Length (1wd = 16 bits)	Required Element Type	Example: (Date is 04/28/92)
DATE-DMY	6 Bytes	DATE	280492
DATE-DMYY	8 Bytes	DATE	28041992
DATE-MDY	6 Bytes	DATE	042892
DATE-MDYY	8 Bytes	DATE	04281992
DATE-YMD	6 Bytes	DATE	920428
DATE-YYMD	8 Bytes	DATE	19920428
DATE-DDY	5 Bytes	DATE	11892
DATE-DDYY	7 Bytes	DATE	1181992
DATE-YDD	5 Bytes	DATE	92118
DATE-YYDD	7 Bytes	DATE	1992118
DATE-DM	4 Bytes	DATE	2804
DATE-MD	4 Bytes	DATE	0428
DATE-DD	3 Bytes	DATE	118
DATE-MY	4 Bytes	DATE	0492
DATE-MYY	6 Bytes	DATE	041992
DATE-YM	4 Bytes	DATE	9204
DATE-YYM	6 Bytes	DATE	199204

TIME-ITEM Table (All Formats)

Time Keyword	Required Length (1wd = 16 bits)	Required Element Type	Example: (Time is 3.43 pm/24secs)
TIME-HM24	4 Bytes	TIME	1543
TIME-HMS24	6 Bytes	TIME	154324
TIME-HM12	5 Bytes	TIME	0343P
TIME-HM24-D	5 Bytes	TIME	15:43
TIME-HMS24-D	8 Bytes	TIME	15:43:24
TIME-HP12-D	6 Bytes	TIME	03:43P
TIME-HM24-I	5 Bytes	TIME	
TIME-HMS24-I	8 Bytes	TIME	
HPTIME	4 Bytes	TIME	

You can specify the following three item options when you are defining an item itself:

- UNIQUE KEY
- KEY
- OCCURS *n*

If the record is either an IMAGE dataset or a KSAM file, you can label an item as a KEY.

If the item is an array, you can specify the number of occurrences in this array.

```
ITEM turnover PACKED SIZE 8 OCCURS 12
```

(turnover is an array with 12 occurrences representing the turnover of every month of a year.)

The REDEFINED Statement

Any item in a record/item relationship may contain sub-items, meaning that you can describe a parent/ child relationship.

With the REDEFINED statement you can specify the item as a sequence of other items.

The REDEFINED statement consists of the keywords REDEFINED and BY, followed by one or more ITEM statements and the final keyword, END.

```
ITEM order-date  DATE-MD
 REDEFINED BY
  ITEM order-date-month
  ITEM order-date-day
 END
```

Order-date is a Date-type item, 4 Bytes in size, stored in the form month/day. The REDEFINED statement splits it into its components *month* and *day*, and the items ORDER-DATE-MONTH and ORDER-DATE-DAY are defined in ELEMENT-statement strings of two characters or digits in length.

If the redefined item (parent item) is a key, its size must be equal to the sum of the sizes of the sub-items (child items). If the redefined item (parent item) is not a key, its size must be equal to or longer than that of the sub-items together (child items).

Note

You cannot redefine items which are arrays with an occurrence of greater than one.

The BUILD Statement

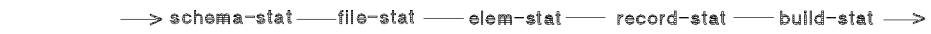
With the BUILD statement you give BRWGEND instructions to build the HP ALLBASE/BRW dictionary file, BRWDIC, from the dictionary source-file which you have generated, (in our examples, the GENDEF file).

The BUILD statement is the last one in the process of generating the dictionary source file and consists only of the keyword BUILD.

The BRWDIC can only be built if no error occurs during the analysis of the file containing the statements you created.

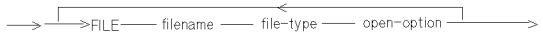
The following page shows an illustration of the syntax of the dictionary source file that you generate.

User-Generated Dictionary Source-File

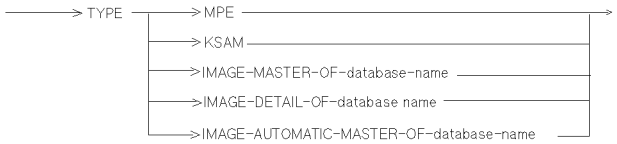


schema-stat:

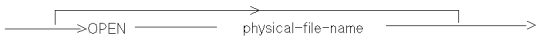
file-stat:



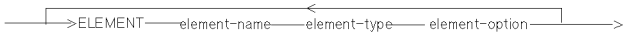
file-type:



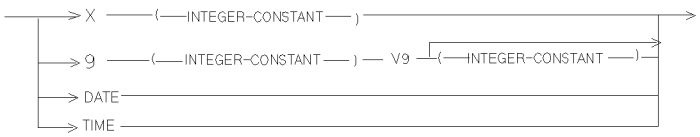
open-option



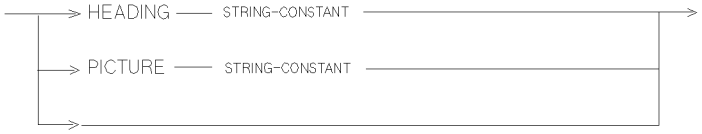
elem-stat:



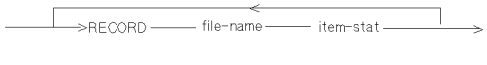
element-type:



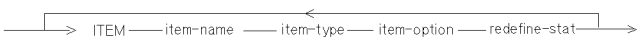
element-option:



record-stat:

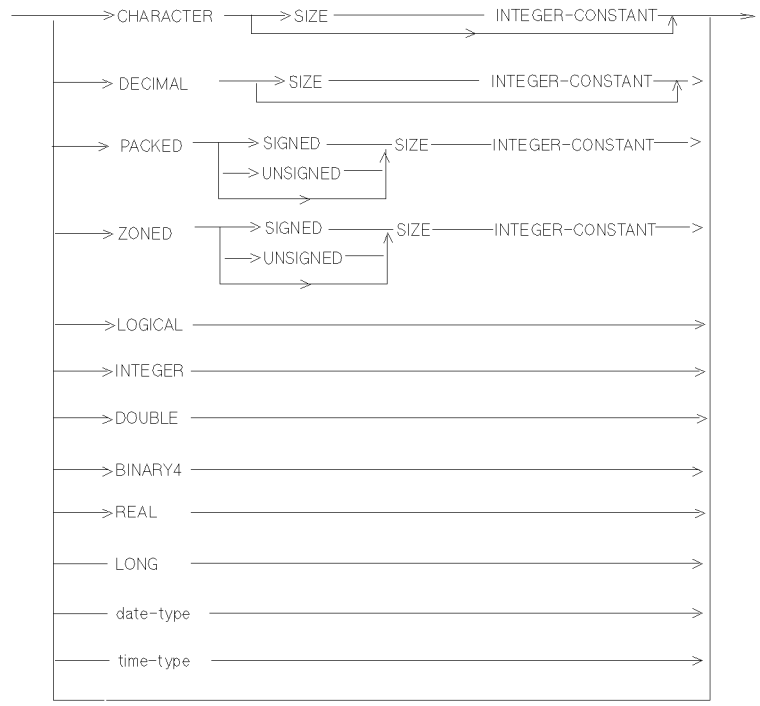


item-stat:

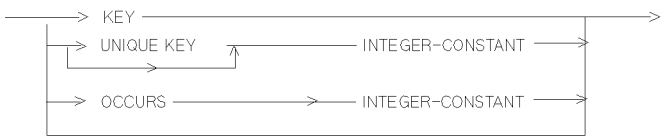


DICTSRC1

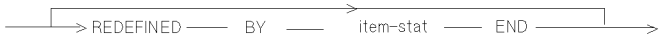
item-type:



item-option:



redefine-stat



build-stat

date-type, field-type

see keyword in date- and time-table (2.d)

file-name, element-name, item-name, physical-field-name

An identifier can be 1 to 20 chars long, must begin with a letter and can be followed by one of the following: A - Z, a - z, 1 - 9, -, _, #, %, ', ".

DICTSRC2

Creating A BRWDIC Dictionary File With the User-Generated Dictionary

Once you have generated the source file, you can create the HP ALLBASE/BRW dictionary file, BRWDIC.

To create a BRWDIC with a User-Generated Dictionary

1. Run the BRWGEND program:

```
: RUN BRWGEND.PUB.SYS
```

2. Respond to the following prompts:

```
Name of Dictionary Specification File:
Name of Print File                    :
Use Redefined (Parent) Items (N/Y)    :
Use Names up to 20 Characters (Y/N)   ?
```

   For the dictionary specification file, type the name of the file that contains the statements you defined: for example GENDEF.

   For the print file, type the name of the file to which you want BRWGEND to write any error messages or warnings.

   If you do not want BRWGEND to write the list to a file, press Return, and warnings and messages are printed to the screen.

   To use redefined (parent) items in the BRWDIC, type Y (yes) at the propmt. Type N (no), or press Return if you do not wish to include them.

   To use names with up to 20 characters, type Y or press Return for this prompt. Type N (no) if you want to use only names with a maximum length of 16 characters. If you use this option, names longer than 16 characters are truncated and checked to make sure that the truncated names are still unique. If you want to run reports with BRW/V, type N for no. BRW/V cannot use names longer than 16 characters.

After analyzing the input, BRWGEND counts the number of errors and warnings found and prints that figure on the screen.

```
* * * ERRORS 0          Warnings 0

END OF PROGRAM
```

If the result of the error count is zero (0), BRWGEND creates the HP ALLBASE/BRW dictionary file BRWDIC. (The number of warnings is irrelevant.)

Changes in Dictionary Files

This section describes the consequences when a data dictionary is modified.

Modifying BRWDIC

You cannot directly modify a BRWDIC dictionary file. If a data dictionary has been modified, you must run the appropriate program (BRWD3000, BRWAPPD, or BRWSD) to create a new BRWDIC file from the modified data dictionary.

The BRWDIC file is time-stamped upon creation. There fore, when a new BRWDIC is created, a new time stamp is added.

Automatically Recompiled Reports

You can enable HP ALLBASE/BRW to automatically recompile your reports by changing the Execution Values default using the BRWSETUP program. (See Chapter 17, "Configuration and Security" for more information about the BRWSETUP program.)

Each time a report is run, HP ALLBASE/BRW compares the time-stamp of the report compilation and the time-stamp of the BRWDIC file. If the BRWDIC file was created after the report was last compiled, the BRWEXEC program, (that part of HP ALLBASE/BRW that executes reports), automatically recompiles the report before execution, provided you changed the defaults in the configuration. This guarantees that the BRWEXEC program always uses the actual formats of datasets and files.

Caution

If the layout of the source tables has been changed (for example, an item used in the report has been deleted), you need to recompile the BRWDIC from the updated data dictionary to ensure correct execution of the report.

You can insulate the report from some changes to the source tables by requesting item security on the Tune Access screen. Item security will cause HP ALLBASE/BRW to only ask for items from the source table that are specifically used by the report. In the case of the source tables having changes that don't directly affect those items, you may be able to run the report without any errors. There could be a slight performance cost for using item security. See Chapter 19 for more about performance considerations.

Automatic Upgrade of the Specification File

HP ALLBASE/BRW checks the creation time-stamp of the BRWDIC dictionary file used for the report. If the dictionary file was prepared after the last modification of the report specification file, HP ALLBASE/BRW automatically upgrades the file while loading the report.

The changes made during this upgrade are listed on the Select Report screen. You can print out the listing by pressing the terminal (Print) key. It can be helpful to have this information and a specification file listing to identify what changes (if any) must be made manually.

Manual Upgrades to Specification Files

Some problems can occur that cannot be solved by either the automatic recompilation or the automatic specification file upgrades. For example, if you change an item's type from numeric to string, you must manually change all calculation formulae in which the item occurs. Similarly, if you change the length of an item and you want to print the item with the new length, you must modify the report specification file manually.

19

Performance Considerations

Improve execution performance for your reports

At a Glance

For every report produced with HP ALLBASE/BRW, report execution performance is influenced by the following basic factors:

- the amount of data to be printed in the report
- the amount of data in the datasets, files or HP SQL tables accessed by the report
- how data from datasets, files or HP ALLBASE/SQL tables is combined (joined or merged)
- calculations and formatting performed during report execution

In addition to these basic factors, you can affect execution performance in other ways *without* impairing the original reporting requirements.

To improve performance, you can do the following:

- Reduce the amount of data to the subset needed for printing in the report, as early as possible.
- Reduce the number of physical block transfers from and to the disks.
- Reduce the number of head movements on the disks; that is, provide an efficient access profile on the disks.

This chapter covers

- influencing performance
- data access methods
- using the compile listing
- execution statistics
- effective calculations
- table calculated items vs layout calculated items

Influencing Performance

What Your System Can Do For Performance

Proper system configuration can have an important impact on all database and I/O on your system. Consult your system reference manuals for information about system configuration.

Datasets and files spread over several disks can improve performance of a report when it accesses multiple datasets or files at the same time, which is when you use keyed access.

What HP ALLBASE/BRW Can Do For Performance

With HP ALLBASE/BRW you can tune a report's data access for best execution performance without changing the relational data access definition. This means that the report always prints the same data, regardless of the sequence in which the datasets or files are accessed, which keys are used and whether serial access is used. See Chapter 3, "Understanding Relational Data and Tables," and Chapter 6, "Defining Data Access," for more information about the relational data access definition.

In addition to tuning single reports, you can improve overall performance of multiple reports by using the multi-pass reporting features described in Chapter 12, "Multi-Pass Reporting and Output Files."

To find the optimal data access method for a single report, use the following description of how HP ALLBASE/BRW performs data access and the hints about data access methods and when to use them.

Access Blocks

A report is executed in one or more phases, called access blocks. Each access block inputs data, processes and optionally sorts it, and then outputs the data. The last access block prints the report and creates the output files (if any). The previous access blocks create temporary work files used by other access blocks.

Compiler Optimizations

The HP ALLBASE/BRW compiler translates the relational access definition into the access blocks needed for the selected access methods. In addition, it optimizes the data access, so that:

- data selections are performed as early as possible
- table calculated items are calculated as early as possible
- the number of items is reduced to only those used in the report

The compiler generates a listing which shows how the relational access definition was translated and the optimizations which have been performed. This listing is discussed in more detail later in this chapter.

Data Access Methods

This section describes the data access methods that you can use to improve performance. The emphasis here is on access methods for reports using joined source tables.

Data Access Methods For Joins

Joining datasets, files or HP ALLBASE/SQL tables means that their records are combined such that records are joined where the common items in the relation definition have identical values. This is described in Chapter 6, Defining Data Access.

HP ALLBASE/BRW provides two different data access methods for joins, which can be used singly or in combination. These are discussed below.

Data Access Methods For Merges

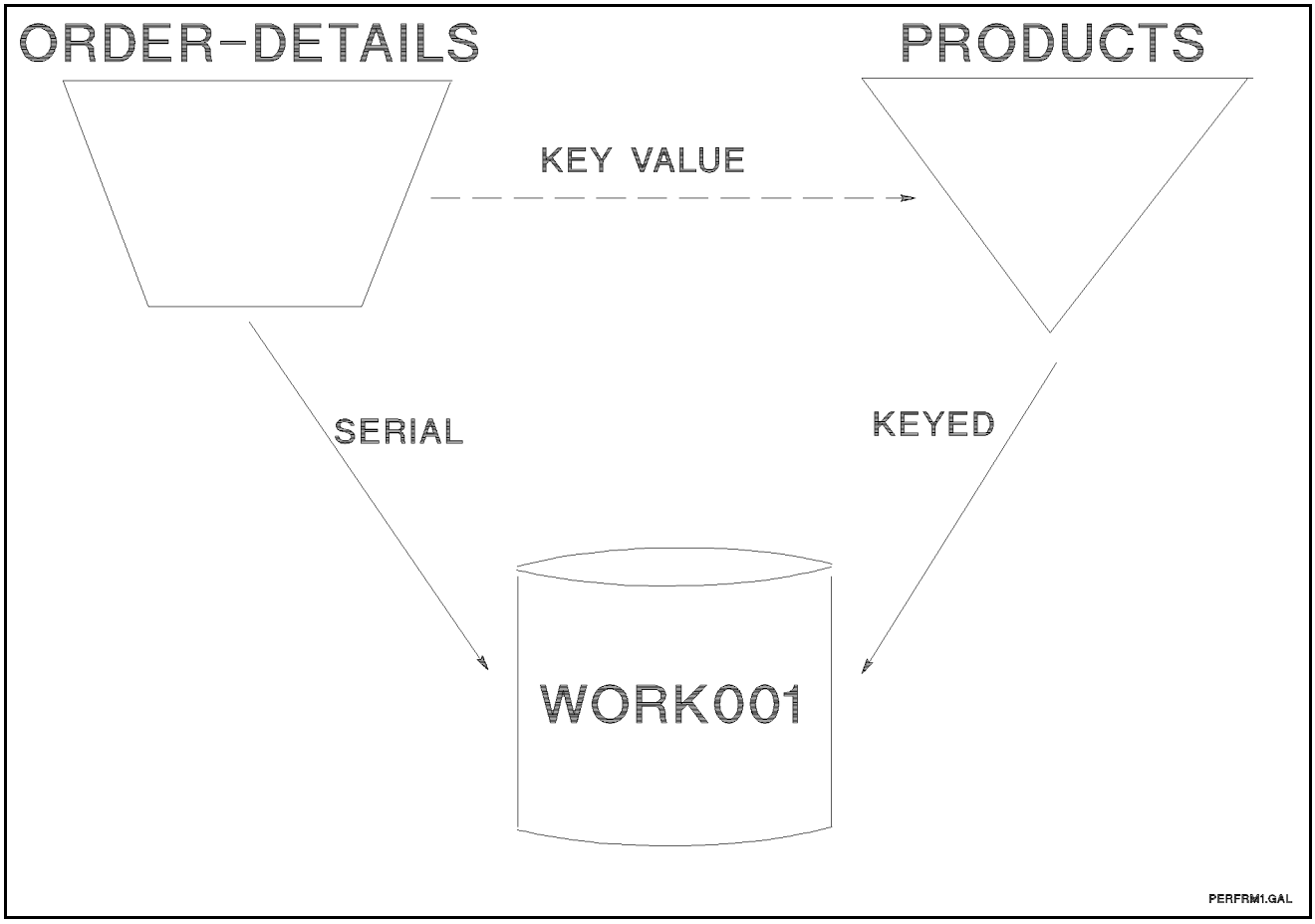
HP ALLBASE/BRW uses serial access to HP ALLBASE/SQL tables, IMAGE datasets, KSAM and MPE files which are to be merged. Datasets, files or HP ALLBASE/SQL tables files are accessed in the access sequence specified on the Tune Access screen.

Keyed Access For Joins

Keyed access can be used for IMAGE datasets and KSAM files which are joined. In addition, IMAGE datasets, KSAM and MPE files can be accessed via logical record number, which is also a type of key. When a key is specified, HP ALLBASE/BRW uses:

- chained access to IMAGE detail datasets
- calculated access to IMAGE master datasets
- keyed access to KSAM files

- direct access if the key is the item *table-rec-number*



Keyed Access for a Joined Table

Since keyed access causes random access to datasets and files, it results in good execution performance, **if any of the following are true:**

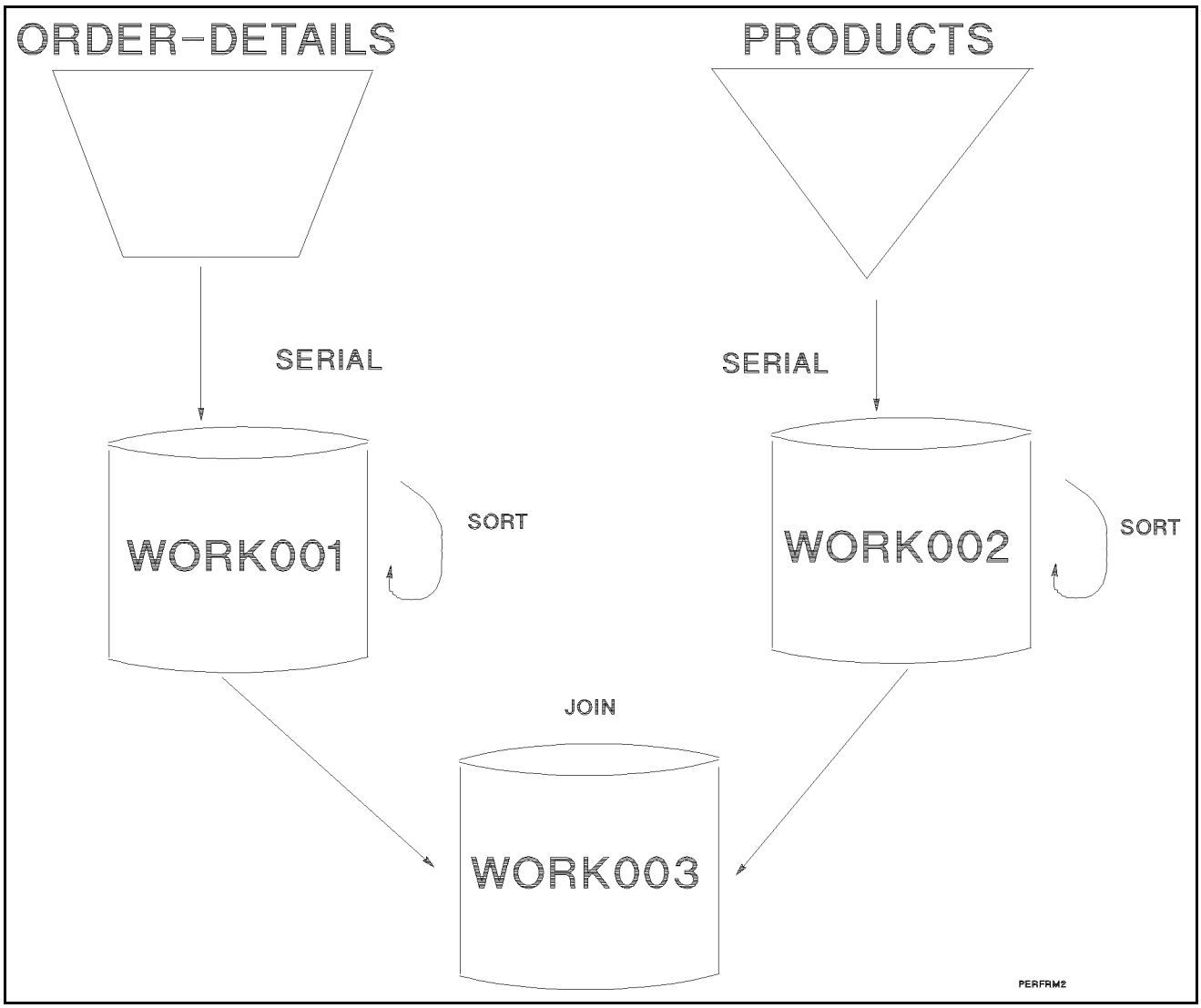
- The report selects a small amount of data out of a large amount, for example, 5,000 out of 300,000 records.
- The accessed datasets or files are small enough to be buffered to a high degree by the disk caching.

Serial Access For Joins

Serial access can be used for IMAGE datasets, HP ALLBASE/SQL tables, KSAM or MPE files which are joined. For best effects, HP ALLBASE/BRW uses the following method to perform serial access with joined tables:

- All datasets, files or HP ALLBASE/SQL queries are copied into temporary workfiles. During the copy, all possible selections and table calculated items are performed on single datasets, files or queries. Also, only items actually used in the report are written to the temporary work files.

- All temporary work files are sorted by common items in the relation definitions.
- The temporary work files are joined by reading them in parallel in one pass.



Serial Access

Although serial access requires three passes over the data, it may be necessary if common items do not have keys.

If the selections in the report can be performed when each dataset or file is read, serial access causes a more effective access profile to the disks (minimum head movements, fewer block transfers) compared to keyed access, which causes more block transfers and head movements on the disks (see *Keyed Access*, above).

For a small amount of data, serial access is slower than keyed access because of the additional passes over the data.

Access Sequence For Joins

The sequence in which the datasets, files, or HP ALLBASE/SQL tables are accessed can also influence the report execution performance. You should choose an access sequence that can reduce the amount of data as early as possible. See the compile listing, discussed later in this chapter, for when and which selections are performed.

For keyed access, the data selections should be performed as early as possible; that is, datasets or files which contain items used for selections should be accessed first.

For serial access, the access sequence does not have as much impact on execution performance, but you can follow the same rules as for keyed access.

Access Sequence For HP ALLBASE/SQL Tables

If HP ALLBASE/SQL tables have different access sequence numbers, a query is sent to HP ALLBASE/SQL for each table. HP ALLBASE/SQL tables with the same access sequence number are combined to one HP ALLBASE/SQL query.

In general, it is more efficient to combine all HP ALLBASE/SQL tables from a single DBEnvironment into a single query by giving them all the same access sequence number on the Tune Access screen. This allows HP ALLBASE/SQL to apply internal optimization and internal SQL indexes to join the tables in the most efficient way.

If separate queries are given for each HP ALLBASE/SQL table, HP ALLBASE/BRW will have to save the results of these queries into temporary files and sort these temporary files on the common relation items before combining the tables.

Caution

There are exceptions to the general rule that multiple HP ALLBASE/SQL tables should be combined into a single SQL query.

If the common relation items joining all the tables in the HP ALLBASE/BRW report do not include items directly relating two SQL tables, HP ALLBASE/BRW will generate an SQL query such as the example shown on the following page.

```
SELECT SQLA.item1,SQLB.item2 FROM SQLA,SQLB;
```

Since there is no WHERE clause, HP ALLBASE/SQL is forced to combine every record in SQLA with every record in SQLB, a condition called a *production*. Productions are very inefficient for tables with a large number of records. For instance, if SQLA has 5000 records and SQLB has 10000 records, the SQL SELECT statement listed above would return 50,000,000 records (5000 times 10000)!

In this case, it is more efficient to use separate access sequence numbers to generate two SQL SELECT statements such as the following:

```
SELECT SQLA.item1 FROM SQLA;
```

and

```
SELECT SQLB.item2 from SQLB;
```

Let HP ALLBASE/BRW join this information to the non-SQL data in the report.

When combining HP ALLBASE/SQL data with HP IMAGE/3000 or KSAM data, it is usually best to use the access sequence numbers in the Tune Access screen to load the HP ALLBASE/SQL tables first.

The Tune Access screen does not accept key items for HP ALLBASE/SQL tables, so putting them first allows keyed joins by giving key items for the following HP IMAGE/3000 or KSAM tables.

Using the Compile Listing

HP BRW/XL's compiler generates a listing that shows how the relational data access definition was translated and the optimizations that have been performed.

Example

Below are two examples of how you can use the compile listing for a report.

For these examples, the report uses the table PRINT-TABLE which joins the datasets PRODUCTS and ORDER-DETAILS.

The Define Table screen for this join looks like this:

```
HP ALLBASE/BRW                  Define Table                  Report: ORDPRODS

    Table  PRINT-TABLE          Type  J   J = Join  M = Merge

 Source Table(for SQL preceded by'Owner.')  Location (DB/DB Env/File)  Password
 PRODUCTS                                   TOYDB.PUB
-ORDER-DETAILS                             -TOYDB.PUB                 -
```

The report contains these definitions:

- a **selection condition**, as shown here:

```
MONTH OF (ORDER-DETAILS.SHIPMENT-DATE) = ?MONTH

AND

PRODUCTS.PRODUCT-LINE SATISFIES ?PRODUCT-LINE
```

- the **table calculated item** *on-stock-amount* with the following formula:

```
PRODUCTS.PRICE * PRODUCTS.QUANTITY
```

- the **table calculated item** *ordered-amount* with the following formula:

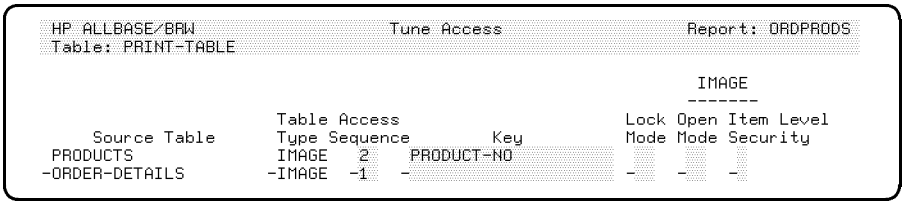
```
PRODUCTS.PRICE * ORDER-DETAILS.QUANTITY
```

- the **table calculated item** *month-words* with the following formula:

```
?MONTH OF ("JAN", "FEB", "MAR", "APR", "MAY", "JUN"
          "JUL", "AUG", "SEP", "OCT", "NOV", "DEC")
```

Example 1: Using Keyed Access For This Report

To use keyed access for this report, the following data access methods are specified on the Tune Access screen:



The listing generated by the compiler looks like this:

```
D A T A   A C C E S S   S T R U C T U R E
=========================================
PRINT-TABLE  (WORK001) join of:
------------------------------
   ORDER-DETAILS, TOYDB.PUB  serial
   PRODUCTS, TOYDB.PUB  key:  PRODUCT-NO

T A B L E   C A L C U L A T I O N S   A N D   S E L E C T I O N S
=================================================================
PRINT-TABLE  (WORK001)
   ORDER-DETAILS,  TOYDB.PUB

      Selection:  month_of  (ORDER_DETAILS.SHIPMENT_DATE) = ?MONTH

   PRODUCTS, TOYDB.PUB

      Calculated items : ON-STOCK-AMOUNT  ORDERED-AMOUNT

      Selection : PRODUCTS.PRODUCT-LINE  satisfies  ?PRODUCT-LINE

C O N S T A N T   C A L C U L A T E D   I T E M S
================================================
MONTH-WORDS
```

The DATA ACCESS STRUCTURE section on this listing shows that PRINT-TABLE creates the work file WORK001, which contains

the result of the join. The join is performed by reading the dataset ORDER-DETAILS serially and the dataset PRODUCTS by the key *product-no*.

The TABLE CALCULATIONS AND SELECTIONS section shows what is performed when each data set is read:

When the dataset ORDER-DETAILS is read, the following is performed:

```
MONTH_OF (ORDER-DETAILS.SHIPMENT-DATE) = ?MONTH
```

When a record from the dataset PRODUCTS is read, the following is performed:

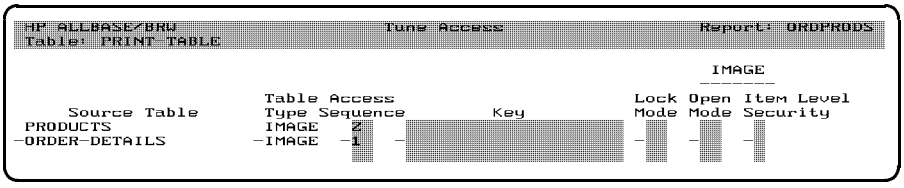
- Calculation of the items *on-stock-amount* and *ordered-amount*
- Selection of the following:

```
PRODUCTS.PRODUCT-LINE SATISFIES ?PRODUCT-LINE
```

The CONSTANT CALCULATED ITEMS section shows that the calculated item *month-words* has a constant value for the whole report (because it is calculated only from constants and parameters), so it is calculated only once.

Example 2: Using Serial Access for This Report

In this example, the datasets are accessed serially. The Tune Access screen below shows the access sequence:



The compiler generates the following compile listing:

```
D A T A   A C C E S S   S T R U C T U R E
========================================
PRINT-TABLE   (WORK003) join of:
-------------------------------
   WORK001
      PRODUCTS, TOYDB.PUB  serial

   WORK002
      ORDER-DETIALS, TOYDB.PUB  serial


T A B L E   C A L C U L A T I O N S   A N D   S E L E C T I O N S
================================================================
PRINT TABLE  (WORK003)
    WORK001

     WORK002

        Calculated items:  ORDERED-AMOUNT

WORK001
    PRODUCTS, TOYDB.PUB

           Calculated items : ON-STOCK-AMOUNT

           Selection : PRODUCTS.PRODUCT-LINE  satisfies ?PRODUCT-LINE
WORK002
    ORDER-DETAILS, TOYDB.PUB

           Selection : month-of (ORDER-DETAILS.SHIPMENT-DATE) = ?MONTH

C O N S T A N T   C A L C U L A T E D   I T E M S
================================================
MONTH-WORDS
```

The DATA ACCESS STRUCTURE section of the listing shows that the table PRINT- TABLE creates a work file WORK003, which contains the result of the join. The join is performed by reading the work files WORK001 and WORK002. WORK001 is created by reading the dataset PRODUCTS serially; WORK002 by reading ORDER-DETAILS serially.

The TABLE CALCULATIONS AND SELECTIONS section shows that the item *ordered-amount* is calculated when the work files are joined. Selections and calculation of the item *on-stock-amount* are performed when the datasets are read, which is earlier during execution of the report.

The CONSTANT CALCULATED ITEMS section is the same as in Example 1.

Example 3: Accessing HP ALLBASE/SQL

Below is the Tune Access screen from "Combining Data from Different Sources" of Chapter 6. In this example, the two HP ALLBASE/SQL tables default to the same access sequence number.

```
HP ALLBASE/BRW                    Tune Access                    Report: PARTPROD
Table: PART-PRODUCT

                                                                    IMAGE
                                                                    -------
                          Table Access                          Lock Open Item Level
     Source Table         Type Sequence        Key              Mode Mode Security
 PRODUCTPARTS             SQL     1
-PARTS                   -SQL    -1   -                          -    -    -
 PRODUCTS                 IMAGE   2    PRODUCT-NO                      Lock Mode:
```

The compile listing on the next page shows that when the access sequence number is the same, only one HP ALLBASE/SQL query is generated for both tables.

Selection conditions for HP ALLBASE/SQL do not appear under TABLE CALCULATIONS AND SELECTIONS when HP ALLBASE/SQL makes the selection. Instead, selections that can be performed by HP ALLBASE/SQL show up in the SQL query included at the bottom of the compile listing.

By assigning different access sequence numbers to HP SQL tables, separate queries are generated and HP ALLBASE/BRW performs the join.

```
D A T A   A C C E S S   S T R U C T U R E

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

ROOT (WORK001) join of:

----------------------

   CUSTOMERS,EXPLSQL.EXPLDB.TESTSQL  SQL

   INVOICES,EXPLSQL.EXPLDB.TESTSQL  SQL

   ORDERS,EXPLDB.EXPLDB.TESTSQL  key: ORDER-NO



T A B L E   C A L C U L A T I O N S   A N D   S E L E C T I O N S

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

ROOT (WORK001)

   CUSTOMERS,EXPLSQL.EXPLDB.TESTSQL

   INVOICES,EXPLSQL.EXPLDB.TESTSQL

   ORDERS,EXPLDB.EXPLDB.TESTSQL

      Selection: ORDERS.ORDER-DATE = ?ORDERDATE


G E N E R A T E D   S Q L   Q U E R I E S

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

ROOT (WORK001)

   Query Number: 1


   SELECT  T1.CUSTOMER_NAME,T2.ORDER_NO

   FROM    MGR@TESTSQL.CUSTOMERS  T1,MGR@TESTSQL.INVOICES T2

   WHERE   (T1.CUSTOMER_NO=t2.CUSTOMER_NO) AND (T1.CUSTOMER_NAME=?CUSTOMER):
```

Execution Statistics

During report execution, HP ALLBASE/BRW optionally provides statistics on the amount of time each part of the report needed (CPU and elapsed time), and how many records have been read, written or sorted.

The following are the associated execution statistics for the compile listings for Examples 1, 2, and 3.

Example 1: Statistics of Keyed Access

```
Access Block 1:
                                      $records  cpu-sec     elapsed-sec
                                      --------  -------     -----------

   read-dset serial ORDER-DETAILS, TOYDB    98     0.518          2.201
   read-dset calc   PRODUCTS, TOYDB         23     0.281          2.864
   write            WORK001                 23     0.282          0.902
   sort             WORK001                 23     1.101          3.932
   process                                         1.432          3.992
                                                   =======    ==========
   time totals for block                           3.614         13.891

Access Block 2
                                      $records  cpu-sec     elapsed-sec
                                      --------  -------     -----------

   read-mpe serial  WORK001                 23     0.235          0.325
   print            PFK                     40     0.909          1.425
   process                                         0.830          1.026
                                                   =======    ==========
   time totals for block                           1.974          2.776

Report Info :



                                                   cpu-sec     elapsed-sec
                                                   =======     ===========
   time totals for report                           5.588          16.667
```

Example 2: Statistics of Serial Access

```
Access Block 1:
                                        $records  cpu-sec     elapsed-sec
                                        --------  -------     -----------
   read-dset serial PRODUCTS, TOYDB        30     0.279          0.380
   write            WORK001                30     0.336          0.681
   sort             WORK001                30     0.942          3.473
   process                                        0.966          1.450
                                                  =======     ==========
   time totals for block                          2.523          5.984

Access Block 2
                                        $records  cpu-sec     elapsed-sec
                                        --------  -------     -----------
   read-mpe serial  ORDER-DETAILS, TOYDB   98     0.500          0.657
   write            WORK002                23     0.284          0.734
   sort             WORK002                23     1.065          2.665
   process                                        0.813          0.907
                                                  =======     ==========
   time totals for block                          2.662          4.963

Access Block 3
                                        $records  cpu-sec     elapsed-sec
                                        --------  -------     -----------
   read-mpe serial  WORK001                30     0.265          0.380
   read-mpe search  WORK002                23     0.358          0.587
   write            WORK003                23     0.315          0.497
   sort             WORK003                23     1.150          2.541
   process                                        0.961          0.989
                                                  =======     ==========
   time totals for block                          3.049          4.994

Access Block 4
                                        $records  cpu-sec     elapsed-sec
                                        --------  -------     -----------
   read-mpe serial  WORK003                23     0.235          0.314
   print            PFS                    40     0.867          1.220
   process                                        0.840          0.950
                                                  =======     ==========
   time totals for block                          1.942          2.484


Report Info :

                                                  cpu-sec     elapsed-sec
                                                  =======     ============
   time totals for report                         10.176          18.425
```

Example 3: Statistics of HP ALLBASE/SQL Access

```
Access Block 1:

                                  #records    cpu-sec     elapsed-sec

                                  ---------   ---------   -----------

 read-sql         QUERY    1,EXPLSQL        4      0.686         0.815

 read-dset chain  ORDERS,EXPLDB             4      0.031         0.039

 write            WORK001                   4      0.079         0.272

 process                                           1.556         3.469

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

 time totals for block                            2.352         4.595

 (0     segment loads for block)


Access Block 2

                                  #records    cpu-sec     elapsed-sec

                                  ---------   --------    -----------

 read-mpe   serial WORK001                4      0.147         0.146

 print             TESTLIST               4      0.109         0.500

 process                                           0.038         0.038

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

 time totals for block                            0.294         0.684

 (0     segment loads for block)




Report Info :


                                              cpu-sec       elapsed-sec

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

 time totals for report                          2.646           5.279
```

Effective Calculations

This section discusses some rules for effective calculations, especially with more complicated reports. For more details about Calculations, refer to Chapter 15.

The Calculation Language for HP ALLBASE/BRW is expression-oriented. That is, every formula has a resulting value — the value of the calculated item or function.

Programming languages like COBOL or PASCAL are statement-oriented. You specify both the control flow and the data flow explicitly. You define which data goes into which variables and when.

For example in PASCAL you can write:

```
a := b + c;
e := a / d;
```

By the sequence in which you write the calculation, you explicitly specify that `a` is to be calculated before `e`.

To do the same calculation in HP ALLBASE/BRW, you define two calculated items in any sequence:

- item A with the formula: `B + C`
- item E with the formula: `A / D`

The formulas are similar to the right side of the assignment statements in PASCAL, but the difference is that you do not specify the sequence in which A and E will be calculated. HP ALLBASE/BRW does that for you by analyzing the formulas of the calculated items and using the most efficient sequence for the calculation. You don't have to know anything about HP ALLBASE/BRW's internal control flow.

When you use HP ALLBASE/BRW's calculation language, there are ways to use it very efficiently and other ways which are not so efficient, in much the same way as with programming languages. The following gives you some insight into how HP ALLBASE/BRW calculations work and some hints about how to define calculations effectively.

How Calculated Items Work

Calculated items are items which get their values by the execution of an algorithm defined in the calculated item's formula. Their purpose is to produce values which are not contained in any of the source tables, as such, but can be derived from other items. A calculated item can be used in the report like any other item.

In a table every item or every component of an array item has a single value per record. The same is true for calculated items, too. Therefore, HP ALLBASE/BRW calculates them only once per record. If you have defined calculated items which use other calculated items in their formulas, as in the example above in which E uses A in its formula, HP ALLBASE/BRW's compiler analyzes

these dependencies and automatically generates code such that A is calculated before E.

If A is used in more than one other calculated item, for example in E, F and G, A is only calculated once before E, F and G. These items then only use the result value of A. This means that HP ALLBASE/BRW generates very effective code if calculated items are used by other calculated items.

HP ALLBASE/BRW does not permit cycles or recursion in calculations. For example, if you define the calculated items:

```
item R   with the formula:  S + 1
item S   with the formula:  R + 1
```

where R uses S and S uses R, HP ALLBASE/BRW's compiler could not determine which item would be calcu lated first. In this case you would get a message that there is a `Cycle in the calculated item formula` when the report is compiled.

How Functions Work

Functions are used to define common formulas. They can only be used in other formulas and cannot, for example, be printed or used for sorting like an item. This is because functions need values for their function arguments which can only be specified in formulas. Function arguments are like function parameters in PASCAL. They give you a lot of flexibility when you define function formulas.

On the other hand, function arguments do not allow the calculation of functions to be optimized in the way HP ALLBASE/BRW optimizes calculated items. If a function is used in three calculated items, the function is calculated three times because it may have different function arguments and therefore three different results. HP ALLBASE/BRW does not determine if two or more function calls use arguments of the same value.

Beware of putting too much into a function formula. Here is an example of a function that could be made simpler:

function name	**argument**
X	(a, b, mode)

The formula for this function is defined as:

```
IF mode = 1 THEN a + b
            ELSE a - b
```

The function is used in two formulas for calculated items:

item name	**with the formula:**
SUM	X (k, l, 1)
DIFFERENCE	X (m, n, 2)

Instead of the more complex function (X), it would be better to use two simpler functions, one for the addition and one for the subtraction:

function and arguments	with the formula:
ADD (a, b)	a + b
SUB (a, b)	a - b

Then you could use the two functions in a simpler way with the calculated items:

item name	with the formula:
SUM	ADD (k, l)
DIFFERENCE	SUB (m, n)

Function X is also not efficient in terms of performance, since the IF THEN ELSE must be evaluated each time the function is calculated. You can avoid this by using the functions ADD and SUB. If you have functions with many IF THEN ELSEs, determine whether you can divide them into simpler, separate functions.

In this discussion about functions, we have talked primarily about *using* functions and not about *calling* functions. This is because there is really no mechanism to call functions in HP ALLBASE/BRW. Instead, a function in HP ALLBASE/BRW is more like a macro than like a function in PASCAL, because HP ALLBASE/BRW's compiler fills in the function's code every time a function is used. So, if a function is used in 50 calculated items, the function's code is duplicated 50 times!

You can reach HP ALLBASE/BRW's limits, (otherwise not easy to achieve), by using complex functions in many calculated items or in other functions. If this occurs, HP ALLBASE/BRW's compiler aborts with the following message:

```
Max. number of segments exceeded
```

which means that too much code will be generated (> 504 KBytes).

Therefore, keep function formulas small if you plan to use the functions in many places. Better still, when possible use calculated items instead of functions.

As with calculated items, cycles (recursions), are not allowed in functions.

Rules For Effective Calculations

In report writing, using effective calculations means achieving good performance for calculations. Since the calculations may need to be done very often in a report, depending on the volume of data, calculation formulas could be performed several thousand or even several hundred thousand times.

In a programming language you achieve good performance by structuring the code well. You can do this in HP ALLBASE/BRW too, to get calculations that have good performance and are easy to maintain.

The most important points are listed below:

- **RULE 1**

  Avoid repeating identical blocks of code.

  In a programming language, you would *not* write code with a block that is identical, like the following:

```
a := c * (d - e + 4 * (f / (100 - g)));
b := h * (d - e + 4 * (f / (100 - g)));
```

  Instead, you would use a temporary variable:

```
i := d - e + 4 * (f / (100 - g));
a := c * i;
b := h * i;
```

  You can also use temporary variables in HP ALLBASE/BRW. You can define a calculated item, for example:

item name	**with the formula:**
I	`D - E + 4 * (F / (100 - G))`

  Then you can use the new item in the other calculated items:

item name	**with the formula:**
A	C * I
B	H * I

  Note that you do not need to specify that I is to be calculated before A and B; HP ALLBASE/BRW recognizes this. You also do not need to print the item I; you can use it just as a temporary variable.

  In a programming language, you can use a function as another way of structuring, such as:

```
INTEGER FUNCTION x (y : integer);
BEGIN
 x := y * (d - e + 4 * (f / (100 - g)));
END;
```

  You can use this function to calculate the values for `a` and `b`:

```
a := x (c);
b := x (h);
```

  This structuring provides you with easily maintained formulas, but not with good performance, since the whole function formula is calculated twice. If you use a function in HP ALLBASE/BRW, performance will be even worse because not only will the function be calculated twice, but its code will also be duplicated and therefore will require more memory.

- **RULE 2**

  Try to use calculated items instead of functions.

  Sometimes structuring may not be so easy. For example, if you have the following calculated items:

  item name **with the formula:**

  A

```
IF     r = 3  AND  s = 0
 OR  r = 4  AND  s <> 4
 OR  r = 5  AND  s <> 17
 OR  r > 7  AND  s = 3
 OR  r < 2  AND  s <= 4
THEN   c + 3  ELSE c - 3
```

  item name **with the formula:**

  B

```
IF     r = 3  AND  s = 0
   OR  r = 4  AND  s <> 4
   OR  r = 5  AND  s <> 17
   OR  r > 7  AND  s = 3
   OR  r < 2  AND  s <= 4
THEN   h + 2  ELSE h - 2
```

  The largest parts of both formulas, the IF conditions, are identical.

  It is not so simple to avoid the duplicated code in these formulas, but you can do so by using a calculated item as a temporary variable. See the following example:

  item name **with the formula:**

  COND

```
IF    r = 3  AND  s = 0
  OR  r = 4  AND  s <> 4
  OR  r = 5  AND  s <> 17
  OR  r > 7  AND  s = 3
  OR  r < 2  AND  s <= 4
THEN  1  ELSE  0
```

  Then use COND in the formulas of A and B:

  item name **with the formula:**

  A

```
IF cond = 1 THEN c + 3 ELSE c - 3
```

  item name **with the formula:**

  B

```
IF cond = 1 THEN h + 2 ELSE h - 2
```

Table Calculated Items or Layout Calculated Items?

You should consider the performance differences between table calculated items and layout calculated items if (BUT ONLY IF) you have many calculations in a report.

Optimizing Table Calculated Items

Table calculated items are calculated as early as possible, that is, as soon as the items used in their formulas become available during the data access phases of a report. In the early data access phases, there are far fewer records than after the tables are joined. Therefore, calculations can be performed less frequently if they are performed after just one, or a few, datasets or files have been read.

For example: in the database TOYDB, each record in the dataset PRODUCTS has several corresponding records in the dataset ORDER-DETAILS because, typically, there are several orders per product. If you join both datasets in a report, you will get the values of the items of PRODUCTS repeated for each related ORDER-DETAILS record. Suppose you defined a table calculated item ON-STOCK-VALUE which uses items from the dataset PRODUCTS. HP ALLBASE/BRW calculates ON-STOCK-VALUE just after a record from the dataset PRODUCTS has been read and the local selection has been performed.

Optimizing Selections

If you tune the data access of the report so that PRODUCTS is read first and then ORDER-DETAILS is read in keyed mode, or if both datasets are read serially, the item ON-STOCK-VALUE will be calculated much less frequently than when it is calculated for every detail in the report. But, if you have defined a selection on an item from dataset ORDER-DETAILS, you should access ORDER-DETAILS first, because early selections save more time than early calculation of calculated items.

Selections are also optimized so that they are performed as early as possible. If a selection or relation condition contains the logical operator AND, for example, `X = Y AND Z = 40`, the condition is divided into separate, smaller selections, `X = Y` and `Z = 40`, and each of these is optimized individually. The compile listing informs you when HP ALLBASE/BRW actually performs the selections and calculates the table calculated items. Table calculated items are not necessarily calculated in the table where they were defined, especially if the report has nested tables.

Table calculated items which use only parameters and constants but no other items in their formulas are calculated only once in the report. The compile listing shows them as `Constant Calculated Items`.

It is a good idea to use table calculated items if you use them in the report layout, but do not use those items from which they are calculated. This can make the record length of HP ALLBASE/BRW's internal work files much smaller. For example

in HPFA (HP Financial Accounting) the ACCOUNT-MASTER dataset contains about a hundred items that hold the account budget, debit and credit values for 28 periods. In most HPFA reports only the account budget and the account balance for the current period and their corresponding cumulative values are printed. After calculating the four table calculated items, HP ALLBASE/BRW does not need the values of the hundred items and so throws them away. This reduces the record length of the work file by more than 1000 bytes and makes sorting faster.

Optimizing Layout Calculated Items

Layout calculated items, on the other hand, are preferable if you have calculated items in addition to other items. For example, if you calculate the budget/balance variance and print it along with the items from which it was calculated, it is better to define it as a layout calculated item. This keeps the record length of the workfile smaller. However, if you want to select or sort on this calculated item, you need to define it as a table calculated item.

HP ALLBASE/BRW does not optimize layout calculated items automatically. But you can do this manually. For example, if you print the invoices for customers, you typically print the customer information in a break header and footer. The break headings and footings are printed less frequently than the details of the invoices. If you have a layout calculated item which you print only in the break header or footer, you can specify CALC in the corresponding ColCalc field on the Define Lines screen. This tells HP ALLBASE/BRW that you want this item to be calculated only when this lineset is actually printed (and not for every detail).

Specifying CALC as above also tells HP ALLBASE/BRW that you want to use the other items in the calculated item's formula with the same column calculation and numeric precision as specified for this lineset. For example, if you have a layout calculated item PERCENT-VARIANCE with the formula `100 * BUDGET / BALANCE`, and if you specify TOTAL for the items BUDGET and BALANCE and CALC for PERCENT-VARIANCE, HP ALLBASE/BRW uses the total values of BUDGET and BALANCE to calculate PERCENT-VARIANCE. This is exactly what you want.

20

Intrinsics

The description and operation of HP ALLBASE/BRW intrinsics

At a Glance

You can use the intrinsics described in this chapter to build HP ALLBASE/BRW reports into your application programs. Your application programs can initiate and control the execution of reports. The two types of intrinsics are called Request Report Intrinsics and Execution Report intrinsics respectively.

This chapter covers

- refreshing the terminal
- request report intrinsics
- execution report intrinsics

Refreshing the Terminal

The BRWSTARTREQUEST intrinsic calls BRWSTART.PUB.SYS, which switches your terminal to block-mode. Your application program will have to refresh the terminal after control has been passed from BRWSTART back to your application. The method used to refresh your terminal varies depending upon whether your application program uses block mode or character mode.

Character Mode

If your application uses character mode, call VOPENTERM to turn your terminal to block mode before using BRWSTARTREQUEST. After control is returned to your application program, you should call VCLOSETERM to switch your terminal back to character mode. If your application uses function key labels, you must redisplay the labels after leaving BRWSTARTREQUEST.

Block Mode

If your application uses block mode, (that is, it uses VPLUS), you might want the screen that was displayed before the call to BRWSTARTREQUEST to be redisplayed after returning from the intrinsic. To do this you will have to first clear and reset the terminal, using $REFRESH as the nfname parameter value in a call to VGETNEXTFORM. (See the VPLUS manual for a full description). Then you must display the previous form using VSHOWFORM. If you have defined your own function key labels, you will have to display them before calling VSHOWFORM.

The following PASCAL program fragment gives an example of the process for character mode.

```
TYPE fmt_vp_comarea = RECORD

  cstatus     : short_int;
  language    : short_int;
  comarea_len : short_int;
  filler      : ARRAY [1..50] OF short_int;
    .
    .

 END; {RECORD}

 VAR
  vp_comarea  : fmt_vp_comarea;
  terminal    ; PACKED ARRAY [1..8] OF CHAR;

 BEGIN
  Terminal := 'Testterm';

  WITH vp_comarea DO
  BEGIN
    cstatus  := 0;
    language := 5;
    comarea_len := 60;
  END;

  VOPENTERM (vp_comarea,terminal);
  BRWSTARTREQUEST (REQUEST_COMAREA, parm_defaults);
  VCLOSETERM (vp_comarea);

 { Then re-display the function key labels and screen}

 END.
```

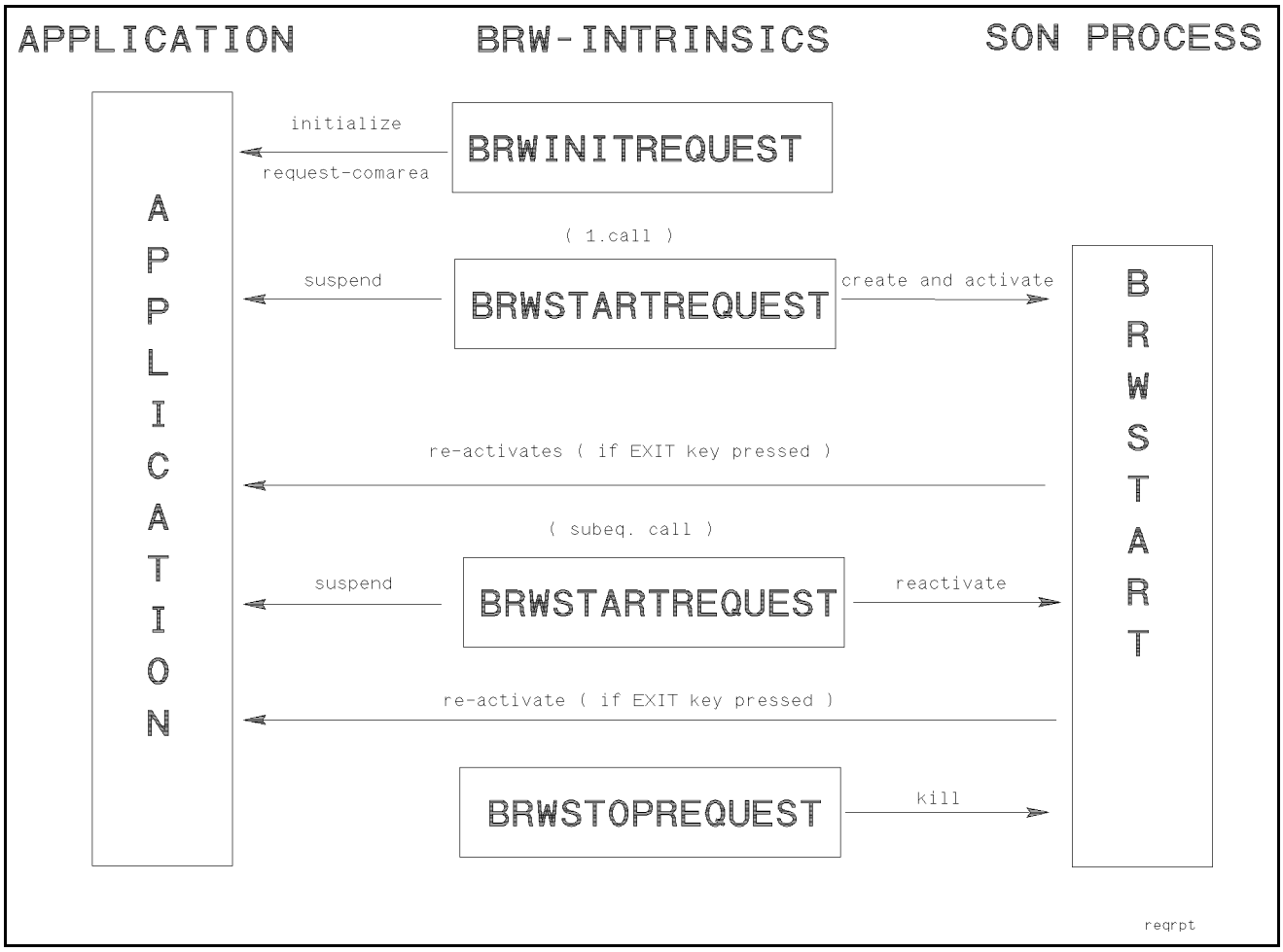
The following PASCAL program fragment gives an example of the process for block mode.

```
TYPE fmt_vp_comarea = RECORD

 cstatus     : short_int;
 language    : short_int;
 comarea_len : short_int;
  .
  .
 nfname      : PACKED ARRAY [1..16] OF CHAR;
  .

END; {RECORD}

VAR
 vp_comarea  : fmt_vp_comarea;

BEGIN

 BRWSTARTREQUEST (REQUEST_COMAREA, parm_defaults);
 vp_comarea.nfname := '$REFRESH';
 VGETNEXTFORM (vp_comarea);

{ Then re-display the function key labels }

 VSHOWFORM (vp_comarea)

END.
```

Request Report Intrinsics

HP ALLBASE/BRW includes a program called BRWSTART which allows you to request HP ALLBASE/BRW or report job files, using the screens set up by HP ALLBASE/BRW for the purpose. You can use the request report intrinsics described here to call BRWSTART and therefore to access the Request Report, Select Report, and Online Review screens from within your application program.

The following diagram shows how the request report intrinsics operate.



This table is a quick overview of the request report intrinsics:

Intrinsic Name	Description
BRWINITREQUEST	This intrinsic initializes the parameter REQUEST_COMAREA. Parameter REQUEST_COMAREA is used by all the request report intrinsics.
BRWSTARTREQUEST	This intrinsic creates a son process and runs the BRWSTART program. It can also pass information about the default values of selections and parameters. The actual values are entered by the user on the HP ALLBASE/BRW screens called by BRWSTART.
BRWSTOPREQUEST	This intrinsic halts the BRWSTART program. A program that calls the Request Report intrinsics must allocate a data area for communication with the intrinsics. This data area is the parameter REQUEST_COMAREA. This parameter (or items within it) is used by all the Request Report intrinsics. The structure of the REQUEST_COMAREA is shown below the description of the BRWINITREQUEST intrinsic. All parameters are integer-array reference parameters and have word addresses.

The next sections give the syntax and other information about these intrinsics.

BRWINITREQUEST

Call BRWINITREQUEST like this:

```
BRWINITREQUEST (REQUEST_COMAREA)
```

This intrinsic initializes the common area (REQUEST_COMAREA) prior to the first call to BRWSTARTREQUEST. You must set the parameter item comarea_len to 100 before calling BRWSTARTREQUEST.

REQUEST_COMAREA

REQUEST_COMAREA is a 100-byte array with the following structure:

```
return_status   :   2 byte integer;
error_parm      :   2 byte integer;
comarea_len     :   2 byte integer;
exec_filename   :  36 bytes;
start_screen    :   2 byte integer;
listf_id        :   8 bytes;
listf_group     :   8 bytes;
process_info    :  40 bytes.
```

Each item of the parameter REQUEST_COMAREA is discussed in detail below.

return_status : 2 bytes;

Integer to which the intrinsic status is returned. It is set to 0 if the call is successful, or to a non-zero value if an error occurs; possible return_status values are described in the section titled Error Handling.

error_parm : 2 bytes;

If an error occurs, this integer contains additional error information; possible error-parm values are described in the section titled Error Handling.

comarea_len : 2 bytes. (Set to 100);

You must set comarea_len to 100 before calling BRWINITREQUEST. Comarea_len is an integer containing the length in bytes of the allocated area (that is, the REQUEST_COMAREA parameter) that can be used by the HP ALLBASE/BRW Request Report intrinsics.

exec_filename : 36 bytes;

Byte array containing the name of the Report or the report job file.

start_screen : 2 bytes;

Integer that indicates whether the BRWSTART program should begin with the Select Report Screen, the Request Report Screen or the Online Review Screen.

```
0 = start with Request Report
1 = start with Select Report
2 = start with Online Review
```

listf_id : 8 bytes;

Byte array containing a string pattern for the file designator. That is, a filename or a fileset name with wildcard characters.

listf_group : 8 bytes;

Byte array containing a string pattern for the group designator. That is, a group name with or without wildcard characters.

process_info : 40 bytes;

For use by the intrinsics only. Contains information for process handling.

BRWSTARTREQUEST

Call BRWSTARTREQUEST like this:

```
BRWSTARTREQUEST (REQUEST_COMAREA, parm_defaults)
```

This intrinsic starts the BRWSTART program.

REQUEST_COMAREA

REQUEST_COMAREA is a 100-byte array with the following structure:

```
return_status   :   2 byte integer;
error_parm      :   2 byte integer;
comarea_len     :   2 byte integer;
exec_filename   :  36 bytes;
start_screen    :   2 byte integer;
listf_id        :   8 bytes;
listf_group     :   8 bytes;
process_info    :  40 bytes.
```

You must set the following items from the parameter REQUEST_COMAREA before calling BRWSTARTREQUEST (the structure of REQUEST_COMAREA is described under BRWINITREQUEST):

comarea_len: set to 100

exec_filename: Report or report job file name.

Set to the name of the report or report job file to be loaded when using the Request Report screen. This item is required to use the Online Review screen, otherwise you can leave it blank.

start_screen: 0, 1 or 2

Use the start_screen item to choose the first screen you want to display. Various screens can be called depending on the start_screen value. For example:

```
start_screen = 0
```

With the value set at 0, the Request Report screen is called first. You can then go to the Online Review screen.

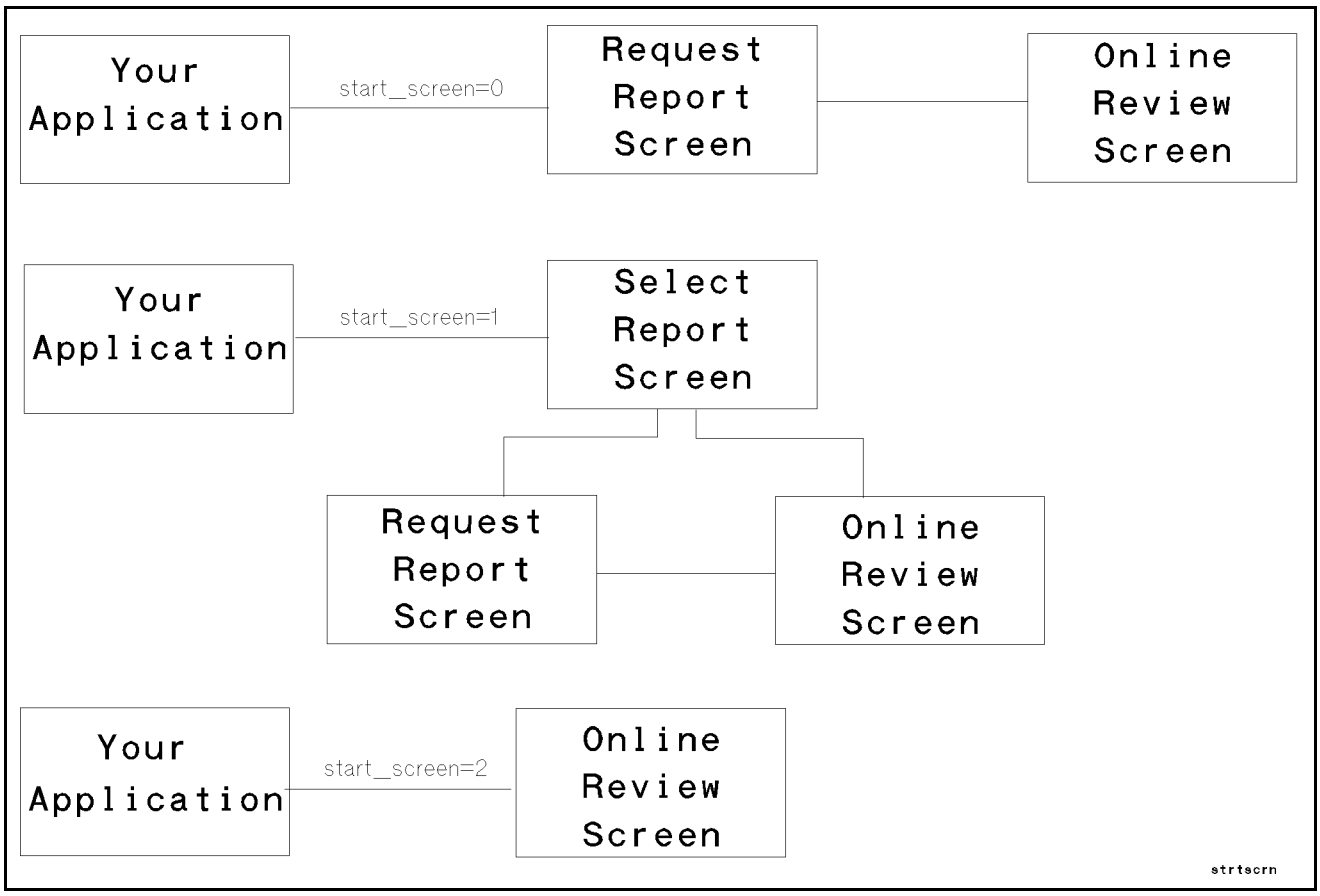
```
start_screen = 1
```

With the value set at 1, the Select Report screen is called first. You can then go to either the Request Report Screen or the Online Review screen directly. (You can also go to the Online Review screen from the Request Report Screen).

```
start_screen = 2
```

With the value set at 2, the Online Review screen is called first. No other screens can be called from the Online Review screen.

The illustration below shows the start_screen parameter and where you enter the HP ALLBASE/BRW screens for each parameter value.



listf_id: file or fileset name

Set to the file designator of the reports to be listed if the first screen is the Select Report screen.

listf_group: group or groupset name

Set to the group designator of the Reports to be listed if the first screen is the Select Report screen.

process_info: set by prior call to BRWINITREQUEST.

PARM_DEFAULTS

PARM_DEFAULTS is an integer array parameter.

The array structure is:

```
   nbr_parms    :   2 byte integer;
Then, for each parameter
      parm_name       :  20 bytes;
      default_value   :  55 bytes;
      filler          :   1 byte;
```

The array size is calculated using the formula below.

```
No. of bytes = (max number of parameters * 76) + 2
```

`nbr_parms`	an integer specifying the number of parameter descriptions
`parm_name`	a byte array holding the parameter name
`default_value`	a byte array holding the value of the parameter
`filler`	not used

PARM_DEFAULTS allows the calling program to inform HP ALLBASE/BRW about the defaults to be used for certain selections (for example, for selection on ACTUAL-PERIOD or DIVISION). If the actual report does not contain all of these parameters, any superfluous parameters are ignored.

Note

See REFRESHING THE TERMINAL earlier in this chapter for information on how to refresh your terminal after calling BRWSTARTREQUEST.

BRWSTOPREQUEST

Call BRWSTOPREQUEST like this:

```
BRWSTOPREQUEST (REQUEST_COMAREA)
```

This intrinsic stops the BRWSTART program.

`REQUEST_COMAREA` REQUEST_COMAREA is described under BRWINITREQUEST. The parameter item process_info is set by prior call to BRWSTARTREQUEST. You do not have to set any parameter items

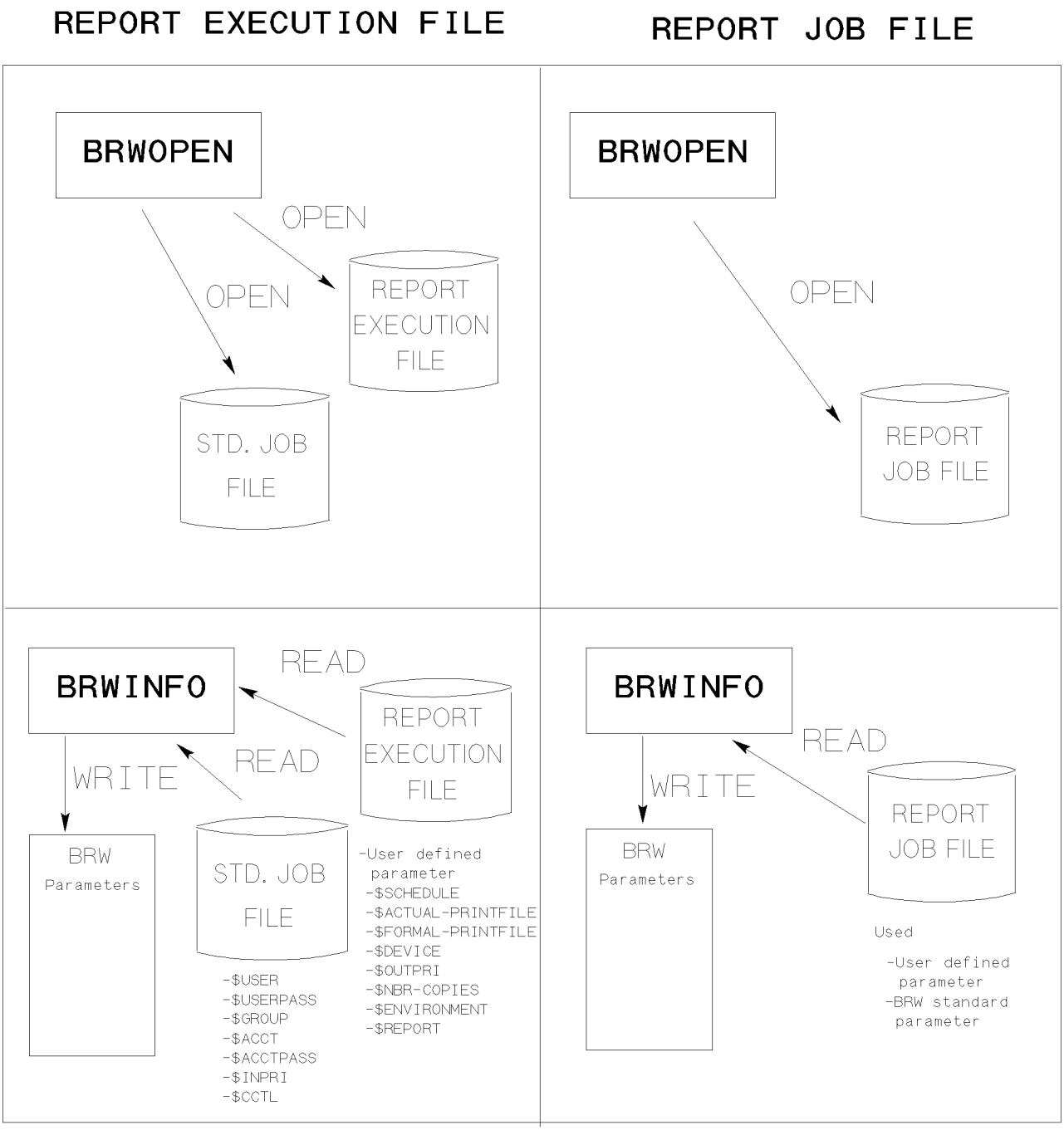
If you keep the BRWSTART program alive when returning to the application program, it provides a fast switch to the BRWSTART screens for all subsequent calls to BRWSTARTREQUEST.

Execution Intrinsics

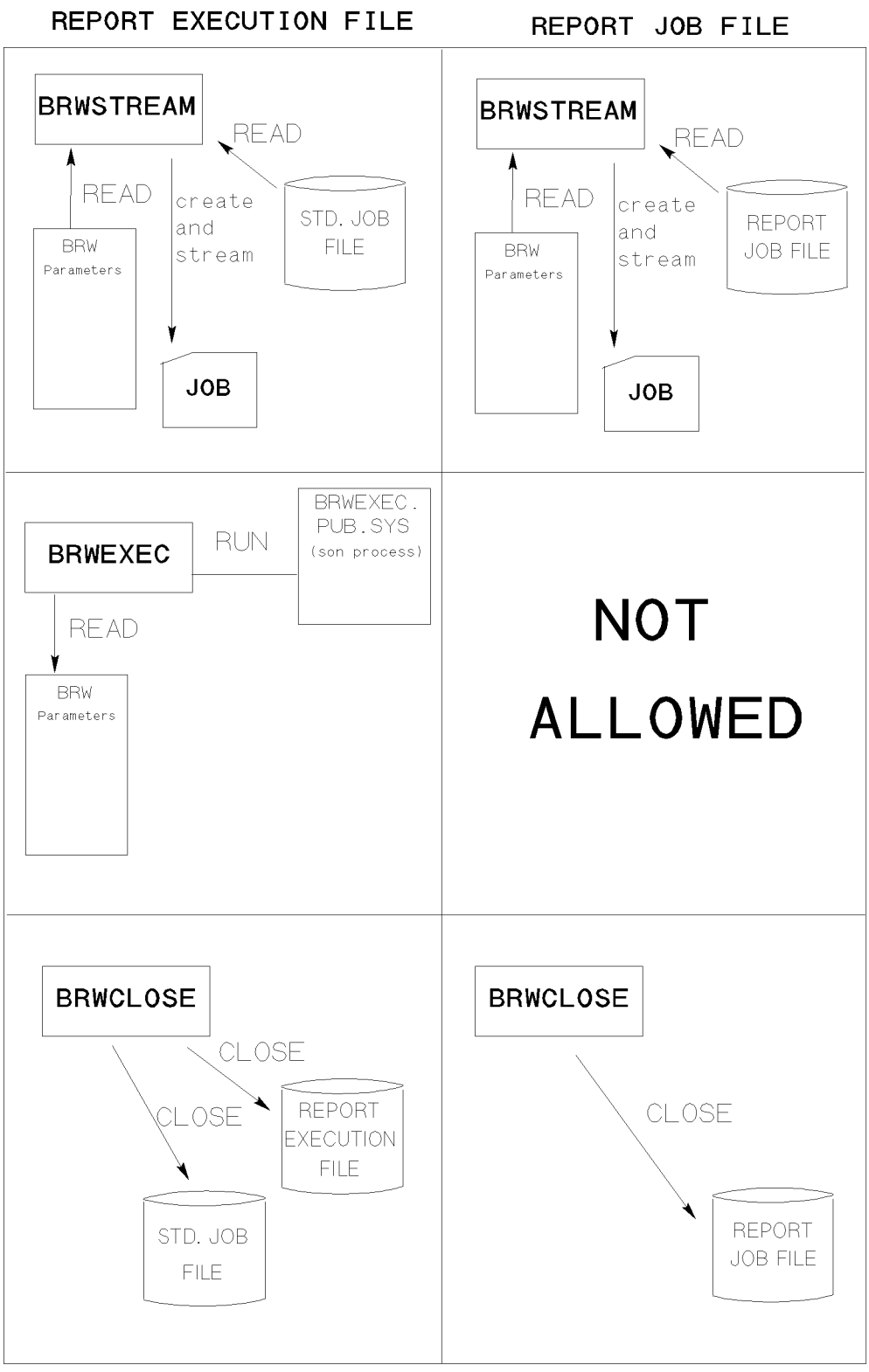
The following table gives a brief description of each HP ALLBASE/BRW intrinsic:

Intrinsic Name	Description
BRWOPEN	The intrinsic BRWOPEN opens a file or files and returns information about the files. The files opened can be *either* a report execution file and the standard job file, *or* a report job file. You determine which files are opened by supplying *either* the name of a report execution file, *or* the name of a report job file, in the FILENAME parameter when calling BRWOPEN. For example, if you call BRWOPEN with a FILENAME parameter of an Execution file, REPORT1.EXEC, BRWOPEN will open REPORT1.EXEC and the standard job file, and return information about those files in the parameter BRW_COMAREA. However, if you called BRWOPEN with a FILENAME parameter of a report job file, REPJOB1.BRWJOB, the intrinsic would only open the file REPJOB1.BRWJOB, and return information about that file in BRW_COMAREA. The information in BRW_COMAREA will be used by the other execution intrinsics.
BRWINFO	This intrinsic is called using the value in BRW_COMAREA obtained from the previous call of BRWOPEN. BRWINFO will obtain a full description of each parameter in *either* the report execution file and standard job file, *or* a report job file. The choice of files, of course, depends on the files opened by BRWOPEN. The full description will include such information as each parameter's name, type, result type, whether the parameter is required or not. The parameter descriptions will be held in the second parameter to intrinsic BRWINFO, called BRW_PARAMETERS.
BRWSTREAM	This intrinsic uses the parameter information returned by the previous call to BRWINFO, plus the file information returned by the call to BRWOPEN, *either* to stream the standard job file, which executes the report execution file, *or* to stream the report job file. The choice of files, of course, depends on the files opened by BRWOPEN.
BRWEXEC	This intrinsic executes a report execution file on line, by creating a son process that runs BRWEXEC.PUB.SYS. This intrinsic uses the information about the report execution file returned by BRWOPEN, and the parameter information returned by BRWINFO, when running BRWEXEC.PUB.SYS. Note that you *cannot* use this intrinsic with a report job file.
BRWCLOSE	This intrinsic closes the job file opened by BRWOPEN. The file closed will be *either* the standard job file, *or* the report job file. The choice of files closed, of course, depends on the files opened by BRWOPEN.

The following diagrams illustrate how each intrinsic works.



intrin1


REPORT EXECUTION FILE
REPORT JOB FILE
BRWSTREAM
READ
READ
create and stream
STD. JOB FILE
BRW Parameters
JOB
BRWSTREAM
READ
READ
create and stream
REPORT JOB FILE
BRW Parameters
JOB
BRWEXEC
RUN
BRWEXEC.PUB.SYS (son process)
READ
BRW Parameters
NOT ALLOWED
BRWCLOSE
CLOSE
CLOSE
REPORT EXECUTION FILE
STD. JOB FILE
BRWCLOSE
CLOSE
REPORT JOB FILE

intrin2

BRWOPEN

Use the BRWOPEN intrinsic to open an existing report execution file and the standard job file, or a report job file. The intrinsic checks to see if the file given as the call argument is a report job file or an execution file.

The job file (report or standard job file) is kept open for use by other intrinsics. The report execution file is closed, and all the information it contains that will be needed by the other intrinsics is kept in the BRW_COMEREA field.

Call BRWOPEN like this:

```
BRWOPEN (brw_comarea, filename)
```

The parameters have the following characteristics:

BRW_COMAREA — BRW_COMAREA has a 332 byte array with the following structure:

```
return_status   :   2 byte integer;
error_parm      :   2 byte integer;
for system use  : 328 bytes.
```

BRW_COMAREA is set by the BRWOPEN intrinsic. Return_status is an integer that is set to 0 if the call is successful, or to a non-zero value if an error occurs. Error_parm returns additional error information; all values of return_status and error_parm are described in the section Error Handling.

FILENAME — 36-byte character array that must contain the report execution file name or the report job filename. The filename must be followed by a terminating blank character.

It is a good idea to fully qualify the execution file name. For example, use REPORT1.PUB.APPLICS rather than REPORT1 or REPORT1.PUB. This will avoid a problem if you use a file equation for the printfile when the execution file name and the report printfile name are the same.

BRWINFO

Use this intrinsic to get information about the parameters in a report or report job file. The information necessary to access such files will have been returned in the parameter BRW_COMAREA by your previous call to BRWOPEN. The BRWINFO intrinsic will return information on both HP ALLBASE/BRW Standard Parameters (such as $USER, $USERPASS) and user-defined parameters.

Call BRWINFO like this:

```
BRWINFO (brw_comarea, brw_parameters)
```

Where the arguments have the following characteristics:

BRW_COMAREA — A 332 byte array with the following structure:

```
return_status       :   2 bytes;
error_parm          :   2 bytes;
for system use      : 328 bytes.
```

BRW_COMAREA has the same characteristics as in intrinsic BRWOPEN. That is, you use the values in BRW_COMAREA that were returned by the previous call to BRWOPEN. You must have made a successful call to intrinsic BRWOPEN before calling BRWINFO.

BRW_PARAMETERS — An integer array parameter. Its structure is:

```
max_num_parms       :    2 byte integer;
actual_num_parms    :    2 byte integer;
```

Then, for each parameter:

```
parm_name           :    20 bytes;
parm_type           :     2 byte integer;
parm_result_type    :     2 byte integer;
result_length       :     2 byte integer;
parm_mode           :     2 byte integer;
upshift             :     2 byte integer;
parm_value          :    55 bytes;
reserve             :     1 byte.
```

The array size is calculated using the formula below.

```
No. of bytes =  (max_num_parms * 86) + 4
```

The intrinsic returns the actual number of parameters found in the job template header or report execution file, and the

BRW_PARAMETERS array is filled with information about each parameter.

Each item of the parameter BRW_PARAMETERS is discussed in detail below.

max_num_parms : 2 bytes;

This item is an integer and is REQUIRED by the intrinsic. You must set max_num_parms to the maximum number of parameters you allow in a job template header or in the report execution file, that is, those parameters defined between BEGINPARMS and ENDPARMS or defined within the report specification, plus the HP ALLBASE/BRW Standard Parameters ($USER, $USERPASS, $GROUP, $GROUPPASS, $ACCT, $ACCTPASS, $REPORT, $SCHEDULE, $FORMAL-PRINTFILE, $DEVICE, $OUTPRI, $NBR-COPIES, $INPRI, $CCTL, and $ENVIRONMENT).

actual_num_parms : 2 bytes

Integer in which the BRWINFO intrinsic returns the actual number of parameters found in the job template header or report execution.

parm_name : 20 bytes;

Name of the parameter.

parm_type : 2 bytes;

Integer that indicates the type of the parameter; that is, either Single-value, List- of-values, or Comparison Predicate.

```
0 = Single Value
1 = List of Values
3 = Comparison Predicate
```

parm_result_type : 2 bytes;

Integer that indicates the type of the parameter result; that is, either an HP ALLBASE/BRW Standard Parameter, or user-defined parameters of type String, Date, Time, Integer, Numeric Fixed, or Real.

```
0 = HP ALLBASE/BRW Standard Parameter
1 = String
2 = Date
3 = Time
4 = Integer
5 = Numeric Fixed
6 = Real
```

result_length : 2 bytes

If the result is of type String, this is an integer showing the length of the string.

```
parm_mode : 2 bytes;
```

Integer that indicates whether the parameter is Required or Frozen.

```
0 = Parameter is neither Required nor Frozen
1 = The parameter value is Frozen (cannot be
    altered)
2 = A parameter value is Required
```

upshift : 2 bytes;

Integer that indicates whether any letters in the parameter should be shifted to upper-case.

```
0 = characters are not shifted to upper-case
1 = characters are shifted to upper-case
```

parm_value : 55 bytes;

Byte array containing the value of the parameter.

reserve : 1 byte;

Not used.

Note

Note 1

If the parameter is an HP ALLBASE/BRW Standard Parameter of type $USER, $GROUP, $ACCT, $USERPASS, $GROUPPASS, or $ACCTPASS, parm_name will be set to the standard parameter name (for example, $USER). The parm_value of the HP ALLBASE/BRW Standard Parameters $USER, $GROUP, or $ACCT will be set to the log-on value. The parm_value of an HP ALLBASE/BRW Standard Parameter password ($USERPASS, $GROUPPASS, or $ACCTPASS) will be set to spaces.

Note 2

If you use BRWINFO to read parameter information from a report job file, the items $FORMAL- PRINTFILE, $DEVICE, $OUTPRI, $CCTL, $ENVIRONMENT, and $COPIES will all be read from the value in the $PRINTFILE parameter.

BRWSTREAM

Use this intrinsic to stream a batch report job. The intrinsic will pass a value for each parameter in BRW_PARAMETERS to the appropriate parameter name in the job. If the job is streamed successfully, the job number is returned in error_parm. If not, the appropriate error number is returned (see Error Handling).

Call BRWSTREAM like this:

```
BRWSTREAM (brw_comarea, brw_parameters)
```

where the call arguments have the same characteristics as in the intrinsic BRWOPEN.

BRW_PARAMETERS contains the parameters to be passed to the job template body when the batch job is created. The BRW_PARAMETERS array is read for each parameter that must be passed to the job template body. An error is returned if a needed parameter is not found in this array.

Note

The item actual_num_parms must be set to the actual number of parameters contained in the parameter array. This will be automatically set in your call previous to BRWSTREAM was BRWINFO, with the same BRW_COMAREA. Otherwise you must set the actual_num_parms yourself.

BRWEXEC

Use the BRWEXEC intrinsic to execute a report on- line. You cannot use this intrinsic with report job files.

Call BRWEXEC like this:

```
BRWEXEC (brw_comarea, brw_parameters)
```

where the call arguments have the same characteristics as in BRWOPEN. BRW_PARAMETERS contains the parameters to be passed to the son process which runs the program BRWEXEC.PUB.SYS.

The only HP ALLBASE/BRW Standard Parameters used are:

$REPORT
$ACTUAL-PRINTFILE
$FORMAL-PRINTFILE
$DEVICE
$OUTPRI
$NBR-COPIES
$ENVIRONMENT
$CCTL

Execution statistics are not displayed when calling the BRWEXEC intrinsic.

BRWCLOSE

Use the BRWCLOSE intrinsic to close the HP ALLBASE/BRW files which are opened using the BRWOPEN intrinsic.

Call BRWCLOSE like this:

```
BRWCLOSE (brw_comarea)
```

where BRW_COMAREA has the same characteristics as in the other intrinsics.

Error Handling

If an error occurs, the intrinsics return a non-zero value in the first 2-byte integer (the return_status parameter item) of the BRW_COMAREA or REQUEST_COMAREA parameters. The second 2-byte integer of these parameters (the error_parm parameter item) provides additional error information.

This additional information can be the file system error code (FSERR), the command interpreter error number (CIERR), the record number in the job template file (REC#), or the job control word (JCW). Possible values for parameter items return_status and error_parm are listed in the following table.

↓ Return Status		Intrinsic affected:							
Meaning		BRW-INIT	BRW-START	BRW-STOP	BRW-OPEN	BRW-INFO	BRW-STREAM	BRW-EXEC	BRW-CLOSE
0	No error The intrinsic executed correctly. The job number of the successfully streamed job is returned in BRW_COMAREA.ERROR_PARM.	X	X	X	X	X	X	X	X
1	Illegal address for BRW_PARAMS					X	X	X	
2	Illegal address for PARM-DFLTS		X						
3	Illegal address for filename				X				
4	The value of COMAREA_LEN is less than 100. You must set this field to the size, in bytes, of the REQUEST_COMAREA before you call BRWINITREQUEST.	X	X	X					
5	The blank character that is required at the end of the filename (in the 36-byte array FILENAME used in BRWOPEN) is missing.				X				
6	The value of the field BRW_PARAMETERS.MAX_NUM_PARMS is less than or equal to 0. (The value must be greater than 0).		X						
7	The value of the field BRW_PARAMETERS.MAX_NUM_PARMS is less than or equal to 0. (The value must be greater than 0).					X	X	X	
8	The array BRW_PARAMETERS.PARM_ARRAY is too small. Set the size of this array to the number of parameters using BRW-PARAMETERS.MAX_NUM_PARMS and try again.					X			
9	The file that you have tried to open is neither a report job file or report execution file.				X				
10	FOPEN failed on the file. The FSERR is returned in BRW_COMAREA.ERROR_PARM.				X				

↓ Return Status		Intrinsic affected:							
Meaning		BRW-INIT	BRW-START	BRW-STOP	BRW-OPEN	BRW-INFO	BRW-STREAM	BRW-EXEC	BRW-CLOSE
11	If you requested BRWOPEN to open a report execution file, it will also try to open a standard job file. It looks for BRWJOB in this order: BRWJOB; BRWJOB.PUB; BRWJOB.PUB.SYS; BRWJxxx.PUB.SYS (xxx is the Native Language Support number if the JCW NLDATALANG is set); BRWJ000.PUB.SYS. The FSERR is returned in BRW_COMAREA.ERR_PARM.				X				
12	FCLOSE failed to close the file. The FSERR is returned in BRW_COMAREA.ERROR_PARM.				X				
13	FCLOSE failed to close the standard job file. (In the order given for 11 above.)								X
14	An error occurred when reading the report job file. The FSERR is returned in BRW_COMAREA.ERROR.PARM.					X			
15	An error occurred when reading the standard job file. The FSERR is returned in BRW_COMAREA.ERROR_PARM.					X			
15	The intrinsic FCONTROL that rewinds the standard job file failed. The FSERR is returned in BRW_COMAREA.ERROR_PARM.					X			
17	An error occurred when reading the Job Template Header of the standard job file or report job file. (For example, an unknown word was found in the header).					X			
18	The intrinsic FCONTROL that rewinds the report job file failed. The FSERR in returned in BRW_COMAREA.ERROR_PARM.		X			X	X		
19	A syntax error was found in a parameter definition in the Job Template Header. The record number of the record where the error was found is returned in BRW_COMAREA.ERROR_PARM.					X	X		
20	An unexpected end-of-file marker was found when reading the report job file of standard job file. The FSERR is returned in BRW_COMAREA.ERROR_PARM.					X			
21	You have called one of the intrinsics BRWSTREAM, BRWEXEC, or BRWINFO without previously opening the BRW files by calling BRWOPEN.					X	X	X	

↓ Return Status		Intrinsic affected:							
	Meaning	BRW-INIT	BRW-START	BRW-STOP	BRW-OPEN	BRW-INFO	BRW-STREAM	BRW-EXEC	BRW-CLOSE
22	An error occurred when creating the temporary file TRJOB0.						X		
23	An error occurred when closing the temporary file TRJOB0.						X		
25	An error occurred when reading the temporary file TRJOB0. The FSERR is returned in BRW_COMAREA.error_parm.						X		
26	A syntax error was found when substituting parameter values for parameters. The record number of the record where the error was found is returned in BRW_COMAREA.error_parm.						X		
27	A parameter was found in the Job body that is neither defined in the Job template header nor in a Standard parameter. The record number of the record where the error was found is returned in BRW_COMAREA.error_parm.						X		
28	An error occurred when the son process creates its temporary files. The FSERR is returned in BRW_COMAREA.error_parm.		X				X		
29	An error occurred when the son process opens its temporary files. The FSERR is returned in BRW_COMAREA.error_parm.						X	X	
30	An error occurred when the son process reads its temporary files. The FSERR is returned in BRW_COMAREA.error_parm.						X		
31	An error occurred when the son process writes to its temporary files. The FSERR is returned in BRW_COMAREA.error_parm.		X					X	
32	The intrinsic tried to read the job number of the streamed job from the temporary file BRWTEMP but failed to find it.						X		
33	An error occurred when calling the intrinsic FINDJCW. The status from FINDJCW is returned in BRW_COMAREA.error_parm.						X		

↓ Return Status		Intrinsic affected:							
Meaning		BRW-INIT	BRW-START	BRW-STOP	BRW-OPEN	BRW-INFO	BRW-STREAM	BRW-EXEC	BRW-CLOSE
34	An error occurred when calling the MPE command STREAM via the COMMAND intrinsic. The CIERR is returned in BRW_COMAREA.ERROR_PARM.						X		
35	An error occurred when calling the intrinsic CREATE PROCESS. The error message number from CREATEPROCESS is returned in BRW_COMAREA.ERROR_PARM.		X				X	X	
36	BRWSTARTREQUEST tries to activate the son process BRWSTART, but BRWSTART does not exit. That is, BRWSTART ought to exist but does not!		X						
37	The son process BRWSTART terminated abnormally.		X						
38	The BRWEXEC program encountered an error when executing.							X	
39	An error occurred when calling the intrinsic KILL to kill the son process BRWSTART.			X					
40	An error occurred when calling the intrinsic NLGETLLANG. The status from NLGETLANG is returned in BRW_COMAREA.error_parm.				X				
41	An error occurred when calling the intrinsic COMMAND to call the MPE commands FILE and RESET. The error code from the COMMAND intrinsic is returned in BRW_COMAREA.error_parm.							X	
42	Execution file read failed. Error returned because of a damaged or faulty report execution file. Build a new execution file and start again.					X			
43	Execution file version error. Error returned because of a damaged or faulty report execution file. Build a new execution file and start again.					X			
44	Execution file wrong type. Error returned because of a damaged or faulty report execution file. Build a new execution file and start again.					X			
45	Wrong type of report. Error returned because of a damaged or faulty report execution file. Build a new execution file and start again.					X			
46	Section number error. Error returned because of a damaged or faulty report execution file. Build a new execution file and start again.					X			
47	Default parameter error. Error returned because of a damaged or faulty report execution file. Build a new execution file and start again.					X			
48	Selection set error. Error returned because of a damaged or faulty report execution file. Build a new execution file and start again.					X			

21

HP ALLBASE/BRW and HP BRW/V

Comparing HP ALLBASE/BRW to HP BRW/V

At a Glance

HP ALLBASE/BRW is the version of Business Report Writer that creates reports that run in native mode on an HP 3000 series 9*xx*, the series of computers that use the MPE XL operating system.

HP BRW/V is the version of Business Report Writer used on any HP 3000 running MPE V, that is the HP 3000 series 3*x*, 4*x*, 5*x*, 6*x*, or 7*x*.

This chapter describes the relationships between HP ALLBASE/BRW and HP BRW/V. You need only read this chapter if you satisfy one or more of the following conditions:

- You have upgraded from an HP 3000 series 3*x*, 4*x*, 5*x*, 6*x*, or 7*x* to an HP 3000 series 9*xx*, and wish to run old reports on your new computer.
- You want to use the HP 3000 series 9*xx* computer to develop reports that will later be used on an HP 3000 series 3*x*, 4*x*, 5*x*, 6*x*, or 7*x* computer.

Note

For the purposes of this chapter, HP 3000 series 3x, 4x, 5x, 6x, or 7x computers are denoted by their operating system name, MPE V, and HP 3000 series 9xx computers are denoted by their operating system name, MPE XL. Reports written on MPE V computers will be denoted *HP BRW/V* reports, and reports written on MPE XL computers will be denoted HP ALLBASE/BRW.

To summarize, you can run any HP BRW/V report on HP ALLBASE/BRW, and you can compile some HP ALLBASE/BRW reports so that they will run on any HP 3000 running MPE V V-Delta-1 MIT (version G.03.01) or later.

An HP ALLBASE/BRW report that accesses an HP ALLBASE/SQL database cannot be compiled to run on an MPE V system. An HP ALLBASE/BRW report that uses names greater than 16 characters requires modifications before it can be compiled to run on MPE V systems. This is important if you develop reports on HP ALLBASE/BRW and run them on HP BRW/V.

This chapter covers

- comparing HP ALLBASE/BRW and HP BRW/V
- transferring reports between MPE V and MPE XL
- MPE V to MPE XL in Native Mode
- MPE V to MPE XL in Compatibility Mode
- MPE XL to MPE V

Comparing HP ALLBASE/BRW and HP BRW/V

There are three notable differences between HP BRW/V and HP ALLBASE/BRW:

- HP ALLBASE/BRW allows for 20 characters in a name field (that is, table, item), compared with 16 in HP BRW/V.
- HP ALLBASE/BRW supports access to HP ALLBASE/SQL tables, whereas HP BRW/V does not.
- HP ALLBASE/BRW supports two Real number types, HP 3000 Reals and IEEE Reals, whereas HP BRW/V supports only HP 3000 Reals

You can tell whether you are running HP ALLBASE/BRW or HP BRW/V from the left-hand corner of the banner line (the top line of each screen).

File names in HP ALLBASE/BRW and HP BRW/V

To allow the operation of both *HP BRW/V* and *HP ALLBASE/BRW* on MPE XL systems, the names of all Business Report Writer program and data files for *HP BRW/V* are different from the names of the files for HP ALLBASE/BRW, (except for the program REMPTY.PUB.SYS). Report specification files, dictionary output files, intermediate report files, and print files are interchangeable between *HP BRW/V* and *HP ALLBASE/BRW*. The list on the following page shows the equivalent files and their function; the files are supplied with HP ALLBASE/BRW and are held in PUB.SYS.

File codes in HP ALLBASE/BRW and HP BRW/V

The file codes and file mnemonics for HP ALLBASE/BRW files are the same as those for HP BRW/V files, with the following two exceptions.

- native mode execution files in HP BRW/V have a file code of 1438 and a file mnemonic of REXMN
- the BRWDIC file in HP BRW/V has a file code of 1437 and a file mnemonic of RDICN

The following table shows the HP ALLBASE/BRW files names compared to the HP BRW/V files:

HP BRW/V (Compatibility Mode)	HP BRW/XL (Native Mode)	Description of File
BRW	BRWXL	Report Specification Program
RAPPDIC	BRWAPPD	Application Dictionary Interface
RCOMP	BRWCOMP	Report Compile Program
RCONVERT	BRWCONV	REPORT/V and INFORM/V Conversion Program
RDIC3000	BRWD3000	Dictionary/V Interface Program
RDICLIST	BRWDLIST	RDIC and BRWDIC List Program
RLIST	BRWLIST	Report Specification List Program
RSDEXT	BRWSDEXT	Command File to add SYSDIC Extensions
RSYSDIC	BRWSD	System Dictionary Interface Program
RCONF000	BRWC000	Standard Configuration File
REMPTY	REMPTY	Program to empty Intermediate Report Files
REXEC	BRWEXEC	Report Execution Program
RJOB000	BRWJ000	Standard Job File
RSETUP	BRWSETUP	BRW Configuration Program
RSORT	-	No longer needed in MPE XL
RSTART	BRWSTART	Report Request Program
RSTREAM	BRWSTRM	(internal program only)
RLIT000	BRWL000	BRW Literals File
RDLIT000	BRWDL000	Dictionary Interface Literals File
RFORM000	BRWF000	BRW Forms File
RMSG000	BRWM000	BRW Message File

The above files are supplied with HP BRW/XL and reside in PUB.SYS. However, you can define dictionary extract files, customized configuration files and customized standard job files. These files can be held in any group on your system.

The following list shows such files and their functions.

HP BRW/V (Compatibility Mode)	HP BRW/XL (Native Mode)	Description of File
RDIC	BRWDIC	Dictionary Extract File
RCONF	BRWCONF	Customized Configuration File
RJOB	BRWJOB	Customized Standard Job File

Modes

HP ALLBASE/BRW runs in native mode or in compatibility mode. Native mode is simply MPE XL. Any report created using HP ALLBASE/BRW on an MPE XL computer will automatically be in native mode.

Compatibility mode emulates MPE V on MPE XL. Therefore compatibilty mode allows you to run existing report execution files without recompilation. Of course, in compatibility mode you do not get the full benefits of MPE XL.

You cannot run an existing HP BRW/V report execution file in native mode; however, you can easily create a new HP ALLBASE/BRW report execution file by compiling the HP BRW/V report specification file on your MPE XL computer. Then, when you run the HP ALLBASE/BRW report execution file, you will get the same report with the full performance benefits of MPE XL.

The "Running HP BRW/V Reports on HP ALLBASE/BRW" section describes how HP ALLBASE/BRW chooses the correct mode for running HP ALLBASE/BRW or HP BRW/V reports.

Transferring Reports between MPE V and MPE XL

You can transfer reports from HP BRW/V to HP ALLBASE/BRW. However, you must recompile an HP BRW/V report with HP ALLBASE/BRW if you want to obtain the full performance benefits of an MPE XL system. It is also possible to transfer reports from HP ALLBASE/BRW to HP BRW/V provided the report does not reference an HP ALLBASE/SQL table and does not contain names longer than 16 characters.

MPE V to MPE XL (to run in Native Mode)

In order to run an existing MPE V report in native mode on MPE XL, you must create a new HP ALLBASE/BRW report execution file by compiling the HP BRW/V report specification file on the MPE XL.

To transfer from MPE V for native mode on MPE XL

1. Store the following from the MPE V system and restore them on the MPE XL system:
   a. the data files
   b. the data dictionary
   c. the report specification files
   d. any report job files
   e. any *amended* configuration files and standard job files
2. Run the appropriate BRWDIC generating program (BRWD3000.PUB.SYS, BRWAPPD.PUB.SYS, BRWSD.PUB.SYS, BRWACCSD.PUB.SYS or BRWGEND.PUB.SYS) to create a BRWDIC file from the restored MPE V data dictionary.

   If you have MPE or KSAM files that contain real numbers, you must tell HP ALLBASE/BRW that these numbers are in HP 3000 format by adding the longname !REAL-HP3000 (for *HP Dictionary/V*) or the element-subtype HP 3000 (for *HP System Dictionary*) to the item's definition in the data dictionary before creating the BRWDIC file. See chapter 18, "Preparing a Data Dictionary."

Note 

You can remove the data dictionaries as soon as you have created the BRWDIC file.

3. Edit the report job files so that the program filenames refer to HP ALLBASE/BRW programs.
4. Edit the amended standard job files so that any program filenames refer to HP ALLBASE/BRW programs.

5. Rename the following files:
   a. standard job files from RJOB to BRWJOB
   b. the configuration files from RCONF to BRWCONF
6. Using the programs BRWXL.PUB.SYS or BRWCOMP.PUB.SYS, compile the report specification files to create HP ALLBASE/BRW report execution files

The reports are now ready to run in MPE XL native mode.

MPE V to MPE XL (to run in Compatibility Mode)

To tranfer from MPE V to MPE XL for Compatibility Mode

1. Store the data from the MPE V system and restore it on the MPE XL system.
2. Store the report execution files from the MPE V system and restore them on the MPE XL system.
3. Store any report job files from the MPE V system and restore them on the MPE XL system.
4. Store any amended configuration files and standard job files from the MPE V system and restore them on the MPE XL system.

The reports are now ready to run in MPE XL compatibility mode.

Note

The above instructions assume that you only want to run the HP BRW/V reports in compatibility mode on MPE XL. If you want to modify or compile the reports in compatibility mode on your MPE XL system, you must also transfer the report specification files and RDIC file from the MPE V system to the MPE XL system.

MPE XL to MPE V

To use an HP ALLBASE/BRW report from MPE XL on an MPE V system, you should first verify that the report will be compatible on MPE V.

If you do development on the MPE XL system, and only intend to execute the reports on MPE V, you don't need to copy the data dictionaries or specification files. Just create RDIC on the MPE XL with the compatibility mode RDIC generating program, then use BRW.PUB.SYS or RCOMP.PUB.SYS on MPE XL to make the report execution file.

You don't have to have copies of the databases on both machines.

To verify compatibility for a report

1. Go to the HP ALLBASE/BRW Select Report screen or the Global Values screen of BRWSETUP.
2. Choose (Verify) to check whether a report accesses an SQL database or if it contains names with a length greater than 16 characters.

Reports that access an SQL database cannot be used on MPE V. If a report contains names with a length greater than 16 characters, they must be changed to 16 characters or less.

If no SQL files are accessed and no names are greater than 16 characters in length, you get a messages that the report has been verified. You can transfer the report to MPE V.

If conflicts are detected, warning messages appear on the screen and are also written to a temporary file called RVERIFYM. These are the warning messages:

```
SQL accessed
Name > 16 chars
```

You can print the warning messages by pressing (Print).

Note

In addition to the compatibility checks done by using `Verify`, you need to be aware that HP BRW/V does not support IEEE-type real numbers.

To transfer a compatible report from MPE XL to MPE V

1. When there are no conflicts in the report, store the following from the MPE XL system and restore it on the MPE V system:
   a. the data files
   b. any report job files
   c. any amended configuration files and standard job files
2. Edit these file so that the program filenames are those of HP BRW/V:
   a. report job file
   b. amended standard job files
3. Rename these files:
   a. the standard job files from BRWJOB to RJOB
   b. the configuration files from BRWCONF to RCONF
4. Run the appropriate RDIC generating program (RDIC3000.PUB.SYS, RAPPDIC.PUB.SYS, RSYSDIC.PUB.SYS, RACCSDIC.PUB.SYS, or RGENDIC.PUB.SYS) to create an RDIC file from the data dictionary.

   You can remove the data dictionaries as soon as you have created the RDIC file.

5. Using BRW.PUB.SYS or RCOMP.PUB.SYS on HP ALLBASE/BRW, compile the report specification files to create HP BRW/V report execution files.
6. Store the HP BRW/V report execution files from the MPE XL system and restore them on the MPE V system.

The reports are now ready to run on the MPE V system.

Note

The MIT version of the MPE V operating system where you restore the execution file must be V-Delta-1 MIT (G.03.01) or later.

Running HP BRW/V Reports on MPE XL

You can run HP BRW/V reports on your MPE XL system in either compatibility mode or native mode. The choice depends on the program you execute.

To run HP ALLBASE/BRW in native mode on MPE XL

1. Run the execution program for HP ALLBASE/BRW:

   ```
   RUN BRWEXEC.PUB.SYS
   ```

   or BRWSTART.PUB.SYS, or BRWXL.PUB.SYS

2. At the `Execution File:` prompt, type the name of an HP ALLBASE/BRW report execution file:

An error message appears if you try to use a HP BRW/V report execution file with BRWEXEC.PUB.SYS.

To run HP BRW/V in compatibility mode on MPE XL

1. Run the execution program from HP BRW/V:

   ```
   RUN REXEC.PUB.SYS
   ```

   or, RSTART.PUB.SYS, or BRW.PUB.SYS

2. Supply the name of an HP BRW/V report execution file at the 'Execution File:' prompt.

An error message appears if you try to use a HP ALLBASE/BRW report execution file with REXEC.PUB.SYS.

Error Messages

The following error messages may be displayed when you are operating in the BRW/V environment. The error messages are listed in alphabetical order and are followed by a brief explanation.

Note

These errors can be detected before entering *BRW/V* by using the Verify function key from the *HP ALLBASE/BRW* Select Report screen.

Configuration file is incomplete

This error message may occur when running the program RSTART, RCOMP, REXEC, or RSETUP. This message could also occur when using the Compile Report screen or the Report Request screen. It indicates that the configuration file has not been renamed.

To solve this problem, rename BRWCONF to RCONF.

File contains names with more than 16 characters

This error message may occur from the Select Report screen. It indicates that the report specification contains one or more names longer than 16 characters.

To solve this problem, use the Verify function key in HP ALLBASE/BRW to obtain a list of names that are too long. Shorten them to 16 or less than 16 characters.

File contains names with more than 16 characters

***** ERROR: Referenced access file *filename* is incompatible.**

This error message may occur when you are attempting to compile a report by running the RCOMP program or using the Compile Report screen. It indicates that the report uses the access definitions of another report which contains names longer that 16 characters.

To solve this problem, either (1) remove the name of the referenced file and define your own access, (2) reference a different file, or (3) use HP ALLBASE/BRW and shorten the names in the referenced access file to 16 or less than 16 characters.

File contains names with more than 16 characters

***** ERROR: Specification file *filename* is incompatible.**

This error message may occur when attempting to compile a report by running the RCOMP program or using the Compile Report screen. It indicates that the specification file references names longer than 16 characters.

To solve this problem, use the Verify function key in HP ALLBASE/BRW to get a list of the names that are too long. Shorten the names to 16 or less that 16 characters.

File contains SQL ACCESS definitions

This error message may occur from the Select Report screen. It indicates that the report references an HP ALLBASE/SQL table. It is not possible to access an HP ALLBASE/SQL table using HP BRW/V.

To solve this problem, use HP ALLBASE/BRW to remove the HP ALLBASE/SQL table access definitions.

File contains SQL ACCESS definitions

***** ERROR: Referenced access file *filename* is incompatible**

This error message may occur when you are attempting to compile a report by running the RCOMP program or using the Compile Report screen. It indicates that the report points to an access definition file which references an HP ALLBASE/SQL table. It is not possible to access an HP ALLBASE/SQL table using *BRW/V*.

To solve this problem, either (1) remove the name of the referenced file and define your own access, (2) reference a different file, or (3) use HP ALLBASE/BRW and remove the HP ALLBASE/SQL table access definition in the referenced access file.

File contains SQL ACCESS definitions

***** ERROR: Specification file *filename* is incompatible**

This error message may occur when you are attempting to compile a report by running the RCOMP program or using the Compile report screen. It indicates that the specification file points to an HP ALLBASE/SQL table. It is not possible to access an HP ALLBASE/SQL table using *HP BRW/V*.

To solve this problem, use HP ALLBASE/BRW to remove the HP ALLBASE/SQL table access definitions.

Referenced access file contains names with more than 16 characters

This error message may occur from the Modify Report screen. It indicates that the report uses the access definitions of another report that contains names longer than 16 characters.

To solve this problem either, (1) remove the name of the referenced file and define your own access, (2) refer to a different file, or (3) use HP ALLBASE/BRW and shorten the names in the referenced access file to 16 or less than 16 characters.

Referenced access file contains SQL ACCESS definitions

This error message may occur from the Modify Report screen. It indicates that the report points to an access defintion file which references an HP ALLBASE/SQL table. It is not possible to access an HP ALLBASE/SQL table using *BRW/V.*

To solve this problem, either (1) remove the name of the referenced access HP ALLBASE/SQL table and define your own access, (2) reference a different file, or (3) use HP ALLBASE/BRW to remove the HP ALLBASE/SQL table access definitions in the referenced access file.

Transfer Reference

Data

MPE files, KSAM files, IMAGE databases, intermediate report files, and dictionary output files can be used without amendment in either HP ALLBASE/BRW or HP BRW/V. Data from HP ALLBASE/SQL tables cannot be accessed in HP BRW/V.

Names In HP ALLBASE/BRW names can be up to 20 characters in length, but in HP BRW/V they are restricted to 16 characters.

Configuration Files

You can transfer configuration files from HP BRW/V to HP ALLBASE/BRW as long as you rename the file BRWCONF. For example, if you defined a special configuration file for group DATAIN in account ADMIN, the file will be called RCONF.DATAIN.ADMIN. You can restore this file onto your MPE XL system, but you must rename it BRWCONF before it will be used by HP ALLBASE/BRW.

To transfer configuration files from HP ALLBASE/BRWto HP BRW/V

1. Press the Verify function key on the Global Values screen to make sure no global function to be referenced in an HP BRW/V report contains names longer than 16 characters.
2. Rename the file from BRWCONF to RCONF.

HP ALLBASE/BRW Intrinsics

HP ALLBASE/BRW intrinsics have the same characteristics as HP BRW/V intrinsics except that:

- The BRW_COMAREA parameter used by the HP ALLBASE/BRW execution intrinsics (BRWOPEN, BRWINFO, BRWSTREAM, BRWEXEC, and BRWCLOSE) requires 332 bytes.

  (The BRW_COMAREA parameter used with the equivalent HP BRW/V intrinsics requires at least 300 bytes, but it is not a problem to use the larger number of bytes for both. That way you can have one program that works for both systems).

- The element comarea_len of the parameter REQUEST_COMAREA (used by the *HP ALLBASE/BRW* request report intrinsics BRWINITREQUEST, BRWSTARTREQUEST, and BRWSTOPREQUEST) must be set to 100. (The element comarea_len of the parameter REQUEST_COMAREA used with the equivalent HP BRW/V intrinsics must be at least 50).

So if you transfer programs that use HP BRW/V intrinsics to MPE XL, consult Appendix D, HP ALLBASE/BRW Intrinsics, and alter the parameter length(s).

Report Job Files

You must alter any appropriate program names when transferring report job files between HP BRW/V and HP ALLBASE/BRW. (See the list of different program file names near the beginning of this chapter.) For example, a report job file for HP BRW/V might contain:

```
.
.
.
$PRINTFILE
!CONTINUE
!RUN REXEC.PUB.SYS
salesan.rexec
sales-area = "WESTERN"
!EOD
.
.
```

Before using this report job file with HP ALLBASE/BRW, you must amend the file as below:

```
.
.
.
$PRINTFILE
!CONTINUE
!RUN BRWEXEC.PUB.SYS
salesan.rexec
sales-area = "WESTERN"
!EOD
.
```

Note that the REXEC program has been changed to BRWEXEC. You must also make sure that the execution file salesan.rspec was compiled under HP ALLBASE/BRW.

Standard Job Files

You can transfer a standard job file from HP BRW/V to HP ALLBASE/BRW and vice versa so long as you rename the files and alter any program names within it. For example, if you defined a special HP BRW/V standard job file for group DATAIN in account ADMIN, the file will be called RJOB.DATAIN.ADMIN. You can restore this file onto your MPE XL system, but you must rename it BRWJOB before it will be used by HP ALLBASE/BRW. You must also alter any appropriate program names.

22

Converting Reports

Use BRWCONV to upgrade Report/V or Inform/V reports

At a Glance

The HP ALLBASE/BRW program BRWCONV upgrades Report/V or Inform/V reports to HP ALLBASE/BRW reports by creating an HP ALLBASE/BRW specification file from the source file.

This chapter describes how to use the BRWCONV program.

This chapter covers

- required software
- running BRWCONV
- incompatibilities
- translation rules
- example reports

Required Software

To run BRWCONV, you must have the following software installed on your system in addition to HP ALLBASE/BRW:

REPCOMP.PUB.SYS	The Report/V compiler. It is used by BRWCONV to verify Report/V reports before the conversion.
INFCNV.PUB.SYS	Required to convert Inform/V reports.

Location of BRWDIC, REPCOMP, and INFCNV

BRWCONV looks for a BRWDIC file in BRWDIC. PUB.*logon-account*. The two files REPCOMP.PUB.SYS and INFCNV.PUB.SYS require a *HP Dictionary/V* data dictionary in the PUB group of the logon-account (DICT.PUB.*logon-account*). You can always, of course, use file equations if you want to hold the dictionary or BRWDIC file elsewhere.

Using BRWCONV

This section describes the steps you must follow to run the BRWCONV program. In addition to this description, the steps are illustrated in the conversion example at the end of this chapter.

To convert reports

1. Define files and elements in the data dictionary.
2. Verify that item names correspond to HP ALLBASE/BRW syntax.
3. Identify DATE and TIME items.
4. Create the BRWDIC file.
5. Create group sets if desired.
6. Run BRWCONV.

The following sections describe each step in detail.

Define Files, Items, and Elements

Make sure that all items used in the reports you are converting are defined in the data dictionary. Report/V allows elements to be defined within a report. *HP ALLBASE/BRW* always looks in BRWDIC for item definitions so all elements, including child elements, used in a report must be defined in the dictionary. (Report/V uses only *HP Dictionary/3000*).

There are two ways to tell which items and files you must define in the dictionary:

- Look through the reports you want to convert for DEFINE statements. Any *count, type, size, dec, storage, parent-name,* or*position* parameters contain element information that you must define in the dictionary. ACCESS statements contain information about the files that must be defined.

  Or ...

- you can compile the Report/V reports with the DEFN option to get a list of all items defined in each report. Use the DICT option to check the dictionary for definitions. Any elements that are not defined in either the report *or* the dictionary and any files accessed that are not defined in the dictionary will cause compilation errors.

Verify Item Names

Verify that all item names correspond to the *HP ALLBASE/BRW* name syntax. Item names in *HP ALLBASE/BRW* can be up to 20 characters long. In addition they must start with a $, @, # or letter. Item names cannot contain points, commas, round parentheses or spaces. You can use the dictionary alias name to give an item a legal *HP ALLBASE/BRW* name. (There are also a few restrictions, detailed under the section IMCOMPATIBLE FEATURES, on the Report/V item names that *HP ALLBASE/BRW* can recognize.)

Define Date and Time Items

Use the dictionary long-name to identify Date and Time items. In Report/V, dates and times are ordinary String items. *HP ALLBASE/BRW* allows greater flexibiliy for these items. To allow the *HP ALLBASE/BRW* options, modify the long-names as described in Chapter 18 of this manual.

Create BRWDIC

Run the appropriate dictionary interface program to create a BRWDIC file. See Chapter 18 of this manual.

If your dictionary is in another account and you want to create a BRWDIC in the account where the data is located, you can use file equations:

```
:file DICT.PUB = DICT.PUB.OTHER
:run BRWD3000.PUB.SYS
```

Create a Group for Sets of HP ALLBASE/BRW Reports

The BRWCONV program can convert sets of Report/V or Inform/V files. If you want to convert a set of files, you must create a group to hold the new *HP ALLBASE/BRW* specification files before you run the BRWCONV program.

Run BRWCONV

Follow the steps below to convert the reports.

Note

If you want to leave the BRWCONV program at any time, do not type a response to the prompt. Press Return.

To convert a report

1. Run the program BRWCONV.

   ```
   :RUN BRWCONV.PUB.SYS
   ```

   BRWCONV prompts you for the type of reports you want to convert.

   ```
   Report/V or Inform/V to BRW Conversion Program
   Enter ? for Help.
   Convert from Report/V or from Inform/V (R/I) ?
   ```

2. Type R for Report/V reports or I for Inform/V reports.

   If you enter a report type, BRWCONV prompts you for the name of the report you want to convert.

3. Type the report name.

   ```
   Report/V Source File(s) = @.REPSCRC
   ```

   You can use wildcards to specify sets of files. (If you are converting sets of files, you must create a group to contain the HP ALLBASE/BRW reports before you run BRWCONV.)

   When converting Inform/V files, enter the file name or fileset name only, that is, do not include the II that prefixes all INFORM

filenames. For example, if you want to convert the INFORM file IIREP01, just enter REP01 when BRWCONV prompts for the filename.

BRWCONV prompts you for the names of the HP ALLBASE/BRW specification file and execution file (or the groups, if you are converting a set of reports).

4. Type the execution file or specification file name and press Return.

```
Group for BRW Specification Files = BRWSPEC
Group for BRW Execution Files = BRWEXEC
```

If you specify an execution file name or group, the converted report(s) are compiled into the specified file(s).

When you convert sets of reports, the HP ALLBASE/BRW files get the same name as the Report/V files, but are located in the appropriate group.

Verification of Report/V Reports

Before it converts a report, BRWCONV calls the Report/V compiler to make sure the report is valid. If the compilation fails, an error message is issued and the report is not converted.

If the compilation succeeds, BRWCONV issues the message "verified" and proceeds with the conversion. If a file with the name of the HP ALLBASE/BRW specification file already exists, BRWCONV will issue the following prompt after the 'verified' message.

```
verified
BRW specification file SPECLIST.RSPEC exists.  Replace it (N/Y)?
```

If you enter N (or simply press Return), BRWCONV aborts the conversion of the report (conversion of other reports in a set of reports will not be affected).

If you enter Y, the old file is purged, and a new one created if the report conversion is successful.

You can avoid getting this prompt if you run BRWCONV with PARM=1 (see next page). Parm=1 causes all files with names identical to the HP ALLBASE/BRW specification file to be purged without getting this prompt.

File Lockwords/Passwords

During the conversion, BRWCONV prompts you for database passwords and file lockwords. If you specify passwords during report conversion, they are included in the HP ALLBASE/BRW specifications.

To convert reports with file passwords

1. Follow the same steps as when converting reports without file passwords.

   After the verification of the report, BRWCONV prompts you for the passwords or lockwords.

2. Respond to the prompts with the appropriate lockword or password.

   Here is an example of the prompts for file lockwords and database passwords:

```
CUSTLIST.REPSRC
  verified
     Password for database TOYDB =
  converted
  compiled
CUSTSTAT.REPSRC
  verified
     Password for database TOYDB =
  converted
  compiled
```

The passwords do not appear on the screen when you type them.

Note

When you convert sets of reports that use the same database or file, you can enter the password for the first prompt, then press Return to subsequent prompts that require the the same password.

To define passwords at execution time

1. Run BRWCONV with PARM=1 as shown here:

```
:RUN BRWCONV.PUB.SYS;PARM=1
```

The PARM=1 option suppresses the password prompt and the purge files prompt. This makes it easy for you to run BRWCONV as a batch job to convert all of your reports.

Error Messages (RWARN)

Any exceptions that occur during the conversion, such as items which are not defined in the dictionary or features which do not convert, are written to a file called RWARN. You can use this file to determine whether you must make any manual modifications to the HP ALLBASE/BRW specifications. RWARN has a maximum size of 10000 records. In the unlikely event of 10000 errors, the RWARN file will fill up, and any other warnings found will be lost (unless you enlarge the file).

BRWCONV does *not* change the Report/V reports. It creates new files for the HP ALLBASE/BRW specifications. Therefore, if BRWCONV warns you that a feature was not translated correctly, you can either make the necessary correction to the Report/V report and run BRWCONV again, or you can modify the new HP ALLBASE/BRW report.

Inform/V Reports

You can use BRWCONV to convert Inform/V reports if you have the program INFCNV.PUB.SYS on your system. BRWCONV translates all Inform/V features, so no manual modifications are necessary.

On-line Help Facility

BRWCONV includes an on-line help facility that you can access by entering a question mark (?) as the response to any prompt. (Except, of course, prompts for passwords!)

Incompatible Features

BRWCONV does not translate all Report/V features. Some are not available or not necessary in HP ALLBASE/BRW and some must be modified manually after you convert the reports. This section lists those features which do *not* convert automatically and tells you which ones require manual modification.

BRWCONV can translate all Inform/V features.

The following table shows the incompatible features between Report/V and HP ALLBASE/BRW.

Names	BRWCONV assumes that all item names in a Report/V report obey the following rules: ■ The item name begins with a $, @, #, or letter. ■ The item name is a maximum of 20 alphanumeric characters. ■ The item name does NOT contain a decimal point (.) or colon (:). ■ The item name is not one of the following: MAXI MUM, MINIMUM, TOTAL, AVERAGE, COUNT, VARIANCE, RANGE, STDERR, or STDDEV. If the Report/V item name contains a point (.) or colon (:), you cannot use the data dictionary alias name. You will have to alter the name in the Report/V source file and also in the data dictionary before you can run BRWCONV.
Functions	The functions STDERR, VARIANCE, RANGE, and STDDEV are ignored.
CCTL !LIST !PAGE !LIST LINE=0 NOCRLF DISPLAY !NOLIST OPTION NODUPLICATE	Not available in *HP BRW/XL*. These commands are ignored.
$CPU	HP ALLBASE/BRW produces detailed run-time statistics including CPU time. This option is ignored.
REPORT	Does not apply in HP ALLBASE/BRW.
ACCESS	There can only be a maximum of 14 files or datasets in a report. Two or more datasets cannot have the same name. For example, you cannot convert a report that uses data from a dataset called CUSTOMERS in database TOYDB, and also from a dataset called CUSTOMERS in database OTHERDB.
SORT	If a report has more than 9 sort levels, only the first 9 levels are translated. The rest are ignored.

SELECT	BRWCONV cannot convert Report/V generic constants if they contain a ^ character and, at the same time, they are to be used in relations with the operators: >, <, <=, and >=. That is, BRWCONV will convert all SELECT statements in relations that use the operators = or <>, and/or that do not include generic constants containing an ^ character. Note that you can use generic constants with the ^ character if the ^ character appears at the end of the name. Only when the ^ character appears within a name, and any of the operators >, >=, <, <= are used, can BRWCONV not con vert the generic constant. For example, assuming a relational operator >, >=, <, <= is used: AB^ will be converted AB^^ will be converted A^B will NOT be converted A^B^ will NOT be converted
JOIN	JOIN is not translated. A JOIN command is treated as a SPACE command.
NEED	By default, HP ALLBASE/BRW uses NEED for all lines in a lineset. NEED across linesets is not possible. NEED is ignored.
PAGE	If used in the middle of a lineset or used to force several blank pages, PAGE is ignored. Otherwise PAGE is translated.
ROW	If used at the beginning of a lineset, ROW is trans lated. Otherwise it is ignored.
BLOCK- BLOCKEND IF THEN ELSE LET OPTION SUPPRESS	These features are implemented differently in BRW. Manual modification is necessary after upgrade.
DEFINE(ITEM)	Items from databases or files must be defined in the dictionary for use in HP ALLBASE/BRW. Only heading texts, edit masks and calculated items are trans lated. Note that you must also define an item's alias name (if any) in the data dictionary. BRWCONV cannot take the alias name directly from the DEFINE(item) statement.
OPTION LANG	Language is part of *HP BRW/XL*'s configuration file, not a part of a report. This option is ignored.

Translation Rules

This table contains a list of Report/V commands and a description of how they are translated into HP ALLBASE/BRW specifications.

Commands	HP ALLBASE/BRW Specifications
ACCESS	If given, the access statement is used to define data access for the report. If no ACCESS statement exists, BRWCONV obtains the access specifications from the Report/V compiler. The table PRINT-TABLE is created. It has type JOIN and contains all files specified in the ACCESS statement as source tables. The table is tuned to access the source tables in the same sequence as they appear in the ACCESS statement. KEY, CHAIN, and MATCH options are used to determine table relations and the type of access to the source tables. Elements listed in the LIST option are selected for projection. The MODE option is used to determine the DBOPEN mode. The default is 5. There are a maximum of 14 files or datasets in a report. Two or more datasets cannot have the same name. For example, you cannot convert a report that uses data from a dataset called CUSTOMERS in database TOYDB, and also from a dataset called CUSTOMERS in database OTHERDB.
DEFINE(*item*)	Only heading texts, edit masks, and calculated items are translated. Items with a COMPUTE parameter become numeric table calculated items with the formula given in the arithmetic expression. You must also define an item's alias name (if any) in the data dictionary. BRWCONV cannot take the alias name directly from the DEFINE(*item*) statement.
DETAIL	Used as the detail lineset.
GROUP(n)	The item with sort level n is printed with suppressed repetition of equal values.
GROUP(n) SUMMARY	Used as the break level n footing.
GROUP(n) TITLE	Used as the break level n heading.
OPTION NOBANNER	The default page heading is not created.
OPTION NOHEAD	No column headers are created for the items.
OPTION WIDTH=n	Used as report width if no PWIDTH is specified.
OPTION DEPTH=n	Used as page length if no PDEPTH is specified.
OPTION PWIDTH=n	Used as report width.
OPTION PDEPTH=n	Used as page length.

Commands	HP ALLBASE/BRW Specifications
PAGE FOOTING	Used as page footing lineset.
PAGE HEADING	Used as page heading lineset.
REPORT SUMMARY	Used as report footing.
REPORT TITLE	Used as report heading.
SELECT	A selection set SET, which contains the selections from all select statements, is created. If an element is named without selection criteria, a Comparison Predicate parameter with the same name as the item is created. The selection term *item* `SATISFIES ?`*item* is added to the Selection Condition and connected to any other selection terms via AND or OR as specified. The parameter is added to the Selection Set.
SORT(*n*)	The report is sorted by the item specified in the SORT statement. If multiple SORT statements exist, the priority of the sort items is determined by *n*. If a report contains more than 9 sort levels, only those with the 9 highest priorities are translated.
!INCLUDE	The file to be included is expanded into the file where it is referenced.
$DATELINE	DATE and TIME are used.
$PAGE	PAGE is used.
$TIME	TIME is used.
$TODAY	DATE is used.
AVERAGE(*item-name* [,*break-num*])	The item is printed with the ColCalc AVG[*break-num*].
ALIGN	The item is positioned in the same column as in the detail lineset.
CENTER	The print field of the item is centered on the line.
COL=*number*	The print field of the item starts at position *number*.
EDIT=*edit-mask*	Item edits that correspond to *edit-mask* are created for the item.
HEAD=*char-string*	*Char-string* is used as the column header (unless OPTION NOHEAD is specified).
JOIN=*number*	JOIN=*number* is treated as SPACE=*number*. SPACE=*number* is described later in this table.
LEFT	Justify left.
LINE=*number*	The item is placed on a new line. If number is greater than one, *n*-1 blank lines are inserted first. If number is zero, LINE is ignored.

Commands	HP ALLBASE/BRW Specifications
LNG=*number*	LNG is ignored if the item has an edit-mask defined either using a lineset, or in DEFINE(*item*). If no such edit-mask is defined, the print field of the item has length *number*. If edit-masks are defined in both DEFINE(item) and lineset, the lineset edit-mask is used.
NOHEAD	Suppresses a header for the item.
NOSIGN	A layout calculated item is created. It is named NOSIGN-*n* where *n* is a number that makes the name unique. The formula is: `IF` *item* `< 0 THEN NO_VALUE ELSE` *item* The item edits for this item must specify a dash as the NO_VALUE character.
PAGE=*number*	If used with the first item or text in the lineset, it translates to Page Before the lineset. If used with blank text at the end of a lineset, it translates to Page After. If PAGE is used in the middle of a lineset, it is ignored. *Number* is ignored.
RIGHT	Right justify.
ROW=*number*	Only translated if it is used with the first item or text in a lineset. Defines absolute positioning of the lineset on line *number*.
SPACE[=*number*]	The item is positioned *number* spaces after the previous item on the line.
TRUNCATE	The length of the print field is truncated if the item does not fit on the line. The default is to position the item on a new line.
ZERO or ZEROES	Prefill with character "0" is used.

MAXIMUM(*item-name* [,*break-num*])	The item is printed with the ColCalc MAX[*break-num*].
MINIMUM(*item-name* [,*break-num*])	The item is printed with the ColCalc MIN[*break-num*].
TOTAL(*item-name* [,*break-num*])	The item is printed with the ColCalc TOTAL[*break-num*].
COUNT(*item-name* [,*break-num*])	If the item is numeric, it is printed with ColCalc COUNT[*break-num*]. Otherwise a calculated item with type Integer is created and used with COUNT instead of the original item. The formula for the calculated item is: `IF` *original-item* `= NO_VALUE THEN NO_VALUE ELSE 1` The name of the calculated item is COUNT-*n* where *n* is a number that makes the name unique.

Example Report Conversion

This section contains a step-by-step example of how to convert a Report/V report into an HP ALLBASE/BRW report. The source for the Report/V report is:

```
REPORT PRODFR;
DISPLAY "TEST REPORT FROM TOYDB";
OPTION NOHEAD,NOBANNER;

ACCESS ORDER-DETAILS (TOYDB.PUB),
           LIST= (PRODUCT-NO, QUANTITY, SHIPMENT-DATE);
       PRODUCTS (TOYDB.PUB), KEY=PRODUCT-NO,
           LIST=(PRODUCT-NAME, PRICE, PRODUCT-LINE);

DEFINE(ITEM) $TIME,        EDIT="24HH:MM":
             $TODAY;
             PRODUCT-NO,   EDIT="^^-^^^-^":
             SALES-VALUE P(12,2), COMPUTE= (PRICE) * (QUANTITY):
             SHIPMENT-DATE, ENTRY-"Date Range";

sort(1) product-line: product-no;

PAGE HEADING
   "Report: PRODFC", LINE=2:
   "Date:",COL=105:
   $TODAY, COL=111:
   "Time:", COL=121:
   $TIME,   COL=127:
   "Page:", COL=121, LINE:
   $PAGE,   COL=129,LGN=3:

   "Production Forecast Report", CENTER, LINE=2:
   "-------------------------", CENTER, LINE;

GROUP(1) TITLE
   "Product Line:",  COL=2,LINE=2:
   PRODUCT-LINE,     COL=16:
   "-------------",  COL=2,LINE:
   "Product Number", COL=16, LINE=2:
   "Product Name",   COL=34:
   "Quantity",       COL=65:
   "Unit Price",     COL=79:
   "Sales Value",    COL=96:
   "--------------", COL=16, LINE:
   "-------------",  COL=34:
   "----------",     COL=65:
   "-----------",    COL=79:
   "------------",   COL=96;

BLOCK
  IF (product-line) = 50 THEN
   LET (PRICE) = .95 * (PRICE)
   ELSE
    LET (PRICE) = (PRICE);
BLOCKEND

GROUP(2) SUMMARY
   PRODUCT-NO,       COL=16:
   PRODUCT-NAME,     COL=36:
   TOTAL(QUANTITY),  COL=65:
   PRICE,            COL=79, EDIT= "*******, ^^":
   TOTAL(SALES-VALUE), COL=94, EDIT= "Z,ZZZ,ZZ^,^^";

GROUP(1) SUMMARY
   "--------",       COL=65:
   "-------------",  COL=95:
   TOTAL(QUANTITY),  COL=65,LINE:
   TOTAL(SALES-VALUE), COL=95, EDIT = "$,$$$,$$^,$$":
   " ", PAGE;

SELECT SHIPMENT-DATE AND PRODUCT-LINE;
```

The printed report is shown below:

```
Report: PRODFC                                                  Date: 03/20/92  Time:  08:23
                                                                                Page:     1

                                    Production Forecast Report
                                    --------------------------

Product Line: 50
------------
             Product Number    Product Name                Quantity     Unit Price      Sales Value
             --------------    ------------                --------     ----------      -----------
             61-288-F          FILE, FINE                     784       *****11.02        8,639.68
             61-288-M          FILE, MIDDLE                   356       *****11.59        4,126,04
             61-288-R          FILE, ROUGH                    480       *****12.64        6,067.20
             68-401-F          EMERY PAPER FINE               414       *****23.56        9,653.84
             61-288-M          EMERY PAPER MTD.               545       *****23.56       12,840.20
             61-288-M          EMERY PAPER RGH                895       *****23.28       20,835.60
                                                             --------                    -----------
                                                               3474                      $62,262.56
```

Define Data

The first step in report conversion is to make sure that all elements and files have been defined in the dictionary. You can use the DICTDBM procedure SHOW FILE or compile the report with the DEFN option as follows:

```
:run REPCOMP.PUB.SYS

SOURCE FILE>  TOYREP

LIST FILE>

CONTROL>  DEFN, DICT
```

The DICT option means that the compiler will check for dictionary definitions of elements and files used in the report. Any elements not defined in either the report or the dictionary, and any files accessed which are not defined in the dictionary, will cause compilation errors.

The compile listing is printed to your terminal.

The last section of the listing is the Data Definition section:

```
DATA ITEM DEFINITIONS:

     $PAGE                                I (   4,  0,  2)
     $TIME                                T (  64,  0,  2)
     $TODAY                               D (  64,  0,  2)
  D  PRICE                                P (   7,  2,  4)
  D  PRODUCT-LINE                         X (   2,  0,  2)
  D  PRODUCT-NAME                         X (  16,  0, 16)
  D  PRODUCT-NO                           X (   6,  0,  6)
  D  QUANTITY                             I (   4,  0,  2)
     SALES-VALUE                          P (  12,  2,  6)
  D  SHIPMENT-DATE                        X (   4,  0,  4)

CODE FILE STATUS: REPLACED


0 COMPILATION ERRORS
PROCESSOR TIME=00.00.09
  ELAPSED TIME=00:00:30
```

This listing indicates by the D in the first column that all items except $PAGE, $TIME, $TODAY, and SALES VALUE are defined in the dictionary. The first three items are standard Report/V items that BRWCONV will replace with the appropriate standard HP ALLBASE/BRW items. SALES VALUE, as seen in the DEFINE statement in the report source, is a calculated item. BRWCONV translates calculated items.

The ACCESS statement caused no compilation errors so the files accessed are defined in the dictionary.

Since all necessary items and files are in the dictionary, no action is necessary.

Verify Names

All of the items used in the report are legal HP ALLBASE/BRW names, so no action is necessary.

Define Date and Time Items

Use DICTDBM to modify the long name for SHIPMENT-DATE to !DATE-MD.

Create BRWDIC

Run BRWDIC3000.PUB.SYS. Answer Y to the prompt asking if Parent/Group items should be included.

Convert the Report

After completing the preparation steps, you are ready to convert the report using the steps given in the section in this appendix called Running BRWCONV.

BRWCONV issues a message that the report has been converted with warnings. The file RWARN contains these warnings:

```
Report/V or Inform/V to BRW Conversion Program
File :  TOYREP2

*** WARNING: REPORT/v DISPLAY statement ignored
              SHIPMENT-DATE, ENTRY="Date Range";
                                                 ^
*** WARNING: ENTRY option in DEFINE(ITEM) ignored

*** WARNING: REPORT/V BLOCK statement ignored
```

The Report/V DISPLAY statement prints a message to the terminal when the report is run. It has no effect on the final report, so you can ignore this warning.

The ENTRY option specifies a prompt that tells the user to enter parameter values. HP ALLBASE/BRW displays parameter names on the Request Report screen instead. You can also ignore this warning.

The BLOCK-BLOCKEND statement has no equivalent in HP ALLBASE/BRW so BRWCONV cannot translate it. To see what effect ignoring the statement has, you can compile and run the new report. The printed report looks like this:

```
Report: PRODFC                                                  Date: 03/20/92  Time:  16:58
                                                                                Page:     1

                                   Production Forecast Report
                                   --------------------------
                              *** WARNING(S) occurred during Report Conversion

Product Line: 50
------------
             Product Number    Product Name                 Quantity     Unit Price      Sales Value
             --------------    ------------                 --------     ----------      -----------
             61-288-F          FILE, FINE                        784       *****11.60        9,094.40
             61-288-M          FILE, MIDDLE                      356       *****12.20        4.343.20
             61-288-R          FILE, ROUGH                       480       *****13.30        6,384.00
             68-401-F          EMERY PAPER FINE                  414       *****24.80       10,267.20
             61-288-M          EMERY PAPER MTD.                  545       *****24.80       13,516.00
             61-288-M          EMERY PAPER RGH                   895       *****24.50       21,927.50
                                                               --------                     -----------
                                                                  3474                      $65,532.30
```

Notice the differences between this report and the original Report/V report. You must manually modify the HP ALLBASE/BRW report

to eliminate the differences, which were caused by the ignored BLOCK-BLOCKEND statement.

The Report/V BLOCK-BLOCKEND statement modifies the price of products in product line 50. To duplicate this in the new report, modify the HP ALLBASE/BRW report as follows:

- Define a calculated item, ADJ-PRICE, with the formula:

```
If PRODUCT-LINE = 50
then PRICE * .95
else PRICE
```

- Modify the formula for SALES-VALUE so that it is computed based on ADJ-PRICE instead of PRICE.
- Print out ADJ-PRICE instead of PRICE in the level two break footer.
- Delete the warning statement from the Report Header.

The modified HP ALLBASE/BRW report prints exactly the same report as the original Report/V report.

Glossary

access block
A unit of data access, when the report is executed. Locks on databases or files are only held during execution of one access block.

Produces an intermediate work file, or if it is the last access block, the final report.

The compile program BRWCOMP splits a table into several access blocks, or combines several tables into one access block, depending on the selected access methods.

access method
Specifies how to retrieve the data, that is, the sequence in which the source tables are to be accessed; the method to be used for each (serial or keyed) and if the source table is to be locked during access.

access table
The result table when you combine one or more source tables in HP ALLBASE/BRW. The access table is a logical table defined within your report, it is not an actual table (file) on the system. You can use one or more access tables as source tables within the same report. For instance, if you join two source tables to create an access table, you can in turn use that access table as a source table and join or merge it with another table in the report.

argument
Provides a value, when an expression is calculated; can be an item, function, parameter, constant, or another expression.

array
Combination of several similar items (called array items or array components) under one name.

break
Occurs when the value of a sort item (or part of it) changes. For each sort level a break heading and break footing lineset can be specified.

BRWACCSD
The program used to create a BRWDIC dictionary file from HP Access Central Dictionary.

BRWAPPD
The program used to create an BRWDIC dictionary file from HP Application Dictionary.

BRWCOMP
See *report compile program*.

BRWCONF
The system configuration file you use (depending on your user interface language) in the account or group you want to configure. You create the BRWCONF file when you copy it from BRWC000.PUB.SYS.

See *configuration file*.

BRWCONV
See *report conversion program*.

BRWDIC
See *dictionary file*.

BRWD3000
The program used to create a BRWDIC dictionary file from HP Dictionary/V.

BRWEMPTY
Program used to clear intermediate report files. Filename BRWMPTY.PUB.SYS

BRWEXEC
See *report execution program*.

BRWGEND
The program used to create a BRWDIC dictionary file from a user-generated file containing dictionary statements.

BRWLIST
See *report list program*.

BRWSD
The program used to create a BRWDIC dictionary file from HP System Dictionary.

BRWSETUP
See *configuration program*.

BRWSTART
See *report request program*.

calculated item
Is calculated from other items, parameters or constant values. Provides a single value each time a calculation is performed.

Also called calc item.

calculation formula
An expression which results in a numeric, string, date or time value. The formula of a calculated item or a function.

child
The parent element in a source table may be subdivided into several child elements. In this way, included parts of the parent element can be referenced separately. See also, *composite search element.*

column calculations
Vertical calculations, which summarize data for a break, page or report. Can total, average, find a minimum, find a maximum, count or calculate. You can set the source and the reset level for column calculations.

Results of column calculations can be used as arguments within layout calculations or layout functions.

common item
Same item occurs in several source tables. Are used to define relations.

Only records in which each common item has identical values in the source tables will be joined together. Can have different names in the source tables.

Need not be key items, that is, do not have to be HP TurboIMAGE/XL search items or KSAM keys.

compile
Compiling the report collects the records you want in your report, performs any calculations you specified, and organizes the report according to the layout you defined. The compile program BRWCOMP splits a table into several access blocks, or combines several tables into one access block, depending on the selected access methods.

composite search element
A search element that consists of several components. To define a composite search element you must first define an element with the name of the search element. Then use element/element relations to describe the components of the composite search element. The composite search element becomes a parent element, and the components become its child elements.

condition
A formula, which is either true or false, depending on given values. Usually comparisons combined with boolean operators. The formula of a selection condition, relation condition, line or lineset suppress condition.

configuration file
The file name is BRWC000. Contains defaults to be used in RSPEC, configuration for BRWEXEC, and global functions.

HP ALLBASE/BRW looks for the file in the following order:

1. BRWCONF
2. BRWCONF.PUB
3. BRWCONF.PUB.SYS
4. BRWC*nnn*.PUB.SYS, where *nnn* is the number returned by the Native Language Support intrinsics.
5. BRWC000.PUB.SYS

configuration program
Used to configure HP ALLBASE/BRW. The file name is BRWSETUP.PUB.SYS.

core set
The System Dictionary contains a number of predefined entity types, relationship types, relationship classes, and attributes, known as the core set. These structures are created when the dictionary is initialized.

data interchange format files (DIF)
A standard file format used by different applications to exchange data. Visical, Lotus 1-2-3 and Charting Gallery can read DIF files.

data manipulation
Extracting information from source tables and applying the selection conditions and calculations needed to produce the final access table that you use to generate the report.

DBA
Database Administrator. The individual responsible for the creation and maintenance of HP TurboIMAGE/XL or HP ALLBASE/SQL databases, including granting and revoking access authority.

database
Physical storage of application data. Consists of one or more datasets for an HP TurboIMAGE/XL database, or tables and views for an HP ALLBASE/SQL database. Provides various access methods for use to retrieve data.

database administrator
See *DBA*.

database environment
(DBEnvironment) A collection of files for one or more logical HP ALLBASE/SQL databases.

data dictionary
Central data storage for data definitions. Contains definitions of items, datasets, databases, files, keys, passwords, and so on.

HP Dictionary/V, HP System Dictionary, HP Application Dictionary, HP Access Central Dictionary, and user-generated dictionaries can be used with HP ALLBASE/BRW. For each account in which HP ALLBASE/BRW is to be used, a dictionary file must be created first by using the programs BRWD3000, BRWSD, BRWAPPD, BRWACCSD, or BRWGEND.

data set
Physical storage of source data in an HP TurboIMAGE/XL database; the two types of data sets are master and detail.

detail
The lineset printed for each record in the final access table.

dictionary file
Contains definitions of items, databases, data sets, files of one account. The file name is BRWDIC. HP ALLBASE/BRW can access multiple dictionary files in one report.

dictionary output file
Saves data for later processing. The file code is ROUTD. Data is stored in external form as defined in the data dictionary.

exception condition
An exception condition occurs when the value of an item, function, parameter, or expression is invalid. When an exception condition occurs, HP ALLBASE/BRW issues an execution warning message, which is printed at the end of the report.

execution file
Used to produce the report. The file code is REXNM. Cannot be modified. Contains the report specifications in compiled form, similar to a program file.

expression
A nested formula in parentheses.

file
Physical storage of a source table that is not in a database.

final access table
The table used for the report layout. If several tables are nested, it is the table on top of table hierarchy.

footing
Data printed at the bottom of a page, report or after a detail. Lineset for a break, page or report.

Can be used to print column calculations for that break, page or report. Can be used to print data from the previous detail, or previous lineset if it is a page footing.

format
Description of the items of a table (name, type, length, position in record, number of array components). The format of each dataset or file used in a report must have been specified in the dictionary.

The format of intermediate report files is included in these files (self-describing files). They cannot be specified in the dictionary.

formula
Calculated items, functions and conditions are expressed in a formula. The calculation language is used to express formulas.

function
An expression frequently used in other formulas. Can have up to nine function arguments.

function argument
A placeholder for an actual expression supplied when the function is used in another formula.

heading
Data printed at the top of a page, report or before a detail. Lineset for a break, page or report. Can be used to print data for the next detail, or the next lineset if it is a page heading. Can print carry forward column calculations from the previous footing of same type.

HP ALLBASE/SQL
Database management system used on the HP3000 that supports the relational data model. SQL = Structured Query Language.

HP ALLBASE/SQL authority
A privilege given to a user to perform specific database operations such as connecting to a DBEnvironment or accessing tables or views. The DBA is responsible for granting and revoking authority.

HP ALLBASE/SQL CONNECT
A command used to access a DBEnvironment. The user must have a CONNECT or DBA authority to the specified DBEnvironment.

HP ALLBASE/SQL query
HP ALLBASE/BRW builds a SELECT command to retrieve data from one or more tables or views. This process of retrieving the data and the command are referred to as an HP ALLBASE/SQL Query.

HP ALLBASE/SQL SELECT
A command that retrieves data from one or more tables or views. The retrieved data is in the form of a table, called the result table or query result.

HP ALLBASE/SQL table
Data in the relational database is organized into a table consisting of rows and columns. A row is a record and a column is an item.

HP ALLBASE/SQL table owner
Within HP ALLBASE/SQL a table or view is accessed by specifying both the owner and table name.

HP ALLBASE/SQL view
A table derived by placing a "window" over one or more tables to let users access only certain data.

HP ALLBASE/SQL system catalog
Contains information on data structures of HP ALLBASE/SQL databases.

HP ALLBASE/SQL autostart mode
An autostart flag is set to determine how the DBEnvironment is opened. It must be set to "on" to allow the CONNECT command that is always used by HP ALLBASE/BRW.

HP ALLBASE/SQL user mode
When set to "multi", more than one user and program can use the DBEnvironment.

HP Application Dictionary
Data dictionary used with HP applications.

HP TurboIMAGE/XL
Database management system used on the HP3000 that supports the network data model.

intermediate report file
(IRF) Saves data for later reporting. The file code is ROUTI.

Used for reporting on results (exception reporting). Used to optimize production reporting cycles (reduces redundant database access).

Data is stored in HP ALLBASE/BRW internal format.

IMAGE
The term IMAGE is sometimes used to represent both HP IMAGE/3000 and HP TurboIMAGE/XL databases.

item
A column in a table. Can get its values from database, file or from calculations. Can be printed, used for sort or selection, written to an output file, used in calculation formulas. If a

calculated item, provides a single value each time the calculation is performed.

item edit
Rule for how an item value is to be formatted for printing.

item types
Are:

S = string
N = fixed point numeric
I = Integer
R = floating point real
D = date
T = time

join
Method of combining source tables in which the columns of two or more tables are joined side-by-side.

KSAM
Keyed sequential access method on the HP3000 system.

KSAM file
Allows sequential and keyed access to the data in the KSAM file. Always consists of two MPE files: a data file and a key file.

HP ALLBASE/BRW supports KSAM files with fixed length records.

layout calculated item
A calculated item defined within the report layout. Is calculated when the report is formatted; can be calculated from column calculations; can be printed, written to an output file, or used to calculate other layout calc items. Cannot be used for sort or selection.

Also called layout calc item.

layout functions
Are defined within the report layout; can be used in other formulas defined in the report layout.

Are similar to program subroutines.

line
A printed line on a page; can consist of item values and fixed text.

lineset
A unit printed when a certain event occurs in a report, for example: a detail, break, new page, or the beginning or ending of the report. Can consist of a single line of text or items, multiple lines or no lines. Pagination and output file records also relate to a lineset.

link
See *join*.

literals
A text constant of up to 16 characters that is stored in BRWDIC and can be referenced by a symbolic name.

logical-expessions
See conditions

lock mode
Used to secure a lock on an IMAGE dataset, HP ALLBASE/SQL, table, KSAM or MPE file. You can specify a lock mode on the HP ALLBASE/BRW Tune Access screen.

merge
Method of combining two similar tables by stacking (appending) one on the other.

MPE XL
Operating system on the HP3000 computer.

MPE file
Sequential file supported by MPE. HP ALLBASE/BRW supports standard MPE files with fixed length records. Special MPE files (RIO, MSG, CIR, and so on) are not supported by HP ALLBASE/BRW.

multi-pass reporting
Execution of a sequence of reports and optional application programs.

numeric precision
Determines the precision of an item's value when it is used in further processing.

Consists of a number and method code. The number indicates the least significant digit of the precision as the 10th exponent of this digit. The method code indicates how superfluous digits are to be treated: rounded (R), truncated (T) or ceiled (C).

open join
Method of combining source tables in which source tables are joined and all records from one source table are reported, even if these have no correspond ing records in the other source tables. See also *join*

outer join
See *open join*.

output file
Saves data for later reporting or processing. Can be specified for each lineset; can be an intermediate report file or a dictionary output file.

page
A page of the printed report. Page heading and footing linesets can be defined to be printed on each page.

Other linesets can be printed as they are, or they can be positioned at an absolute position of the page.

pagination
The control of the page layout. A new page can be specified before or after a lineset; or, a lineset can be split if it does not fit completely on rest of page. A lineset can be positioned absolute on a page.

parameter
A constant which gets its value at report execution time and which makes data selection for a report more flexible.

parameter types
Are:

P = comparison predicate
S = single value
L = list of values

parent element
The parent element in a source table may be subdivided into several child elements. In this way, included parts of the parent element can be referenced separately. See also, *composite search element.*

positioned item
Positioned by line number within lineset, position within line, and length. Not all items in the report layout need to be positioned (printed), for example, sorts or column calculations do not require printing.

print field
The space in a lineset in which the value of an item is printed. Specified by line number within lineset, position within line, and length. The item values can be formatted using item edits before they are printed in the print field.

print file
Contains the printed report; can be a spool file, a device file or a permanent file on disk. The file code is PRINT. Has fixed length records and can be either CCTL or NOCCTL.

projected items
When source tables are combined, their items are projected to the newly formed table. A projected item can be given an alias name, excluded from projection, and given numeric precision; these attributes are only in effect *outside* the item's source table.

Quoted Basic Files
Files used as data transfer files by some standard programs, including Lotus 1-2-3, RBASE 5000, DBASE, and BASIC programs. DSG/3000 can also read Quoted BASIC files as free-formatted data files. In BASIC, quotes are used to delimit strings.

record
A row in a table. Each record in the final access table will be printed as one detail lineset.

record number
See *TABLE-REC-NUMBER*.

referenced access file
A report specification file that contains access definitions used by another report.

relation
Defines the rules for joining source tables; does *not* specify the access method to be used to retrieve the data.

relation condition
Selects a subset of records from joined source tables; is defined within a table.

Uses the calculation language.

relational model
Viewing database information as a collection of tables that contain items, without considering internal data formats or access patterns.

report
Data printed on paper or to disk. Can also mean the definition of the report (report layout, data access, calculations to be used to create the report, and so on).

report compile program
Used to compile a report. The file name is BRWCOMP.PUB.SYS. Can be used stand-alone or within the report specification program.

report conversion program
Used to convert Report/V and Inform/V reports to HP ALLBASE/BRW specification files. The filename is BRWCONV.PUB.SYS.

report execution program
Used to produce a report. The file name is BRWEXEC.PUB.SYS. Can be used stand-alone or within the report specification program.

report job file
For producing multi-pass reports; used for bundling the production of multiple reports. Is created using HP EDITOR/3000 or any compatible editor; the optional file code is RJOB.

report layout
The way the report looks in its final form, after all calculations have been performed, rows and columns have been placed on the page, and so on.

report list program
Used to list the specifications of a report. The file name is BRWLIST.PUB.SYS. Can be used stand-alone or within the report specification program.

report request program
A program used to request HP ALLBASE/BRW reports. The filename is BRWSTART.PUB.SYS. Can be used stand-alone.

report specification file
When you have finished creating the basic report, you have a report specification file that contains all the specifications for your report.

report specification program
To specify, list, compile, and execute a report. The filename is BRWXL.PUB.SYS. Invokes the report compile, execution, and list programs.

result table
A table that is formed by HP ALLBASE/BRW by accessing one or more datasets, tables, or files. The result table used for the report layout is called the final access table.

root source table
First source table in a sequence of joins with keyed access.

row calculation
Calculations using layout calculated items; specified by adding a calculated item to a report and then defining a formula for that item.

sample layout
A picture of the report layout, in which the print fields are shown as X's or 9's for numeric fields. X's and 9's can be changed with the program BRWSETUP.PUB.SYS.

selection
Selects a subset of the data in the final access table.

selection condition
Criteria to select a subset of data from the final access table. Is expressed as a condition formula and uses the calculation language. Can be flexible by using parameters.

selection criteria
Control the amount of data which goes into a report, as well as when and how the report is printed.

selection set
One group of data selection criteria used for a report. Multiple selection sets can be defined in a specification file, but the report must be compiled once for each selection set. Allows data-dependent security. Can contain default values for parameters, print file and report schedule.

sort
A report can be sorted by up to nine items; each item in ascending or descending order.

source table
Can be an IMAGE dataset, an HP ALLBASE/SQL table, a MPE, KSAM or IRF file or an HP ALLBASE/BRW table defined in the report.

specification file
Used for creation and modification of a report. The file code is RSPEC. Must be compiled into an execution file to produce the report. Contains the report specifications.

standard functions
Predefined functions supplied by HP ALLBASE/BRW.

standard job file
Used as report job file, when a single report is started in batch. The file name is BRWJ000.

HP ALLBASE/BRW looks for the file in the order below:

1. BRWJOB
2. BRWJOB.PUB
3. BRWJOB.PUB.SYS
4. BRWJ*nnn*.PUB.SYS, where *nnn* is the number returned by the Native Language Support intrinsics.
5. BRWJ000.PUB.SYS

The standard job file can be modified using HP EDITOR/3000 or any compatible editor.

suppress condition
Criteria to suppress printing of a line or lineset; can be specified for each line or lineset. Is expressed as a condition formula and uses the calculation language.

syntax diagram
Describes the syntax of the HP ALLBASE/BRW calculation language in a graphic form.

table
A collection of data in which the rows are records and the columns are items. Basis for the report layout or other tables; joins or merges the source tables. Tables can be nested, thus can consist of other tables as source tables.

table calculated item
Is a calculated item defined within a table. After it is projected from that table, it becomes a normal item. Is calculated when the data is retrieved. Can be printed, used for sort and projection, written to an output file, used to calculate other items.

Also called table calc item.

table function
Defined for a table. Can be used in other formulas defined in the same table. Similar to a program subroutine.

TABLE-REC-NUMBER
The logical record number defined by HP ALLBASE/BRW. Source tables that are IMAGE data sets, KSAM files, or MPE files always have the item TABLE-REC-NUMBER included in the result table, HP ALLBASE/SQL tables do not.

text
Fixed text printed on a line.

transaction isolation level
Applies to HP ALLBASE/SQL databases. Is the level of concurrent access that is allowed during a transaction.

transaction priority
Applies to HP ALLBASE/SQL databases. Is an integer from 0 to 255 specifying the priority of a transaction. The transaction with the largest priority number is aborted to remove a deadlock.

value
A value in the mathematical sense. Each record con tains one value per item.

VPLUS
User interface system on the HP3000 system. Used in the report specification program.

Index

C

D

M

O

U