Advantage™ VISION:Inform®

User Guide

4.0

Computer Associates®

IFBUG040.PDF/D31-001-011

Contents

Chapter 1: Introduction

Chapter 2: Using Quick Query

Chapter 3: General Concepts

Chapter 4: Using VISION:Bridge

Chapter 5: Using Immediate Response

Chapter 6: Data Selection and Field Processing

Chapter 7: Advanced Topics

Chapter A: Paging and Viewing

Chapter B: Glossary of the CUSTOMER Database

Chapter C: Examples of Editing Numeric Data

Chapter D: Reference Summary

Index

Chapter

1

Introduction

This guide provides information for end users and data processing professionals who use Advantage™ VISION:Inform® for their reporting needs. The inexperienced user can use this book as a learning guide and the experienced user can use it as a reference.

This guide describes commands and their use, and presents practical applications for using these commands. Often you will use VISION:Inform for similar types of applications in your own installation.

What is VISION:Bridge?

VISION:Bridge™ is the 3270 platform that uses VISION:Inform for extracting data, formatting data, and creating displays and reports. VISION:Bridge provides the means for a user with an IBM® 3270 terminal or 3270 emulator to connect to VISION:Inform. Once connected, the user can create queries using VISION:Bridge features known as the Full Screen Editor, Quick Query, and Immediate Response.

VISION:Inform and VISION:Bridge work together to provide users with access to virtually any file or data management system supported on IBM host computers. Accessing the data you want can be easily accomplished through a IBM 3270 terminal or the Batch Simulator.

VISION:Bridge is an easy-to-learn, easy-to-use, natural language for you to develop queries to retrieve data. You create these queries with an IBM 3270 terminal, or terminal emulator, and submit them for processing in the background area of your host computer.

The queries you submit are automatically batched together with other queries and optionally processed during non-peak hours. Batching queries and subsequently processing them together alleviates unnecessary computer and database access bottlenecks.

You can view the reports you generate at your terminal, at someone else's terminal, or have them printed. Your reports can also be formatted for use by other software tools, or they can be formatted in HTML for use by web browsers.

What is Quick Query?

Quick Query is a feature of VISION:Bridge that you use to get reports quickly. Quick Query uses only one structured screen; everything you need is right in front of you, with no navigation across screens required. You do not have to learn how to structure queries or how to write commands. The query is automatically built by pointing and picking from a list of available options.

Quick Query supports many reporting capabilities, including selection criteria, sorting, grouping, summarization, and page titles. Online Help is also available to show you what to do.

What is Immediate Response?

Immediate Response is a feature of VISION:Bridge. It uses the same natural language as VISION:Bridge for developing queries to retrieve and instantaneously display the data you want to see. Your queries are created from a terminal and processed online. You can display data with either an entire query or with just an output-generating statement. You do not require a background processor.

What is the Batch Simulator?

This means of using VISION:Bridge provides the batch environment user with the same powerful data extraction and reporting capabilities available to the online user. The only difference between the Batch Simulator and VISION:Bridge or Immediate Response is that the interactive, full screen editing facilities are not available for developing queries. You use the same natural language for queries, but process the queries in batch (using batch JCL), providing the scheduled execution of stored queries.

Contacting Computer Associates

For technical assistance with this product, contact Computer Associates Technical Support on the Internet at esupport.ca.com. Technical support is available 24 hours a day, 7 days a week.

Chapter

2

Using Quick Query

All About Quick Query

The Quick Query feature of VISION:Bridge is the formatted approach to creating a query quickly. It is easy to develop a query in a "pick and choose" fashion, because you do not need to know the detailed syntax of the query language or the exact spelling of data names and databases. All files, logical data views, and databases available through VISION:Bridge are also available through Quick Query. Its purpose is to assist you so that you can quickly and easily:

- Pick and choose the fields for your reports.
- Group and sort columns of data.
- Calculate totals, counts, averages, maximums, and minimums for columns of data.
- Provide selection criteria.
- Make quick changes to existing reports.

You can quickly develop queries using the Quick Query panel shown in Figure 2-1. It contains the specifications for a quick and easy report. See a sample of the resulting report in Figure 2-2.

```
Aprqqm2                Computer Associates - Quick Query
PR Query: FINANCE4     DataView: FINANCE                    Report Width:  70
  Title 1: PROFIT CENTER BUDGETS VS ACTUAL - AUGUST
        2: DIRECT LABOR & MATERIALS - PRODUCT SUMMARY
Report Fields - Find: %                          Summary Only? N  More:
Summary   Group  1    Group  2     Group  3     Column  4      Column  5
          ___ ACCTDIV ___ PRODCODE ___ ACCTNO1  ___ ACTAMTBD   ___ ACTAMTAC
  Total                                       G2  PRODCODE   G2  PRODCODE
  Count                                       ___            ___
    Max                                       G2  PRODCODE   G2  PRODCODE
    Min                                       ___            ___
    Avg                                       ___            ___
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1           ACCTMM     EQ   08
                 L2    AND    ACCTNO1    EQ   5150
                 L3    OR     ACCTMM     EQ   08
                 L4    AND    ACCTNO1    EQ   5510
Available Fields    - Like: %                              More:   +
___ ACCOUNT    ___ ACCTCNT    ___ ACCTDATE    ___ ACCTDD     ___ ACCTDESC
___ ACCTDIV    ___ ACCTMM     ___ ACCTNO      ___ ACCTNO1    ___ ACCTNO2
___ ACCTYY     ___ ACTAMTAC   ___ ACTAMTBD    ___ ACTCHGMM   ___ ACTCHGPD

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-1 A Quick Query

```
MAR 22, 2001    PROFIT CENTER BUDGETS VS ACTUAL - AUGUST        PAGE 1
                DIRECT LABOR & MATERIALS - PRODUCT SUMMARY

-------------------------------------------------------------------------
             ACCOUNT   PRODUCT  ACCOUNT     ACCOUNT        ACCOUNT
             DIVISION   CODE    NUMBER   AMOUNT BUDGET  AMOUNT ACTUAL
-------------------------------------------------------------------------
                21      1011     5150          2,100          5,143
                                 5510          3,100          5,934

PRODCODE TOTAL  21      1011                   5,200         11,077
         MAX.                                  3,100          5,934

                        1012     5150          2,100          1,795
                                 5510          3,100          2,693

PRODCODE TOTAL  21      1012                   5,200          4,488
         MAX.                                  3,100          2,693

                        1013     5150          2,100          1,057
                                 5510          2,700          1,586

PRODCODE TOTAL  21      1013                   4,800          2,643

/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/

MAR 22, 2001    PROFIT CENTER BUDGETS VS ACTUAL - AUGUST        PAGE 6
                DIRECT LABOR & MATERIALS - PRODUCT SUMMARY

-------------------------------------------------------------------------
           ACCOUNT   PRODUCT  ACCOUNT     ACCOUNT        ACCOUNT
           DIVISION   CODE    NUMBER   AMOUNT BUDGET  AMOUNT ACTUAL
-------------------------------------------------------------------------
GRAND    TOTAL                                57,000         62,066
         MAX.                                  3,600          5,934
```

Figure 2-2 The Resulting Report

Figure 2-3 shows an empty Quick Query panel with the following panel areas:

Heading	Specifies the query name and the database.
Title	Specifies two title lines for the report.
Field and Summary window	Specifies what fields go in which columns, as well as summary, sort, and grouping information.
Data Selection	Provides selection criteria for including only the data you want in your report.
Point and Pick window	Provides a scrollable window to select query names, databases, or data names depending on context.
Message line	Displays system diagnostic and informational messages.
Command area	For entering a command instead of pressing a PF key.

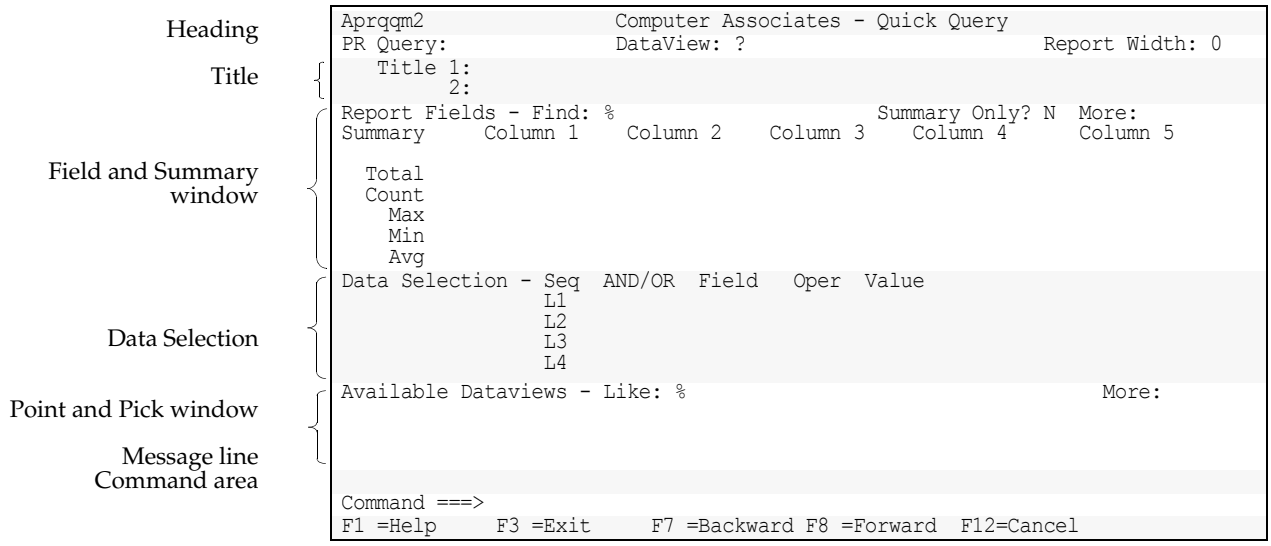


Figure 2-3 An Empty Quick Query Panel

The Quick Query panel appears when you log on to VISION:Inform and select Option 4 (Quick Query) from the Main Menu. Option 5 (Quick Query Immediate Response) is only slightly different.

For more information. see the section Quick Query Immediate Response. Figure 2-4 shows the Main Menu with Option 4 entered.

```
Menu2                  Computer Associates - Main Menu

4 Enter one of the following VISION:Inform or VISION:Bridge Options:


  VISION:Inform Options

  1. Operation Facilities                (Background Processor Status)
  2. Administration Facilities           (Profile development)
  3. Report Facilities                   (Report Handling)

  VISION:Bridge Options

  4. Quick Query                         (Assisted Query Development)
  5. Quick Query Immediate Response      (Assisted Query Development)
  6. Standard Query Processing           (Submit, Delete, Edit
                                          Queries and Stmts)
  7. Immediate Response Query Processing (Run Queries and Immed Mode)


Command ===>
F1 =Help     F12=Cancel
```

Figure 2-4 The Main Menu

Note:

- In JCL and panels, **bold** **underlined** text indicates user entries.
- In JCL and panels, **bold** entries indicate previous entries.

To create a new query:

1. Give the query a name by typing a 1- to 8-character alphanumeric name in the name field of the header area (Figure 2-5).

Point and Pick window

```
Aprqqm2                 Computer Associates - Quick Query
PR Query: mynew1      DataView:                          Report Width:   0
  Title 1:
        2:
Report Fields - Find: %                          Summary Only? N  More:
Summary     Column 1     Column 2     Column 3     Column 4     Column 5
  Total
  Count
    Max
    Min
    Avg
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1
                 L2
                 L3
                 L4
Available Queries   - Like: %                                    More:   +
_   ABCDREN      _   DB2DEPT     _   DB2ITEMX     _   DB2OLYMX     _   DB2QR471
_   FINANCE0     _   FINANCE1    _   FINANCE2     _   FINANCE3     _   FINANCE4
_   FORMTPE1     _   FRED1       _   IN6DEPT      _   IN6EMPE1     _   IN6EMPX

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-5 Creating a Query

If you choose a query name which already exists, that query and its specifications display on the Quick Query panel.

You can also select an existing query from the scrollable list in the Point and Pick window.

2. Press Enter for a list of database names. They appear in the scrollable Point and Pick window (Figure 2-6).

```
Aprqqm2                Computer Associates - Quick Query
PR Query: MYNEW1       DataView: ?                        Report Width:   0
  Title 1:
        2:
Report Fields - Find: %                          Summary Only? N  More:
Summary     Column 1     Column 2     Column 3      Column 4     Column 5
  Total
  Count
    Max
    Min
    Avg
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1
                 L2
                 L3
                 L4
Available Dataviews - Like: %                                    More:   +
_   ADMIN       _   CARS        _   CUSTFILE     _   CUSTLDV       _   CUSTLDV1
_   CUSTOMER    _   DB2CUST     _   DB2DEPT     _   DB2EMPE      _   DB2ITEM
_   DB2LDVPS    _   DB2LDV01    _   DB2LDV02    _   DB2OLYM      _   DB2OLYMM
                                                                  _
EN00** ITEM MYNEW1   NOT IN FOREGROUND LIBRARY.
Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-6 Creating a Query

The "?" in the Dataview field at the top of the panel reminds you to select a database for the next step. You must use the Point and Pick window to select a database.

3. Select a database for processing.

To select a database, type an 'S' in the space to the left of its name in the scrollable list down in the Point and Pick area as shown in Figure 2-7.

```
Aprqqm2                Computer Associates - Quick Query
PR Query: MYNEW1       DataView: ?                        Report Width:   0
  Title 1:
        2:
Report Fields - Find: %                          Summary Only? N  More:
Summary     Column 1    Column 2     Column 3      Column 4      Column 5
  Total
  Count
    Max
    Min
    Avg
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1
                 L2
                 L3
                 L4
Available Dataviews - Like: %                                    More:   +
s   FINANCE



Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-7 Selecting a Database

4. After you select a database, press Enter to display the field names in the database, as shown in Figure 2-8.

```
Aprqqm2                Computer Associates - Quick Query
PR Query: MYNEW1       DataView: ?                        Report Width:   0
  Title 1:
        2:
Report Fields - Find: %                          Summary Only? N  More:
Summary     Column 1     Column 2     Column 3     Column 4     Column 5
  Total
  Count
    Max
    Min
    Avg
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1
                 L2
                 L3
                 L4
Available Dataviews - Like: %                                    More:   +
_   ACCOUNT     _  ACCTCNT     _  ACCTDATE    _  ACCTDD      _  ACCTDESC
_   ACCTDIV     _  ACCTMM      _  ACCTNO      _  ACCTNO1     _  ACCTNO2
_   ACCTYY      _  ACTAMTAC    _  ACTAMTBD    _  ACTCHGMM    _  ACCTCHGPD

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-8 Displaying Field Names

You are now ready to create the specifications for the columns of data to appear in the final results. You will be interacting with the Field Summary window and the Point and Pick windows.

Figure 2-9 shows the Field Summary window and the Point and Pick window.

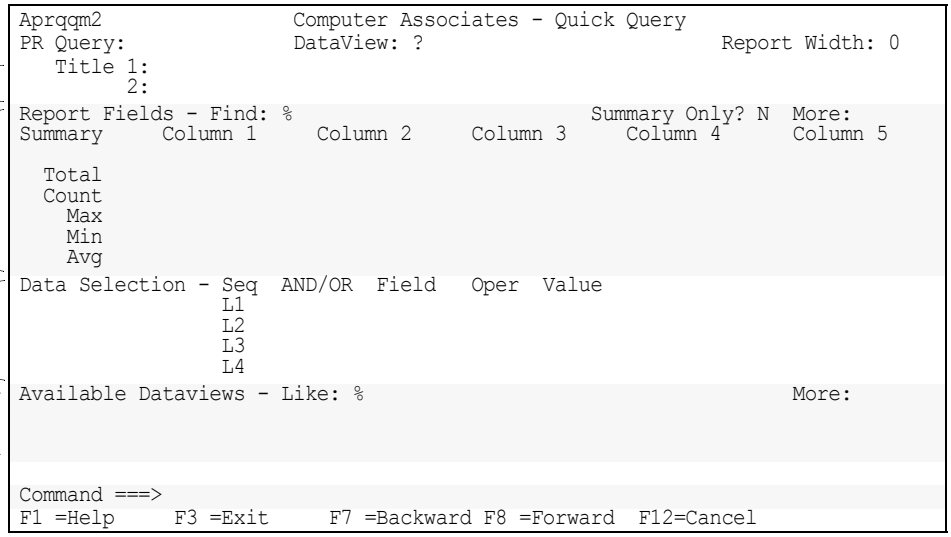


Figure 2-9 Part of a Quick Query Panel

To select a field, enter a column specification into the 3-character data entry field to the left of the appropriate field name in the Point and Pick window. There are three kinds of column specifications to choose from:

Cnn — Specifies a column of data (such as C05). In the example (Figure 2-12), column 5 of the report will be values for ACTAMTAC. (You can omit the leading zero.)

Gnn — Specifies a grouping field (such as G02). A grouping field produces control breaks in a report. You can also summarize other fields by this grouping field. In Figure 2-14, control breaks will be taken on column 1 and 2, and totals and counts of column 5 will be taken when column 5 changes value. (You can omit the leading zero.)

Onn — Sorts this column in order (such as O3). (You can omit the leading zero.)

? — Provides an additional line of field descriptions (if available), in the message area when you press Enter. The ? does not select the field.

Figure 2-10 and Figure 2-11 show the specifications for the first three columns of a report.

```
Aprqqm2               Computer Associates - Quick Query
PR Query: FINANCEP    DataView: FINANCE                Report Width:   0
  Title 1: PROFIT CENTER SUMMARY
        2: BUDGET ACTUALS
Report Fields - Find: %                        Summary Only? N  More:
Summary     Column 1     Column 2     Column 3     Column 4     Column 5

  Total
  Count
    Max
    Min
    Avg
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1
                 L2
                 L3
                 L4
Available Fields    - Like: PR%                                  More:
___ PRODCODE     ___ PRODCOD1   o3_ PROFDEP     ___ PROFDES    g2__ PROFDIV
g1_ PROFGRP      ___ PROFNO


Command ===>
F1 =Help     F3 =Exit     F7 =Backward  F8 =Forward  F12=Cancel
```

Figure 2-10 Specifying Columns for a Report

```
Aprqqm2               Computer Associates - Quick Query
PR Query: FINANCEP    DataView: FINANCE                Report Width:  30
  Title 1: PROFIT CENTER SUMMARY
        2: BUDGET ACTUALS
Report Fields - Find: %                        Summary Only? N  More:
Summary     Group  1     Group  2     Order  3     Column 4     Column 5
        ___ PROFGRP ___ PROFDIV ___ PROFDEP
  Total                             ___
  Count                             ___
    Max                             ___
    Min                             ___
    Avg                             ___
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1
                 L2
                 L3
                 L4
Available Fields    - Like: PR%                                  More:
___ PRODCODE     ___ PRODCOD1   ___ PROFDEP     ___ PROFDES    ___ PROFDIV
___ PROFGRP      ___ PROFNO


Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-11 Specifying Columns for a Report

Figure 2-12 and Figure 2-13 show specification of the last two columns of this report. Notice that the Field and Summary window now contains your specifications as column headings appearing just below the column specification type. The fields you select appear with the appropriate headings above the columns.

```
Aprqqm2                Computer Associates - Quick Query
PR Query: FINANCEP     DataView: FINANCE               Report Width:  30
  Title 1: PROFIT CENTER SUMMARY
        2: BUDGET ACTUALS
Report Fields - Find: %                          Summary Only? N  More:
Summary     Group 1     Group 2     Order 3     Column 4       Column 5
        ___ PROFGRP ___ PROFDIV ___ PROFDEP
  Total                         ___
  Count                         ___
    Max                         ___
    Min                         ___
    Avg                         ___
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1
                 L2
                 L3
                 L4
Available Fields    - Like: AC%                                  More:   +
___ ACCOUNT     ___ ACCTCNT     ___ ACCTDATE    ___ ACCTDD      ___ ACCTDESC
___ ACCTDIV     c4_ ACCTMM      ___ ACCTNO      ___ ACCTNO1     ___ ACCTNO2
___ ACCTYY      c5_ ACTAMTAC    ___ ACTAMTBD    ___ ACTCHGMM    ___ ACTCHGPD

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-12 Specifying Columns for a Report

```
Aprqqm2                Computer Associates - Quick Query
PR Query: FINANCEP     DataView: FINANCE                 Report Width:  54
  Title 1: PROFIT CENTER SUMMARY
        2: BUDGET ACTUALS
Report Fields - Find: %                          Summary Only? N  More:
Summary     Group 1     Group 2     Order 3     Column 4       Column 5
        ___ PROFGRP ___ PROFDIV ___ PROFDEP ___ ACCTMM     ___ ACTAMTAC
  Total                         ___           ___            ___
  Count                         ___           ___            ___
    Max                         ___           ___            ___
    Min                         ___           ___            ___
    Avg                         ___           ___            ___
Data Selection - Seq  AND/OR  Field Oper  Value
                 L1
                 L2
                 L3
                 L4
Available Fields    - Like: AC%                                  More:   +
___ ACCOUNT     ___ ACCTCNT     ___ ACCTDATE    ___ ACCTDD      ___ ACCTDESC
___ ACCTDIV     ___ ACCTMM      ___ ACCTNO      ___ ACCTNO1     ___ ACCTNO2
___ ACCTYY      ___ ACTAMTAC    ___ ACTAMTBD    ___ ACTCHGMM    ___ ACTCHGPD

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-13 Specifying Columns for a Report

Figure 2-10 through Figure 2-13 also show you that you can input more than one column specification in an interaction.

Figure 2-14 shows entry of the summary specifications for column 5.

```
Aprqqm2                Computer Associates - Quick Query
PR Query: FINANCEP     DataView: FINANCE                  Report Width:  54
  Title 1: PROFIT CENTER SUMMARY
        2: BUDGET ACTUALS
Report Fields - Find: %                            Summary Only? N  More:
Summary     Group  1    Group  2    Order  3      Column 4      Column 5
        ___ PROFGRP ___ PROFDIV ___ PROFDEP   ___ ACCTMM    ___ ACTAMTAC
  Total                          ___           ___           g2_
  Count                          ___           ___           g2_
    Max                          ___           ___           ___
    Min                          ___           ___           ___
    Avg                          ___           ___           ___
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1
                 L2
                 L3
                 L4
Available Fields    - Like: AC%                                  More:   +
___ ACCOUNT     ___ ACCTCNT     ___ ACCTDATE    ___ ACCTDD      ___ ACCTDESC
___ ACCTDIV     ___ ACCTMM      ___ ACCTNO      ___ ACCTNO1     ___ ACCTNO2
___ ACCTYY      ___ ACTAMTAC    ___ ACTAMTBD    ___ ACTCHGMM    ___ ACTCHGPD

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-14 Entering Summary Specifications

Figure 2-15 (which displays after you press Enter) shows that the name of the grouping field (PROFGRP) which now appears corresponding to the rows for Total and Count.

```
Aprqqm2                Computer Associates - Quick Query
PR Query: FINANCEP     DataView: FINANCE                Report Width:  66
  Title 1: PROFIT CENTER SUMMARY
        2: BUDGET ACTUALS
Report Fields - Find: %                              Summary Only? N  More:
Summary     Group  1    Group  2    Order  3    Column 4      Column 5
        ___ PROFGRP ___ PROFDIV ___ PROFDEP ___ ACCTMM    ___ ACTAMTAC
  Total             ___          ___                      G2_ PROFDIV
  Count             ___          ___                      G2_ PROFDIV
    Max             ___          ___                      ___
    Min             ___          ___                      ___
    Avg             ___          ___                      ___
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1
                 L2
                 L3
                 L4
Available Fields    - Like: AC%                                  More:  +
___ ACCOUNT     ___ ACCTCNT     ___ ACCTDATE    ___ ACCTDD      ___ ACCTDESC
___ ACCTDIV     ___ ACCTMM      ___ ACCTNO      ___ ACCTNO1     ___ ACCTNO2
___ ACCTYY      ___ ACTAMTAC    ___ ACTAMTBD    ___ ACTCHGMM    ___ ACTCHGPD

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-15 Entering Summary Specifications

Scrolling

Place the cursor in the Field and Summary window first. Use the standard F8 (forward) key and F7 (backward) key to scroll columns in the Field and Summary window which you specify, but are not presently showing in the window.

Likewise, use the Forward and Backward keys to display the list of names in the Point and Pick window by placing the cursor in that window.

Using the More: Field

The More: field displays:

More: +	"+" if there are more columns or names in the forward direction.
More: -	"-" if you reached the end in the forward direction and there are now more columns or fields in the backward direction.
More: - +	Both "-" and "+" indicate that you are in the middle and there is more information available by scrolling either backward or forward.
More:	A blank in a More: field indicates that only the fields or columns shown are available; there is no need to scroll the window.

A confirmation message (LK03** The Point And Pick Window has been scrolled Forward.) appears in the Message line to tell you what was scrolled during your last interaction (Figure 2-16).

```
Aprqqm2                Computer Associates - Quick Query
PR Query: FINANCEP  24   DataView: FINANCE                Report Width:  66
  Title 1: PROFIT CENTER SUMMARY
        2: BUDGET ACTUALS
Report Fields - Find: %                          Summary Only? N  More:
Summary     Group  1     Group  2     Order  3      Column 4      Column 5
        ___ PROFGRP ___   PROFDIV ___   PROFDEP ___   ACCTMM     ___ ACTAMTAC
  Total             ___           ___           ___              G2  PROFDIV
  Count             ___           ___           ___              G2  PROFDIV35
    Max             ___           ___           ___              ___
    Min             ___           ___           ___              ___
    Avg             ___           ___           ___              ___
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1
                 L2
                 L3
                 L4
Available Fields    - Like: %                                    More: - +
___ ACTCHGPD    ___ ACTCHGYY    ___ DATECNT     ___ DESCACNO    ___ DESCDEP
___ DESCDIV     ___ DESCDV      ___ DESCGRP     ___ DESCPDCD    ___ PRODCODE
___ PRODCOD1    ___ PROFDEP     ___ PROFDES     ___ PROFDIV     ___ PROFGRP
LK03** The Point And Pick Window has been scrolled Forward.
Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-16 A Quick Query Message

Number of Columns and Column Wrap

You can have from 1 through 99 columns on a report. The only restriction is a practical one — columns wrap if they cannot fit on a single line. Column headings also wrap.

- Make sure that all the columns selected will fit on one logical page.
- No one field can be wider than a line.
- The Report Width: field is a simple guideline — compare this with the width of your panel or printer. When adjusting selections to fit lines on a page, remember to include the three lines of column heading, as well as the title lines in your count.

Pattern Matching with the Find: Field

You can use pattern matching as a shortcut to paging and scrolling.

- To search for columns, use the Find: field located just below the two title lines.
- To search for a specific field name or column number in the Field and Summary window, enter that value into the Find: field and press the appropriate function key (F8 for forward and F7 for backward).

Figure 2-17 shows the value 7 entered in the Find: field to make the column 7 specifications appear in the Field and Summary window.

```
Aprqqm2               Computer Associates - Quick Query
PR Query: FINANCE4     DataView: FINANCE                Report Width: 147
  Title 1: PROFIT CENTER BUDGETS VS ACTUAL - AUGUST
        2: DIRECT LABOR & MATERIALS - PRODUCT SUMMARY
Report Fields - Find: 7%                        Summary Only? N  More:   +
Summary     Group  1    Group  2    Group  3     Column 4       Column 5
          ___ ACCTDIV ___ PRODCODE___ ACCTNO1    ___ ACTAMTBD   ___ ACTAMTAC
  Total                                        G2  PRODCODE   G2  PRODCODE
  Count                                        ___            ___
    Max                                        G2  PRODCODE   G2  PRODCODE
    Min                                        ___            ___
    Avg                                        ___            ___
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1           ACCTMM     EQ   08
                 L2    AND    ACCTNO1    EQ   5150
                 L3    OR     ACCTMM     EQ   08
                 L4    AND    ACCTNO1    EQ   5510
Available Fields    - Like: %                                   More:   +
___ ACCOUNT     ___ ACCTCNT     ___ ACCTDATE    ___ ACCTDD      ___ ACCTDESC
___ ACCTDIV     ___ ACCTMM      ___ ACCTNO      ___ ACCTNO1     ___ ACCTNO2
___ ACCTYY      ___ ACTAMTAC    ___ ACTAMTBD    ___ ACTCHGMM    ___ ACTCHGPD

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-17 Using the Find: Field

Figure 2-18 shows the results after pressing F8. Column 7 becomes the leftmost column in the Field and Summary window.

Note also that a confirmation message (LK11** The Report Window has been repositioned to the specified Column Number.) appears in the Message line informing you what happened.

```
Aprqqm2               Computer Associates - Quick Query
PR Query: FINANCE4     DataView: FINANCE                Report Width: 147
  Title 1: PROFIT CENTER BUDGETS VS ACTUAL - AUGUST
        2: DIRECT LABOR & MATERIALS - PRODUCT SUMMARY
Report Fields - Find: 7%                        Summary Only? N  More: -
Summary     Column 7     Column 8     Column 9     Column 10     Column 11
          ___ ACCTDESC
  Total ___
  Count ___
    Max ___
    Min ___
    Avg ___
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1           ACCTMM     EQ   08
                 L2    AND    ACCTNO1    EQ   5150
                 L3    OR     ACCTMM     EQ   08
                 L4    AND    ACCTNO1    EQ   5510
Available Fields    - Like: %                                   More:   +
___ ACCOUNT     ___ ACCTCNT     ___ ACCTDATE    ___ ACCTDD      ___ ACCTDESC
___ ACCTDIV     ___ ACCTMM      ___ ACCTNO      ___ ACCTNO1     ___ ACCTNO2
___ ACCTYY      ___ ACTAMTAC    ___ ACTAMTBD    ___ ACTCHGMM    ___ ACTCHGPD
LK11** The Report Window has been repositioned to the specified Column Number.
Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-18 Using the Find: Field

Likewise, you could enter the column name ACCTDESC in the Find: field, press F8, and obtain the same result.

```
Aprqqm2                Computer Associates - Quick Query
PR Query: FINANCE4     DataView: FINANCE                Report Width: 147
  Title 1: PROFIT CENTER BUDGETS VS ACTUAL - AUGUST
        2: DIRECT LABOR & MATERIALS - PRODUCT SUMMARY
Report Fields - Find: ACCTDESC%                 Summary Only? N  More: -
Summary     Column 7     Column 8     Column 9     Column 10    Column 11
          ___ ACCTDESC  Total ___
  Count ___
    Max ___
    Min ___
    Avg ___
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1           ACCTMM     EQ   08
                 L2    AND    ACCTNO1    EQ   5150
                 L3    OR     ACCTMM     EQ   08
                 L4    AND    ACCTNO1    EQ   5510
Available Fields    - Like: %                                  More:   +
___ ACCOUNT     ___ ACCTCNT     ___ ACCTDATE    ___ ACCTDD      ___ ACCTDESC
___ ACCTDIV     ___ ACCTMM      ___ ACCTNO      ___ ACCTNO1     ___ ACCTNO2
___ ACCTYY      ___ ACTAMTAC    ___ ACTAMTBD    ___ ACTCHGMM    ___ ACTCHGPD
LK10** Next occurrence of a field matching the Find pattern has been found.
Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-19 Using the Find: Field

Search Characters

You can enter two special characters in the Find: field and Like: field. They are:

_	The underline means any single character.
%	The percent sign means any set of characters.

Search Patterns

You can have the following patterns:

C%	Means C followed by anything (such as all names beginning with the letter C).
R_ _	Means R followed by any two characters (such as all 3-lettered names beginning with the letter R).
%Q%	Means any characters, Q, then any characters (such as, any name with a Q in it).

You can enter a pattern of the above symbols and press either F8 (search forward) or F7 (search backward) to position on the first occurrence of a column that matches the search pattern. Each successive press of the function key retrieves the next column matching the pattern.

When you reach the end, past column 99, further entries wrap around the list in a circular fashion beginning again at column 1. A confirmation message explaining what you have retrieved then displays in the Message line.

In a similar fashion, you can control scrolling in the Point and Pick window by the use of the Like: field.

Note that the Point and Pick window displays a list of available data views, query names, or field names in alphabetical order.

The same special pattern characters are used with the Like: field as are used with the Find: field. With the Like: field, however, the pattern is used as a filter and only the names that match the pattern appear in the window.

Figure 2-20 shows that only those field names beginning with the letters “ACCT" were selected by the pattern ACCT%.

```
Aprqqm2                Computer Associates - Quick Query
PR Query: FINANCE3     DataView: FINANCE                  Report Width:  76
  Title 1: PROFIT CENTER BUDGETS VS ACTUAL - AUGUST
        2: DIRECT LABOR AND DIRECT MATERIALS
Report Fields - Find: %                          Summary Only? N  More:
Summary     Group  1    Group  2     Column 3      Column 4      Column 5
         ___ ACCTNO1 ___ ACCTDIV ___  ACTAMTBD ___  ACTCHGPD ___  ACTAMTAC
  Total                          G1   ACCTNO1   G1  ACCTNO1   G2  ACCTDIV
  Count                          ___            ___            ___
    Max                          ___            ___            ___
    Min                          ___            ___            ___
    Avg                          G2  ACCTDIV    ___            G2  ACCTDIV
Data Selection - Seq  AND/OR  Field   Oper  Value
                 L1           ACCTMM   EQ   08
                 L2    AND    ACCTNO1  EQ   5150
                 L3    OR     ACCTMM   EQ   08
                 L4    AND    ACCTNO1  EQ   5510
Available Fields    - Like: ACCT%                              More:
___ ACCTCNT     ___ ACCTDATE     ___ ACCTDD      ___ ACCTDESC    ___ ACCTDIV
___ ACCTMM      ___ ACCTNO       ___ ACCTNO1     ___ ACCTNO2     ___ ACCTYY

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-20 Using the Like: Field

Data Selection Specification

You can enter additional selection criteria in the form of AND/OR logical expressions in the Data Selection area.

Figure 2-21 shows a request for information from the FINANCE database. The selection is for records belonging to division 23 and only those whose budget is greater than 5000.

Data Selection area

```
Aprqqm2                Computer Associates - Quick Query
PR Query: FINANCE3     DataView: FINANCE                 Report Width:  76
  Title 1: PROFIT CENTER BUDGETS VS ACTUAL - AUGUST
        2: DIRECT LABOR AND DIRECT MATERIALS
Report Fields - Find: %                        Summary Only? N  More:
Summary     Group 1     Group 2     Column 3     Column 4     Column 5
          ___ ACCTNO1   ___ ACCTDIV ___ ACTAMTBD ___ ACTCHGPD ___ ACTAMTAC
  Total                           G1  ACCTNO1  G1  ACCTNO1  G2  ACCTDIV
  Count                           ___          ___          ___
    Max                           ___          ___          ___
    Min                           ___          ___          ___
    Avg                           G2  ACCTDIV  ___          G2  ACCTDIV

Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1           ACCTDIV    EQ    23
                 L2     AND   ACTAMTBD   GT    5000
                 L3
                 L4
Available Fields    - Like: %                                  More:  +
___ ACCOUNT     ___ ACCTCNT     ___ ACCTDATE    ___ ACCTDD      ___ ACCTDESC
___ ACCTDIV     ___ ACCTMM      ___ ACCTNO      ___ ACCTNO1     ___ ACCTNO2
___ ACCTYY      ___ ACTAMTAC    ___ ACTAMTBD    ___ ACTCHGMM    ___ ACTCHGPD

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-21 Entering Logical Expressions

You can enter up to four logic statements in the Data Selection area.

- Each is of the form “compare this field name to this value."
- Each of the logic statements is connected to the previous one by a logic connector (OR by default).
- You can change OR to AND by typing over it.

Select field names from those in the Point and Pick window by placing an L followed by a number (1, 2, 3, or 4) in the data entry field to the left of the field you will use to make the comparison. Figure 2-22 shows an L1 in front of the field PROFDIV.

```
Aprqqm2                Computer Associates - Quick Query
PR Query: FINANCE3     DataView: FINANCE                Report Width:  76
  Title 1: PROFIT CENTER BUDGETS VS ACTUAL - AUGUST
        2: DIRECT LABOR AND DIRECT MATERIALS
Report Fields - Find: %                          Summary Only? N  More:
Summary     Group  1    Group  2    Column 3      Column 4      Column 5
        ___ ACCTNO1  ___ ACCTDIV ___ ACTAMTBD ___ ACTCHGPD ___ ACTAMTAC
  Total                          G1  ACCTNO1   G1  ACCTNO1   G2  ACCTDIV
  Count                          ___           ___           ___
    Max                          ___           ___           ___
    Min                          ___           ___           ___
    Avg                          G2  ACCTDIV   ___           G2  ACCTDIV
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1
                 L2
                 L3
                 L4
Available Fields    - Like: PROF%                              More:
___ PROFDEP     ___ PROFDES     L1_ PROFDIV     ___ PROFGRP     ___ PROFNO


Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-22 Entering Logical Expressions

Figure 2-23 shows the partial statement built by pressing Enter. Notice that the first statement (L1) does not need a connector. Notice also that the EQ operator has been supplied in the Oper field automatically.

```
Aprqqm2         Computer Associates - Quick Query
PR Query: FINANCE3     DataView: FINANCE                Report Width:  76
  Title 1: PROFIT CENTER BUDGETS VS ACTUAL - AUGUST
        2: DIRECT LABOR AND DIRECT MATERIALS
Report Fields - Find: %                          Summary Only? N  More:
Summary     Group  1    Group  2    Column 3      Column 4      Column 5
        ___ ACCTNO1 ___ ACCTDIV ___ ACTAMTBD ___ ACTCHGPD ___ ACTAMTAC
  Total                          G1  ACCTNO1   G1  ACCTNO1   G2  ACCTDIV
  Count                          ___           ___           ___
    Max                          ___           ___           ___
    Min                          ___           ___           ___
    Avg                          G2  ACCTDIV   ___           G2  ACCTDIV
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1           PROFDIV    EQ
                 L2
                 L3
                 L4
Available Fields    - Like: PROF%                              More:
___ PROFDEP     ___ PROFDES     ___ PROFDIV     ___ PROFGRP     ___ PROFNO

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-23 Entering Logical Expressions

The set of operator values and their meanings are:

EQ — equal	GT — greater than
GE — greater than or equal to	LE — less than or equal to
NE — not equal	LT — less than

If you need a different operator, type over the supplied value of EQ with the appropriate value from the above list.

Each line in the Data Selection area is protected from data entry until the comparison field is selected from the Point and Pick window. The process for building the data selection criteria is:

1. Select one or more comparison fields from the Point and Pick window in one interaction (press Enter).
2. Supply the values corresponding to each.
3. Change any logical AND/OR connector or comparison operator as appropriate.
4. Press Enter.

The system prompts you with a diagnostic message if you do not supply a value before going on to scroll the Point and Pick window to get another comparison field.

Summarizing Fields

Five types of summaries are available. Each type specifies a calculation to be made over a group of records.

Gn Operator

Specify which fields to use as control breaks by selecting them with a Gn operator from the Point and Pick window.

The five summaries available with Quick Query are:

Total	—	The algebraic sum of a group of numeric fields.
Count	—	The number of detail records within the group.
Max	—	The maximum value of that column within the group.
Min	—	The minimum value of that column within the group.

Avg — The average value of a numeric field.

You can specify any or all of these types with summaries for a column. You can specify each to be taken over the same or a different group. Each of the values is output to your report when a change in the value of the “group by" field is detected.

Figure 2-24 shows the specifications for the report in Figure 2-25.

```
Aprqqm2                Computer Associates - Quick Query
PR Query: FINANCE4     DataView: FINANCE                  Report Width:  70
  Title 1: PROFIT CENTER BUDGETS VS ACTUAL - SEPTEMBER
        2: REVENUES - PRODUCT SUMMARY
Report Fields - Find: %                          Summary Only? N  More:
Summary     Group  1     Group  2     Group  3     Column 4      Column 5
        ___ ACCTDIV   ___ PRODCODE ___ ACCTNO1    ACTAMTBD      ACTAMTAC
  Total                                          G2 PRODCODE   G2 PRODCODE
  Count
    Max                                          G2 PRODCODE   G2 PRODCODE
    Min                                          ___           ___
    Avg                                          ___           G3 ACCTNO1
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1           ACCTMM     EQ   09
                 L2    AND    ACCTNO1    EQ   0000
                 L3
                 L4
Available Fields    - Like: %                                    More:   +
___ ACCOUNT     ___ ACCTCNT     ___ ACCTDATE    ___ ACCTDD       ___ ACCTDESC
___ ACCTDIV     ___ ACCTMM      ___ ACCTNO      ___ ACCTNO1      ___ ACCTNO2
___ ACCTYY      ___ ACTAMTAC    ___ ACTAMTBD    ___ ACTCHGMM     ___ ACTCHGPD

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-24 Entering Field Summaries

```
MAR 22, 2001      PROFIT CENTER BUDGETS VS ACTUAL - SEPTEMBER                PAGE 1
                          REVENUES - PRODUCT SUMMARY

-------------------------------------------------------------------------------
                ACCOUNT   PRODUCT  ACCOUNT     ACCOUNT        ACCOUNT
                DIVISION   CODE    NUMBER   AMOUNT BUDGET  AMOUNT ACTUAL
-------------------------------------------------------------------------------
                   21      1011     0000          20,000         19,542

ACCTNO1  AVG.      21      1011     0000                         19,542

PRODCODE TOTAL     21      1011                   20,000         19,542
         MAX.                                     20,000         19,542
         AVG.                                                    19,542

                           1012     0000          10,000          9,123

ACCTNO1  AVG.      21      1012     0000                          9,123
PRODCODE TOTAL     21      1012                   10,000          9,123
         MAX.                                     10,000          9,123
         AVG.                                                     9,123

P/N
```

Figure 2-25 A Report With Summaries (Page 1 of 2)

```
/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\//\/\/\/\/\/
MAR 22, 2001      PROFIT CENTER BUDGETS VS ACTUAL - SEPTEMBER                PAGE 8
                         REVENUES - PRODUCT SUMMARY

-------------------------------------------------------------------------------------
                ACCOUNT   PRODUCT  ACCOUNT     ACCOUNT        ACCOUNT
                DIVISION    CODE    NUMBER   AMOUNT BUDGET  AMOUNT ACTUAL
-------------------------------------------------------------------------------------
GRAND    TOTAL                                   118,500        124,036
         MAX.                                     20,000         23,968
         AVG.                                                    11,276
```

Figure 2-25   A Report With Summaries (Page 2 of 2)

Notice that there is a grand summary at the end of the report. If you request a "summary only" report by specifying a Y in the Summary Only? field (located just below the two title lines in Figure 2-24), then only the summary lines display and none of the detail.

Figure 2-26 is the corresponding Summary Only report corresponding to Figure 2-25.

```
MAR 22, 2001      PROFIT CENTER BUDGETS VS ACTUAL - SEPTEMBER      PAGE 1
                          REVENUES - PRODUCT SUMMARY

---------------------------------------------------------------------------
                ACCOUNT   PRODUCT  ACCOUNT     ACCOUNT         ACCOUNT
                DIVISION   CODE    NUMBER   AMOUNT BUDGET  AMOUNT ACTUAL
---------------------------------------------------------------------------
ACCTNO1  AVG.      21       1011     0000                         19,542

PRODCODE TOTAL     21       1011                  20,000          19,542
         MAX.                                     20,000          19,542
         AVG.                                                     19,542

ACCTNO1  AVG.      21       1012     0000                          9,123

PRODCODE TOTAL     21       1012                  10,000           9,123
         MAX.                                     10,000           9,123
         AVG.                                                      9,123

ACCTNO1  AVG.      21       1013     0000                          5,469
PRODCODE TOTAL     21       1013                   5,500           5,469
         MAX.                                      5,500           5,469

P/N

/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/

MAR 22, 2001      PROFIT CENTER BUDGETS VS ACTUAL - SEPTEMBER      PAGE 6
                          REVENUES - PRODUCT SUMMARY

---------------------------------------------------------------------------
                ACCOUNT   PRODUCT  ACCOUNT     ACCOUNT         ACCOUNT
                DIVISION   CODE    NUMBER   AMOUNT BUDGET  AMOUNT ACTUAL
---------------------------------------------------------------------------
GRAND    TOTAL                                   118,500         124,036
         MAX.                                     20,000          23,968
         AVG.                                                     11,276
```

Figure 2-26   A Summary Only Report With Grand Summaries

Modifying Queries and Making Changes

You can type over any specification, except those created by selecting from the Point and Pick window. You can choose another field and replace one already chosen by reusing the specific column specification with a different field.

For example, you could change the summary groupings for column 5 by PRODCODE to groupings by PROFNO simply by selecting PROFNO from the Point and Pick window with a G2 (see Figure 2-27).

```
Aprqqm2                Computer Associates - Quick Query
PR Query: FINANCE4     DataView: FINANCE               Report Width:  70
  Title 1: PROFIT CENTER BUDGETS VS ACTUAL - AUGUST
        2: DIRECT LABOR & MATERIALS - PRODUCT SUMMARY
Report Fields - Find: %                        Summary Only? N  More:
Summary     Group 1     Group 2      Group 3     Column 4      Column 5
        ___ ACCTDIV ___ PRODCODE ___ ACCTNO1 ___ ACTAMTBD  ___ ACTAMTAC
  Total                                      G2  PRODCODE  G2  PRODCODE
  Count                                      ___           ___
    Max                                      G2  PRODCODE  G2  PRODCODE
    Min                                      ___           ___
    Avg                                      ___           ___
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1           ACCTMM     EQ   08
                 L2    AND    ACCTNO1    EQ   5150
                 L3    OR     ACCTMM     EQ   08
                 L4    AND    ACCTNO1    EQ   5510
Available Fields    - Like: PR%                              More:
___ PRODCODE    ___ PRODCOD1    ___ PROFDEP    ___ PROFDES    ___ PROFDIV
___ PROFGRP     g2_ PROFNO

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-27 Changing Summary Grouping

Notice that all the other columns in the report grouped by PRODCODE now show grouped by PROFNO (see Figure 2-28).

```
Aprqqm2                Computer Associates - Quick Query
PR Query: FINANCE4     DataView: FINANCE              Report Width:  73
  Title 1: PROFIT CENTER BUDGETS VS ACTUAL - AUGUST
        2: DIRECT LABOR & MATERIALS - PRODUCT SUMMARY
Report Fields - Find: %                        Summary Only? N More:
Summary     Group 1     Group 2     Group 3      Column 4     Column 5
          ___ ACCTDIV ___ PROFNO  ___ ACCTNO1  ___ ACTAMTBD ___ ACTAMTAC
  Total                                        G2  PROFNO   G2  PROFNO
  Count                                        ___          ___
    Max                                        G2  PROFNO   G2  PROFNO
    Min                                        ___          ___
    Avg                                        ___          ___
Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1           ACCTMM     EQ   08
                 L2    AND    ACCTNO1    EQ   5150
                 L3    OR     ACCTMM     EQ   08
                 L4    AND    ACCTNO1    EQ   5510
Available Fields    - Like: P%                                  More:
___ PRODCODE    ___ PRODCOD1    ___ PROFDEP    ___ PROFDES    ___ PROFDIV
___ PROFGRP     ___ PROFNO

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-28 Changing Summary Grouping

Rearranging Column Numbers

To rearrange column numbers, use the following specifications:

Xnn Place in the input area in front of the field name designated for a column. Switches column n with the current column.

Mnn Same placement as described above. Moves the current column on top of column n.

D When a D is placed over the Ln in the Data Selection or the Gn summary specification, the current line specification is deleted.

You can make these changes any time you have the query loaded in the Quick Query panel. Quick Queries are stored internally in a different format from those created with the Source Processing panel (Option 6 from the Main Menu).

Changing a Quick Query with the Full Screen Editor

You can change a Quick Query through the Full Screen Editor in Option 6, but once you do this, it is no longer available as a Quick Query.

Note: Once you change a Quick Query with the Full Screen Editor, it is no longer considered a Quick Query item. Only Quick Query items show in the Point and Pick window query list.

You can do this, however, to take advantage of the FORMAT statements or provide more data selection specification than is available through the Quick Query panel. Note that you can change the comment lines which are the sample FORMAT statement in the query to be processed.

```
Aprqqm2                Computer Associates - Quick Query
PR Query: FINANCE4     DataView: FINANCE              Report Width:  73
  Title 1: PROFIT CENTER BUDGETS VS ACTUAL - AUGUST
        2: DIRECT LABOR & MATERIALS - PRODUCT SUMMARY
Report Fields - Find: %                        Summary Only? N More:
Summary     Group 1     Group 2     Group 3     Column 4      Column 5
        ___ ACCTDIV ___ PROFNO  ___ ACCTNO1 ___ ACTAMTBD  ___ ACTAMTAC
  Total                                     G2  PROFNO    G2  PROFNO
  Count                                     __            __
    Max                                     G2  PROFNO    G2  PROFNO
    Min                                     ___           ___
    Avg                                     __            ___
Data Selection - Seq  AND/OR  Field   Oper  Value
                 L1           ACCTMM   EQ   08
                 L2    AND    ACCTNO1  EQ   5150
                 L3    OR     ACCTMM   EQ   08
                 L4    AND    ACCTNO1  EQ   5510
Available Fields    - Like: P%                              More:
___ PRODCODE    ___ PRODCOD1    ___ PROFDEP    ___ PROFDES    ___ PROFDIV
___ PROFGRP     ___ PROFNO

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-29 A Quick Query

Figure 2-30 shows the actual query built through the Quick Query panel in Figure 2-29.

```
Editor2 ----- Computer Associates - Editor ---- Name: FINANCE4 Type: QUERY
                                                              More:     +

000100 QUERY DATABASE FINANCE
000200 REPORT ACCTDIV PROFNO ACCTNO1 ACTAMTBD ACTAMTAC ,
000300 IF ACCTMM EQ '08' AND ACCTNO1 EQ '5150' OR ACCTMM EQ '08' AND ACCTNO1
000400 EQ '5510' GRANDSUM
000500 GROUP BY ACCTDIV PROFNO ACCTNO1
000600 ORDER BY ACCTDIV PROFNO ACCTNO1
000700 TOTAL ACTAMTBD BY PROFNO
000800 MAX ACTAMTBD BY PROFNO
000900 TOTAL ACTAMTAC BY PROFNO
001000 MAX ACTAMTAC BY PROFNO
001100 TITLE  'PROFIT CENTER BUDGETS VS ACTUAL - AUGUST'
001200 TITLE  'DIRECT LABOR & MATERIALS - PRODUCT SUMMARY'
001300 ; FORMAT HEIGHT  WIDTH  DATEPOS {UL|UR|UM|LL|LR|LM|NO},
001400 ; PAGEPOS {UR|UL|UM|LL|LR|LM|NO},
001500 ; TITLEPOS {TOP|BOTTOM},
001600 ; LABELS {SUPPRESS|NOSPACE|SPACE},
001700 ; HEADINGS {YES|NO},
001800 ; DATEFMT {TODAY|DATE|ISDATE|JULIAN},
001900 ; LINES {0|1|2|3|4|5|6|7|8} BORDER '-' PAGEGRP {YES|NO}
002000 END QUERY

Command ===>
F1 =Help     F3 =Exit     F5 =Rfind    F6 =Rchange  F7 =Backward F8 =Forward
F10=Left     F11=Right    F12=Cancel
```

Figure 2-30 Query Built Through the Quick Query Panel

```
Editor2 ------ Computer Associates - Editor ----- Name: FINANCE4 Type: QUERY
                                                              More:

000100 QUERY DATABASE FINANCE
000200 REPORT ACCTDIV PROFNO ACCTNO1 ACTAMTBD ACTAMTAC ,
000300 IF ACCTMM EQ '08' AND ACCTNO1 EQ '5150' OR ACCTMM EQ '08' AND ACCTNO1
000400 EQ '5510' GRANDSUM
000500 GROUP BY ACCTDIV PROFNO ACCTNO1
000600 ORDER BY ACCTDIV PROFNO ACCTNO1
000700 TOTAL ACTAMTBD BY PROFNO
000800 MAX ACTAMTBD BY PROFNO
000900 TOTAL ACTAMTAC BY PROFNO
001000 MAX ACTAMTAC BY PROFNO
001100 TITLE  'PROFIT CENTER BUDGETS VS ACTUAL - AUGUST'
001200 TITLE  'DIRECT LABOR & MATERIALS - PRODUCT SUMMARY'
001300 FORMAT  DATEPOS UR,
001400 PAGEPOS LM,
001500 TITLEPOS TOP,
001800 DATEFMT JULIAN
002000 END QUERY

Command ===>
F1 =Help     F3 =Exit     F5 =Rfind    F6 =Rchange  F7 =Backward F8 =Forward
F10=Left     F11=Right    F12=Cancel
```

Figure 2-31 The Changed Query

Figure 2-31 shows the changed query (by way of the Source Processing panel). Figure 2-32 shows the changes in the report.The Command Line

```
                PROFIT CENTER BUDGETS VS ACTUAL - AUGUST        01.082
              DIRECT LABOR & MATERIALS - PRODUCT SUMMARY

-------------------------------------------------------------------------------
                ACCOUNT     PROFIT    ACCOUNT     ACCOUNT        ACCOUNT
                DIVISION  CENTER NO.  NUMBER   AMOUNT BUDGET  AMOUNT ACTUAL

-------------------------------------------------------------------------------
                   21      10213200    5150          2,100          5,143
                                                     2,100          1,795
                                                     2,100          1,057
                                                     2,100          1,131

PROFNO   TOTAL     21      10213200                  8,400          9,126
         MAX.                                        2,100          5,143

                           10213300    5510          3,100          5,934
                                                     3,100          2,693
                                                     2,700          1,586
                                                     3,600          2,035

PROFNO   TOTAL     21      10213300                 12,500         12,248
         MAX.                                        3,600          5,934

                                PAGE 1
P/N
```

Figure 2-32 The Resulting Quick Query Report

The function keys shown at the bottom of the Quick Query panel are the shortcut single-key equivalents to typing the commands on the Command line.

The following function keys are the default function key assignments delivered with your system. The default function keys and their explanations follow:

F1	Help	Displays a context sensitive Help panel based on the cursor location.
F3	Exit	Exits the Quick Query panel and saves the query if it validates.
F7	Backward	Browses backward in the window where your cursor is located.
F8	Forward	Browses forward in the window where your cursor is located.
F12	Cancel	Discontinues work on this panel and returns to the Main Menu. F12 is the default for CICS
F24	Cancel	F24 is the default Cancel key for IMS.

In addition, there are four commands that have no corresponding default function key equivalents and must be typed on the Command line.

CLEAR — The CLEAR command clears all input fields and returns to the initial conditions for the panel.

RENAME — The RENAME command has no operands. When you enter RENAME in the Command line, the current query name is blanked out, and you can re-enter a different name. Subsequently, that name becomes the name of the query. This does not affect the query if it is already saved. The next time you save the query, it will have the new name.

SAVE — The SAVE command validates and saves the query without exiting from the Quick Query panel.

SUBMIT — The SUBMIT command displays the system Submit panel (see Figure 2-33).

You supply the appropriate information and press F3. This submits the query for processing.

For more information on how to use this panel, see the section Submit Panel. When you exit the Submit panel, you return to the Quick Query panel.

```
Submit2                    Computer Associates - Submit
                           Name: FINANCE4 Type: QUERY   OwnerId: SYSTEM

 To Submit the query enter EXIT
 To leave the panel without submitting the query enter CANCEL

Class                              10          (Enter  1-14 )
Maximum number of roots            ________    (Enter 1-99999999)
Maximum number of lines            ____        (Enter 1-9999)
Route to batch printer?            _           (Y)
Format for logical terminal        ________    (Terminal Id)
Generate query source with report  _           (Y)
Output Format                      1           (1 - Standard Report
                                                2 - HTML
                                                3 - Tab Delimited
                                                4 - Comma Delimited
                                                5 - Plain Text)
Override Query Output Format       _           (Y)



Command ===>
F1 =Help     F3 =Exit     F12=Cancel
```

Figure 2-33 The Submit Panel

Quick Query Immediate Response

Note: You start Quick Query Immediate Response by selecting Option 5 from the Main Menu.

Figure 2-34 shows a Quick Query panel for an Immediate Response query.

```
Aprqqm2                Computer Associates - Quick Query
IR Query: FINANCIR     DataView: FINANCE               Report Width: 53
  Title 1: THIS IS AN EXAMPLE OF AN IMMEDIATE RESPONSE QUERY
        2: THE INFORMATION IS REQUESTED FROM THE FINANCE DATABASE
Report Fields - Find: %                                        More:
           Column 1     Column 2     Column 3     Column 4     Column 5
       ___ PROFGRP  ___ PRODCODE ___ PROFDEP  ___ ACCTDATE ___ ACTAMTBD



Data Selection - Seq  AND/OR  Field     Oper  Value
                 L1
                 L2
                 L3
                 L4
Available Fields    - Like: %                                  More:   +
___ ACCTCNT     ___ ACCTDATE    ___ ACCTDD      ___ ACCTDESC    ___ ACCTDIV
___ ACCTMM      ___ ACCTNO      ___ ACCTNO1     ___ ACCTNO2     ___ ACCTYY
___ ACTAMTAC    ___ ACTAMTBD    ___ ACTCHGMM    ___ ACTCHGPD    ___ ACTCHGYY

Command ===>
F1 =Help     F3 =Exit     F7 =Backward F8 =Forward  F12=Cancel
```

Figure 2-34 The Quick Query Immediate Response Panel

Note: You can obtain a hard copy of an Immediate Response Query by issuing the RUN command for a stored Immediate Response Quick Query in the Batch Simulator input stream.

You can only run Immediate Response queries using the RUN command from Option 7 (Immediate Response Query Processing) in the Main Menu.

Immediate Response queries are designed to instantaneously retrieve small amounts of database information. When you run an Immediate Response query, your terminal waits until the information you requested is ready and then displays it at your terminal.

Figure 2-35 shows the command to run a Quick Query.

```
?:
run financir
_
_
_
_
_
_
_
_
```

Figure 2-35 Running a Query

Figure 2-36 shows the Quick Query Immediate Response report.

```
             THIS IS AN EXAMPLE OF AN IMMEDIATE RESPONSE QUERY
         THE INFORMATION IS REQUESTED FROM THE FINANCE DATABASE
PROFGRP         PRODCODE         PROFDEP       ACCTDATE          ACTAMTBD
  10              1011             3100         013192             20,000
                                                022892             20,000
                                                033192             20,000
                                                043092             20,000
                                                053192             20,000
                                                063092             20,000
                                                073192             20,000
                                                083192             20,000
                                                093092             20,000
                                                103192             20,000
                                                113092             20,000
                                                123192             20,000
                1011                            013192              1,000
                                                022892              1,000
                                                033192              1,000
                                                043092              1,000
                                                053192              1,000
P/N
/\/\/\/\/\//\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\/\
            THIS IS AN EXAMPLE OF AN IMMEDIATE RESPONSE QUERY
        THE INFORMATION IS REQUESTED FROM THE FINANCE DATABASE
PROFGRP         PRODCODE         PROFDEP       ACCTDATE          ACTAMTBD
  10              3012             3400         113092              3,200
                                                123192              3,200
                  3013                          013192              2,900
                                                022892              2,900
                                                033192              2,900
                                                043092              2,900
                                                053192              2,900
                                                063092              2,900
                                                073192              2,900
                                                083192              2,900
                                                093092              2,900
                                                103192              2,900
                                                113092              2,900
                                                123192              2,900
NR06** END OF DISPLAY, DB CALL COUNT=     13, PAGE COUNT=   39
?:
_
```

Figure 2-36 Quick Query Immediate Response Report

In CICS, use BMS paging commands to page through the results. The exact commands vary according to your installation standards.

In IMS, press the key specified in the lower left corner of the screen (PA1 or PA2) to page through the results.

Quick Query for Immediate Response is exactly the same as non-Immediate Response, except that with Immediate Response there is no summarization available.

Chapter

3

General Concepts

Before You Start

Before beginning a VISION:Bridge session, you need a user ID or user profile. If security is an issue at your installation, you will also need a password and be familiar with your user profile.

User ID and Profile

Your VISION:Inform system administrator provides you with a profile which is an internal list of the types of data available to you. The information in your profile pertains to a specific user ID. However, multiple users can log on simultaneously with the same user ID or profile. You cannot use VISION:Bridge without a user ID or profile.

Environments

VISION:Bridge runs in several different environments. To you, the environment is transparent. As you use VISION:Bridge, all commands and features work the same regardless of the environment in which they runs. What differs is how you tell the environment that you want to use VISION:Bridge. Just how you do this depends on the environment and the particular computer installation you are using. This involves entering the appropriate command at your terminal. Your VISION:Inform system administrator can help you with this.

Logon Panel

Once you tell the system that you want to use VISION:Bridge, the Logon panel displays, as shown in Figure 3-1. You can now log on to VISION:Bridge by entering your user ID and password (provided by your VISION:Inform system administrator).

```
Logon2                    Computer Associates - Logon


                  Welcome to VISION:Inform Release 4.0.

                   Please Enter Your User ID and Password:

                     User ID  . . . . ______
                     Password . . . .         (if password protected)


               +-----------------------------------------------+
               |  Proprietary and confidential information of  |
               |    Computer Associates International, Inc.    |
               |  Use restricted by written license agreement. |
               |              (c) 1980, 2001                   |
               |    Computer Associates International, Inc.    |
               |  as an unpublished work. All rights reserved. |
               +-----------------------------------------------+

Command ===>
F1 =Help     F3 =Exit
```

Figure 3-1 Logon Panel

Main Menu

After entering your user ID and password, the Main Menu displays, as shown in Figure 3-2.

```
Menu2                Computer Associates - Main Menu

_ Enter one of the following VISION:Inform or VISION:Bridge Options:



VISION:Inform Options

1. Operation Facilities                  (Background Processor Status)
2. Administration Facilities             (Profile Development)
3. Report Facilities                     (Report Handling)

VISION:Bridge Options
4. Quick Query                           (Assisted Query Development)
5. Quick Query Immediate Response        (Assisted Query Development)
6. Standard Query Processing             (Submit, Delete, Edit
                                          Queries and Stmts)
7. Immediate Response Query Processing   (Run Queries and Immed Mode)



Command ===>
F1 =Help     F12=Cancel
```

Figure 3-2 Main Menu

Use the Main Menu to select the option you to work with in this session:

- Use Options 3 through 7 for VISION:Bridge.
- The VISION:Inform system administrator uses Options 1 and 2.

Source Processing Panel

To work with queries and groups of statements, select Main Menu Option 6 (Standard Query Processing). This displays the Source Processing panel. From the Source Processing panel, you can:

- Submit queries for processing.
- Delete items from the foreground library.
- Create and update items of type QUERY and STMTS.

The Source Processing panel is shown in Figure 3-3.

```
Source2 -------------- Computer Associates - Source Processing----------------
                                                                      More:
   Name      Type     Owner    Last Used      Name     Type     Owner    Last Used
_ ACCTS     QUERY    SAVANNAH 06/15/01
_ ITEMCNT   QUERY    SAVANNAH 06/15/01
_ ITEMLST   QUERY    USER1    06/15/01
_ MONTHLY   QUERY    USER1    06/15/01
_ MYQUERY   QUERY    USER1    06/15/01
_ VVVV      QUERY    SYSTEM   06/15/01
_ WEEKLY    QUERY    USER1    06/15/01
_ PART1     STMTS    USER1    06/15/01
_ PART2     STMTS    USER1    06/15/01
_ PART3     STMTS    USER1    06/15/01
_ SPECIAL   STMTS    USER1    06/15/01
_ TEST1     STMTS    SAVANNAH 06/15/01
_ TEST2     STMTS    SAVANNAH 06/15/01


Command ===>
F1 =Help     F7 =Backward     F8 =Forward      F12=Cancel
```

Figure 3-3 Source Processing Panel

This panel displays the name of each item in the foreground library and the item type, the owner of the item (the user ID that saved the item), and the date that the item was last used.

Note: You can shorten command line entries (commands and keywords) to the shortest string which uniquely identifies the item.

An underscore precedes each item. This underscore is where you specify what you want to do with the item. The following codes are valid:

E To edit the item. This displays the Editor panel containing the specified item.

S To submit the item for processing. This displays the Submit panel where you can specify options before submitting the item for processing.

D To delete the item from the foreground library (you cannot delete an item owned by someone else). After entering D to delete an item, a prompt asks you to confirm the request by re-entering the D. This confirmation process helps prevent accidental deletions.

Code only one entry at a time.

Creating a Query or Group of Statements

To create a new item, move the cursor to the Command line and type:

CREATE name type

where the parameters are:

CREATE is the command.

name is the name of the item.

type is the item type. It can be QUERY or STMTS.

The CREATE command displays the Editor panel where you can create the specified item.

Positioning to a Specific Display

To position the display to a specific entry (QUERY/STMTS), move the cursor to the Command line and type:

LOCATE name {STMTS}

where the parameters are:

LOCATE is the command.

name is the name of the item.

STMTS is an optional parameter for STMTS type items. QUERY is the default.

Other commands available on the Source Processing panel are HELP, FORWARD, BACKWARD, and CANCEL.

- Use HELP to display a Help panel based on the location of the cursor.
- Use FORWARD to scroll the window toward the bottom of the data.
- Use BACKWARD to scroll the window toward the top of the data.
- Use CANCEL to cancel any pending commands and return to the Main Menu.

Editor Panel

Display the Editor panel by selecting an item to edit or specifying an item to create on the Source Processing panel.

- Use this panel to create or change source items.
- Scroll the data up, down, left, or right.
- Change the data by typing over the displayed data and by using line commands and primary commands.
- You enter line commands directly over the line number of the lines to be affected (see the section Line Commands).
- Enter primary commands on the Command line (see the section Primary Commands).

When you create a new item, the Editor panel is initially empty, as shown in Figure 3-4.

Editor Panel Components

The Line Command area on the left side of the panel contains . . . . . for a new item. Sequence numbers replace these marks when you enter information on the lines.

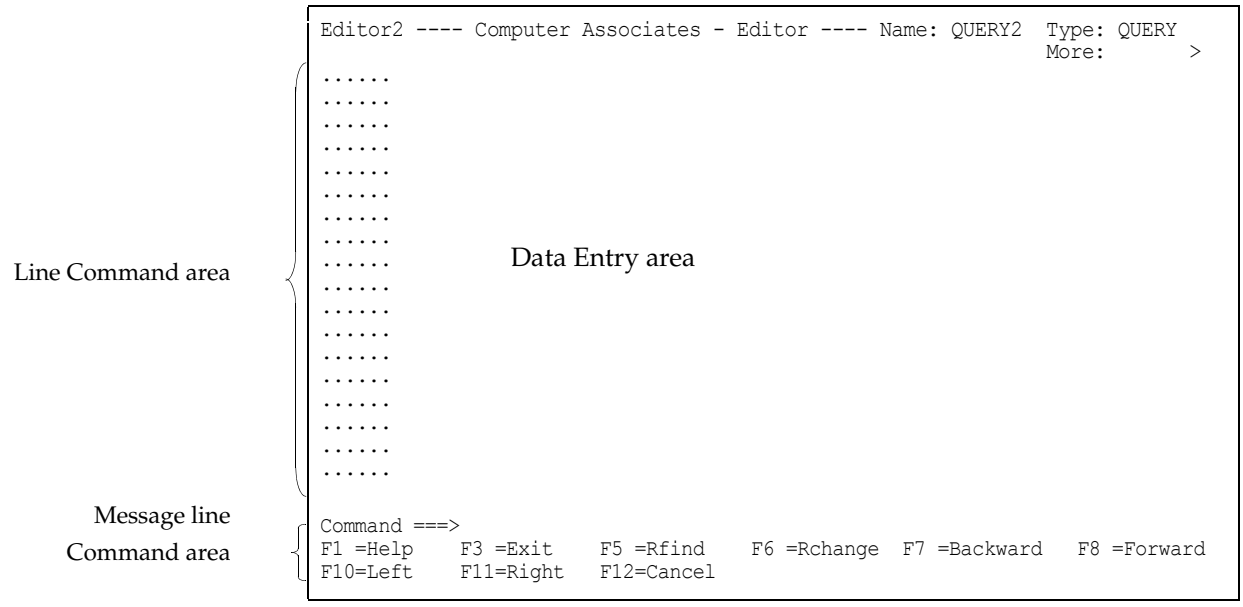


Figure 3-4 Creating a New Item

This is also where you can specify line commands. The area to the right of the Line Command area is the Data Entry area. This is where you enter the source for the item being edited.

The last four lines of the panel are the Command area. The Command area includes a message line, the Command line, and two function key lines.

- The message line displays informational and diagnostic messages during your edit session.
- You enter primary commands on the Command line. .
- Function key lines contain the function keys that are available for use on the panel.

For a description of the function keys, see the section Primary Commands. When you are editing an existing item, the Editor panel displays the source of the item and sequence numbers, as shown in Figure 3-5.

```
Editor2 ---- Computer Associates - Editor ---- Name: ITEMLST  Type: QUERY
                                                              More:       >
000100 QUERY CUSTOMER
000200 REPORT CUSTNO ORDERNO ITEMNAME
000300 TITLE 'ITEMS ORDERED'
000400 END REPORT
000500 END QUERY









Command ===>
F1 =Help    F3 =Exit    F5 =Rfind   F6 =Rchange   F7 =Backward   F8 =Forward
F10=Left    F11=Right   F12=Cancel
```

Figure 3-5 The Editor Panel

The first line of the panel contains the name of the item and the item type (QUERY or STMTS). The second line contains the scrolling information More: field, which indicates the presence of data:

< Left of the Data Entry area shown.

> Right of the Data Entry area shown.

- Above the Data Entry area shown (backward).

+ Below the Data Entry area shown (forward).

Note that blanks are considered data, and all edited items are fixed length 132 character records.

Scrolling

To view the area that is not currently shown, enter the appropriate scrolling command. The scrolling commands are: LEFT, RIGHT, BACKWARD, and FORWARD. You can also use these commands with optional parameters. For instance:

FORWARD n

Use the FORWARD command n parameter to specify how many lines forward the window should scroll. n is a number from 1-999999.

LEFT M

Use the LEFT command m parameter to scroll the window left as far as possible. M stands for maximum.

Scrolling commands are primary commands and are assigned to function keys. For a list of primary commands and applicable function keys, see the section Primary Commands.

Line Commands

Line commands affect only a single line or a block of lines. You enter line commands by typing them in the Line Command area on one or more lines and pressing Enter. Figure 3-6 lists all of the line commands along with a description.

Command	Description
A	Identifies the line after which to insert moved/copied lines. Used in conjunction with the M line command, the C line command, or the COPY primary command.
B	Identifies the line before which to insert moved/copied lines. Used in conjunction with the M line command, the C line command, or the COPY primary command.
C	Copies this line to the location specified by the A or B line command.
Cn	Copies n (where n is a number from 1 - 99999) lines to the location specified by the A or B line command.
CC	Use in pairs to delimit a block of lines to be copied to the location specified by the A or B line command.

Figure 3-6 Line Commands (Page 1 of 2)

Command	Description
D	Deletes this line.
Dn	Deletes n (where n is a number from 1 - 99999) lines.
DD	Use in pairs to delimit a block of lines to be deleted.
I	Inserts a blank line after this line.
In	Inserts n (where n is a number from 1 - 99999) blank lines after this line.
M	Moves this line to the location specified by the A or B line command.
Mn	Moves n (where n is a number from 1 - 99999) lines to the location specified by the A or B line command.
MM	Use in pairs to delimit a block of lines to be moved to the location specified by the A or B line command.
R	Repeats this line on the next line.
Rn	Repeats this line n (where n is a number from 1 - 99999) times immediately following the line on which Rn is entered.
RR	Delimits the block of lines to be repeated immediately following the second RR command.
RRn	Delimits the block of lines to be repeated n (where n is a number from 1 - 9999) times immediately following the second RRn command. (You can enter the n on either or both of the RR commands.)

Figure 3-6 Line Commands (Page 2 of 2)

Primary Commands

Primary commands affect the entire item being edited. Enter primary commands in one of the following ways:

- By entering the command on the Command line (Command ===>) and pressing Enter.
- By pressing a function key that has been defined to execute a specific command.

When you enter a command on the Command line, you need not enter the entire command. Enter only the minimum number of characters that uniquely identify the command. For example, you enter CH for the CHANGE command, but not the letter C.

The following sections describe all of the primary commands available in the VISION:Bridge Editor along with a description of each. The default function key is also shown, when applicable.

BACKWARD Primary Command (F7)

Scrolls the window toward the top of the Data Entry area. You can include optional parameters of M (maximum) or n (where n is a number from 1 - 999999).

CANCEL Primary Command (F12)

Ends the current edit session without saving the item, and returns to the calling panel.

CHANGE Primary Command

Changes the first or ALL occurrences of one text string to a second text string.

The command syntax is:

CHANGE string1 string2 (ALL)

If the cursor is in the Data Entry area, the search begins at the cursor position.

If the cursor is not in the Data Entry area, the search starts at the beginning of the Data Entry area.

string1 and string2

Use the single (') or double (") quotation marks as delimiters to surround string1 and string2. They are only necessary when you use these delimiting characters in the data or when blanks are part of either string.

- ==>CHG in the Line Command area mark changed lines.
- The lengths of string1 and string2 can be unequal.
- The lengths of string1 and string2 are limited by the length of the Command line area for your terminal.

All

If you specify ALL, all occurrences of the specified string are changed.

If you do not specify ALL, only the first occurrence changes.

COPY Primary Command

Copies a query or collection of statements from the foreground library to the location specified by the A or B line command. (The A or B line command is not required if the item being edited is empty.)

The command syntax is:

COPY name type

where name is the name of the item you want to copy, and type is either QUERY or STMTS.

CREATE Primary Command

Saves the item in the current edit session under a different name/type as specified in the command.

The command syntax is:

CREATE name type

where name is the name of the item you want to create, and type is QUERY or STMTS.

Validates the item being saved for syntax and internal consistency unless the type is STMTS.

DATAVIEW Primary Command

Displays a pop-up panel with a list of data views.

EXIT Primary Command (F3)

Saves the current item and returns to the calling panel. If there are validation errors, a message displays telling you that you must correct the errors or use the CANCEL command to exit the session.

FIELDS Primary Command

Displays a pop-up panel with a list of fields for the current data view.

FIND Primary Command

Finds the first occurrence of a text string in the item being edited.

The command syntax is:

FIND string

If the cursor is in the Data Entry area, the search begins at the cursor position.

If the cursor is not in the Data Entry area, the search starts at the beginning of the Data Entry area.

Use single (‘) or double (") quotation marks to surround the string. They are only necessary when you use these delimiting characters in the data or when blanks are part of the string.

Note: The length of the string is limited by the length of the Command line for your terminal.

FORWARD Primary Command (F8)

Scrolls the window forward toward the bottom of the data. You can include optional parameters of M (maximum) or n (where n is a number from 1 - 999999).

HELP Primary Command (F1)

Displays a Help panel based on the location of the cursor. (For example, if the cursor is on the Command line, the Help panel contains Command line information.)

LEFT Primary Command (F10)

Scrolls the window toward the left margin. You can include optional parameters of M (maximum) or n (where n is a number from 1 - 999999).

LOCATE Primary Command

Locates a specific line of data based on the parameter specified.

The command syntax is:

LOCATE n (n is 1 - 999999)
LOCATE ERR
LOCATE CHG

If you specify n and an exact match of that line number cannot be found, the line with the number closest to but less than that number will be the new first line in the Data Entry area.

If you specify ERR, the position of the cursor is on the next line with the ==>ERR (validation error) flag.

If you specify CHG, the position of the cursor is on the next line with the ==>CHG (changed data) flag.

LOCATE Primary Command Different Formats

A different format of the LOCATE primary command is used with the following panels:

- Source Processing panel (See the section LOCATE name {STMTS}.)
- Dataview panel (See the section Using the LOCATE Primary Command in the Dataview Pop-up Panel.)
- FIELDS panel (See the section Using the Fields Pop-up Panel.)

RCHANGE Primary Command (F6)

Repeats the last entered CHANGE command. (See the section CHANGE Primary Command.)

RENUMBER Primary Command

Renumbers all lines. The first line of the data is sequence number 100; the remaining lines increment by 100.

RESET Primary Command

Clears any pending line commands, messages, validation error messages, and ==>CHG indicators.

RFIND Command (F5)

Searches for the string entered in the last FIND or CHANGE command. (See the FIND command in the section FIND Primary Command or CHANGE command in the section CHANGE Primary Command.)

RIGHT Primary Command (F11)

Scrolls the window toward the right margin. You can include optional parameters of M (maximum) or n (where n is a number from 1 - 999999).

SAVE Primary Command

Saves the item without terminating the edit session in the foreground library under the name and type as shown in the first line of the panel. Validates the item prior to the save unless you specify the ASIS parameter.

SUBMIT Primary Command

Displays the Submit panel where you can submit the query in the current edit session for background processing.

VALIDATE Primary Command

Validates the query in the current edit session. (STMTS cannot be validated.) A message and ==>ERR display in the line command area if syntax or internal consistency errors are found.

Help Panels

Online Help panels are available to give you specific information regarding the panel you are currently using. To request Help, press F1 or enter HELP on the Command line.

The information in the Help panel is based on the current cursor location. That is, if the cursor is on the Line Command area of the Editor panel, the Help panel contains Line Command information. Or, if the cursor is on the Main Menu, information displays regarding the Main Menu.

Help panels are available from all VISION:Bridge panels, except the Command Input panel.

Figure 3-7 shows the Help panel displayed from the Main Menu.

```
Help2          Computer Associates - Help: Main Menu (body area)

A valid option (1-7) must be entered in the single field provided on this
screen. Valid VISION:Inform options are as follows:

  1 - Operation Facilities - For status updates on background processors, and
      for determining the System Maintenance Level.
  2 - Administration Facilities - For creating, updating, and deleting User
      Profiles.
  3 - Report Facilities - For Query status, and Query output processing.

Valid VISION:Bridge options are as follows:

  4 - Quick Query - For assisted development of VISION:Bridge queries.
  5 - Quick Query Immediate Response - For assisted development of
      interactive queries.
  6 - Standard Query Processing - For submitting queries, and for creating,
      updating, and deleting items of type QUERY and STMTS.
  7 - Immediate Response Query Processing - For interactive query processing.

Command ==>
F12=Cancel
```

Figure 3-7 Help Panel

To exit from any Help panel and return to the previous panel, press F12 or enter CANCEL on the Command line.

Diagnostic Messages

VISION:Bridge checks queries for syntax and consistency. When it finds a problem, it issues a message. For example, assume you enter the query in Figure 3-8 and then enter the SAVE command.

```
Editor2 ---- Computer Associates - Editor ---- Name: INAMTS   Type: QUERY
                                                              More:

000100 QUERY DATABASE CUSTOMER
000200 REPORT CUSNAM CUSTNO ORDERNO INAMT
000300 TITLE 'INVOICE AMOUNT
000400 END REPORT
000500 END QUERY









Command ===> save
F1 =Help    F3 =Exit     F5 =Rfind   F6 =Rchange  F7 =Backward   F8 =Forward
F10=Left    F11=Right    F12=Cancel
```

Figure 3-8 Saving a Query with Errors

The resulting panel is shown in Figure 3-9.

Note:

- The Quick Query panels display similar messages.
- During a Command input session or Immediate Response session, messages display immediately following the line with the problem.

```
Editor2 ------ Computer Associates - Editor ------ Name: INAMTS  Type: QUERY
                                                                 More:

==>ERR REPORT CUSNAM CUSTNO ORDERNO INAMT
QM03** Field name CUSNAM  is not defined.
==>ERR TITLE 'INVOICE AMOUNT
SX06** CLOSING QUOTE FOR CHARACTER CONSTANT NOT FOUND
PS04** REQUIRED VALUE FOR IDENTIFIER LINE     IS MISSING
000400 END REPORT
000500 END QUERY




IY02** Item INAMTS   type QUERY    contains errors and was not saved.
Command ===>
F1 =Help    F3 =Exit     F5 =Rfind   F6 =Rchange  F7 =Backward    F8 =Forward
F10=Left    F11=Right    F12=Cancel
```

Figure 3-9 Diagnostic Messages

VISION:Bridge responds by moving the first line that does not validate to the top of the panel and placing an indicator (==>ERR) in the line command of each such line. Directly beneath each line is a message describing what is wrong.

In this example, the REPORT statement specifies a field called CUSNAME. VISION:Bridge checks this name against an internal database glossary and did not find it.

- The first message indicates that the name does not exist. (The correct field name is CUSTNAME.)
- The next two messages inform the user of a missing quotation mark on the TITLE line.
- The message at the bottom of the panel indicates that the query can not be saved, because it contains errors.

The CUSTOMER Database

Most of the examples in this manual produce reports from a database called CUSTOMER. A database is a set of related data stored together in a computer.

The CUSTOMER database contains order, shipping, and billing information pertaining to the customers of a hypothetical company. It contains information such as the names, addresses, and phone numbers of customers, the items they have ordered and their costs, shipping information for the items, and so on.

Figure 3-10 shows the structure of the CUSTOMER database. The tree-like structure is a way of showing how the various data elements relate to one another. The names you see in the boxes are field names. Field names tie your requests for information directly to the information in the computer. They identify the VISION:Bridge the data you need.

For example, CUSTNAME is the field name that identifies all the customer names in the database. When you tell VISION:Bridge you want to report CUSTNAME, VISION:Bridge knows you want to see all the customer names.

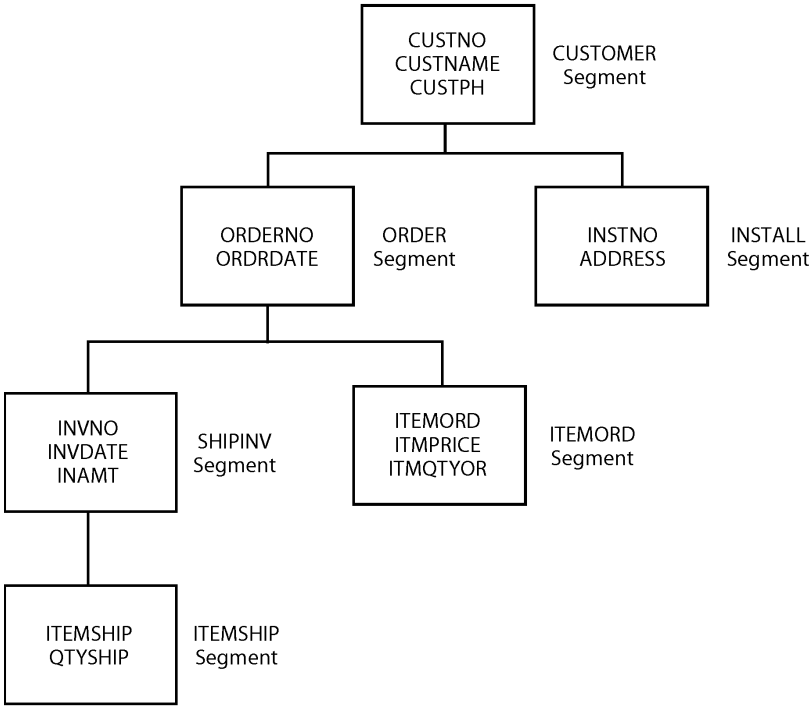


Figure 3-10 CUSTOMER Database Hierarchy

Do not be concerned about the details of the database for now. Look at the field names and become familiar with them. You will see them in the examples throughout this manual. You will get a detailed overview of the CUSTOMER

database in Chapter 7, *Advanced Topics* where the discussion of VISION:Bridge data structures occur. For a complete diagram of the CUSTOMER database see Figure 7-7.

Glossaries

The file definition provides useful information about the database. This information is available to you in the form of a glossary, an alphabetical list of the database fields and their characteristics. VISION:Bridge provides online glossaries, which you can view on the terminal.

GLOSSARY is a VISION:Bridge command available through Option 1 (Operational Facilities) or Option 3 (Report Facilities) on the Main Menu. See the discussion of the GLOSSARY command in Chapter 4, *Using VISION:Bridge* in the section GLOSSARY Command and in Chapter 5, *Using Immediate Response* in the section GLOSSARY Command.

Glossary Information

Using the GLOSSARY command, you can:

- Find the spelling of a field name.
- Determine in which segment the field is.
- Determine the field size and type.
- Find the alternate field name.
- Find the field description.

Glossary Sample

As shown in Figure 3-11, the online glossary lists the fields in alphabetical order, with the characteristics of each field.

```
CUSTOMER - IMS DATABASE FILE   DATE CREATED - 01.111   TIME - 12.25.125

   NAME    TYPE   START  LENGTH   DEC  OUT-LEN  KEY  SEGNAME   PARENT

ADDLIN1    CHAR      22     25           25          INSTALL   CUSTOMER
   FIRST LINE OF THE ADDRESS
ADDLIN2    CHAR      47     25           25          INSTALL   CUSTOMER
   SECOND LINE OF THE ADDRESS
ADDRESS    CHAR      22     50           50          INSTALL   CUSTOMER
   FIRST & SECOND LINES OF THE ADDRESS
CHECKNO    CHAR      76      7            7          SHIPINV   ORDER
   CHECK_NUMBER
   THE PRE-PRINTED CHECK NUMBER
CUSTNAME   CHAR       6     30           30     S    CUSTOMER  CUSTOMER
   CUSTOMER_NAME
   CUSTOMER/COMPANY NAME
CUSTNO     CHAR       1      5            6     1    CUSTOMER  CUSTOMER
   CUSTOMER_NUMBER
   A FIVE-DIGIT CUSTOMER NUMBER
CUSTPH     CHAR      36     10           10          CUSTOMER  CUSTOMER
   PHONE_NUMBER
   CUSTOMER PHONE NUMBER INCLUDING AREA CODE
FRTCOST    PACKD     23      4    2      10          SHIPINV   ORDER
   FREIGHT_COST
   COST OF SHIPMENT
INAMT      PACKD     54      4    2      10          SHIPINV   ORDER
   INVOICE_AMOUNT
   AMOUNT OF INVOICE
```

Figure 3-11 Online Glossary (Partial)

An explanation of the glossary abbreviations is available in Figure 3-12.

Glossary Abbreviations

Heading	Entry	Meaning
NAME	name	Name of the field.
TYPE	CHAR	Characters.
	PACKD, ZONED, FIXED, or FLOAT	Numbers.
	VARBL	Variable length.
START	number	Field start location in segment.
LENGTH	number	Length of field in bytes.
DEC	0 - 9	Number of decimal places in a numeric field.
OUT-LEN	number	Column width on a report.
KEY	blank or number	Key field designation for a segment.
SEGNAME	name	Name of segment field is in.
PARENT	name	Name of parent segment in the hierarchy.

The alternate field name and field description are listed below these entries if they are included in the file definition (see Figure 3-11).

Figure 3-12 Online Glossary Entries

Display Glossary Information From the Full Screen Editor

Glossary for a data view can also be obtained during a full screen editor session for queries by entering the FIELDS primary command in the Editor panel. The query must contain at least a valid QUERY statement and have been validated prior to using this command. The system responds with the following panel listing the fields for the current data view.

```
Editor2 ------- Computer Associates - Editor ------ Name:  AQUERY  Type: QUERY
                                    +-------------------------------------+
000100 QUERY FINANCE                | Fields2 Computer Associates - Fields|
000110 SELECT IF ACCOUNT_NUMBER GT '10  | Dataview: FINANCE      More:     +  |
000200 REPORT PROFIT_CENTER_ACCOUNT_NU  | _ ACCOUNT                         |
000300 END REPORT                   | _ ACCOUNT_AMOUNT                    |
000400 END QUERY                    | _ ACCOUNT_BUDGET_AMOUNT             |
                                    | _ ACCOUNT_CHARGE_MONTH              |
                                    | _ ACCOUNT_CHARGE_PERIOD             |
                                    | _ ACCOUNT_CHARGE_YEAR               |
                                    | _ ACCOUNT_DATE                      |
                                    | _ ACCOUNT_DAY                       |
                                    | _ ACCOUNT_DESCRIPTION               |
                                    | _ ACCOUNT_MONTH                     |
                                    | _ ACCOUNT_NUMBER                    |
                                    | _ ACCOUNT_TYPE                      |
                                    | _ ACCOUNT_YEAR                      |
                                    | _ ACCTCNT                           |
                                    | _ DATECNT                           |
                                    |                                     |
                                    | Command ===>                        |
Command ===>                        | F1 =Help     F7 =Backward           |
F1 =Help     F2 =Save     F3 =Exit  | F8 =Forward  F12=Cancel             |
F7 =Backward F8 =Forward  F9 =Validate  +-------------------------------------+
```

Figure 3-13 Editor Fields Panel

Obtain detailed information for a specific field by placing an 'S' next to the desired field and pressing ENTER. The system displays the Fields Detail panel as shown in Figure 3-14.

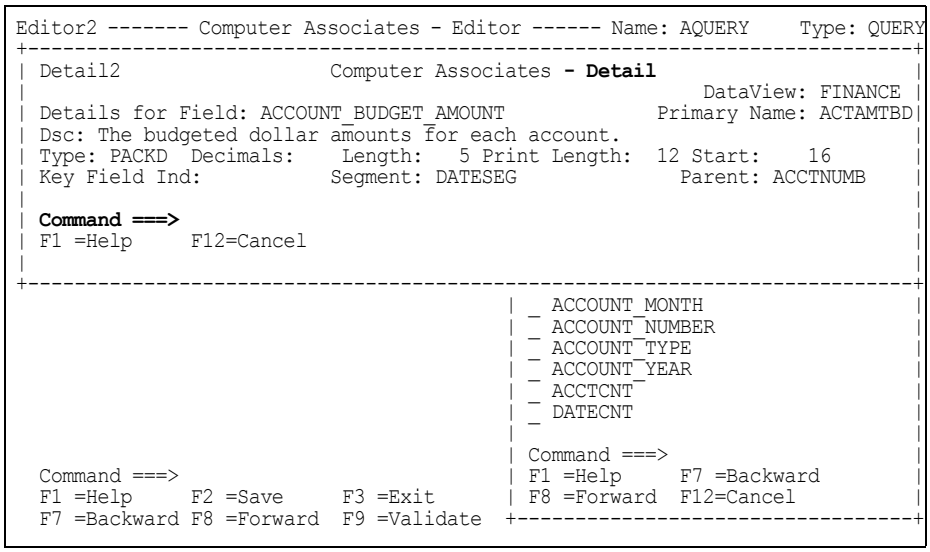


Figure 3-14 Fields Detail Panel

Chapter

4

Using VISION:Bridge

What Can VISION:Bridge Do?

With VISION:Bridge, you can access data stored in many different types of databases and files regardless of its database structures or the format of the data. You retrieve this data through the specification of queries. The requested data is automatically formatted into reports that are returned to the originating terminal, to another logical terminal, to a printer, or to the communication file for future action.

VISION:Bridge produces attractive and easy-to-read reports. You can easily modify the queries that generate reports to meet individual requirements with VISION:Bridge data editing capabilities.

VISION:Bridge features include:

- A simple, easy-to-learn query language, which provides a variety of reporting functions.
- An online syntax checker, which checks queries and provides helpful diagnostic messages.
- Intelligent data selection capabilities.
- Sorting and control break commands to generate easy-to-use reports.
- Summary commands, such as counts, totals, accumulations, averages, percent, and ratios.
- Built-in summary functions for data selection based upon the summarization of a field.
- Arithmetic processing commands, facilitating calculations and the selection of data based upon arithmetic calculations.
- Automatic lookup and retrieval of data stored in tables.
- An interactive full screen editor, which means you can save for later modification or execution.
- Support of IMS, DB2®, and all commonly used databases, as well as standard MVS® files.
- A logical data view feature providing the capability of defining logical views involving more than one file or database.

- Multi-level security through the use of individual user profiles, which define access to data at the user, database, segment and field, or individual record levels.
- The creation of subfiles from data residing in files and databases.
- Extensive report handling and processing options through user defined batching and class specifications.
- Ability to call external user written routines for specialized functions.
- Ability to generate reports in formats suitable for use with a web browser, or other software products.

System Overview

With VISION:Bridge, you can develop queries to obtain the data you want to see. The queries you create consist of VISION:Bridge commands that tell VISION:Bridge what you want and how you want to see it.

By using VISION:Bridge, you can obtain data without concerning yourself with:

- The type of file in which it is stored.
- How to get the data out of the file.
- How to execute your query.
- Specifics about the environment where your query executes.

Separating you from the environment is one of the major strengths of VISION:Bridge. With VISION:Bridge, the environment becomes the responsibility of your VISION:Inform system administrator, freeing you to satisfy your processing and reporting needs with a minimum of effort.

Figure 4-1 shows VISION:Bridge within your computer.

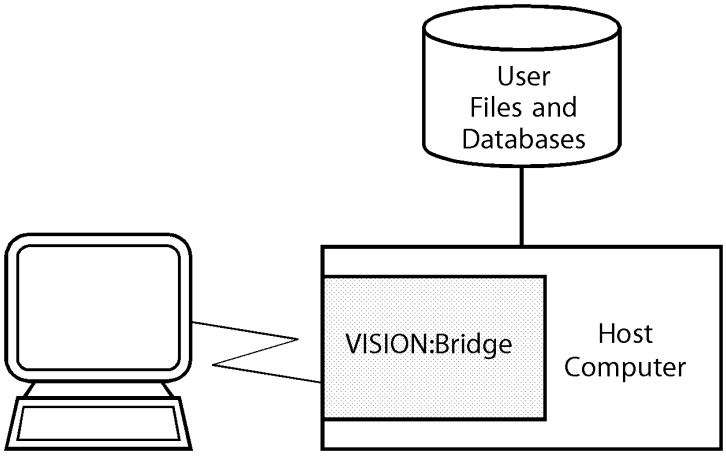


Figure 4-1 VISION:Bridge in Your Computer

This chapter describes how your queries are processed and how your reports are handled. You will also learn about system components and how they work with one another to process your queries and return your reports.

How VISION:Bridge Works

VISION:Bridge operates in many different environments. It must function according to the rules of the environment. VISION:Bridge relieves you of the responsibility of knowing these rules.

The basic premise of VISION:Bridge is that you create and submit queries through an online terminal. Your computer processes these queries in a background area. When the report you request is ready, VISION:Bridge automatically returns it to your terminal. You can send your report to another terminal or to a printer.

Figure 4-2 illustrates how VISION:Bridge operates.

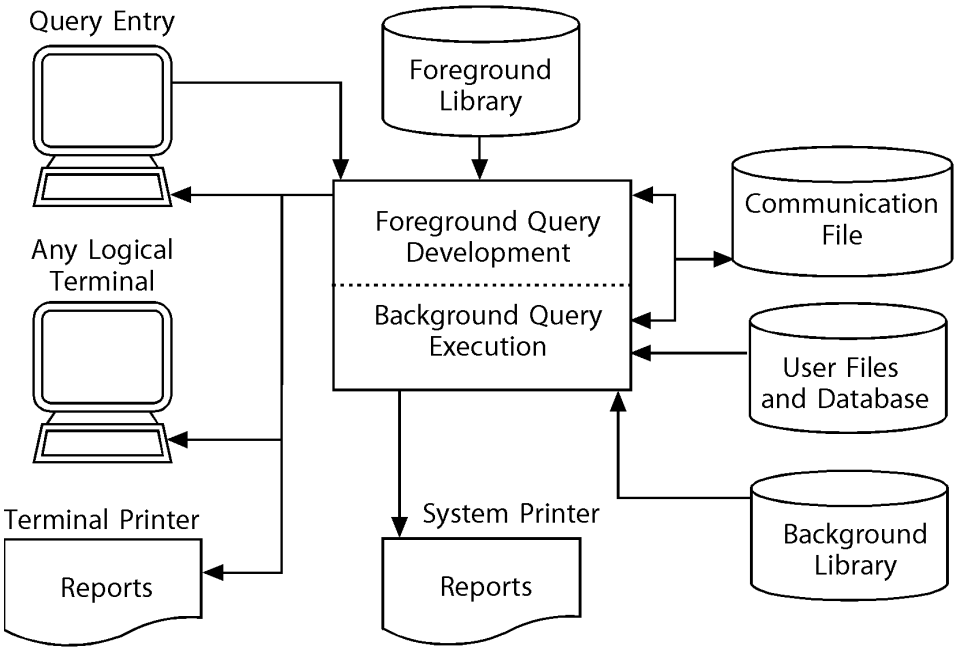


Figure 4-2 How VISION:Bridge Operates

There are, however, some exceptions.

Immediate Response

In certain online environments, VISION:Bridge actually processes your query, not in a background area but in the online area itself through the Immediate Response function.

Batch Simulator

There is also a version of VISION:Bridge that operates completely in a background area. In this version, both the creation and processing of queries takes place in a background area through the Batch Simulator.

System Components

VISION:Bridge consists of the following components:

- Foreground Processor
- Background Processor
- Foreground library
- Background library
- Communication file

The components that VISION:Bridge uses at any one time depends on the environment in which it is operating.

Figure 4-3 illustrates the components of VISION:Bridge.

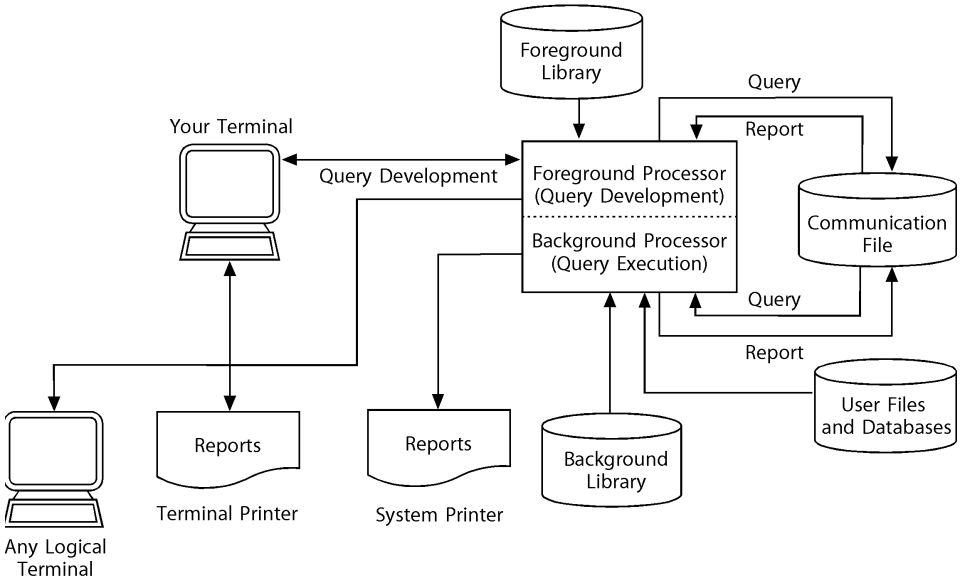


Figure 4-3 VISION:Bridge Components

Foreground Processor

The Foreground Processor is the component that maintains a dialog with you when you are at your terminal. You provide instructions for the Foreground Processor. The Foreground Processor is known as an application program. The environment in which it operates treats it just like other application programs under its control.

Start the Foreground Processor the same way as you start other application programs in a particular environment. Once started, you can create queries, edit them, submit them for processing, and display the output they produce, all from your online terminal.

The Foreground Processor checks your queries for syntax and internal consistency.

- Syntax problems occur when you make a mistake entering a VISION:Bridge command. You will learn more about commands and their syntax later in this chapter.
- Internal consistency problems usually occur when you request some item of data that is not defined.

The Foreground Processor displays a diagnostic message when it detects a problem. Diagnostic messages consist of one or more lines of text which describes the problem. You use the Foreground Processor to correct the problem and resubmit your query.

In addition to diagnostic messages, the Foreground Processor also issues two other types of messages.

- Informational messages provide information about your online session. Unlike diagnostic messages, which require that you take some action, informational messages require no action on your part.
- Prompting messages display during an online session when the Foreground Processor wants you to provide instructions. Prompting messages require that you take some action.

For more information, see the section Diagnostic Messages.

Background Processor

The Background Processor is the component that processes your query. It is an application program that operates in the batch portion of your host computer. The Background Processor prepares your queries for processing and translates them.

Then, it proceeds to retrieve the data requested in your query. It manipulates the data according to your query instructions.If you want data to be arranged in a particular sequence, a sort program sorts the data into that sequence.

After data retrieval and manipulation, the Background Processor formats your report according to your query specifications. Completed reports display at your terminal. You can also send the reports to some other terminal or to your system printer. The Background Processor also does the allocation and formatting of reports destined for OS/390 subfiles or for other software programs and web browsers.

Classes

One or more Background Processors can be active in your system at any time. They process queries according to classes. Classes are a numerical way for your VISION:Inform system administrator to categorize queries. These values range from 1 - 14. The VISION:Inform system administrator assigns priorities to these classes. You can assign queries classes or you can accept the default. A Background Processor can have one or several classes of queries that it can process. Before the Background Processor can process your query, there must be a Background Processor active to process the class you use for your query.

Assigning classes to Background Processors is an individual preference that varies from installation to installation. Your VISION:Inform system administrator is the person who implements classes in your installation.

Background Processor Operation

Your VISION:Inform system administrator also decides the manner in which Background Processors operate. They can remain active indefinitely and process your queries on a first in first out basis. Or, the system administrator can activate the Background Processor on an as needed basis. Using a Background Processor this way means that it starts when queries for its particular class are waiting; it processes them and then terminates.

Batching

The Background Processor groups your queries with other queries. This is called batching and improves performance and efficiency. When several queries request data from the same database, it is very time consuming if each one accesses the database separately. It is more efficient to access the database once and formulate several reports, than to access the data numerous times.

The Background Processor looks at the queries and determines whether they can be batched. There are certain requirements that your query must meet before it can be batched. If it meets the criteria for batching, it is assembled with other queries using the same database.

Figure 4-4 illustrates the flow of queries and reports.

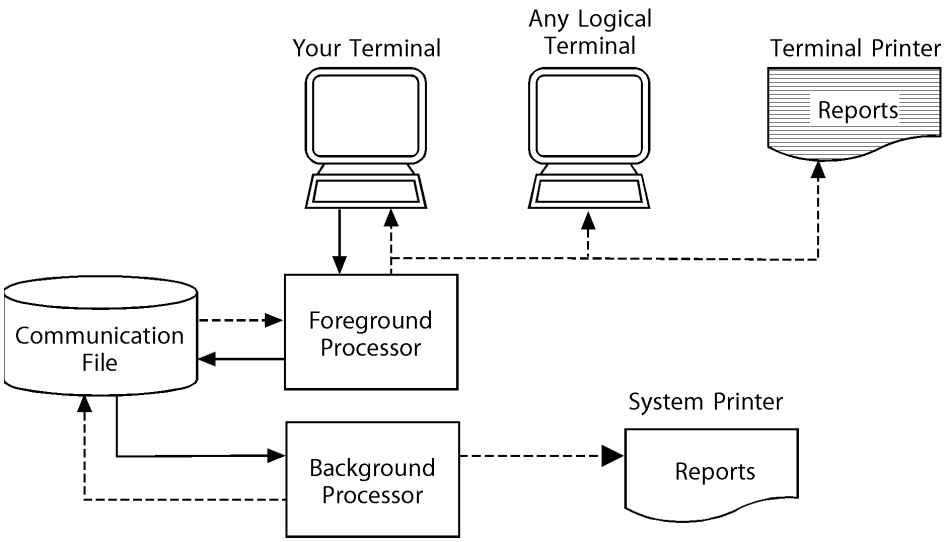


Figure 4-4 The Flow of Queries and Reports

Communication File

You submit queries from the Foreground Processor, and the Background Processor processes them. Whenever you submit a query, the communication file becomes the storage medium for storing your submitted queries and reports.

Foreground Library

The Foreground Processor uses the foreground library for storing the queries you create from your terminal. Saving a query means that you can call upon it numerous times to get the data you want.

This library also contains information, called file definitions, about the databases and files from which you can obtain information. They describe the contents of the databases and files your queries can access. Your VISION:Inform system administrator gives this information to VISION:Bridge so that all you have to do in your query is request the items you want to see.

The Foreground Processor uses these file definitions to verify that the items you request are available. If they are not currently defined, they are not available. This is considered invalid and, as a result, the Foreground Processor displays an appropriate diagnostic message.

Another important function of this database or file is to provide security to your system. Only you and other authorized users can use VISION:Bridge through a user profile. It contains information about which users can use the system, the files available to them, the data available to them and what kinds of things they can do. Defining the user profile is another function of your VISION:Inform system administrator.

Background Library

The Background Processor uses the background library when it processes your query.

The background library contains definitions of the files that you can access in your queries. Unlike the foreground library, it does not contain any stored queries or security information.

Language Overview

You develop VISION:Bridge queries by specifying a series of English-like statements. Use these statements to tell VISION:Bridge what to do.

The statements you specify resemble sentences; but unlike sentences, they do not require stringent punctuation. The only punctuation you use are blank spaces and commas. Sometimes, commas separate items for clarity. You always use commas to continue a statement on the next line.

This section describes the statements you will use, their various components, and the rules for their usage.

VISION:Bridge Command Statement Syntax

A command is an order to VISION:Bridge (such as QUERY a database or REPORT data). For VISION:Bridge to recognize your command and its keywords, follow the syntax rules for using the commands and keywords.

The general syntax for a statement is:

COMMAND (KEYWORD$_1$ (operand$_1$)) $\left[\left\{\begin{array}{l}\underline{\text{KEYWORD}_2} \\ \text{KEYWORD}_3\ \text{operand}_3\end{array}\right\}\right]\ldots$

Text, which follows the syntax, describes each element in a command statement.

COMMAND

Begin each statement with a command.

- Commands can be followed by one or more parameters.
- Some commands do not have parameters.

KEYWORD operand

Parameters provide more information for the command.

- Parameters can be composed of one or more keywords or keywords followed by keyword operands.
- Parameters are optional or required.
- Some keywords do not have operands.

Conventions

Note: *Parameters can also be referred to as identifiers, modifiers, or options.*

- Brackets [ ] indicate an optional parameter.
- Parameters enclosed in braces { } represent options from which you can select one or more.
- Entries printed in **UPPERCASE** in this font type are system commands, parameters, or keywords.
- Entries printed in **lowercase** in this font type are keyword operands (informational entries, such as field names) that you provide.
- Many keywords are optional if their parameters are entered in the order specified in the format. The description indicates whether or not a parameter is optional.
- Underlined entries are defaults.
- Separate commands, parameters, and operands from one another by one or more blanks.

  You can use a comma in place of, or in addition to, a blank to improve readability. Commas are optional and are not processed or interpreted, unless one is the last character on a line to indicate continuation.
- You can continue statements that do not fit on one line onto subsequent lines by terminating each line, except the last, with a comma.

Positional Order

As you become familiar with the commands, you will know when you can leave out the keywords and when they are required. Keywords are helpful when you are not familiar with the positional order of a command. By specifying the keyword, you can specify the part of the statement in any order following the command.
For instance, the following statements are specified in their positional order without keywords and with keywords in any order:

```
COUNT ORDERNO BY CUSTNO
```
(positional order)

or

```
COUNT BY CUSTNO ITEM ORDERNO
```
(any order with keyword)

Expressions

You can use relational and arithmetic expressions by themselves or as part of another statement.

Relational Expressions

The following illustrates the use of relational expressions in a statement.

```
REPORT CUSTNO CUSTNAME IF CUSTNAME EQ 'TULSA TIMES'
```

or

```
IF CUSTNAME EQ 'TULSA TIMES'
```

IF designates the start of the relational expression whether it is in a statement by itself or part of another statement. Relational expressions test for a specific condition. In the REPORT statement, the report is only generated if the relational expression is true.

The second IF statement causes the statements following it to be executed if the condition is true.

Arithmetic Expressions

In the following statements, you will see how arithmetic expressions are used:

```
REPORT CUSTNO CUSTNAME ITMQTYOR * 1.1
```

The presence of the arithmetic operators (+, -, *, /) tells you that the expression performs an arithmetic operation. You can use this type of expression in a statement with other expressions or by itself in the following way:

```
LET ITMTOT = ITMQTYOR * 1.1
```

This LET statement places the result of the arithmetic operation (ITMQTYOR * 1.1) in a new field called ITMTOT.

Constants

In the previous examples, note the use of a literal constant “TULSA TIMES” and a numeric constant 1.1.

Constants are data values that you specify in a statement. The value never changes. Constants are unlike variables, whose values can change during a query.

Combining Expressions

You can combine both relational and arithmetic expressions in a single statement. The following examples illustrate combining these type of expressions:

```
REPORT CUSTNAME IF ITMQTYOR * 1.1 GT INAMT
```

In the first example, CUSTNAME is only reported when the result of the multiplication is greater than the value in the field INAMT.

```
LET ITMTOT = ITMQTYOR * 1.1,
    IF CUSTNAME EQ 'TULSA TIMES'
```

In the second example, the result of the multiplication is placed in the field ITMTOT if the customer is the TULSA TIMES.

Note: Appendix D, *Reference Summary*, contains detailed specifications for each of the *VISION:Bridge* commands.

You can use many combinations of expressions. All of them are described in detail in the sections that follow.

Queries

You now know the format of the statements that make up queries. This gives you a good idea of what a query looks like. The following example illustrates a typical query:

```
QUERY CUSTOMER
SELECT IF ORDERNO GE 0100
REPORT CUSTNO CUSTNAME ORDERNO
TITLE 'CUSTOMER ORDER LIST'
END REPORT
END SELECT
END QUERY
```

In the example, notice that queries always begin with a QUERY command and end with an END QUERY command.

Between these two commands, specify the statements that determine the data, how it is manipulated, and the format in which it is reported.

SELECT Groups

Queries contain groups of statements that perform a particular function. To select the data you want to see, use the SELECT group. It begins with a SELECT command and ends with an END SELECT command. All statements in the SELECT group are applied to the data made available by the SELECT statement.

IF Groups

The same is true for the IF group. It begins with the IF command and ends with the END IF command. Statements appearing between these two are applied to the data made available by the IF statement.

REPORT Groups

The query above contains a single REPORT group within a SELECT group. The SELECT statement limits the data available to just those records where ORDERNO is greater than or equal to 0100. The REPORT statement tells what fields to report. A single SELECT group can have many REPORT groups under its control and each one will be subject to criteria specified on the SELECT statement.

Subsequent sections contain more information about these groups and how they function.

Introducing VISION:Bridge Commands

This section contains a list and brief description of each of the VISION:Bridge commands and functions. You can also find descriptions of each command and function in Appendix D, *Reference Summary*.

Note:

- *For Immediate Response commands, see Chapter 5, Using Immediate Response.*
- *Summary commands (AVG, COUNT, CUM, MAX, MIN, PCT, RATIO, and TOTAL) operate at the group or entire file level.*
- *Summary functions (AVG, COUNT, MAX, MIN, and TOTAL) operate at the record level.*
- *Summary functions can also be referred to as built-in summary functions or summaries.*

These commands, as well as some other words, are reserved keywords. For a complete list of reserved words, as well as a technique for using these words as field names, see Appendix D, *Reference Summary*.

Commands and Functions	Description
AVG	Computes and reports the average of all valid occurrences of the specified fields.
AVG (fieldname)	Summary function to compute the average of all valid occurrences of the specified field.
CALL	Executes a user written program. See your VISION:Inform system administrator for more information.
;comments	Not a command. You can place comments on any query line after a semicolon (;).
COUNT	Counts and reports the number of valid occurrences of the specified fields.
COUNT (fieldname)	Summary function to count the number of valid occurrences of the specified field.
CUM	Computes and reports the cumulative total of all valid occurrences of the specified fields.
ELSE	Keyword used following an IF relational expression to specify that the next command is performed when the IF expression is false.

Commands and Functions	Description
END IF	Designates the end of an IF group.
END QUERY	Designates the end of a query.
END REPORT	Designates the end of a REPORT group.
END SELECT	Designates the end of a SELECT group.
EXTRACT	Generates a subfile.
FORMAT	Indicates that automatic report formatting is being overridden.
GLOSSARY	Displays the information in a database file definition.
GROUP	Causes a field to print only when its value changes.
IF	Indicates that statements following this statement execute if the relational expression in this statement is true.

Note: For information on ALL, AND, ANY, HAS, HAVE, OR, and TO, see Chapter 6, *Data Selection and Field Processing*.

Commands and Functions	Description
IF (ALL)	Indicates that statements following this statement execute if the relational expression is true for all the occurrences of the specified field in the current record.
IF (AND)	Connects two or more relational expressions and executes the statements following them when all the relational expressions are true.
IF (ANY)	Indicates that statements following this statement execute if the relational expression is true for any of the occurrences of the specified field in the current record.
IF ALL (HAVE)	Indicates that statements following this statement execute if all the occurrences of the specified field or fields specified in the relational expressions in this statement are true in the current record.

Commands and Functions	Description
IF ANY (HAS/HAVE)	Indicates that statements following this statement execute if any occurrences of the field or fields specified in the relational expressions in this statement are true in the current record.
IF (OR)	Connects two or more relational expressions and executes the statements following them when any of the tests are true.
IF (TO)	Indicates that statements following this statement execute if any occurrences of the specified field in the relational expression fall within the range specified by the TO.
ITEM	Overrides the default editing specifications for reports.
LET	Assigns a value to the field (database or temporary) following the LET command.
LIST	Displays a list of statements.
LISTLIB	Displays the database names or query names in the foreground library by category, in alphabetical order.
MAX	Computes and reports the maximum value of all valid occurrences of each specified field.
MAX (fieldname)	Summary function to compute the maximum valid value of the specified field.
MIN	Computes and reports the minimum value of all valid occurrences of each specified field.
MIN (fieldname)	Summary function to compute the minimum valid value of the specified field.
ORDER	Outputs data in an ascending or descending sequence (which can be different than the way it is stored in the database).
PCT	Computes and reports the percentages of one field value compared to another field value.
PF	Isolates a portion of a character field to be used in a query statement. “PF” stands for partial field.
PSTATUS	Displays the status of one or more Background Processors.

Commands and Functions	Description
PURGE	Delete queries that are currently executing, waiting for execution, ready for viewing, held, or disabled due to an execution problem.
QSTATUS	Displays the status of queries submitted by this user ID.
QUERY	The first statement of every query. Names the database or logical data view from which you will produce a report.
QUIT	Terminates the command input session and returns you to the Main Menu.
RATIO	Computes and reports the ratio of one field value compared to another field value.
REPORT	Generates a report for the fields listed on the statement and based upon the criteria specified (if any).
ROUTE	Directs the query output to a specific destination.
SELECT	Establishes the range of database segments or fields that your query accesses.
SET	Defines a temporary field with explicit attributes.
TITLE	Specifies the lines of text that comprise the title of your report.
TOTAL	Computes and reports the total of all the valid occurrences of all the fields listed on the command.
TOTAL (fieldname)	Summary function to compute the sum of all the valid occurrences of the specified field.
VIEW	Requests viewing of the output for a query or queries.

A Sample VISION:Bridge Query

Figure 4-5 shows a simple VISION:Bridge query and the report it produces. There are several important things to observe.

Query

```
QUERY CUSTOMER
REPORT CUSTNO CUSTNAME ORDERNO
TITLE 'A SIMPLE QUERY'
END REPORT
END QUERY
```

Output

```
DEC. 11, 2001              A SIMPLE QUERY                  PAGE 1
-----------------------------------------------------------------
CUSTOMER      CUSTOMER                                     ORDER
NUMBER          NAME                                       NUMBER
-----------------------------------------------------------------
00001       TULSA TIMES                                    07321
00001       TULSA TIMES                                    08432
00013       ESSEX ENTERPRISES                              01142
00013       ESSEX ENTERPRISES                              03619
00013       ESSEX ENTERPRISES                              09541
00028       AMERICAN RING AND WIRE SERVICE                 11211
00043       BANK OF THE PEOPLE                             23468
00048       STONEWELL STATE SYSTEMS                        06573
00048       STONEWELL STATE SYSTEMS                        06781
00089       GOING PLACES INSURANCE COMPANY                 03921
00089       GOING PLACES INSURANCE COMPANY                 13842
00089       GOING PLACES INSURANCE COMPANY                 14821
00115       SUNNY GRAPE GROWERS                            00284
00115       SUNNY GRAPE GROWERS                            01391
```

Figure 4-5 Sample VISION:Bridge Query and Report

- There are five simple statements in the query. Each statement provides an instruction to VISION:Bridge.
- Notice the format of the report. The date is in the upper left hand corner of the page, the page number is in the upper right corner, and the title is centered at the top of the page. The information is listed from left to right across the page in the order specified on the REPORT statement. This is the default report format. As you will learn, you can override the default format and produce reports with almost any format you like.
- Notice that some of the information is repeated. The customer number for Tulsa Times, for example, is listed twice. So is the name Tulsa Times. This is because Tulsa Times had two orders in the database. VISION:Bridge printed the customer number and customer name for each of the orders it found. This is characteristic of the way VISION:Bridge processes structured databases.

Occasionally, reports are complex and hard to read if too much information is printed. As you will learn, it is easy to prevent this with a command that suppresses the printing of repetitive data.

A Sample VISION:Bridge Session

This section describes a VISION:Bridge session from log on to log off. You will see every command statement and keystroke required to produce the query and report.

Before You Begin a Session

To begin a VISION:Bridge session, you need a user ID. If security is an issue at your installation, you will probably need a password.

Your VISION:Inform system administrator sets up your profile with your user ID, password, and an internal list of the types of data available to you, such as databases, segments, and fields within a database. The information in your profile pertains to a specific user ID. However, multiple users can log on simultaneously with the same user ID. You cannot use VISION:Bridge without a user ID and profile.

Starting the Session

VISION:Bridge is a computer program that can run in several different environments. To you, the user, the environment is transparent. Once you start VISION:Bridge, it looks the same and all the commands and features work the same regardless of the environment in which it runs.

What differs is how you tell the environment to use VISION:Bridge. How you do this depends on the environment and the particular computer installation you are using (for example, CICS, IMS, or TSO). Enter the appropriate command at your terminal. Your VISION:Inform system administrator can provide assistance.

Once you tell the computer you want to use VISION:Bridge the Logon panel displays, as shown in Figure 4-6. Log on to VISION:Bridge.

```
Logon2                    Computer Associates - Logon


                  Welcome to VISION:Inform Release 4.0.

                   Please Enter Your User ID and Password:

                      User ID  . . . . _____
                      Password . . . .        (if password protected)


               +-------------------------------------------------+
               |  Proprietary and confidential information of    |
               |    Computer Associates International, Inc.      |
               |  Use restricted by written license agreement.   |
               |                (c) 1980, 2001                   |
               |    Computer Associates International, Inc.      |
               |  as an unpublished work. All rights reserved.   |
               +-------------------------------------------------+

Command ===>
F1 =Help     F3 =Exit
```

Figure 4-6 VISION:Bridge Logon Panel

After you receive your user ID and password from your VISION:Inform system administrator, enter it on the Logon panel (Figure 4-6) to display the Main Menu (Figure 4-7).

```
Menu2                    Computer Associates - Main Menu

6 Enter one of the following VISION:Inform or VISION:Bridge Options:


  VISION:Inform Options

  1. Operation Facilities                (Background Processor Status)
  2. Administration Facilities           (Profile development)
  3. Report Facilities                   (Report Handling)

  VISION:Bridge Options

  4. Quick Query                         (Assisted Query Development)
  5. Quick Query Immediate Response      (Assisted Query Development)
  6. Standard Query Processing           (Submit, Delete, Edit
                                          Queries and Stmts)
  7. Immediate Response Query Processing (Run Queries and Immed Mode)

Command ===>
F1 =Help     F12=Cancel
```

Figure 4-7 Main Menu

Use the Main Menu to select the option with which you will be working in this session. The reporting commands are available in Option 6 (Standard Query Processing).

Select Option 6 for Standard Query Processing (as shown in Figure 4-7) for VISION:Bridge to display the Source Processing panel. From this panel, you can:

- Submit queries for processing.
- Delete queries.
- Start the Editor panel to update queries or create new ones.

The Source Processing panel displays, as shown in Figure 4-8. Since you are creating a new query and not updating an existing query, enter the following CREATE command on the Command line:

```
CREATE SAMPLE QUERY
```

```
Source2 ---------- Computer Associates - Source Processing -----------
                                                              More:
    Name     Type   Owner    Last Used    Name   Type   Owner     Last Used
_ ACCTS     QUERY  SAVANNAH  09/15/01   _ ITEMCNT   QUERY  SAVANNAH  09/15/01
_ ITEMLST   QUERY  USER1     09/15/01
_ MONTHLY   QUERY  USER1     09/15/01
_ MYQUERY   QUERY  USER1     09/15/01
_ WEEKLY    QUERY  USER1     09/15/01
_ PART1     STMTS  USER1     09/15/01
_ PART2     STMTS  USER1     09/15/01
_ PART3     STMTS  USER1     09/15/01
_ SPECIAL   STMTS  USER1     09/15/01
_ TEST1     STMTS  SAVANNAH  09/15/01
_ TEST2     STMTS  SAVANNAH  09/15/01
_



Command ===> create sample query
F1 =Help    F7 =Backward  F8 =Forward  F12 =Cancel
```

Figure 4-8 Source Processing Panel

This command identifies the item name as SAMPLE and the item type as QUERY. After pressing the Enter key, the Editor panel displays, as shown in Figure 4-9.

```
Editor2 ------ Computer Associates - Editor ----- Name: SAMPLE   Type: QUERY
                                                                 More:
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''
''''''

Command ===>
F1 =Help     F3 =Exit   F5 =Rfind   F6 =Rchange   F7 =Backward   F8 =Forward
F10=Left     F11=Right  F12=Cancel
```

Figure 4-9 An Empty Editor Panel

When you create a new query, the Editor panel is initially empty, as shown above. You start typing your query. After each line of the query, press the Tab key twice to move to the next line. (If you press Enter, the Line Command area is replaced by sequence numbers. To correct this, move the cursor to the Line Command area and type an i to insert a line after the current line.)

Then type:

```
query customer
report custno custname orderno
title 'a simple query'
end report
end query
```

After entering the last line of the query, press Enter. Notice that the Line Command area now contains sequence numbers, and the query displays in uppercase letters, as shown in Figure 4-10.

```
Editor2 ------ Computer Associates - Editor ------ Name: SAMPLE   Type: QUERY
                                                                  More:
000100 QUERY CUSTOMER
000200 REPORT CUSTNO CUSTNAME ORDERNO
000300 TITLE 'A SIMPLE QUERY'
000400 END REPORT
000500 END QUERY









Command ===>
F1 =Help     F3 =Exit    F5 =Rfind  F6 =Rchange   F7 =Backward F8 =Forward
F10=Left     F11=Right   F12=Cancel
```

Figure 4-10 The Sample Query

To save the query and exit the panel, type the EXIT primary command or press the function key assigned to it (F3s in Figure 4-10). The Source Processing panel displays with the new query in the list and a message displays that indicates that the query was saved. From this panel you can now submit the query for processing. Type an S in front of the new query, SAMPLE, as shown in Figure 4-11.

```
Source2 ------- Computer Associates - Source Processing ------
                                                          More:
    Name     Type     Owner      Last Used    Name    Type    Owner

 Last Used
_ ACCTS     QUERY    SAVANNAH    09/15/01
_ ITEMCNT   QUERY    SAVANNAH    09/15/01
_ ITEMLST   QUERY    USER1       09/15/01
_ MONTHLY   QUERY    USER1       09/15/01
_ MYQUERY   QUERY    USER1       09/15/01
S SAMPLE    QUERY    USER1       09/15/01
_ WEEKLY    QUERY    USER1       09/15/01
_ PART1     STMTS    USER1       09/15/01
_ PART2     STMTS    USER1       09/15/01
_ PART3     STMTS    USER1       09/15/01
_ SPECIAL   STMTS    USER1       09/15/01
_ TEST1     STMTS    SAVANNAH    09/15/01
_ TEST2     STMTS    SAVANNAH    09/15/01
_


IY01** Item SAMPLE   type QUERY   has been saved.
Command ===>
F1 =Help      F3 =Exit     F5 =Rfind   F6 =Rchange   F7 =Backward F8 =Forward
F10=Left      F11=Right    F12=Cancel
```

Figure 4-11 Source Processing Panel

Submit Panel

The Submit panel, shown in Figure 4-12, displays when you type an S in front of a query on the Source Processing panel or enter the SUBMIT primary command from the Editor panel.

To submit the query, fill in the entries on the panel (or accept the default values) and enter the EXIT primary command.

```
Submit2                    Computer Associates - Submit
                           Name: SAMPLE   Type: QUERY   OwnerId: USER1

 To Submit the query enter EXIT
 To leave the panel without submitting the query enter CANCEL

Class                              10          (Enter  1-14 )
Maximum number of roots            ________    (Enter 1-99999999)
Maximum number of lines            ____        (Enter 1-9999)
Route to batch printer?            _           (Y)
Format for logical terminal        ________    (Terminal Id)
Generate query source with report  _           (Y)
Output Format                      1           (1 - Standard Report
                                                2 - HTML
                                                3 - Tab Delimited
                                                4 - Comma Delimited
                                                5 - Plain Text)
Override Query Output Format       _           (Y)



Command ===>
F1 =Help     F3 =Exit     F12=Cancel
```

Figure 4-12 Submit Panel

The following text describes each of the entries on the panel shown in Figure 4-12.

Class

The Class entry is the only required entry on the panel and displays with the default value from your profile or the PARMBLK. If your user profile restricts you to submitting queries of a single class, this entry is protected. Valid entries are 1 - 14.

Maximum number of roots

Use this optional entry to specify the maximum number of root segments to be retrieved during the processing of this query. Valid entries are 1 - 99999999.

Maximum number of lines

Use this optional entry to specify the maximum number of lines of output the query is to print. Use this entry to limit output when testing a query. Valid entries are 1 - 9999.

Route to batch printer

Optional entry. Type a Y to format the output and send it to a printer rather than to the communication file. An entry of Y is mutually exclusive with an entry in the Format for logical terminal field (local printer).

Format for logical terminal

Optional entry. Type a logical terminal ID for which the query data is to be formatted before being sent to the communication file. An entry in this field is mutually exclusive with a Y entry in the Route to batch printer field, which sends the report to the system printer.

Generate query source with report

In this optional entry, type a Y to indicate that query statements are to be printed with the query report.

After filling in the entries on this panel (or accepting the default values), type one of the following commands:

EXIT (F3)	Validates the entry fields on the Submit panel and then submits the query for processing. You return to the previous panel, and a confirmation message displays.
CANCEL (F12)	Returns to the previous panel without validating the entries or submitting the query. F12 is the default for CICS.
CANCEL (F24)	F24 is the default for IMS.

Output Format

Use this optional entry to specify alternate report output formats for any reports in the submitted query that do not specify an output format on the FORMAT statement. The acceptable values for this entry are:

Output Format Panel Option	Description
1 – Standard Report	Output report in standard report format
2 – HTML	Output report as an HTML document
3 – Tab Delimited	Output data fields as a tab-delimited file
4 – Comma Delimited	Output data fields as a comma-delimited file
5 – Plain Text	Output report as a plain text file

The Output Format entry works in conjunction with output format specifications on the query's FORMAT statements, and the Override Query Output Format option on the Submit Panel (see below).

Override Query Output Format

Use this optional entry in conjunction with the Output Format entry (see above) to force the Output Format on this Submit Panel to override any output format specification on the query's FORMAT statements.

Reporting

You generate reports by submitting VISION:Bridge queries to the communication file for execution by a background processor. A query is a group of statements, each with a particular format and function that tells VISION:Bridge what data you want to see and how you want it presented.

Examples in this section are of increasing complexity to illustrate the entry and result of each command statement. For more information on additional commands and statement keywords, see the section Additional Capabilities.

This section includes the basic commands for:

- Writing queries that request and format simple reports.
- Saving and submitting your queries.

Commands for management of your VISION:Bridge session and for gathering information about existing queries and databases available to your user ID are in the section Command Input.

Throughout this section you will first see a query and its output, followed by a discussion of the command or commands used in that query.

Statements, Commands, and Parameters

Queries are made up of statements, each of which have a particular format and function. Statements consist of a command followed by parameters. Parameters are composed of keywords which can be followed by operands. Operands can be referred to as user supplied variables. See the section VISION:Bridge Command Statement Syntax.

Note:

- The commands in the adjacent list are in alphabetical order.
- In the sections which follow, the commands are grouped by function.

This section describes the following commands:

Commands	For additional information
AVG, COUNT, CUM	See the section Summary Commands — AVG, COUNT, CUM, MAX, MIN, and TOTAL Commands.
END QUERY	See the section END QUERY Command
END REPORT	See the section END REPORT Command.
GROUP	See the section GROUP Command.
MAX, MIN	See the section Summary Commands — AVG, COUNT, CUM, MAX, MIN, and TOTAL Commands.
ORDER	See the section ORDER Command.
PCT	See the section Summary Command — PCT Command.
QUERY	See the section QUERY CommandLocating Alternate Output Format Report Files.
RATIO	See the section Summary Command — RATIO Command
REPORT	See the section REPORT Command.
TITLE	See the section TITLE Command.
TOTAL	See the section Summary Commands — AVG, COUNT, CUM, MAX, MIN, and TOTAL Commands.

Generating Reports

Note: The following sections show a query followed by its resulting output. Command statements, which are the subject of discussion in the section, appear in **bold** type.

Figure 4-13 illustrates a complete VISION:Bridge query that produces a report of three fields from the CUSTOMER database. The report is titled ITEMS ORDERED.

Query

```
QUERY CUSTOMER
REPORT CUSTNAME ORDERNO ITEMNAME
TITLE 'ITEMS ORDERED'
END REPORT
END QUERY
```

Output

```
JUN 08, 2001              ITEMS ORDERED              PAGE 1
-------------------------------------------------------------
       CUSTOMER                ORDER             ITEM
         NAME                  NUMBER            NAME
-------------------------------------------------------------
TULSA TIMES                     07321       M4 PR FORM
TULSA TIMES                     07321       M4 OUTPUT FORM
TULSA TIMES                     07321       M4 BASIC REQ. FORM
TULSA TIMES                     07321       M4 RUN CNTL. FORM
TULSA TIMES                     08432       M4 DIAGNOSTIC MSG.
ESSEX ENTERPRISES               01142       Q MGR.USER GUIDE
ESSEX ENTERPRISES               01142       Q LANG. REF. MAN.
ESSEX ENTERPRISES               01142       Q LANG. DC
ESSEX ENTERPRISES               03619       M4 TABLE DEF FORM
ESSEX ENTERPRISES               03619       M4 FILE DEF FORM
ESSEX ENTERPRISES               03619       M4 TRAN DEF FORM
ESSEX ENTERPRISES               09541       M4 REF MANUAL
ESSEX ENTERPRISES               09541       M4 OPS GUIDE-OS
AMERICAN WIRE AND RING SERVICE  11211       M4 OPS GUIDE- DOS
BANK OF THE PEOPLE              23468       M4 REF MANUAL
STONEWELL STATE SYSTEMS         06573       T4 REF MANUAL
STONEWELL STATE SYSTEMS         06573       T4 CON/FAC MANUAL
STONEWELL STATE SYSTEMS         06781       M4 BASIC REQ FORM
STONEWELL STATE SYSTEMS         06781       M4-405 CLASS
STONEWELL STATE SYSTEMS         06781       M4 DIAGNOSTIC MSG
```

Figure 4-13 Basic Query and Its Report

Locating Alternate Output Format Report Files

When you submit a query that specifies one of the alternate output formats for a report, the report output for that query is not directed to the communications file as standard queries are, but is instead stored in a dynamically allocated OS/390 data set. Summary data for the alternate output format query is written to the communications file, and when you issue the VISION:Inform VIEW command for the query (in READY status), you will see information similar to that shown below:

```
**OUTPUT FOR USER USER1   **
**ID SAMPLE ALTERNATE FMT RPT  **
**QUERY 1234-SAMPLE  ,  CLASS 10**
**REPORT NO.  1, PAGE COUNT        **
**AGAINST DATABASE FINANCE  **
**PROCESSED 08/29/02    AT 17:52:41**
HTM='USER1.SAMPLE.H2941104.HTM', RECS=    32, SIZE=     28K
```

The first six lines are the standard VISION:Inform report header that is part of all report output. The seventh line shows the generated data set name for the alternate output format report file, along with a record count and an approximate file size in bytes.

QUERY Command

QUERY DATABASE database name

The first command of every query must be the QUERY command. The QUERY command tells VISION:Bridge the database from which you want to report. You can get the list of database names to which you have access by using the LISTLIB command, which is discussed in the section Command Input, or the Editor DATAVIEW primary command. Terminate the query with an END QUERY command.

Parameters:

DATABASE DATABASE is an optional keyword.

database name The name of the database to be accessed.

Example.

```
QUERY CUSTOMER
```

The QUERY statement above tells VISION:Bridge to use the CUSTOMER database for this query. The DATABASE keyword is optional. An entry of QUERY DATABASE CUSTOMER produces the same result.

END QUERY Command

END QUERY

The END QUERY command terminates the query. This command has no keywords or operands. Make END QUERY the last statement of your query.

REPORT Command

REPORT ITEM name(s) IF logical expression SUMMARY GRANDSUM EMPTYFLD value BLKLNSUP

Use the REPORT statement to request data fields to be output to a report. The VISION:Bridge statements to produce an individual report are called a REPORT group. A REPORT group begins with the REPORT command.

A single query can generated any number of reports (up to 255) provided they all access the database listed on the QUERY statement and each report is defined by a REPORT group.

ITEM Indicates that literal data and names of fields to be output follow. ITEM is optional if the field names of values to be output follow the REPORT command.

name(s) Field names or values to be output on the report.

- Fields appear on the report in the sequence they are listed on this statement.
- A field name contains a maximum of eight characters — an alphabetic character followed by up to seven alphanumeric characters.

See the section Defining Files for more information on field naming conventions.

IF Discussed in Chapter 6, *Data Selection and Field Processing*.

logical expression Discussed in Chapter 6, *Data Selection and Field Processing*.

SUMMARY Causes the report to produce summary information only.

GRANDSUM Causes the report to produce automatic grand summaries on a separate last page of the report. This could be the installation default.

EMPTYFLD value Enter one of the following for the handling of zero value numeric fields and blank character fields when requesting COUNT or AVG summaries for a report. If you do not specify EMPTYFLD, zero value numeric fields are included and blank character fields are excluded.

INCLUDE Include the "empty" field occurrences when calculating these summaries.

EXCLUDE Exclude the "empty" field occurrences when calculating these summaries.

BLKLNSUP Suppresses blank lines created by using the GROUP and ORDER statements.

Example.

```
REPORT CUSTNAME ORDERNO ITEMNAME
```

This statement creates a report of the data stored in the fields CUSTNAME, ORDERNO, and ITEMNAME. The data is arranged in the same sequence across the report as the field names are listed on this statement.

END REPORT Command

END REPORT

The END REPORT statement indicates the end of a report group. To produce an additional report from this query, place the statements defining this new report after the END REPORT statement. This statement has no keywords or operands.

TITLE Command

TITLE LINE 'text'

You can place titles on reports using the TITLE statement.

- You can define up to nine title lines for each report.
- Title lines are automatically centered at the top of each page in the report. The first line appears between the date (on the left) and the page number (on the right). If the first title line is too long to fit, it is placed on the next line.
- Make title lines up to 73 characters long.

You can alter the position of the title lines with the FORMAT statement discussed in the section Additional Capabilities.

LINE LINE is an optional keyword.

'text' Character text for title lines. Enclose text strings in single quotation marks. You represent apostrophes within the text of the title by two single quotation marks.

Examples.

Statement 1:
```
TITLE 'ITEMS ORDERED'
```
Title 1:
```
ITEMS ORDERED
```

Statement 2:
```
TITLE 'ITEMS ORDERED' 'IN JANUARY'
```
Title 2:
```
ITEMS ORDERED
 IN JANUARY
```

Statement 3:
```
TITLE 'ITEMS ORDERED'
TITLE 'IN JANUARY'
```
Title 3:
```
ITEMS ORDERED
 IN JANUARY
```

Statement 4:
```
TITLE 'CUSTOMER INVOICES',
'1976-2001' 'SORTED BY DATE'
```

Title 4:
```
CUSTOMER INVOICES
   1976-2001
 SORTED BY DATE
```

Statement 5:
```
TITLE 'MANAGERS'' PRODUCTION REPORT'
```

Title 5:
```
MANAGERS' PRODUCTION REPORT
```

There are a variety of methods for entering titles on reports. Throughout this section, you will see other examples of the TITLE statement in the example queries.

ORDER Command

ORDER BY name(s) DESC name(s)

The ORDER command outputs data in a different sequence than it is stored on the database.

- You can sort up to nine fields in each report.
- Multiple sorting sequence is determined by the order in which the fields are listed on the ORDER statement. The primary sort field (sorted first) is listed first. If another data item is to be sorted within that, enter it second. If a third sort level of sort is required, list it next and so on up to nine.
- The data is sorted in ascending sequence by default.
- You can sort any or all sort fields on the ORDER statement in descending sequence by repeating their field names after the DESC keyword.

To output the fields listed in the ORDER statement, place them on the REPORT statement.

BY Optional keyword if fields to be sorted are listed directly after the ORDER command.

name(s) Fields to be sorted, listed in order of sort significance.

DESC Keyword indicating the fields that follow are to be sorted in descending sequence.

name(s) Fields to be sorted in descending sequence.

Example.

```
ORDER BY CUSTNAME
```

The statement above tells VISION:Bridge to sort the data by CUSTNAME.

Sorting a Field

Note: In the query samples which follow, the **bold** type correlates to the primary topic of discussion.

Figure 4-14 shows the same query illustrated in Figure 4-13 with an ORDER statement added and the title changed. The ORDER statement outputs the data in a sequence different from which it is stored on the database. With no ORDER statement, the data outputs in the order in which it exists on the database.

Query

```
QUERY CUSTOMER
REPORT CUSTNAME ORDERNO ITEMNAME
ORDER BY CUSTNAME
TITLE 'ITEMS ORDERED',
'SORTED BY CUSTOMER'
END REPORT
END QUERY
```

Output

```
JUN 08, 2001          ITEMS ORDERED                  PAGE 1
                    SORTED BY CUSTOMER
------------------------------------------------------------
         CUSTOMER                ORDER        ITEM
           NAME                  NUMBER       NAME
------------------------------------------------------------
AMERICAN WIRE AND RING SERVICE  11211   M4 OPS GUIDE-DOS
BANK OF THE PEOPLE              23468   M4 REF MANUAL
ESSEX ENTERPRISES               01142   Q MGR. USER GUIDE
ESSEX ENTERPRISES               01142   Q LANG. REF. MAN.
ESSEX ENTERPRISES               01142   Q LANG. DC
ESSEX ENTERPRISES               03619   M4 TABLE DEF FORM
ESSEX ENTERPRISES               03619   M4 FILE DEF FORM
ESSEX ENTERPRISES               03619   M4 TRAN DEF FORM
ESSEX ENTERPRISES               09541   M4 REF MANUAL
ESSEX ENTERPRISES               09541   M4 OPS GUIDE-OS
GOING PLACES INSURANCE COMPANY  03921   T4 UTL SYS MANUAL
GOING PLACES INSURANCE COMPANY  13842   T4-202 CLASS
GOING PLACES INSURANCE COMPANY  14821   T4 CON/FAC MANUAL
STONEWELL STATE SYSTEMS         06573   T4 REF MANUAL
STONEWELL STATE SYSTEMS         06573   T4 CON/FAC MANUAL
STONEWELL STATE SYSTEMS         06781   M4 BASIC REQ FORM
STONEWELL STATE SYSTEMS         06781   M4-405 CLASS
STONEWELL STATE SYSTEMS         06781   M4 DIAGNOSTIC MSG
STONEWELL STATE SYSTEMS         06781   M4-604 CLASS
```

Figure 4-14 A Report with One Sorted Field

Sorting Multiple Fields

Figure 4-15 is an example using two sort fields.

Query

```
QUERY CUSTOMER
REPORT CUSTNAME ORDERNO ITEMNAME
ORDER BY CUSTNAME ITEMNAME
TITLE 'ITEMS ORDERED',
'SORTED BY CUSTOMER AND ITEM'
END REPORT
END QUERY
```

Output

```
JUN 08, 2001          ITEMS ORDERED                      PAGE 1
               SORTED BY CUSTOMER AND ITEM
-----------------------------------------------------------------
         CUSTOMER                ORDER       ITEM
           NAME                  NUMBER      NAME
-----------------------------------------------------------------
AMERICAN WIRE AND RING SERVICE  11211   M4 OPS GUIDE-DOS
BANK OF THE PEOPLE              23468   M4 REF MANUAL
ESSEX ENTERPRISES               03619   M4 FILE DEF FORM
ESSEX ENTERPRISES               09541   M4 OPS GUIDE-OS
ESSEX ENTERPRISES               09541   M4 REF MANUAL
ESSEX ENTERPRISES               03619   M4 TABLE DEF FORM
ESSEX ENTERPRISES               03619   M4 TRAN DEF FORM
ESSEX ENTERPRISES               01142   Q LANG. DC
ESSEX ENTERPRISES               01142   Q LANG. REF. MAN.
ESSEX ENTERPRISES               01142   Q MGR. USER GUIDE
GOING PLACES INSURANCE COMPANY  14821   T4 CON/FAC MANUAL
GOING PLACES INSURANCE COMPANY  03921   T4 UTL SYS MANUAL
GOING PLACES INSURANCE COMPANY  13842   T4-202 CLASS
STONEWELL STATE SYSTEMS         06781   M4 BASIC REQ FORM
STONEWELL STATE SYSTEMS         06781   M4 DIAGNOSTIC MSG
STONEWELL STATE SYSTEMS         06781   M4-405 CLASS
STONEWELL STATE SYSTEMS         06781   M4-604 CLASS
STONEWELL STATE SYSTEMS         06573   T4 CON/FAC MANUAL
STONEWELL STATE SYSTEMS         06573   T4 REF MANUAL
```

Figure 4-15 A Report with Two Sorted Fields

Notice the change in the ordering of the item names for each customer.

GROUP Command

GROUP BY name(s) { **SUBTITLE** | **NEWPAGE** }

The GROUP command suppresses printing of repetitive occurrences of field values.

- Entering more than one field name on the GROUP statement indicates multiple levels of grouping (groups within groups).
- Enter field names in order of significance (highest to lowest).
- You can designate up to nine levels of grouping. A blank line is generated following the level 1 group.

Note: Remove blank lines in the report caused by grouping with the BLKLNSUP keyword on the REPORT statement.

Fields listed on the GROUP statement, but not on the REPORT statement, are not reported.

To ensure correct grouping, first sort the fields by using the ORDER statement.

BY Optional keyword.

name(s) The names of the fields to be grouped, in order of significance. Do not group variable length fields.

SUBTITLE A subtitle is printed with each change in value of the grouping field. You must list SUBTITLE fields on the REPORT statement.

NEWPAGE A new page starts with each change in value of the grouping fields.

You must list NEWPAGE fields on the REPORT statement. List grouped fields, not intended to start on a new page, on a separate GROUP statement.

You also use the GROUP command in conjunction with summarization (see the sections Built-In Summary Functions and Summary Commands) and formatting (in the section FORMAT Command, see Example 9. PAGEGRP Keyword)

Example.

```
GROUP BY CUSTNAME
```

This statement only prints the customer name when its value changes.

Note in the section A Report with Two Sorted Fields that the customer name repeats for every occurrence of ORDERNO.

You can eliminate this duplicate data by using the GROUP statement. Figure 4-17 shows the effect of the GROUP statement.

Sorting by Two Fields and Grouping by One Field

Now, the customer name only prints when its value changes.

Query

```
QUERY CUSTOMER
REPORT CUSTNAME ORDERNO ITEMNAME
ORDER BY CUSTNAME ITEMNAME
GROUP BY CUSTNAME
TITLE 'ITEMS ORDERED',
'SORTED BY CUSTOMER AND ITEM',
'GROUPED BY CUSTOMER'
END REPORT
END QUERY
```

Output

```
JUN 08, 2001          ITEMS ORDERED                    PAGE 1
               SORTED BY CUSTOMER AND ITEM
                   GROUPED BY CUSTOMER
-----------------------------------------------------------------
          CUSTOMER              ORDER          ITEM
            NAME                NUMBER         NAME
-----------------------------------------------------------------
AMERICAN WIRE AND RING SERVICE  11211     M4 OPS GUIDE- DOS

BANK OF THE PEOPLE              23468     M4 REF MANUAL

ESSEX ENTERPRISES               03619     M4 FILE DEF FORM
                                09541     M4 OPS GUIDE-OS
                                09541     M4 REF MANUAL
                                03619     M4 TABLE DEF FORM
                                03619     M4 TRAN DEF FORM
                                01142     Q LANG. DC
                                01142     Q LANG. REF. MAN.
                                01142     Q MGR. USER GUIDE

GOING PLACES INSURANCE COMPANY  14821     T4 CON/FAC MANUAL
                                03921     T4 UTL SYS MANUAL
                                13842     T4-202 CLASS

STONEWELL STATE SYSTEMS         06781     M4 BASIC REQ FORM
                                06781     M4 DIAGNOSTIC MSG
                                06781     M4-405 CLASS
                                06781     M4-604 CLASS
                                06573     T4 CON/FAC MANUAL
```

Figure 4-16 Sorting by Two Fields and Grouping by One Field

Sorting by Two Fields and Grouping by Two Fields

Figure 4-17 shows a report grouped by customer names and order numbers, which only print when their values change. They are the two fields on the GROUP statement.

Query

```
QUERY CUSTOMER
REPORT CUSTNAME ORDERNO ITEMNAME
ORDER BY CUSTNAME ITEMNAME
GROUP BY CUSTNAME ORDERNO
TITLE 'ITEMS ORDERED'
END REPORT
END QUERY
```

Output

```
JUN 08, 2001          ITEMS ORDERED                    PAGE 1
-----------------------------------------------------------------
          CUSTOMER               ORDER          ITEM
            NAME                 NUMBER         NAME
-----------------------------------------------------------------
AMERICAN WIRE AND RING SERVICE   11211    M4 OPS GUIDE- DOS

BANK OF THE PEOPLE               23468    M4 REF MANUAL

ESSEX ENTERPRISES                03619    M4 FILE DEF FORM
                                 09541    M4 OPS GUIDE-OS
                                          M4 REF MANUAL
                                 03619    M4 TABLE DEF FORM
                                          M4 TRAN DEF FORM
                                 01142    Q LANG. DC
                                          Q LANG. REF. MAN.
                                          Q MGR. USER GUIDE

GOING PLACES INSURANCE COMPANY   14821    T4 CON/FAC MANUAL
                                 03921    T4 UTL SYS MANUAL
                                 13842    T4-202 CLASS

STONEWELL STATE SYSTEMS          06781    M4 BASIC REQ FORM
                                          M4 DIAGNOSTIC MSG
                                          M4-405 CLASS
                                          M4-604 CLASS
                                 06573    T4 CON/FAC MANUAL
                                          T4 REF MANUAL
SUNNY GRAPE GROWERS              00284    M4 BASIC REQ FORM
```

Figure 4-17 Sorting by Two Fields and Grouping by Two Fields

Commenting Your Queries

You can place comments on any query line. You can add comments to the end of an existing query line or you can make them the entire query line.

; comments

You start a comment with a semicolon (;). The text can be any characters to the end of the statement line.

Whenever VISION:Bridge encounters a semicolon (;) on a query statement, the rest of the line is considered to be a comment.

- If the entire statement is to be a comment, make the semicolon (;) the first non-blank character on the line.
- When you place comment on a continued statement, place the continuation character, a comma (,), to the left of the semicolon (;) starting the comment.

Example.

```
; THIS QUERY HAS A COMMENT FIRST
QUERY CUSTOMER;NO SPACE BEFORE COMMENT
REPORT CUSTNO CUSTNAME ORDERNO
      ; INDENTED COMMENT
ORDER BY CUSTNO  ,  ;PRIMARY REPORT ORDER
ORDERNO    ;SECONDARY REPORT ORDER
END REPORT
END QUERY
```

Displaying Database Information in the Editor Panel

Note: The word “data view” appears as one word in the product.

In the Editor panel, use the DATAVIEW and FIELDS primary commands to obtain information about a particular database definition.

(In the Command Input panel, you can obtain the same information using the VISION:Bridge commands, LISTLIB and GLOSSARY.)

Using the DATAVIEW Primary Command in the Editor Panel

Note: Observe the function keys at the bottom of the panel. It has been customized to assign function keys other than the default assignments.

Figure 4-18 shows the Editor panel with the DATAVIEW primary command.

```
Editor2 ------- Computer Associates - Editor ----- Name: BENTEST Type: Query
                                                                More:      >
000100 QUERY DATABASE FINANCE
000200 REPORT ACCTCNT ACCTDIV ACCTYY ACCTMM
000300 ; FORMAT HEIGHT  WIDTH  DATEPOS {UL¦U¦¦UM¦LL¦LR¦LM¦NO},
000400 ; PAGEPOS {UR¦UL¦UM¦LL¦LR¦LM¦NO},
000500 ; TITLEPOS {TOP¦BOTTOM},
000600 ; LABELS {SUPPRESS¦NOSPACE¦SPACE},
000700 ; HEADINGS {YES¦NO},
000800 ; DATEFMT {TODAY¦DATE¦ISDATE¦JULIAN},
000900 ; LINES {0¦1¦2¦3¦4¦5¦6¦7¦8} BORDER '-' PAGEGRP {YES¦NO},
001000 ; FORMAT HEIGHT  WIDTH  DATEPOS {UL¦UR¦UM¦LL¦LR¦LM¦NO},
001100 ; PAGEPOS {UR¦UL¦UM¦LL¦LR¦LM¦NO},
001200 ; TITLEPOS {TOP¦BOTTOM},
001300 ; LABELS {SUPPRESS¦NOSPACE¦SPACE},
001400 ; HEADINGS {YES¦NO},
001500 ; DATEFMT {TODAY¦DATE¦ISDATE¦JULIAN},
001600 END QUERY


Command ===> dataview
F1 =Help     F2 =Save     F3 =Exit    F4 =Reset  F5 =Rfind F6 =Rchange
F7=Backward  F8=Forward   F9=Validate F10=Left   F11=Right F12=Cancel
```

Figure 4-18 Editor Panel with DATAVIEW Primary Command

Using the Dataview Pop-up Panel

The Dataview pop-up panel, shown in Figure 4-19, displays when you enter the DATAVIEW primary command in the Command area of the Editor panel, shown in Figure 4-18. The Dataview pop-up panel titled, "DataVw2 Computer Associates - DataView", overlays the right side of the Editor panel.

A list of the database definitions associated with the user's profile displays. From this list you can select the definition containing the list of fields you want to see.

```
Editor2 ----- Computer Associates - Editor ----- Name: BENTEST     Type: QUERY
                                                +-------------------------------------+
000100 QUERY DATABASE FINANCE                   | DataVw2 Computer Associates - DataView|
000200 REPORT ACCTCNT ACCTDIV ACCTYY A           |                          More:     +  |
000300 ; FORMAT HEIGHT  WIDTH  DATEPOS          |     Name   Type  Date Promoted        |
000400 ; PAGEPOS {UR¦UL¦UM¦LL¦LR¦LM¦NO          | _ Finance  OS      09/17/01           |
000500 ; TITLEPOS {TOP¦BOTTOM},                 | _ SAMPLE1  LDV     08/15/01           |
000600 ; LABELS {SUPPRESS¦NOSPACE¦SPAC          | _ SAMPLE2  DB2     10/01/01           |
000700 ; HEADINGS {YES¦NO},                     | _ SAMPLE3  OS      10/01/01           |
000800 ; DATEFMT {TODAY¦DATE¦ISDATE¦JU          | _ SAMPLE4  LDV     10/01/01           |
000900 ; LINES {0¦1¦2¦3¦4¦5¦6¦7¦8} BOR          | _ SAMPLE5  LDV     10/01/01           |
001000 ; FORMAT HEIGHT  WIDTH  DATEPOS          | _ SAMPLE6  LDV     10/01/01           |
001100 ; PAGEPOS {UR¦UL¦UM¦LL¦LR¦LM¦NO          | _ SAMPLE7  VSAM    10/01/01           |
001200 ; TITLEPOS {TOP¦BOTTOM},                 | _ SAMPLE8  LDV     10/01/01           |
001300 ; LABELS {SUPPRESS¦NOSPACE¦SPAC          | _ SAMPLE9  LDV     10/01/01           |
001400 ; HEADINGS {YES¦NO},                     | _ SAMPLE10 DB2     10/01/01           |
001500 ; DATEFMT {TODAY¦DATE¦ISDATE¦JU          | _ SAMPLE11 LDV     10/01/01           |
001600 END QUERY                                | _ SAMPLE12 LDV     10/01/01           |
                                                | _ SAMPLE13 LDV     10/01/01           |
                                                |                                       |
                                                | Command ===>                          |
Command ===>                                    | F1 =Help     F7 =Backward             |
F1 =Help     F2 =Save     F3 =Exit              | F8 =Forward  F12=Cancel               |
F7 =Backward F8=Forward   F9=Validate           +---------------------------------------+
```

Figure 4-19 Editor with Dataview Pop-up Panel

Using the LOCATE Primary Command in the Dataview Pop-up Panel

The Dataview pop-up panel supports the LOCATE primary command so that you can find the database definition you want to select. This is very helpful with long database definition lists that require scrolling to find the desired database. The located database name returns at or near the top of the Dataview panel depending on whether you enter a full or partial name. Figure 4-20 shows a sample of the LOCATE primary command.

```
Editor2 ------ Computer Associates - Editor ---- Name: BENTEST      Type: QUERY
                                        +------------------------------------ +
000100 QUERY DATABASE FINANCE           | DataVw2 Computer Associates - DataView|
000200 REPORT ACCTCNT ACCTDIV ACCTYY A   |                          More:     +  |
000300 ; FORMAT HEIGHT  WIDTH  DATEPOS  |     Name   Type  Date Promoted       |
000400 ; PAGEPOS {UR¦UL¦UM¦LL¦LR¦LM¦NO  | _ Finance  OS      09/17/01          |
000500 ; TITLEPOS {TOP¦BOTTOM},         | _ SAMPLE1  LDV     08/15/01          |
000600 ; LABELS {SUPPRESS¦NOSPACE¦SPAC  | _ SAMPLE2  DB2     10/01/01          |
000700 ; HEADINGS {YES¦NO},             | _ SAMPLE3  OS      10/01/01          |
000800 ; DATEFMT {TODAY¦DATE¦ISDATE¦JU  | _ SAMPLE4  LDV     10/01/01          |
000900 ; LINES {0¦1¦2¦3¦4¦5¦6¦7¦8} BOR  | _ SAMPLE5  LDV     10/01/01          |
001000 ; FORMAT HEIGHT  WIDTH  DATEPOS  | _ SAMPLE6  LDV     10/01/01          |
001100 ; PAGEPOS {UR¦UL¦UM¦LL¦LR¦LM¦NO  | _ SAMPLE7  VSAM    10/01/01          |
001200 ; TITLEPOS {TOP¦BOTTOM},         | _ SAMPLE8  LDV     10/01/01          |
001300 ; LABELS {SUPPRESS¦NOSPACE¦SPAC  | _ SAMPLE9  LDV     10/01/01          |
001400 ; HEADINGS {YES¦NO},             | _ SAMPLE10 DB2     10/01/01          |
001500 ; DATEFMT {TODAY¦DATE¦ISDATE¦JU  | _ SAMPLE11 LDV     10/01/01          |
001600 END QUERY                        | _ SAMPLE12 LDV     10/01/01          |
                                        | _ SAMPLE13 LDV     10/01/01          |
                                        |                                      |
                                        | Command ===> locate sample20         |
Command ===>                            | F1 =Help     F7 =Backward            |
F1 =Help     F2 =Save     F3 =Exit      | F8 =Forward  F12=Cancel              |
F7 =Backward F8=Forward   F9=Validate   +------------------------------------ +
```

Figure 4-20 Editor with the Dataview Pop-up Panel and LOCATE Primary Command

Obtaining a Field List

To obtain the field list, shown in Figure 4-22, type an S over the underscore adjacent to the database definition you selected. An S is the only valid entry, and only one entry can be made at any given time.

Using the FIELDS Primary Command in the Editor Panel

When you enter the FIELDS primary command in the Command area of the Editor panel (Figure 4-21), the Fields panel (Figure 4-22) displays with a list of fields for the database associated with the query being edited.

```
Editor2 ------- Computer Associates - Editor ---- Name: BENTEST Type: QUERY
                                                              More:     >
000100 QUERY DATABASE FINANCE
000200 REPORT ACCTCNT ACCTDIV ACCTYY ACCTMM
000300 ; FORMAT HEIGHT  WIDTH  DATEPOS {UL¦U¦¦UM¦LL¦LR¦LM¦NO},
000400 ; PAGEPOS {UR¦UL¦UM¦LL¦LR¦LM¦NO},
000500 ; TITLEPOS {TOP¦BOTTOM},
000600 ; LABELS {SUPPRESS¦NOSPACE¦SPACE},
000700 ; HEADINGS {YES¦NO},
000800 ; DATEFMT {TODAY¦DATE¦ISDATE¦JULIAN},
000900 ; LINES {0¦1¦2¦3¦4¦5¦6¦7¦8} BORDER '-' PAGEGRP {YES¦NO},
001000 ; FORMAT HEIGHT  WIDTH  DATEPOS {UL¦UR¦UM¦LL¦LR¦LM¦NO},
001100 ; PAGEPOS {UR¦UL¦UM¦LL¦LR¦LM¦NO},
001200 ; TITLEPOS {TOP¦BOTTOM},
001300 ; LABELS {SUPPRESS¦NOSPACE¦SPACE},
001400 ; HEADINGS {YES¦NO},
001500 ; DATEFMT {TODAY¦DATE¦ISDATE¦JULIAN},
001600 END QUERY


Command ===> fields
F1 =Help     F2 =Save     F3 =Exit     F4 =Reset F5 =Rfind F6 =Rchange
F7 =Backward F8 =Forward  F9 =Validate F10=Left  F11=Right F12=Cancel
```

Figure 4-21 Editor Panel with FIELDS Primary Command

Using the Fields Pop-up Panel

Validate queries prior to issuing the FIELDS primary command. The Field selection process is the same as described above for the DATAVIEW primary command.

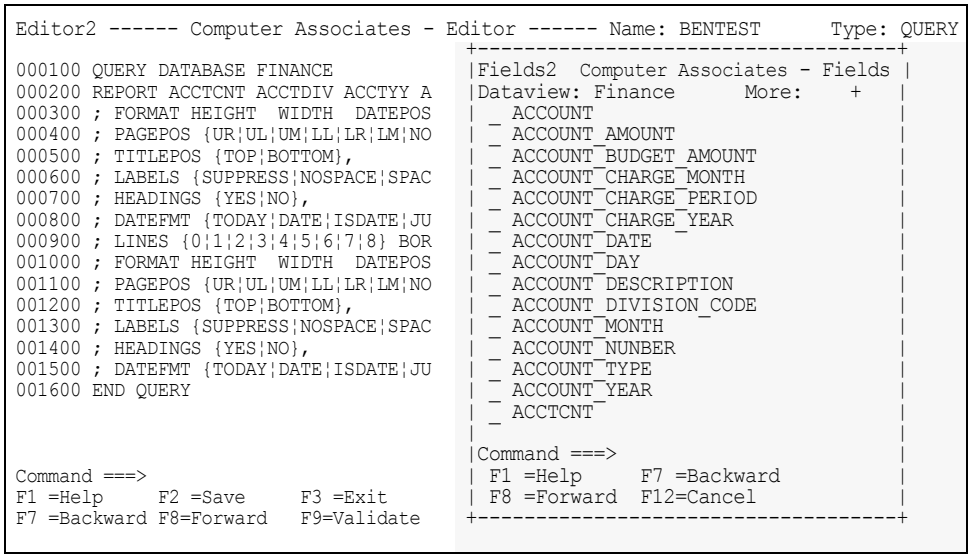


Figure 4-22 Editor with Fields Pop-up Panel

The Fields pop-up panel, titled “Fields2 Computer Associates - Fields”, overlays the right side of the Editor panel.

The Fields pop-up panel also supports the LOCATE primary command, so you can find any given field.

Viewing the Fields Detail Pop-up Panel

To select a field description, type an S over the underscore adjacent to the field name to display the Fields Detail pop-up panel. The Fields Detail pop-up panel, titled “Detail2 Computer Associates - Detail “, overlays the top portion of the screen.

```
Editor2 ------- Computer Associates - Editor ------ Name: BENTEST      Type: QUERY
+-------------------------------------------------------------------------------+
| Detail2                      Computer Associates - Detail                     |
|                                                        DataView:     Finance  |
| Details for Field: ACCOUNT_DIVISION_CODE           Primary Name:     ACCTDIV  |
| Dsc: Each division's two digit identifying code.                              |
| Type: CHAR    Decimals: __  Length: 2    Print Length: 2    Start: 1          |
| Key Field Ind.: _           Segment: ACCTNUMB  Parent:  PROFCENT              |
|                                                                               |
| Command ===>                                                                  |
| F1 =Help     F12=Cancel                                                       |
|                                                                               |
+-------------------------------------------------------------------------------+
001200 ; TITLEPOS {TOP¦BOTTOM},           | _ ACCOUNT_DIVISION_CODE             |
001300 ; LABELS {SUPPRESS¦NOSPACE¦SPACE}  | _ ACCOUNT_MONTH                     |
001400 ; HEADINGS {YES¦NO},               | _ ACCOUNT_NUMBER                    |
001500 ; DATEFMT {TODAY¦DATE¦ISDATE¦JULI  | _ ACCOUNT_TYPE                      |
001600 END QUERY                          | _ ACCOUNT_YEAR                      |
                                          | _ ACCTCNT                           |
                                          |                                     |
                                          | Command ===>                        |
Command ===>                             | F1 =Help     F7 =Backward           |
F1 =Help     F3 =Exit     F5 =Rfind     F | F8 =Forward  F12=Cancel             |
F10=Left     F11=Right    F12=Cancel      +-------------------------------------+
```

Figure 4-23 Editor with Fields Detail Pop-up Panel

The Fields Detail pop-up panel, shown in Figure 4-23 displays with a description of the selected field. The description can include the data view name, the primary name, the long name, the field description, the field type, number of decimals, field length, print length, start position, key field indicator, segment name, and parent segment.

The HELP and CANCEL primary commands are available in the Fields Detail panel command area.

Flags

A flag is an internal indicator defined by VISION:Inform to indicate the existence of certain conditions during the application run. Special words function as flags. Flags are used in processing and reporting statements.

The following flags are available:

Flag	Application	Format
DATE	Records the operating system date in the format: MMM, DD, YYYY.	12 bytes character
EOF	Detects or forces end of file on any sequential input file.	11 bytes character
ISDATE	Records the operating system date in the format:	
	■ YYYYMMDD (logic usage)	8 bytes character
	■ YYYY-MM-DD (formatted reports)	10 bytes character
JULANX	Records the operating system date in the format:	
	■ YYYYDDD (logic usage)	7 bytes character
	■ YYYY.DDD (formatted reports)	8 bytes character
JULIAN	Records the operating system date in the format:	
	■ YYDDD (logic usage)	5 bytes character
	■ YY.DDD (formatted reports)	6 bytes character
LILIAN	Date type field containing the Lilian date as the number of days since the beginning of the Gregorian calendar (October 14, 1582). The valid range of Lilian dates is 1 – 3,074,324 (October 15, 1582 to December 31, 9999). For example, a Lilian date with a value of 152384 converts to the standard date of December 31, 1999.	4 bytes integer (date type)
TIME	Records the time of day a job was started in the format: HH.MM.SS (hours, minutes, seconds).	8 bytes character

Flag	Application	Format
TODAY	Records the operating system date in the format:	
	■ MMDDYY (logic usage)	6 bytes character
	■ MM/DD/YY (formatted reports)	8 bytes character
TODAYX	Records the operating system date in the format:	
	■ MMDDYYYY (logic usage)	8 bytes character
	■ MM/DD/YYYY (formatted reports)	10 bytes character

Command Input

You use the Command Input panel to enter commands for management of your VISION:Bridge session and for gathering information about existing queries and databases available to your user ID. Enter the Command Input panel through Option 3 on the Main Menu.

Once you submit a query for processing, there are additional commands that you can use to check on the status of your query and for handling the output after your queries are processed.

The following commands are discussed in this section:

GLOSSARY	See the section GLOSSARY Command.
LIST	See the section LIST Command.
LISTLIB	See the section LISTLIB Command.
PSTATUS	See the section PSTATUS Command.
PURGE	See the section PURGE Command.
QSTATUS	See the section QSTATUS Command.
QUIT	See the section QUIT Command.
ROUTE	See the section ROUTE Command.
VIEW	See the section VIEW Command.

GLOSSARY Command

```
GLOSSARY  DATABASE database name,
                { SEGMENT segment name }
                { FIELD field name     }
```

The GLOSSARY command displays database definition. You use the GLOSSARY command to provide information about a particular database definition and to see the definition of the entire file, of an individual segment, or a single field.

Note: Glossary information is also available from the Full Screen Editor panel with the FIELDS primary command and Fields Detail panel.

You are restricted to the definitions displayed with the LISTLIB command.

DATABASE	Is an optional keyword if you enter the name of a database immediately after GLOSSARY.
database name	The name of the database for which you are requesting a glossary.
SEGMENT	Indicates that only the portion of the glossary pertaining to a particular segment is to be listed. This is an optional keyword if the segment name appears in the position shown in the format above.
segment name	The name of the segment whose fields are to be displayed.
FIELD	Displays only the information about an individual field. If you select only this option, you must use the FIELD keyword.
field name	The name of the field to be displayed.

Do not use SEGMENT and FIELD in the same GLOSSARY statement.

The following examples show the format of the GLOSSARY statement and the output each statement produces.

Displaying a Database Definition

Example:

```
GLOSSARY CUSTOMER
```

Figure 4-24 is a partial illustration of the output when you use the GLOSSARY statement with only the database (file) definition name.

```
CUSTOMER - IMS DATABASE FILE   DATE CREATED - 01.111   TIME - 12.25.125

  NAME     TYPE  START LENGTH   DEC  OUT-LEN    KEY    SEGNAME   PARENT

ADDLIN1     CHAR   22     25           25              INSTALL   CUSTOMER
  FIRST LINE OF THE ADDRESS
ADDLIN2     CHAR   47     25           25              INSTALL   CUSTOMER
  SECOND LINE OF THE ADDRESS
ADDRESS     CHAR   22     50           50              INSTALL   CUSTOMER
  FIRST & SECOND LINES OF THE ADDRESS
CHECKNO     CHAR   76      7            7              SHIPINV   ORDER
  CHECK_NUMBER
  THE PRE-PRINTED CHECK NUMBER
CUSTNAME    CHAR    6     30           30        S     CUSTOMER  CUSTOMER
  CUSTOMER_NAME
  CUSTOMER/COMPANY NAME
CUSTNO      CHAR    1      5            6        1     CUSTOMER  CUSTOMER
  CUSTOMER_NUMBER
  A FIVE-DIGIT CUSTOMER NUMBER
CUSTPH      CHAR   36     10           10              CUSTOMER  CUSTOMER
  PHONE_NUMBER
  CUSTOMER PHONE NUMBER INCLUDING AREA CODE
FRTCOST     PACKD  23      4    2      10              SHIPINV   ORDER
  FREIGHT_COST
  COST OF SHIPMENT
INAMT       PACKD  54      4    2      10              SHIPINV   ORDER
  INVOICE_AMOUNT
  AMOUNT OF INVOICE
```

Figure 4-24 Using GLOSSARY to Display a Database Definition

Displaying a Segment Definition

Example:

```
GLOSSARY DATABASE CUSTOMER SEGMENT ORDER
```

When the GLOSSARY statement is limited to the ORDER segment of the CUSTOMER database, the following displays.

```
CUSTOMER - IMS  DATABASE FILE   DATE CREATED - 01.111    TIME - 12.25.125

  NAME    TYPE   START   LENGTH  DEC  OUT-LEN   KEY   SEGNAME    PARENT

ORDCMPLT   CHAR   29      1            10             ORDER      CUSTOMER
  IS THE ORDER COMPLETE? (Y OR N)
ORDDUDAT   CHAR   22      6             8             ORDER      CUSTOMER
  DATE THE ORDER IS DUE
ORDERNO    CHAR    1      5             7        1    ORDER      CUSTOMER
  ORDER_NUMBER
  THE FIVE-DIGIT ORDER NUMBER
ORDINVGN    CHAR  28      1            10             ORDER      CUSTOMER
ORDPONUM    CHAR  17      5             8             ORDER      CUSTOMER
  PURCHASE_ORDER_NUMBER
ORDRDATE    CHAR   8      6             8             ORDER      CUSTOMER
  ORDER_DATE
ORPERSON    CHAR  14      3             8             ORDER      CUSTOMER
  ORDER_PERSON
  INITIALS OF PERSON WRITING THE ORDER
SEG20FIL    CHAR  30     10            10             ORDER      CUSTOMER
  UNUSED
*** END OF DEFINITION ***
```

Figure 4-25 Using GLOSSARY to Display a Segment Definition

Displaying a Field Definition

Example:

```
GLOSSARY DATABASE CUSTOMER FIELD CUSTNAME
```

You can also use the GLOSSARY statement, with the FIELD keyword, to display a database field definition. Figure 4-26 illustrates an example where the GLOSSARY statement specifies the CUSTNAME field.

The FIELD keyword is required when requesting information about an individual data field.

```
?:
glossary customer field custname
  NAME     TYPE    START     LENGTH    DEC    OUT-LEN    KEY    SEGNAME    PARENT
CUSTNAME  CHAR        6       30              30         S     CUSTOMER   CUSTOMER
  CUSTOMER_NAME
  CUSTOMER/COMPANY NAME
```

Figure 4-26 Using GLOSSARY to Display a Field Definition

You can obtain a detailed file definition listing with more information about the database, its segments, and fields from your VISION:Inform system administrator.

LIST Command

LIST ITEM name TYPE { **QUERY** / **STMTS** }

LIST displays a collection of statements stored in the foreground library.

ITEM name Is the name of the statements that were previously saved in the foreground library. ITEM is a required keyword.

TYPE { **QUERY** / **STMTS** } Specifies whether the item is a query (QUERY) or a collection of statements (STMTS). If you do not specify this keyword, TYPE QUERY is the default.

Examples:

```
LIST ITEM AUDIT TYPE STMTS

LIST ITEM NAMES
```

LISTLIB Command

LISTLIB { **DATABASE** / **QUERY** / **FDHELP** }

Use the LISTLIB command to list names of items stored in the foreground library. With the LISTLIB command, you can determine which databases you can access.

Note:

- A query list is also available from the Source Processing panel. Select Option 6 (Standard Query Processing) on the Main Menu.
- A data view list is available from the Full Screen Editor panel with the DATAVIEW primary command.

DATABASE Lists all file definition names (databases and logical data views) you can access.

QUERY Lists the names of all queries and collections of statements you can access.

FDHELP Lists all file definitions names that have field descriptions. These descriptions are available to VISION:Journey® for Windows®, VISION:Journey for DOS, Quick Query, and VISION:Bridge users.

Example:

Figure 4-27 and Figure 4-28 show the output generated when you type the LISTLIB command without keywords. All items available to that user ID in the foreground library are listed.

Two screens display. Figure 4-27 lists the data views available to the user. There are two DL/I databases, two non-DL/I databases and one logical data view.

```
DATAVIEW    TYPE    CREATED

BADFIN      NON-DLI    98166
CUSTITEM    LDV        98166
CUSTOMER    DLI        98166
FINANCE     NON-DLI    98166
ITEM        DLI        98166
```

Figure 4-27 LISTLIB Output Showing Data Views

Figure 4-28 displays the queries available to the user. Four are owned by user ID RON and two are owned by user ID SALES.

```
QUERY    OWNER  CREATED   CHANGED     USED

EXAM1    RON    06/16/98  06/16/01   06/16/01
EXPA     RON    06/16/98  06/16/01   06/16/01
LOGQRY   SALES  06/13/98  06/13/01   06/13/01
MSGQRY   SALES  06/13/98  06/13/01   06/13/01
PLANX    RON    06/16/98  06/16/01   06/16/01
TRANX    RON    06/16/98  06/16/01   06/16/01

** END OF LIST **
```

Figure 4-28 LISTLIB Output Showing Queries

If you entered LISTLIB with the keyword QUERY, only the screen in Figure 4-28 displays.

Use the keywords to limit the amount of information you display.

PSTATUS Command

PSTATUS { **NAME background processor(s)** | **ALL** }

The PSTATUS command displays the status of one or more Background Processors that process data views (databases). You can list Background Processors that are currently active and available to process data. In addition, you can list Background Processors that have previously been used, but are not currently active.

NAME — Keyword followed by one or more Background Processor names. The keyword is optional if the first Background Processor name immediately follows PSTATUS.

background processor(s) — A list of one or more Background Processor names that are currently active. The current status of each active Background Processor listed is awaiting or processing.

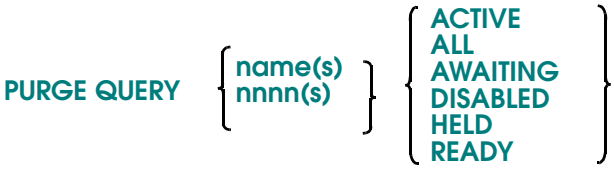
ALL — Displays the status of all Background Processors that have been executed with the current communication file, both those that are currently active and those that have run previously, but are not currently active.

The ALL keyword and a list of Background Processor names are mutually exclusive. If you do not specify either, the status of all currently active Background Processors displays.

Examples.

```
PSTATUS
PSTATUS NAME PAYPROC
PSTATUS ALL
```

PURGE Command



The PURGE command deletes one or more queries awaiting processing, queries already in execution, and queries awaiting viewing. You can also delete the output of queries already processed from the communication file with this command.

This command only effects queries belonging to your user ID.

{QUERY name(s) / QUERY number(s)} Specifies the name or number of the query or queries to be purged. The QUERY keyword is optional.

ACTIVE When you specify a query name, that query is deleted from the communication file as soon as it completes execution.

If there are multiple active queries with the same name, the first one executed is purged.

When used with a query name and ALL, all executing queries with that name are purged after execution.

ALL If specified with no other keywords, all queries with ACTIVE, ALL, AWAITING, DISABLED, HELD, and READY status are purged.

AWAITING When you specify a query name, that query is purged from the processing queue only.

If there are multiple queries with the same name, the first one submitted is purged.

When used with a query name and ALL, all queries waiting to be executed with that name are purged.

DISABLED When you specify a query name, that disabled query is deleted from the processing queue only.

If there are multiple disabled queries with the same name, the first one is purged.

When used with a query name and ALL, all disabled queries with that name are purged.

HELD

When you specify a query name, the held query is deleted from the processing queue only.

If there are multiple held queries with the same name, the first one held is purged.

When used with a query name and ALL, all held queries with that name are purged.

READY

When you specify a query name, the reports from that query are deleted from the communication file.

With multiple queries of the same name, the first one returned to the communication file is purged.

When used with a query name and ALL, all output from queries with that name is deleted.

Examples.

```
PURGE QUERY JBCRPT ALL AWAITING
```

This statement causes all queries awaiting execution with the name of JBCRPT to be purged before execution.

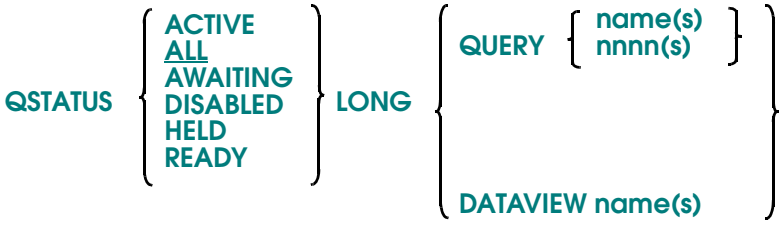
```
PURGE ALL
```

The statement above purges all queries you submitted whether they are awaiting execution, currently executing, or already output to the communication file.

```
PURGE ALL ACTIVE
```

The statement above causes all queries currently executing under your user ID to be purged as soon as processing is complete.

QSTATUS Command



The QSTATUS command provides the status of all queries submitted for processing.

- VISION:Bridge tells you whether your query is awaiting processing (AWAITING), currently being processed (ACTIVE), or is finished processing (READY).
- Two additional query statuses are also possible:
  - HELD indicates that the database is unavailable.
  - DISABLED indicates a problem in query execution.

  See your VISION:Inform system administrator if either HELD or DISABLED appears.

The QSTATUS command also displays the:

- Destination of the report.
- Number of pages the report contains.
- Data view (database) being accessed.
- Class to which the query was submitted.

ACTIVE	Lists queries that are currently being processed.
ALL	Default. This keyword lists queries regardless of status and is mutually exclusive with the above keywords.
AWAITING	Lists queries that have not been selected for processing by a Background Processor.
DISABLED	Lists queries that have been disabled due to a problem encountered during execution.
HELD	Lists queries that have been placed in "hold" status, because a data view required was unavailable.
READY	Lists queries that are completed and ready for viewing.
LONG	Causes a 2-line format of the display to be produced, which includes Background Processor name, submit date and time, and end date and time.
QUERY { **name(s)** / **number(s)** }	Indicates the specific queries to be listed identified by query name or query number. Mutually exclusive with the DATAVIEW keyword.
DATAVIEW name(s)	Identifies the specific name or names of the data views for which queries are to be listed. Mutually exclusive with the QUERY keyword.

Examples.

```
QSTATUS
QSTATUS READY LONG
QSTATUS DATAVIEW FINANCE
```

Sample Output:

```
                          QSTATUS

QUERY #- NAME    STATUS    DEST.  PAGES    DATAVIEW CLASS USERID

4055- RONCOMM    READY     A215       7    CUSTOMER  10
4067- GRANDSUM   READY     N012      12    CUSTOMER  10
0411- GRANDSU2   READY     N012      10    CUSTOMER  10
```

QUIT Command

QUIT

Terminates the command input and displays the Main Menu.

VIEW Command

VIEW QUERY { name(s) / number(s) } { **PURGE** } { **ALL** }

Queries that have finished processing are marked as READY on the output of the QSTATUS command. The reports generated by these queries return to the communication file and are held until you are ready to view them.

Use the VIEW command to display completed reports on your terminal.

- You reference the reports associated with a given query by the query name, or the four digit number, assigned by VISION:Bridge.
- You can retrieve the output from queries one at a time, or you can request more than one query name or number, thereby calling multiple reports to your terminal. In the latter case, they are seen in the order the names, or numbers, were listed on the VIEW statement.
- You also have the option of viewing all available reports to the terminal (one at a time) with the ALL keyword.

Reports that have been viewed remain in the communication file and are available for future viewing. There is no limit on the number of times you can view a report.

Note: For information about paging through reports, see *Appendix A, Paging and Viewing*.

You can delete a report after viewing using the PURGE keyword of the VIEW statement.

QUERY	{ name(s) / number(s) }	Specifies the name or the assigned number of the query that produced the reports you want to view. Obtain the number with the QSTATUS command. You can specify multiple queries.
PURGE		Delete reports from the communication file after viewing.
ALL		Displays all available reports to your screen, one at a time.

Examples.

```
VIEW QUERY PSA123 PURGE
```

This statement displays the report generated by query PSA123 and deletes the report from the communication file after viewing.

```
VIEW ALL
```

The above statement displays all READY queries to the screen, one at a time.

ROUTE Command

ROUTE QUERY { **name(s)** / **nnnn(s)** } { **TO terminal** / **PRINT** } **NOPURGE ALL**

You can send reports to any logical terminal, or local printer, or to the system printer with the ROUTE command.

- Routed reports are deleted from the communication file unless you type the NOPURGE keyword on the ROUTE statement.
- When you route a report to another logical terminal, it remains in temporary storage until someone logs on to that terminal. The report is then output to the screen or printer, although it has already been purged from the communication file.
- Reports routed to printers or other terminals retain their original format and page size.

You must specify TO or PRINT.

QUERY { **name(s)** / **number(s)** } The name, or the VISION:Bridge assigned number, of the query that created the reports to be routed. You can specify multiple entries.

TO terminal Route the report to the specified logical terminal. You can specify only one name.

PRINT Print the report on the system printer.

NOPURGE Do not delete routed reports from the communication file after routing.

ALL Routes all reports available in the communication file to the destination specified.

Examples.

```
ROUTE QUERY QUERY1 TO LTERM2
```

The above statement routes the output from query QUERY1 to logical terminal LTERM2 for viewing. The reports are deleted from the communication file.

```
ROUTE 6578 PRINT NOPURGE
```

This statements routes the reports created by the query number 6578 (assigned by VISION:Bridge) to the system printer. The reports remain on the communication file because of the NOPURGE keyword.

Additional Capabilities

VISION:Bridge provides advanced capabilities with the FORMAT, ITEM, and EXTRACT commands.

After discussion of these capabilities, automatic table lookup, built-in summaries, and summary commands are discussed.

Standard Default Report Format

Reports generated with VISION:Bridge have a standard format.

- The title of the report is centered at the top of the page with the date to the left and the page number to the right.
- Column headings follow the title, date, and page number.
- If you specify summaries, 14 spaces are reserved on the left side of the report page for summary labels.
- The height and width of the report page and the date format are set when you install the system.

You can use the FORMAT command to override these default settings.

FORMAT Command

```
FORMAT  WIDTH number HEIGHT number,
        DATEPOS value PAGEPOS value TITLEPOS value,
        LABELS value HEADINGS value DATEFMT value,
        LINES number BORDER 'x' PAGEGRP value,
        OUTFMT value
```

Use the FORMAT statement to change any of the default report format specifications. When you use the FORMAT command, you override the standard default format specifications.

WIDTH number Enter a number from 80 to the installation maximum (within the limit of the output device).

HEIGHT number Enter the number of lines to appear on each page (2 - 255).

DATEPOS value Enter one of the following for the date position on the report:

UL	Upper left corner, default
UR	Upper right corner
UM	Upper middle of page
LL	Lower left corner
LR	Lower right corner
LM	Lower middle of page
NO	Omit date completely

PAGEPOS value Same choices as DATEPOS; UR is the default. This specifies the position of the page numbers.

TITLEPOS value Enter the TOP or BOTTOM operand to specify where to center the title. TOP is the default.

LABELS value For summary labels enter one of the following:

- **SPACE** Reserve 14 spaces at the left margin for printing labels (default).
- **NOSPACE** Print labels. Columnar data begins at the extreme left. Labels can be partially or totally overlaid.
- **SUPPRESS** Do not print summary labels. Columnar data begins at the extreme left.

HEADINGS value Enter NO to suppress the printing of column headings.

Enter YES to print column headings. YES is the default.

DATEFMT value Enter one of the following to use the current date:

DATE	MMM DD, YYYY format, default
TODAY	MM/DD/YY format
ISDATE	YYYY-MM-DD format
JULIAN	YY.DDD format

To produce a specific date (not the current date), enter the date as integers in the form MMDDYY. 123101 prints as 12/31/01.

LINES number Enter a number from 0 to 8 for the number of blank lines to print between detail lines. Zero (0) is the default.

BORDER 'x' Enter any character in place of x, including one blank. This character prints above and below the column headings instead of the installation default column heading border.

PAGEGRP value Enter NO to suppress the printing of subtitles at top of page. This causes subtitles to print only when they change value. Subtitles always appear at the top of each page unless you specify PAGEGRP NO.

OUTFMT value An optional parameter that indicates the desired output format of the report for the Report Group. Enter one of the following values:

- STD Indicates that the output is to be in standard report format, which is the default.

HTML	Indicates that the output is to be a report in HTML (HyperText Markup Language).
TAB	Indicates that the output is to be data in tab-delimited format.
COMMA	Indicates that the output is to be data in comma-delimited format (also known as Comma Separated Variable, or CSV).
PLAIN	Indicates that the output is to be a report in plain text format.

The OUTFMT keyword of the FORMAT statement applies only to the report for the Report Group that contains it. Output formats specified on the FORMAT statement can be overridden when the query is submitted, using the Submit Panel or SUBMIT command.

Example 1. HEIGHT and WIDTH Keywords

```
FORMAT HEIGHT 25 WIDTH 80
```

In this FORMAT statement, the number of report lines that can appear on each page (or screen display) is set to 25 with the HEIGHT keyword.

- The maximum height that you can specify is 255.
- The default page height is the height of the output device (minus 6 if the device is a printer).

The example defines the width of the report line as 80 characters.

- You can specify any number from 80 to the maximum for your output device with the WIDTH keyword (not to exceed 254).
- The default is the line width of the output device (minus 1 for a terminal).

Use these keywords individually or together to change the dimensions of the output pages to correspond to special forms or various size paper.

Example 2. DATEPOS and PAGEPOS Keywords

```
FORMAT DATEPOS UR PAGEPOS LR
```

The DATEPOS and PAGEPOS keywords move the date and page number to nonstandard positions on the report page. You can also used these keywords to eliminate the date or the page number from printing on a report.

In the FORMAT statement above, the upper right (UR keyword) corner was selected for the date position and the lower right (LR keyword) corner of the report page contains the page number.

If you specify the same position for both the DATEPOS and the PAGEPOS keywords, the VISION:Bridge syntax checker issues a diagnostic message.

The values you can use with DATEPOS and PAGEPOS are listed below:

UL	Upper left
UR	Upper right
UM	Upper middle (centered at top of page)
LL	Lower left
LR	Lower right
LM	Lower middle (centered at bottom of page)
NO	Omit from output

Defaults: DATEPOS-UL
PAGEPOS-UR

Example 3. DATEFMT Keyword

You can change the format of the date on your reports using the DATEFMT keyword. VISION:Bridge gives you the choice of using a standard format or one of three others. Using the format MMDDYY you can also specify any date you want.

Shown below are the different formats produced for reports run on January 21, 2001.

DATE — JAN 21,2001

TODAY — 01/21/01

ISDATE — 2001-01-21

JULIAN — 01.021

If you need a date other than the current date on a report, type that date (MMDDYY) on the FORMAT statement following the DATEFMT keyword. For example:

```
FORMAT DATEFMT 052501
```

The statement causes 05/25/01 to appear as the report date regardless of when the report was produced.

Example 4. TITLEPOS Keyword

```
FORMAT TITLEPOS BOTTOM
```

With the TITLEPOS keyword of the FORMAT statement, you can change the position of the title from the top of the report page to the bottom. The only valid values for this keyword are BOTTOM and TOP (the default). The title is always centered on the report line. For information about the TITLE statement, see the section Reporting.

Example 5. HEADINGS Keyword

```
FORMAT HEADINGS NO
```

This FORMAT statement produces a report with no column headings. YES (the default) and NO are the only valid values for the HEADINGS keyword.

Example 6. LABELS Keyword

When you specify summaries, 14 spaces are reserved at the left margin of the report page for the labels generated by VISION:Bridge. These labels indicate the type of summary function and the field being summarized.

The LABELS keyword specifies the standard format with SPACE and also gives you two other choices.

- The first choice, SUPPRESS, eliminates summary labels. The report data begins at the extreme left side of the page.
- The other choice, NOSPACE, prints summary labels, but the report data begins at the extreme left of the page. This causes the summary labels to be overlaid in some cases.

Value	Labels Printed	14 Spaces Reserved
SPACE	YES	YES
NOSPACE	YES	NO
SUPPRESS	NO	NO

Example 7. LINES Keyword

```
LINES 1
```

This LINES keyword prints one blank line between detail lines on the report. The maximum is eight blank lines between detail lines. The default is 0, no blank lines between detail lines — single spacing.

Example 8. BORDER Keyword

```
BORDER '*'
```

This BORDER keyword prints asterisks above and below the column headings. You can specify any character, including a blank (space bar). Do not use BORDER if you specify HEADINGS NO in the same FORMAT statement.

Example 9. PAGEGRP Keyword

```
PAGEGRP NO
```

This PAGEGRP keyword does not print subtitles at the top of each page. They only print when they change. NO is the only acceptable value for PAGEGRP. When you do not use PAGEGRP, subtitles repeat at the top of each page of the report. For information about subtitles on the GROUP command, see the section Reporting.

ITEM Command

ITEM ENTRY name(s) SPACES value PICTURE pattern HEADING text, NEWLINE NONPRINT CSVQUOTE CSVTRUNC

The ITEM command edits report data to improve its readability. This command overrides the default editing specifications.

- Use this command to increase or decrease the number of blank spaces before data columns. The default is two spaces.
- The HEADING keyword changes column headings and the NEWLINE keyword begins a new output line with the field specified after it.

ENTRY name(s) Enter the names of the fields you want to edit.

- If you place field names immediately after the ITEM command, the ENTRY keyword is optional.
- List each field in only one ITEM statement.
- You must also list these fields on the REPORT statement.
- If the REPORT statement contains multiple references to a given field, the ITEM statement applies to the first reference only.

SPACES value Enter a number from 0 - 99 indicating how many spaces to insert before the data items print on the report.

- The default value is two spaces.
- The SPACES keyword is required.

PICTURE pattern Enter a character-by-character representation of how the output data from the specified field will look. This representation overrides any output edit specifications in the file definition.

- No edit picture can exceed 30 characters.
- There are two edit picture formats: one for numeric data and one for character data.
- The PICTURE keyword is required.

HEADING text Enter character constants (enclosed in quotation marks) that define column headings to print for the associated field.

- Use character constants to create headings for temporary fields defined with the LET or SET statements.
- You can define up to two lines of heading with each line containing from 1 to 14 characters.

NEWLINE Prints the associated data item on a new line. If several field names are on the ITEM statement, each prints on a new line.

NONPRINT Does not print the associated data item. If several field names are on the ITEM statement, none print.

CSVQUOTE Specify CSVQUOTE only for a character field, when the output format for the report will be specified as Comma-delimited or Tab-delimited. This indicator tells the background processor to always enclose character fields within quotes (“”), regardless of whether or not it contains an embedded comma (,).

CSVTRUNC Specify CSVTRUNC only for a numeric field, when the output format for the report will be specified as Comma-delimited or Tab-delimited. This indicator tells the background processor to truncate trailing zeros to the right of the decimal point of the numeric field up to the decimal point.

Specifying Fields to be Edited

ITEM Command ENTRY Keyword

The ITEM command ENTRY keyword specifies the fields to edit using the other keywords of this command. When you specify fields for editing, note:

- You can list database fields, as well as temporary fields, after the ENTRY keyword.
- If you list fields on the ITEM statement, you must also list them on the REPORT statement.
- If you list a field that is edited, using the ITEM statement more than once on the REPORT statement, the ITEM statement applies only to the first reference.
- The ENTRY keyword is optional if you list the fields to be edited directly after the ITEM command.
- When you specify multiple fields on the ITEM statement, the SPACES, PICTURE, HEADING, and NEWLINE keywords apply to all of them.
- In most cases, use one ITEM statement for each field to be edited.

Adjusting Report Data Spacing

ITEM Command SPACES Keyword

With the ITEM command SPACES keyword, you can specify the number of blank spaces between data fields on the report.

- As a default, VISION:Bridge places two blank spaces between report fields.
- You can enter a number from 0 - 99 with the SPACES keyword to specify the number of blank spaces you want in front of the fields listed on report

Example 1.

```
REPORT CUSTNAME CUSTNO ITEMNO
ITEM ENTRY CUSTNAME SPACES 10
```

If a query that contains the statements above, the ITEM statement places 10 spaces in front of the data column for the CUSTNAME field. The CUSTNO and ITEMNO fields each have two spaces before them because no reference is made to those fields on an ITEM statement with the SPACES keyword.

Example 2.

```
REPORT CUSTNAME CUSTNO ITEMNO
ITEM CUSTNAME CUSTNO ITEMNO SPACES 5
```

This example places five spaces in front of each of the fields specified on the ITEM statement. Notice that the ENTRY keyword was omitted on the ITEM statement. You can do this only when you list the fields to be edited directly after the ITEM command.

Example 3.

```
REPORT CUSTNAME CUSTNO ITEMNO
ITEM CUSTNO SPACES 7
ITEM ITEMNO SPACES 12
```

This example produces a report with two spaces (default) in front of the customer name data column, seven spaces before the customer number, and 12 spaces in front of the item number column.

Changing the Report Data Format

ITEM Command PICTURE Keyword

The ITEM command PICTURE keyword changes the format of data fields output to the terminal or to the printer.

If the date on the file is stored as 112101 and you want the date to look like 11/21/01 when it is output, you insert the slashes using the ITEM command PICTURE keyword. For example:

```
ITEM ENTRY DATE PICTURE XX'/'XX'/'XX
```

The pattern XX'/'XX'/'XX inserts the slashes.

If you want to add the character constant “P.O. BOX" in front of the post office box number, you use the PICTURE keyword again.

```
ITEM ENTRY PONUMBER PICTURE 'P.O. BOX ' XXXXX
```

This pattern outputs the data field with the words, P.O. BOX.

Character and Numeric Picture Formats

Note that there are two edit picture formats, one for character data (data type C) and one for numeric data (data types Z, P, and F). Use the GLOSSARY command (see the section Command Input) or the Fields Detail panel to find the data type of a database field.

Leading Dollar Sign

To display a field with two decimal places and a leading dollar sign ($), use numeric editing. For example:

```
ITEM ENTRY PRICE PICTURE $$.99
```

This statement takes the value 03.45 and edits it to $3.45. This also suppresses the leading (0). Appendix C, *Examples of Editing Numeric Data* contains a number of numeric editing examples.

PICTURE Keyword Specifications

The specifications of an edit pattern with the PICTURE keyword overrides any output editing defined on the file definition. No edit picture can exceed 30 characters.

- Rules for character and numeric data editing are in Figure 4-29.
- Rules for numeric data editing are in Figure 4-30.
- For a detailed list of numeric editing patterns, see Appendix C: Examples of Editing Numeric Data.

Edit Picture	Explanation
Character selection (X)	Use the X to indicate that the character in the corresponding position in the source data is to be output. Leading zeros always print for numeric data.
Insert literal data	You can insert any character in the output field, except the installation specified delimiter (#) character. All insertion characters including blanks/spaces are seen by the system as character constants and must be enclosed in quotation marks.
Repeat characters	You can indicate that a character is to be repeated an optional number of times by preceding the character with the number in parentheses. For example, (5) * produces *****.

Figure 4-29 Edit Pictures: Character and Number Data

Edit Picture	Explanation
Digit selection (9)	Each 9 in the edit picture represents an output character position. The output data contains a number from the source field in that position including leading zeros.
Digit selection (Z)	Each Z in the edit picture represents an output character position. ■ The output data contains a number from the source field in that position. ■ If the output character is a leading zero, the zero is suppressed.
Check protection (*)	Each * in the edit picture represents an output character position. ■ The output data contains a number from the source field in that position. ■ If the output character is a leading zero, the zero is replaced by the check protection character (*).
Decimal position indicator (.)	The decimal character represents the decimal position in the edit picture for alignment of columnar data. It also indicates the output position in which the character (.) is inserted.
Grouping character (,)	Each comma in the edit picture represents an output position in which a comma is inserted.
Positive (+) and negative (-)	Use plus and minus signs to position positive and negative signs on the output data (leading, floating, or trailing).
Currency symbol control ($)	The dollar sign indicates the output position into which the currency symbol is inserted (leading or floating).

Figure 4-30 Edit Pictures: Numeric Data

All of the above symbols are installation options and can be represented by other symbols at your installation.

Changing Output Data Headings

ITEM Command HEADING Keyword

The HEADING keyword creates column headings for temporary fields. You can also use this keyword to supply a different heading than the one defined in a database definition. Note the following when specifying column headings:

- You can create one or two line column headings with each line containing from 1 to 14 characters.
- Enclose the text of the column heading in single quotation marks.
- Do not specify fields used as subtitles on an ITEM statement that includes the HEADING keyword.
- If the column heading you define is wider than the data in the field it is defined for, the width of that data column will be equal to the heading.
- Center data fields under column headings. You can override this by defining column headings that include leading or trailing blanks.

Heading Examples

Heading to look like:	ITEM statement used:
CUSTOMER NAME	ITEM CUSTNAME HEADING 'CUSTOMER' 'NAME'
CUSTOMER'S NAME	ITEM CUSTNAME HEADING 'CUSTOMER''S' 'NAME'
CUSTOMER NUMBER	ITEM CUSTNO HEADING 'CUSTOMER' ' NUMBER'

Notice in the second example that an apostrophe is embedded in the column heading by placing two single quotation marks where you want the apostrophe to appear.

Printing Data on a New Line

ITEM Command NEWLINE Keyword

The ITEM command NEWLINE keyword prints data fields on a new report line.

Note: Notice the column headings were eliminated with the FORMAT statement because the column headings associated with the output fields are stacked in the same manner as the data fields.

Figure 4-31 shows a query and the report produced by it, where the NEWLINE keyword is used to print different data fields on each of four print lines. Data items ADDLIN1, ADDLIN2, and BLKLINE each start an additional output line. Because BLKLINE is a temporary field with no value, a blank line is generated after the second address line.

Query

```
QUERY CUSTOMER
SET FIELD BLKLINE TYPE CHAR LENGTH 1
REPORT CUSTNAME ADDLIN1 ADDLIN2 BLKLINE
ITEM ADDLIN1 ADDLIN2 BLKLINE NEWLINE
FORMAT DATEPOS NO PAGEPOS NO HEADINGS NO
END REPORT
END QUERY
```

Output

```
TULSA TIMES
121 UNION ST.
TULSA, OKLAHOMA

TULSA TIMES
1217 STATE ST.
TULSA, OKLAHOMA

ESSEX ENTERPRISES
21120 232 ST.
LOS ANGELES, CA

ESSEX ENTERPRISES
735 AVENUE E
FARGO, NORTH DAKOTA
```

Figure 4-31 Use of the NEWLINE Keyword

Suppressing the Printing of Fields

ITEM Command NONPRINT Keyword

The ITEM command NONPRINT keyword suppresses the printing of fields.

- You can explicitly suppress the printing of any field on the REPORT statement using NONPRINT.
- If you specify a summarized field NONPRINT, only the summary values print; the detail values are suppressed.

You can explicitly suppress printing of fields specified on GROUP and ORDER statements using NONPRINT. They can also be implicitly not printed by not specifying them on the REPORT statement.

Both of the queries shown in Figure 4-32 produce the same report output. The report will be in order by ORDRDATE, but CUSTNO and ORDERNO will be the only fields printing.

Query

```
QUERY CUSTOMER
REPORT CUSTNO ORDERNO
ORDER BY ORDRDATE
TITLE 'REPORT SORTED BY DATE THAT DOES NOT PRINT'
END REPORT
END QUERY
```

Query

```
QUERY CUSTOMER
REPORT CUSTNO ORDERNO ORDRDATE
ORDER BY ORDRDATE
ITEM ORDRDATE NONPRINT
TITLE 'REPORT SORTED BY DATE THAT DOES NOT PRINT'
END REPORT
END QUERY
```

Figure 4-32 Queries With a Non-Printing Field

Creating a Subfile

To create a file that is a subset of the database being processed, use the EXTRACT command. For the command syntax, see the section EXTRACT Command.

Before you can use the EXTRACT command, your system administrator must provide authorization.

Check with your VISION:Inform system administrator as to whether your profile is authorized to use the EXTRACT command and, if so, what subfile names are available to your profile.

- You can report from extracted subfiles in future queries only after you catalog its file definition in the foreground library. The file definition is necessary for telling VISION:Bridge what fields exist on the subfile, what the key field is, and the length of the subfile records.
- If the subfile you extract is constructed of the same fields, in the same order, and has the same key as a subfile previously defined, no new file definition is necessary.

Specifying DEFINE in the EXTRACT Statement

Although creating file definitions is the responsibility of the VISION:Inform system administrator, you could be asked to specify DEFINE in the EXTRACT statement. This creates and outputs the specifications necessary to generate a file definition. The VISION:Inform system administrator can then use these specifications to catalog your file definition in the foreground library.

When you use the DEFINE keyword, the first field specified in the EXTRACT statement is assumed to be the key field for the subfile being created. Consult the VISION:Inform system administrator before using the DEFINE keyword.

EXTRACT Command

EXTRACT NAME subfile name ITEM fieldname(s) DEFINE IF expression

The EXTRACT command extracts data fields and outputs a subfile.

NAME subfile name — Enter the DD name for the subfile. This name must correspond to a subfile name in your profile.

ITEM fieldname(s) — Enter the fields to be output on the subfile. Fields exist on the subfile in exactly the same order as they are entered on the EXTRACT statement.

DEFINE — Writes the file definition statements to an existing data file. This is an optional keyword. Check with your VISION:Inform system administrator before using it.

IF expression — Enter any VISION:Bridge arithmetic, relational, or logical expression.

When creating subfiles, note:

- You cannot create structured subfiles. Subfiles are always sequential (flat) files.
- Fixed length fields must precede variable length fields.
- Fields exist on the subfile in the same order you specify on the EXTRACT statement.
- When you use the DEFINE keyword, the first field you list on the EXTRACT statement is defined as the key field for the subfile.
- You can include temporary fields on a subfile.

Selecting Data for a Subfile

You can place the EXTRACT statement anywhere within a query, except within a report group. A query containing only a QUERY, an EXTRACT, and an END QUERY is acceptable.

```
EXTRACT NAME SUBF030 ITEM CUSTNO,
ORDERNO ITEMNO
```

Use the example EXTRACT statement above to produce a subfile consisting of the fields CUSTNO, ORDERNO, and ITEMNO.

Before accessing this subfile in a query, the VISION:Inform system administrator creates a file definition for this subfile and promotes it to the foreground library.

Query

```
QUERY CUSTOMER
REPORT CUSTNO CUSTNAME ORDERNO ORDRDATE
  ITEM CUSTNO CUSTNAME ORDERNO ORDRDATE SPACES 0
  FORMAT HEADINGS NO DATEPOS NO PAGEPOS NO
END REPORT
EXTRACT SUBF010 CUSTNO CUSTNAME ORDERNO ORDRDATE
END QUERY
```

Figure 4-33 Creating Both a Report and Subfile

The query shown in Figure 4-33 produces both a report and a subfile of the same data. The second through fifth statements create the report, while the sixth statement creates the subfile. This verifies that the proper data is being output to the subfile. It is also helpful in defining the file.

Note:

- The ITEM command SPACES keyword is set to 0.
- This prevents *VISION:Bridge* from placing two spaces between the data fields on the report. No space is put between fields output to a subfile.

The output from the query above is shown in Figure 4-34. The records on the subfile are identical to the report.

Output

```
00001TULSA TIMES                        07321010228
00001TULSA TIMES                        08432010801
00013ESSEX ENTERPRISES                  01142010606
00013ESSEX ENTERPRISES                  03619010817
00013ESSEX ENTERPRISES                  09541011002
00028AMERICAN WIRE AND RING SERVICE     11211010316
00043BANK OF THE PEOPLE                 23468010820
00048STONEWELL STATE SYSTEMS            06573010101
00048STONEWELL STATE SYSTEMS            06781010710
00089GOING PLACES INSURANCE COMPANY     03921010417
00089GOING PLACES INSURANCE COMPANY     13842010610
00089GOING PLACES INSURANCE COMPANY     14821010815
00115SUNNY GRAPE GROWERS                00284010712
00115SUNNY GRAPE GROWERS                01391010525
```

Figure 4-34 The Created Subfile/Report

Example 1. The EXTRACT statement using the DEFINE keyword:

```
EXTRACT SUBF020 CUSTNAME CUSTPH SHIPNO,
        SHIPPER DEFINE
```

This statement produces a subfile containing the customer name, customer phone number, the shipment number, and the shipper.

Note: SUBF020 is the DD name for the subfile and not its data set name. It is also the file definition name for the subfile.

The DEFINE keyword produces a separate file of definition statements for use by the VISION:Inform system administrator in creating the online glossary seen in Figure 4-35.

```
SUBF020 - STANDARD OS FILE DATE CREATED - 01.205   TIME - 13.15.493

  NAME     TYPE  START  LENGTH  DEC  OUT-LEN  KEY   SEGNAME   PARENT
CUSTNAME  CHAR      1    30           30      1    S001L001  S001L001
CUSTPH    CHAR     31    10           10           S001L001  S001L001
SHIPNO    CHAR     41     4            4           S001L001  S001L001
SHIPPER   CHAR     45     3            3           S001L001  S001L001
***END OF DEFINITION***
```

Figure 4-35 A System Glossary of a Subfile

Example 2. The EXTRACT statement using the IF keyword:

```
EXTRACT SUBF040 CUSTNO ORDERNO ORDRDATE,
        IF ORDRDATE GT '010630'
```

This statement produces a subfile record only when the ORDRDATE field is greater than JUNE 30, 2001. The subfile records contain the CUSTNO, ORDERNO, and ORDRDATE fields.

Automatic Table Lookup

VISION:Bridge also provides automatic table lookup. Lookup tables contain information that correspond to database field values in some way. You use them to save space in databases by storing a short field value in the database. When you reference that short field, VISION:Bridge goes to the table and returns a different, usually longer field value, that corresponds to the field value in the database.

In the CUSTOMER database, for example, a field called INSTCODE represents the installation location code. It is one character long. Each value from 1 to 9 represents a different region of the country.

When you define the CUSTOMER database, you create a table that links each value of INSTCODE (the argument) to a region in the country. You retrieve the region by selecting the field INSTLOC (the result). These two fields are linked together in a table that automatically translates the one character code to the appropriate region.

- Figure 4-36 shows this sample customer table.
- Figure 4-37shows a report that references both INSTCODE and INSTLOC. As you can see, the values reported from INSTLOC are more useful for the report and the automatic table lookup saved space in the database.

In the table, suppose you have two columns and nine rows.

- The first column relates to the database INSTCODE field (the argument). It is a one-character field with a value from 1 to 9.
- The second column relates the database field INSTLOC (the result). It is a 16-character field and contains the actual region location name.

When your query requests the INSTCODE and INSTLOC fields, an automatic lookup occurs and the query retrieves the requested data.

1 NORTH REGION
2 SOUTH REGION
3 EAST REGION
4 WEST REGION
5 CENTRAL REGION
6 NORTHEAST REGION
7 SOUTHEAST REGION
8 NORTHWEST REGION
9 SOUTHWEST REGION

Figure 4-36 Sample Customer Table

For more information on automatic table lookup, contact your VISION:Inform system administrator.

Query

```
QUERY CUSTOMER
REPORT CUSTNO INSTNO INSTCODE INSTLOC
IF CUSTNO LT 00030
GROUP BY CUSTNO
END REPORT
END QUERY
```

Output

```
NOV 11, 2001                          PAGE 1

CUSTOMER   INSTNO  INSTCODE       INSTLOC
 NUMBER

00001      0106     8           NORTHWEST REGION
           0347     8           NORTHWEST REGION
00013      0329     9           SOUTHWEST REGION
           5186     8           NORTHWEST REGION
00028      1601     5           CENTRAL REGION
           2702     9           SOUTHWEST REGION
```

Figure 4-37 Using Automatic Table Lookup

Summary Functions Versus Summary Commands

Summary Functions (Built-in Summary Functions)

Summary functions operate at the record level.

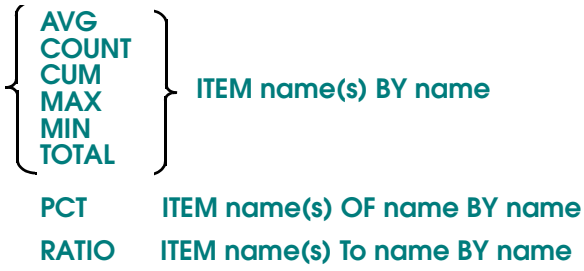
Syntax:

For detailed informations, see the section Built-In Summary Functions.

Summary Commands

Summary commands operate at the group or entire file level.

Syntax:



For detailed information, see the section Summary Commands.

Built-In Summary Functions

You can perform summaries on fields within structured database records using built-in summary functions. The built-in summary function is not applicable when processing records that are not structured (see Chapter 7, *Advanced Topics*). Once a built-in summary is performed, you can use the result for comparison, further calculation, or as output on a report.

The built-in summary functions are AVG, COUNT, MIN, MAX, and TOTAL. Although these functions are similar in nature to the summary commands available as part of the REPORT group (see the section Reporting), the syntax, output format, and level specification are all different. The major difference is that built-in summaries are computed for each report record output, where as the REPORT summaries apply to the entire report (they must be within a REPORT group).

Syntax

Built-in summary syntax requires that the field on which the summary is to be performed be enclosed in parentheses following the function requested. For example, AVG(SALARY).

Output Format

When specified for output on a report (REPORT statement), built-in summaries print in columns like data fields.

Level Specification

With built-in summaries, the level of summary is the parent segment unless otherwise specified. For example, TOTAL(INAMT BY CUSTNO).

If you specify any selection statement, data outside the selection context will not be available, either for built-in summaries, or reporting.

What Built-in Summary Functions Do

The built-in summary functions are AVG, COUNT, MAX, MIN and TOTAL.

AVG(fieldname)	Computes the average of all valid occurrences of the specified item within the context specified in the query.
COUNT(fieldname)	Counts the number of valid occurrences of the specified item within the context specified in the query.
MAX(fieldname)	Determines the maximum valid value of the specified item within the context specified in the query.
MIN(fieldname)	Determines the minimum valid value of the specified item within the context specified in the query.
TOTAL(fieldname)	Computes the sum of all valid occurrences of the specified item for the context specified in the query.

Figure 4-38 is referred to in the explanations of built-in functions for the remainder of this chapter. This figure illustrates the contents of two records from the CUSTOMER database.

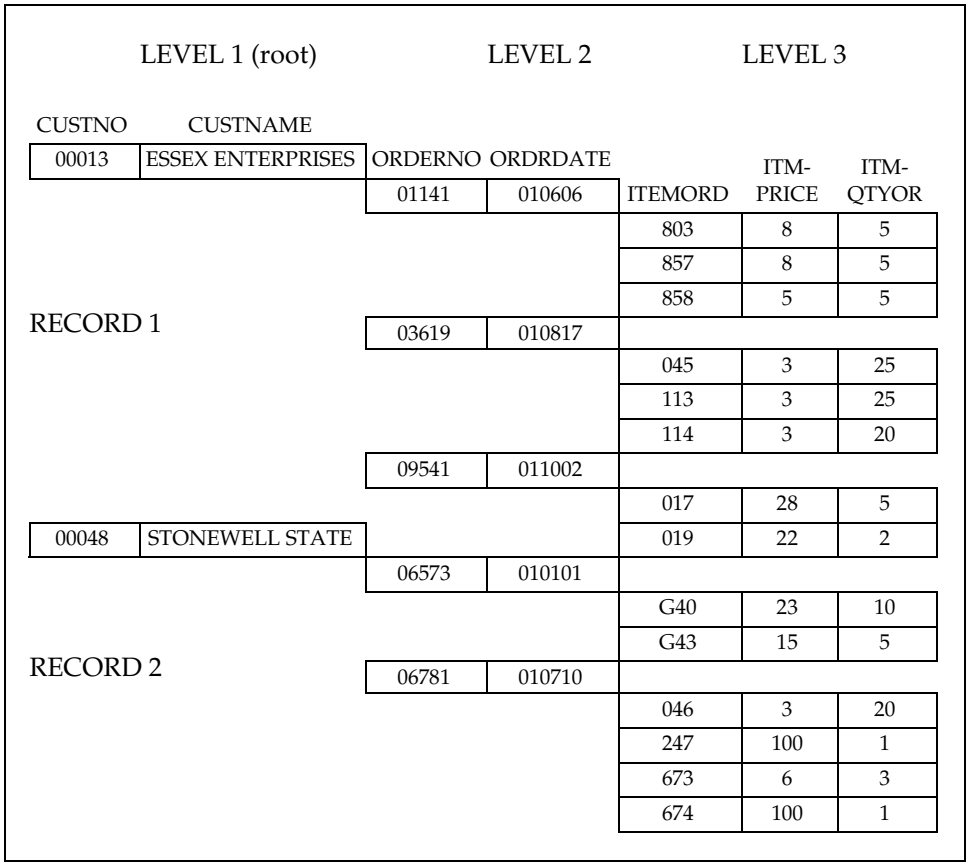


Figure 4-38 Two Records in the Customer Database

Reporting Built-In Summaries

Figure 4-39 shows the query and output for a count of order numbers for each customer.

Query

```
QUERY CUSTOMER
REPORT CUSTNO CUSTNAME COUNT(ORDERNO)
FORMAT DATEPOS NO PAGEPOS NO
END REPORT
END QUERY
```

Output

```
-----------------------------------------------------------------
CUSTOMER            CUSTOMER
 NUMBER               NAME
-----------------------------------------------------------------
00001    TULSA TIMES                              2
00013    ESSEX ENTERPRISES                        3
00028    AMERICAN WIRE AND RING SERVICE           1
00043    BANK OF THE PEOPLE                       1
00048    STONEWELL STATE SYSTEMS                  2
00089    GOING PLACES INSURANCE COMPANY           3
00115    SUNNY GRAPE GROWERS                      2
```

Figure 4-39 Reporting Built-In Summaries

Figure 4-40 shows a method of adding column headings to built-in summaries.

- You add a column heading by setting a temporary field (ORDCNT) to the summary value with the LET statement, then use the ITEM statement to define the heading text.
- You specify the temporary field on the REPORT statement.

Query

```
QUERY CUSTOMER
LET ORDCNT = COUNT(ORDERNO)
REPORT CUSTNO CUSTNAME ORDCNT
ITEM ORDCNT HEADING 'ORDER' 'COUNT'
FORMAT DATEPOS NO PAGEPOS NO
END REPORT
END QUERY
```

Output

```
-------------------------------------------------------------
CUSTOMER    CUSTOMER                          ORDER
 NUMBER       NAME                            COUNT
-------------------------------------------------------------
00001    TULSA TIMES                             2
00013    ESSEX ENTERPRISES                       3
00028    AMERICAN WIRE AND RING SERVICE          1
00043    BANK OF THE PEOPLE                      1
00048    STONEWELL STATE SYSTEMS                 2
00089    GOING PLACES INSURANCE COMPANY          3
00115    SUNNY GRAPE GROWERS                     2
```

Figure 4-40 Adding Column Headings to Built-In Summaries

Looping Considerations When Using Built-In Summaries

Because you can only specify fields from lower level segments for built-in summaries, a looping process takes place. For lower level segments and looping detail, see Chapter 7, *Advanced Topics*.

- If you specify a built-in summary function, the summarization is computed for each occurrence of the summarized field's parent segment unless otherwise specified.

  For example, if the field you specify for a built-in summary function is in level 3, the summarization is done for each of the level 2 segment occurrences in its path. Figure 4-41 illustrates this process. For changing the level of summary, see Figure 4-42.

- Make the built-in summary specification the first reference to the segment in which the field to be summarized resides.

  For example, if the built-in summary field is in level 3, place all fields referenced in the query prior to the built-in summary function in the root or level 2.

The following examples clarify the effect that looping has on built-in summaries.

In Figure 4-41, a total item price is taken for each order, of each customer. This total is taken because item price is in level 3 of the database, and the order segment is its parent at level 2. The total is computed for each occurrence of the order segment even though the order segment is not reported. The CUSTNO field is in the root segment of the database.

Query

```
QUERY CUSTOMER
REPORT CUSTNO TOTAL(ITMPRICE)
FORMAT DATEPOS NO PAGEPOS NO
END REPORT
END QUERY
```

Output

```
----------------------------------------------------------------
CUSTOMER
 NUMBER
----------------------------------------------------------------
00001              12
00001               6
00013              21
00013               9
00013              50
00028              22
00043              28
00048              38
00048             209
00089               7
00089             100
00089              15
00115               9
00115              50
```

Figure 4-41 Totals Taken with Built-In Summaries

Figure 4-42 shows the total ITMPRICE (level 3) computed for each CUSTOMER (level 1).

Query

```
QUERY CUSTOMER
REPORT CUSTNO TOTAL(ITMPRICE BY CUSTNO)
GROUP BY CUSTNO FORMAT DATEPOS NO PAGEPOS NO
END REPORT
END QUERY
```

Output

```
---------------------------------------------------------------
CUSTOMER
 NUMBER
---------------------------------------------------------------
00001                 18
00013                 80
00028                 22
00043                 28
00048                247
00089                122
00115                 59
```

Figure 4-42 Level 2 Totals

Built-In Summaries for Data Selection

Use built-in summaries in IF expressions for data selection. The summary is calculated first and then the comparison is made.

For example, if AVG(QUANTITY) GT 10 is the expression, then the QUANTITY field is averaged and the result is compared to 10. Figure 4-43 is an example of using a built-in summary for data selection. Notice that only those orders with an average quantity greater than 5 are output on the report.

Query

```
QUERY CUSTOMER
REPORT ORDERNO ORDRDATE,
AVG(ITMQTYOR) IF AVG(ITMQTYOR) GT 5
FORMAT DATEPOS NO PAGEPOS NO
END REPORT
END QUERY
```

Output

AVG (ITMQTYOR)

```
---------------------------------------------------------------
```

Figure 4-43 Using a Built-In Summary in an IF Expression

```
ORDER          ORDER
NUMBER         DATE
-------------------------------------------------------------------
07321          010228          10
08432          010801           6
03619          010817          23
06573          010101           7
00284          010712          13
```

Figure 4-43 Using a Built-In Summary in an IF Expression

Summary Commands

You can obtain summary and statistical information about the data on which you are reporting.

In VISION:Bridge there are two ways to prepare this information:

- The built-in summary functions (discussed in the section Built-In Summary Functions).
- The summary commands (discussed in this section).

You can request summaries at group levels or for the entire report. You can also request printing of only summary information. You always use summary commands within a REPORT group.

Note:

- Built-in summary functions operate at the record level.
- Summary commands operate at the group or entire file level.

The types of summarization are listed below:

AVG	Computes an average of all valid occurrences of each field specified.	Page 4-92
COUNT	Counts valid occurrences of each field specified.	Page 4-92
CUM	Reports a cumulative total of valid occurrences of fields specified.	Page 4-92
MAX	Reports the maximum valid value of the fields specified.	Page 4-92
MIN	Reports the minimum valid value of fields specified.	Page 4-92
PCT	Computes a percentage of one field to another.	Page 4-92
RATIO	Reports a ratio of one field value to another.	Page 4-93
TOTAL	Reports a total of all valid occurrences of each field specified.	Page 4-92

Associated REPORT Command Summary Keywords

SUMMARY Add this optional keyword to the REPORT statement to request that only summary information be output on the report.

The SUMMARY keyword suppresses printing of all detail lines and prints only summaries of the fields you specify in the Summary statements.

- You must use at least one of the summary commands (AVG, COUNT, CUM, MAX, MIN, PCT, RATIO, or TOTAL) to receive a summary report.
- See Figure 4-47 for an example of a summary only report. For syntax, see the section REPORT Command.

GRANDSUM Add this optional keyword the REPORT statement to generate automatic grand summaries on the report.

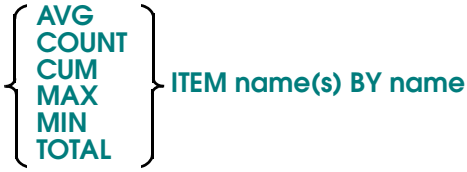
- You must specify at least one summary BY for the report.
- They appear on a separate last page of the report.

An example is shown in Figure 4-45. For the syntax, see the section REPORT Command.

EMPTYFLD Add this optional keyword to the REPORT statement to include or exclude zero value numeric fields and blank character fields from COUNT and AVG summaries.

Place the value INCLUDE or EXCLUDE after the EMPTYFLD keyword. For syntax, see the section REPORT Command.

Summary Commands — AVG, COUNT, CUM, MAX, MIN, and TOTAL Commands



The six summary commands listed above all have the same syntax. The parameters work the same with each command.

ITEM name(s) Data fields to be summarized. You can enter more than one.

- List these fields on the REPORT statement.
- Do not list these fields on the GROUP statement.
- ITEM is an optional keyword.

BY name Calculate and display a summary at a group level.

- If you specify a field name after the BY keyword, you must also specify it on the GROUP statement.
- If you omit the BY keyword and the group level field, the summary is computed for the entire report.

Summary Command — PCT Command

PCT ITEM name(s) OF name BY name

Computes a percentage of one field to another. The PCT command is slightly different than summary commands and has its own syntax.

ITEM name(s) The names of the numerator fields in the percent calculation.

- List these fields on the REPORT statement, but not on the GROUP statement.
- Do not use the same field as numerator in multiple PCT statements. You can use temporary fields of equal value to accomplish this.
- ITEM is an optional keyword.

OF name Indicates the name of the denominator field for the percentage calculation.

- List this field on the REPORT statement, but not on the GROUP statement.
- OF is a required entry on the PCT statement.

BY name When entered, with the name of a field from the GROUP statement, the BY keyword computes and reports summaries at the group level for the specified field.

If you omit the BY parameter and name, the entire report is summarized.

Summary Command — RATIO Command

RATIO ITEM name(s) TO name BY name

Reports a ratio of one field value to another.

ITEM name(s) The numerator fields in the ratio.

- Place these fields on the REPORT statement, but not on the GROUP statement.
- Do not use the same field as numerator in multiple RATIO statements. You create temporary fields of equal value to accomplish this.
- ITEM is an optional keyword.

TO name The denominator field for the ratio.

- List this field on the REPORT statement, but not on the GROUP statement.
- TO is a required keyword.

BY name The name of a field from the GROUP statement to indicate the level at which summaries are to be taken. If you omit the BY keyword and name, the entire report is summarized.

Same Field Numerator in PCT and RATIO

If you use the same field as the numerator in both the PCT and RATIO statements in a single report, both statements have the same denominator field name.

Note: Observe how the columns of data are moved to the right to make room for the words GRAND COUNT.

Example 1. Summary Command — COUNT

Figure 4-44 illustrates a query that counts the number of order numbers on the CUSTOMER database.

Query

```
QUERY CUSTOMER
REPORT CUSTNO ORDERNO
COUNT ORDERNO
END REPORT
END QUERY
```

Output

```
MAR 08, 2001                                          PAGE 1
-------------------------------------------------------------
              CUSTOMER    ORDER
               NUMBER     NUMBER
-------------------------------------------------------------
              00001       07321
              00001       08432
              00013       01142
              00013       03619
              00013       09541
              00028       11211
              00043       23468
              00048       06573
              00048       06781
              00089       03921
              00089       13842
              00089       14821
              00115       00284
              00115       01391
GRAND COUNT               14
```

Figure 4-44 Using the COUNT Command

Example 2. Using COUNT Command and a GROUP Command

Note: Observe that GRANDSUM has been added to the REPORT statement to generate a grand count, as well as a custno count.

The query in Figure 4-45 is simply a modification of the query in Figure 4-44. A GROUP statement has been added: GROUP BY CUSTNO. This statement eliminates the repetition of customer numbers. The addition of BY CUSTNO to the COUNT statement causes the orders to be counted for each customer.

Query

```
QUERY CUSTOMER
REPORT CUSTNO ORDERNO GRANDSUM
GROUP BY CUSTNO
COUNT ORDERNO BY CUSTNO
END REPORT
END QUERY
```

Output

```
JUN 16, 2001                                                    PAGE 4
-------------------------------------------------------------------------
                  CUSTNO  ORDERNO
-------------------------------------------------------------------------
CUSTNO  COUNT     00048         5

                  00050     50001
                            50002

CUSTNO  COUNT     00050         2

                  00089     89001
                            89002
                            89003

CUSTNO  COUNT     00089         3

                  00115     11501
                            11502
                            11503
                            11504

JUN 16, 2001                                                    PAGE 8
-------------------------------------------------------------------------
                  CUSTNO  ORDERNO
-------------------------------------------------------------------------
GRAND   COUNT                 65
```

Figure 4-45 Using the GROUP and COUNT Statements

Example 3. Multiple Summary Statements

Figure 4-46 is an example of using multiple summary statements in a query. Note the TOTAL and PCT commands.

Query

```
QUERY CUSTOMER
REPORT CUSTNO INAMT FRTCOST
GROUP BY CUSTNO
TOTAL FRTCOST BY CUSTNO
PCT FRTCOST OF INAMT BY CUSTNO
FORMAT WIDTH 80
TITLE 'USING TWO SUMMARY FUNCTIONS'
END REPORT
END QUERY
```

Output

```
NOV 03, 2001       USING TWO SUMMARY FUNCTIONS                  PAGE 1
-------------------------------------------------------------------------
                CUSTOMER    INVOICE      FREIGHT
                 NUMBER     AMOUNT       COST
-------------------------------------------------------------------------
                00001        270.00         3.00
                              36.00         2.00

CUSTNO  TOTAL  00001                        5.00
INAMT   PCT.                               1.63%

                00013        105.00        10.00
                             210.00         3.00
                             140.00         2.00

CUSTNO  TOTAL  00013                       15.00
INAMT   PCT.                               3.29%

                00028         22.00         2.00

CUSTNO  TOTAL  00028                        2.00
INAMT   PCT.                               9.09%

                00043         84.00         4.00

CUSTNO  TOTAL  00043                        4.00
INAMT   PCT.                               4.76%

                00048        305.00         6.00
                             260.00         2.00
                              18.00         1.00

CUSTNO  TOTAL  00048                        9.00
INAMT   PCT.                               1.54%
```

Figure 4-46 Using Multiple Summaries

Example 4. Summaries Only

Figure 4-46 is the same as Figure 4-47, except for the SUMMARY keyword on the REPORT statement. This suppresses printing of all detail lines and prints only the summaries of the fields specified in the summary statements, TOTAL and PCT.

Query

```
QUERY CUSTOMER
REPORT CUSTNO INAMT FRTCOST SUMMARY
  GROUP BY CUSTNO
  TOTAL FRTCOST BY CUSTNO
  PCT FRTCOST OF INAMT BY CUSTNO
  FORMAT WIDTH 80
  TITLE 'A SUMMARY ONLY REPORT'
END REPORT
END QUERY
```

Output

```
NOV 03, 2001      A SUMMARY ONLY REPORT                      PAGE 1
------------------------------------------------------------------------
                 CUSTOMER    INVOICE    FREIGHT
                  NUMBER     AMOUNT      COST
------------------------------------------------------------------------
CUSTNO  TOTAL    00001                    5.00
INAMT   PCT.                             1.63%

CUSTNO  TOTAL    00013                   15.00
INAMT   PCT.                             3.29%

CUSTNO  TOTAL    00028                    2.00
INAMT   PCT.                              9.9%

CUSTNO  TOTAL    00043                    4.00
INAMT   PCT.                             4.76%

CUSTNO  TOTAL    00048                    9.00
INAMT   PCT.                             1.54%

CUSTNO  TOTAL    00089                   25.00
INAMT   PCT.                            33.33%

CUSTNO  TOTAL    00115                   13.00
INAMT   PCT.                             5.50%
```

Figure 4-47 A Summary Only Report

Calling External Routines

Use the VISION:Inform CALL facility to initiate external routines from your queries. Through use of the CALL command, queries pass parameters to external routines, which transform the parameters and return results to the queries. You can use these results like any other temporary or database field.

CALL Command

CALL NAME user-routine USING parameters

Note: Make sure that the size and field type of the *VISION:Bridge* parameters agree with those of the user written routine. If the parameters do not agree, an error can occur.

The CALL command executes a user written program that is stored outside of VISION:Bridge.

NAME user-routine Provides VISION:Bridge with the name of the routine to be called. Specify a name up to eight characters long, beginning with an alphabetic character, and followed by alphanumeric characters.

USING parameters Provides the input and output fields the routine needs for execution.

- Parameters are positional. Place parameters in the order that the user routine expects.
- Parameters can be fixed length database fields, temporary fields created with LET or SET statements, numeric constants, or character constants.
- You can specify numeric constants up to 15 digits.
- Make character constants 31 characters or less and enclose them in single quotation marks.

See the *VISION:Inform System Administrator Guide* for more information on the CALL statement parameters.

Example.

```
CALL CONVERT USING SCORE METERS
```

This CALL statement tells VISION:Inform or VISION:Bridge to start the external routine CONVERT using two fields, SCORE and METERS as parameters.

Chapter

5

Using Immediate Response

What Can Immediate Response Do?

With Immediate Response, you can access data stored in databases and files without having to be concerned with their structure or data format. You retrieve data by specifying entire queries, or by specifying output generating statements.

You normally specify queries and output generating statements for processing online. The data you request displays immediately on your terminal in an easy-to-read formatted display. You can also process queries and statements in batch using the Batch Simulator and print them on a system printer.

The features of Immediate Response are as follows:

- A simple, easy-to-learn query language.
- An online syntax checker, which checks queries and provides helpful diagnostic messages.
- Intelligent data selection capabilities.
- Displays sorted in either ascending or descending order.
- Summary command statements with counts, totals, accumulations, and averages.
- Built-in summary functions for data selection based upon the summarization of a field.
- Arithmetic processing commands, facilitating calculations and the selection of data based upon arithmetic calculations.
- An interactive full screen editor with the capability to save queries for later modification or execution.
- An interactive processor, processing output statements and queries as they are entered.
- Support of DL/I databases and VSAM files under IMS and CICS.

- A logical data view capability with definition of logical views involving more than one file or database.
- Multi-level security through the use of individual user profiles, which define access to data at the user, database, segment and field, or individual record levels.
- The ability to process queries interactively.

System Overview

With Immediate Response, you can develop queries to obtain the data you want to see. The queries you create consist of Immediate Response commands that tell Immediate Response what you want and how you want to see it.

By using Immediate Response, you can obtain data without concerning yourself with:

- The type of file in which it is stored.
- How to get the data from the file.
- How to execute your query.
- Specifics about the environment where your query executes.

Separating you from the environment is one of the major strengths of Immediate Response. With Immediate Response, the environment becomes the responsibility of your VISION:Inform system administrator, freeing you to satisfy your processing and reporting needs with a minimum of effort.

Figure 5-1 shows Immediate Response within your computer.

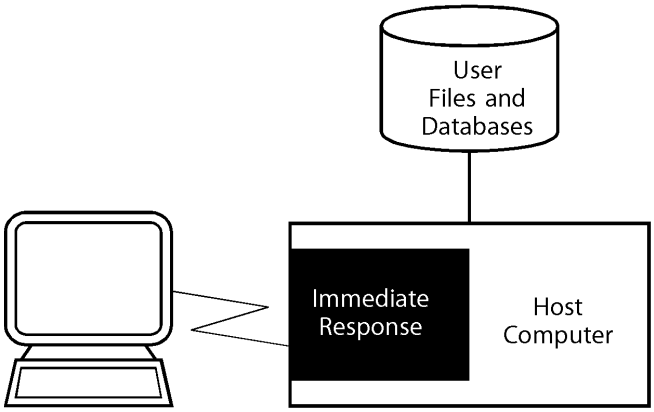


Figure 5-1 Immediate Response in Your Computer

This chapter gives you a better understanding of how Immediate Response works. It describes how your queries are processed and your reports handled. You will also find out about the various components of the system, and how they work with one another to process your queries and return your reports.

How Immediate Response Works

Note: Immediate Response queries are not processed by Background Processors.

The basic operating premise of Immediate Response is that you create queries through an online terminal. The queries you create are processed immediately in the same online area that takes care of talking to your terminal. As soon as the output completes, it displays on your terminal.

Figure 5-2 shows how Immediate Response operates.

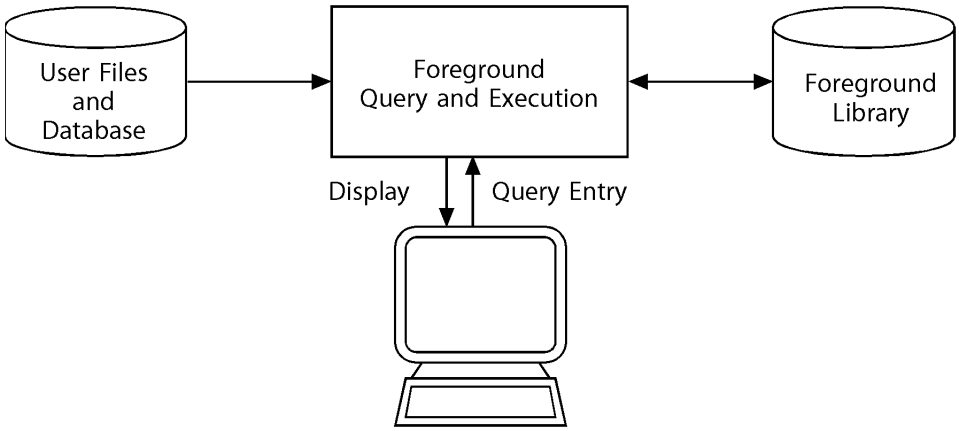


Figure 5-2 How Immediate Response Operates

System Components

Immediate Response consists of the following components:

- The Foreground Processor
- The Foreground Library

Figure 5-3 illustrates the components of Immediate Response. These components are briefly described in the following subsections.

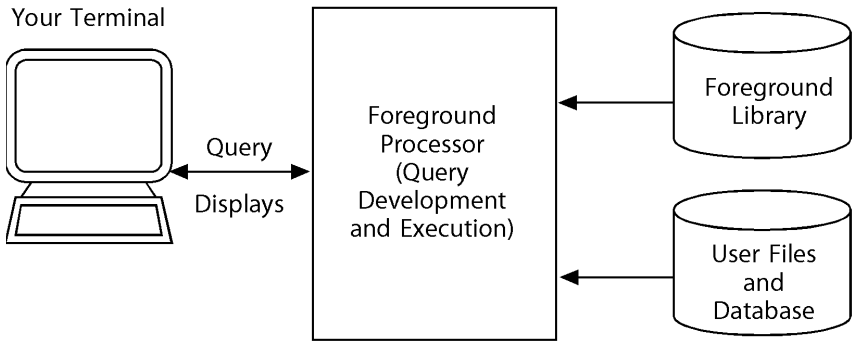


Figure 5-3 Immediate Response Components

Foreground Processor

The Foreground Processor is the component that maintains a dialog with you when you are at your terminal. You provide instructions, in the form of commands, to the Foreground Processor. The Foreground Processor is an application program. The environment in which it operates treats it just like other application programs.

Start the Foreground Processor in the same way you start other application programs in the particular environment. Then, you can create queries, edit them, save them, and process them online.

To process your queries online, the Foreground Processor uses an Immediate Response processor program. The processor retrieves the data you request, performs the necessary manipulations, and formats it into a screen display.

The Foreground Processor checks your queries for syntax and internal consistency.

- Syntax problems occur when you make a mistake entering an Immediate Response command. You will learn more about commands and their syntax later in this section.
- Internal consistency problems usually occur when you request some item of data that is not defined.

The Foreground Processor displays a diagnostic message when it detects a problem. Diagnostic messages consist of one or more lines of text describing the problem. You use the Foreground Processor to correct the problem, and resubmit your query.

In addition to diagnostic messages, the Foreground Processor also issues two other types of messages.

- Informational messages provide information about your online session. Unlike diagnostic messages, which require you to take some action, informational messages require no action on your part.
- Prompting messages displays during an online session when the Foreground Processor wants you to provide instructions. Prompting messages require you to take some action.

When these messages display on your terminal, you can easily identify them. Prompting messages are always followed by a colon, while code letters and two asterisks precede informational and diagnostic messages. You will find a complete list of these messages in *VISION:Inform Messages and Codes*.

Foreground Library

The Foreground Processor uses the foreground library for storing the queries you create from your terminal. You can create a query once, save it, and call upon it numerous times in the future to get the data you want.

This library also contains information, called file definitions, about the databases and files from which you can obtain information. File definitions describe the contents of the databases and files your queries can access. Your VISION:Inform system administrator gives this information to Immediate Response, so that all you have to do in your query is request the fields you want to see.

The Foreground Processor uses these file definitions to verify that the fields you request are available. If the information about the database or data field cannot be found in the foreground library, the Foreground Processor issues a diagnostic message.

Another important function of the library is to provide security to your system. Only you and other authorized users can use Immediate Response through a user profile. It contains information about which users can use the system, the files available to them, the data available to them and what kinds of things they can do. The VISION:Inform system administrator defines and maintains this user profile.

Language Overview

Develop Immediate Response queries by specifying a series of English-like statements. Using these statements, you tell Immediate Response what to do.

The statements you specify resemble sentences; but unlike sentences, they do not require stringent punctuation. The only punctuation you use are blank spaces and commas. Sometimes, you use commas to separate items for clarity. You always use commas to continue a statement on the next line.

This section describes the statements you use, their various components, and the rules for their use.

Immediate Response Statement Syntax

A command is an order to Immediate Response (such as QUERY a database or DISPLAY data). For Immediate Response to recognize your command and its keywords, follow the syntax rules for using the commands and keywords.

The general syntax for a statement is:

Text, which follows the syntax, describes each element in a command statement.

COMMAND

Begin each statement with a command.

- Commands can be followed by one or more parameters.
- Some commands do not have parameters.

KEYWORD operand

Parameters provide more information for the command.

- Parameters can be composed of one or more keywords or keywords followed by keyword operands.
- Parameters are optional or required.
- Some keywords do not have operands.

Keywords

Keyword modify the command.

- Keywords specify options of the command.
- Keywords identify fields or values that the command must act upon.

Most keywords are optional; you can leave them out if you specify the various fields and values in the order the command expects (their positional order). You can specify the above statement without a keyword as follows:

```
DISPLAY CUSTNO CUSTNAME
```

Positional Order

The order in which keywords are presented for each command is the default order. This is important because you can list keywords in any order. But if you change the order, then all keywords for the command are required.

As you become more familiar with the commands, you will know when you can leave out the keywords and when they are required. Keywords are helpful when you are not familiar with the positional order of a command.

By specifying the keyword, you can specify the parts of the statement in any order following the command. For instance, the following statement is specified in positional order without keywords and then with keywords in any order:

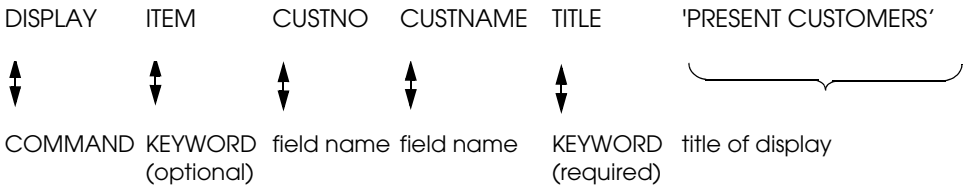
```
COUNT ORDERNO BY CUSTNO  (positional order)
```

or

```
COUNT BY CUSTNO ITEM ORDERNO  (any order with keyword)
```

Example of Immediate Response Statement Syntax

This command is part of the Immediate Response query that produced the display in Figure 5-14:



COMMAND The DISPLAY command tells Immediate Response to display data.

KEYWORD Tells Immediate Response what is coming next, in this case, data items.

field name Tells Immediate Response which data items to display from the database. In this case, the number of each customer (CUSTNO) and the associated name of each customer (CUSTNAME) display.

KEYWORD Tells Immediate Response what is coming next, in this case a title for the display.

title of display With the previous entry of the keyword TITLE, tells Immediate Response that the text in single quotation marks is the title that will appear at the top of each page of this display. Make the length of a title from 1 to the number of characters that will display on the screen.

Altering Command Syntax

The following statement produces a display exactly like the one shown in Figure 5-14. Notice the keyword ITEM is left out. ITEM is an optional entry only if the field names immediately follow the DISPLAY command.

```
DISPLAY  CUSTNO  CUSTNAME,
         TITLE 'PRESENT CUSTOMERS'
```

The next statement also gives the same results as in Figure 5-14. If you use all the keywords (TITLE, ITEM), the keyword and its associated information can be put in any order after the command. You must use optional keywords if you change the order. For the DISPLAY command, ITEM is the only optional keyword.

```
DISPLAY TITLE 'PRESENT CUSTOMERS',
        ITEM CUSTNO CUSTNAME
```

Expressions

You can use relational and arithmetic expressions by themselves or as part of another statement.

Relational Expressions

The following illustrates the use of relational expressions in a statement:

```
DISPLAY CUSTNO CUSTNAME IF CUSTNAME EQ 'TULSA TIMES'
```

IF designates the start of the relational expression. Relational expressions test for a specific condition. In the DISPLAY statement, the display is only generated if the relational expression is true.

Arithmetic Expressions

The following statement illustrates an arithmetic expressions:

```
DISPLAY CUSTNO CUSTNAME ITMQTYOR * 1.1
```

The presence of the arithmetic symbols (+, -, *, /) tells you that the expression performs an arithmetic operation. You can use an arithmetic expression in a statement with other expressions or by itself in the following way:

```
LET ITMTOT = ITMQTYOR * 1.1
```

This LET statement places the result of the arithmetic operation (ITMQTYOR * 1.1) in a new field called ITMTOT.

Constants

In the previous examples, notice the use of the literal constant 'TULSA TIMES' and a numeric constant 1.1.

Constants are data values that you specify in a statement. The value of a constant never changes. Constants differ from variables, whose values can change.

Combining Expressions

You can combine both relational and arithmetic expressions in a single statement. The following examples illustrate combining these types of expressions:

```
DISPLAY CUSTNAME IF ITEMQTYOR * 1.1 GT INAMT
```

In this example, CUSTNAME is only reported when the result of the multiplication is greater than the value in the field INAMT.

```
LET ITEMTOT = ITMQTYOR * 1.1,
    IF CUSTNAME EQ 'TULSA TIMES'
```

In the second example, the result of the multiplication is placed in the field ITEMTOT if the customer is the TULSA TIMES.

You can use many combinations of expressions. All of them are described in detail in the sections that follow. Appendix D, *Reference Summary* contains detailed specifications for each of the Immediate Response commands.

Queries

You are now familiar with the format of the statements that make up queries.

Creating an Immediate Response Query

Immediate Response can select, calculate, and display data from a database. The DISPLAY or QUERY commands tell Immediate Response what you want to display and how you want it displayed.

You use query commands and their keywords to make up query statements. The statements that make up an Immediate Response query include the QUERY command, one or more other query commands (such as DISPLAY, SELECT, LET, SET), and an END QUERY command.

The QUERY command starts a query and specifies the database to be used. The END QUERY command terminates a query.

Note: In the examples which follow, the statement, command, or keyword, which is the subject of interest, appears in **bold** type.

The following Figure 5-4 illustrates a typical Immediate Response query.

```
QUERY CUSTOMER
SELECT CUSTNO IF ORDERNO GE 0100
DISPLAY CUSTNO CUSTNAME ORDERNO,
TITLE 'CUSTOMER ORDER LIST'
END SELECT
END QUERY
```

Figure 5-4 Example of a Typical Immediate Response Query

In the example, you can see that queries always begin with a QUERY command and end with an END QUERY command.

Between these two commands, you specify the statements that determine the data you want to see, how you want to manipulate it, and the order you want to display on your terminal screen.

SELECT Groups

Queries contain groups of statements that perform particular functions. To select the data you want to see, there is the SELECT group. It begins with a SELECT command and ends with an END SELECT command. All the statements in the SELECT group are applied to the data made available by the SELECT statement.

Executing Immediate Response Queries

The statements in your query are executed immediately by the Foreground Processor. You can execute the entire query, or it can execute as soon as you enter an output type statement (for example, DISPLAY).

Executing your query online creates an output display that appears on your terminal. To obtain a printed copy of your output, you can input your query to the Batch Simulator (see your VISION:Inform system administrator for details).

You decide the type of processing that best suits your needs. How urgently you need to see the data and whether you need printed output are the influencing factors in most cases.

Introducing Immediate Response Commands

This section contains a list and brief description of the Immediate Response commands. You can also find descriptions of each command in Appendix D, *Reference Summary*.

Note:

- These are Immediate Response query commands.
- For Immediate Response system commands, see the section System Mode (?:).
- For Immediate Response immediate mode commands, see the section Immediate Mode (IMMED:).
- For Immediate Response checkpoint mode commands, see the section Checkpoint Mode (CKP:).

These commands, as well as some other words, are reserved keywords and are not usually used as field names. See Appendix D, *Reference Summary*, or the keywords in the *VISION:Inform Reference Summary* for the complete list of reserved words, as well as a technique that uses these words as field names.

Command	Description
AVG	Computes and reports the average of all valid occurrences of the specified fields.
COUNT	Counts and reports the number of valid occurrences of the specified fields.
CUM	Computes and reports the cumulative total of all valid occurrences of the specified fields.
DISPLAY	Tells Immediate Response to generate a display for the fields listed on the statement based upon the criteria specified.
END IMMED	Ends the Immediate Response immediate mode and returns to Immediate Response system mode.
END QUERY	Designates the end of a query.
END SELECT	Designates the end of a SELECT group.
IMMED	Puts you in Immediate Response immediate mode, from which you can enter immediate mode commands.
LET	Assigns a value to the field (database or temporary) following the LET command.
QUERY	Tells Immediate Response what database to use.

Command	Description
SELECT	Establishes the range of database segments or fields that your query can access.
SET	Defines a temporary field within your query.
SYSTEM	Returns you to Immediate Response system mode.
TOTAL	Computes and reports the grand total of all the valid occurrences of all the fields listed on the command.

A Sample Immediate Response Session

Figure 5-5 shows a sample Immediate Response report and the statements that produced it. In this section, you can see every command statement and keystroke required to produce the query and display in Figure 5-5.

There are two important things to take note of in Figure 5-5.

- First, there are only two statements in the query.
- Second, the title of the report is centered at the top of the screen. The information is listed from left to right across the page in the order specified in the DISPLAY statement. This is the default report format. The field names appear over each column of data.

Query

```
query customer
display custno custname custph,
        title 'customer list'
```

Output

```
                  CUSTOMER LIST

CUSTNO            CUSTNAME               CUSTPH

00001  TULSA TIMES                      8185551620
00013  ESSEX ENTERPRISES                2335556043
00028  AMERICAN RING AND WIRE SERVICE   0305550101
00043  BANK OF THE PEOPLE               8325554693
00048  STONEWELL STATE SYSTEMS          4345557000
00089  GOING PLACES INSURANCE COMPANY   7385559123
00115  SUNNY GRAPE GROWERS              2345553966
NR06** END OF DISPLAY,DB COUNT=       8,PAGE COUNT=    1
```

Figure 5-5 An Immediate Response Query with the DISPLAY Command

Before You Begin a Session

To begin an Immediate Response session, you need a user ID. If security is an issue at your installation, you need a password.

Your VISION:Inform system administrator sets up your profile with your user ID, password, and an internal list of the types of data available to you, such as databases, segments, and fields within a database. The information in your profile pertains to a specific user ID. However, multiple users can log on simultaneously with the same user ID. You cannot use Immediate Response without a user ID and profile.

Starting the Session

Immediate Response is a computer program that can run in several different environments. To you, the user, the environment is transparent. That is, Immediate Response looks the same and all the commands and features work the same, regardless of the environment in which it runs.

What differs is, how you tell the environment to use Immediate Response. How you do this depends on the environment and the particular computer installation you are using. This means that you enter the appropriate command for VISION:Inform at your terminal. Your VISION:Inform system administrator can help you with this.

Once you tell the computer you want to use VISION:Inform, the Logon panel appears, as shown in Figure 5-6. Log on now to VISION:Inform.

```
Logon2                    Computer Associates - Logon


                 Welcome to VISION:Inform Release 4.0.

                  Please Enter Your User ID and Password:

                     User ID  . . . . ______
                     Password . . . .         (if password protected)


               +-------------------------------------------------+
               |  Proprietary and confidential information of    |
               |    Computer Associates International, Inc.      |
               |  Use restricted by written license agreement.   |
               |                (c) 1980, 2001                   |
               |    Computer Associates International, Inc.      |
               |  as an unpublished work. All rights reserved.   |
               +-------------------------------------------------+

Command ===>
F1 =Help      F3 =Exit
```

Figure 5-6 Logon Panel

Note: Option 5 (Quick Query Immediate Response) assists you in the development of Immediate Response Queries using Quick Query.

After entering your user ID and password (provided by your VISION:Inform system administrator) the Main Menu, shown in Figure 5-7 displays.

```
 Menu2                  Computer Associates - Main Menu

7 Enter one of the following VISION:Inform or VISION:Bridge Options:


  VISION:Inform Options:

  1. Operation Facilities               (Background Processor Status)
  2. Administration Facilities          (Profile Development)
  3. Report Facilities                  (Report Handling)

  VISION:Bridge Options:

  4. Quick Query                        (Assisted Query Development)
  5. Quick Query Immediate Response     (Assisted Query Development)
  6. Standard Query Processing          (Submit, Delete, Edit
                                         Queries and Stmts)
  7. Immediate Response Query Processing (Run Queries and Immed Mode)

Command ===>
F1 =Help     F12=Cancel
```

Figure 5-7 Main Menu

Select an option from the Main Menu. Immediate Response commands are available in Option 6 (Standard Query Processing) and Option 7 (Immediate Response Query Processing). For a description of other options, see Chapter 1, *Introduction*.

Option 6 (Standard Query Processing) displays the Source Processing panel. From this panel:

- Create and update queries.
- Save queries to the foreground library.
- Delete queries from the foreground library.

Select Option 7 (Immediate Response Query Processing) for Immediate Response Query Processing (as shown in Figure 5-7) and the system responds with the ?: prompt on a new panel. You are now ready to begin your Immediate Response session.

The Immediate Response Session

In this session, you create a new query, submit the query for execution, and then exit the session.

In the following example:

- The statements you enter at the terminal are shown in lower case **bold** type.
- The Immediate Response responses called prompts, are shown in upper case letters and include the symbols followed by colons.
  - ?: means that you are in Immediate Response system mode.
  - IMMED: means that you are in Immediate Response immediate mode.
- The symbol < represents pressing Enter on the terminal keyboard.

The example describes each step.

Note: User entries appear in **bold** type.

Statements	Explanation
`?:`	The system initially responds to an entry from the Main Menu with the system prompt (?:).
`immed <`	The IMMED command tells Immediate Response that you want to enter a query at your terminal.
`IMMED:`	■ Immediate Response switches to immediate mode and displays the immediate mode prompt, IMMED:. ■ In Immediate Response mode, the processor checks each statement you enter for proper command syntax and internal consistency of the statements.
`query customer <`	When you see the IMMED: prompt, you enter your QUERY statement and press Enter. Make the QUERY command the first command in a query. The QUERY statement tells Immediate Response the name of the database (CUSTOMER) containing the data you need. Immediate Response checks each statement.
`IMMED:`	■ If Immediate Response finds a problem, it immediately issues a diagnostic message, issues the IMMED: prompt, and waits for you to correct the problem. ■ If Immediate Response does not find a problem, it responds with the IMMED: prompt.

Statement	Explanation
display custno custname custph, <	The DISPLAY statement tells Immediate Response the names of the fields (CUSTNO, CUSTNAME, and CUSTPH) you want to report.
CONT:	The comma at the end of the first line tells Immediate Response that you want to continue the DISPLAY statement on the next line. Immediate Response responds with CONT: (for continue).
title 'customer list' <	Next, you enter the TITLE parameter. The TITLE parameter specifies a title for the report. In the QUERY statement, you told Immediate Response the database name (CUSTOMER), so Immediate Response checks and verifies that the field names that you specify (CUSTNO, CUSTNAME, and CUSTPH) actually exist in the database. If the names you specify cannot be found, Immediate Response displays a diagnostic message. This could mean the field does not exist, or it could mean you spelled the field name incorrectly. In this example, no problems were found so Immediate Response immediately processed your request for information.

Statement	Explanation

```
PA1


              CUSTOMER LIST

CUSTNO            CUSTNAME              CUSTPH
00001  TULSA TIMES                     8185551620
00013  ESSEX ENTERPRISES               2335556043
00028  AMERICAN RING AND WIRE SERVICE  0305550101
00043  BANK OF THE PEOPLE              8325554695
```

When an indicator appears at the bottom of your screen, your report is ready. Page to the next screen, as appropriate, for your environment.

After paging to the next screen, the finished report appears on your terminal screen.

- In IMS, the literal PA1, at the bottom of the terminal screen tells you that your display is ready for viewing. Pressing PA1 on your keyboard brings the first page of the display to your terminal screen.
- In CICS, P/N (or the paging command at your site) appears at the bottom of your screen. Press Enter to retrieve the first page of the display.

`IMMED:` Immediate Response asks for additional input.

`end immed <` There are two ways to exit Immediate Response immediate mode.

- To erase the query, enter the END IMMED command which tells Immediate Response to switch from immediate mode to system mode and erase the query you were working on.
- To save the query so that you can work on it later in Immediate Response immediate mode, enter the SYSTEM command instead of END IMMED.

?: The ?: prompt indicates that you are back in the system mode. This is where you can request that the system display the status of queries or list the contents of the foreground library.

`quit <` The QUIT command ends the session and returns to the Main Menu.

Immediate Response Modes

When you use Immediate Response, you will become familiar with the different modes of operation. Modes represent the system point of view. Each mode provides you with certain capabilities.

When the system is operating from the system point of view (system mode), it knows that your commands will request information stored in the library or the communication file.

Conversely, when the system is in immediate mode, it knows that your commands are exclusively directed towards the immediate, online processing of requests for data.

Immediate Response has three modes of operation: system mode, immediate mode, and checkpoint mode.

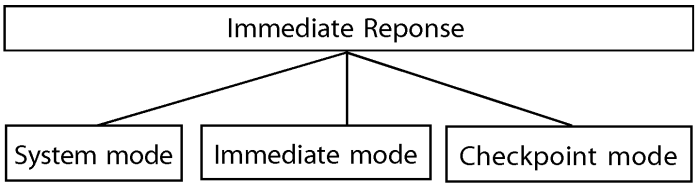


Figure 5-8 Modes of Operation

Different capabilities are available to you in each mode and each mode is distinguished by the type of prompt that the system displays to you on the terminal screen.

The modes are fully connected. That is, by using the proper command you can go directly from any mode to any other mode.

System Mode (?:)

The ?: prompt identifies Immediate Response system mode.

You use the system mode to control stored queries and to provide information about the foreground library. The following is a list and brief description of Immediate Response system mode commands:

Command	Function
GLOSSARY	Displays information about the various databases available to you.
LIST	Lists a query or collection of statements stored in the foreground library.
LISTLIB	Displays the foreground library directory by category (such as queries, statements, and databases).
QUIT	Terminates the session with Immediate Response. Returns immediately to the Main Menu.
RUN	Submits a query stored in the foreground library along with the statements in the immediate work area for execution.

Immediate Mode (IMMED:)

The IMMED: prompt identifies Immediate Response immediate mode.

You use the immediate mode when you want to see the results of your query statements immediately. As soon as you press Enter following a DISPLAY statement, Immediate Response processes the query and returns the requested data to your terminal screen. It is then available for you to view just once. If you clear your screen and move on to another query, the data that was displayed goes away and is no longer available for viewing.

The statements you enter in immediate mode are stored in a work area called the immediate work area. Immediate Response maintains this work area for you. For example, your QUERY statements, data selection statements, and temporary field statements remain in the immediate work area until you enter END QUERY. Then the entire immediate work area is cleared and you can start a new query.

DISPLAY statements, however, are cleared from the work area right after they are processed. That is, if you enter a number of query statements that end with a DISPLAY statement, you get the data requested on the DISPLAY statement, the DISPLAY statement is cleared from the immediate work area, and the rest of the statements remain available and in effect for the next DISPLAY statement you enter.

If you like the look of the report produced in immediate mode you can switch to Option 6 (Standard Query Processing) on the Main Menu, re-enter the statements and save them in the foreground library for future use.

The following are Immediate Response immediate mode commands with a brief description of what each one does.

Command	Function
END IMMED	Returns to the system mode and clears the immediate work area.
GLOSSARY	Displays information about the various databases available to you.
LIST	Lists the contents of the immediate work area.
LISTLIB	Displays the foreground library directory by category (such as queries, statements, and databases).
QUIT	Terminates the session and returns to the Main Menu.
SYSTEM	Switches to the system mode without affecting the immediate work area.

Checkpoint Mode (CKP:)

The CKP: prompt identifies the Immediate Response checkpoint mode. You enter it only from Immediate Response. You cannot directly enter checkpoint mode.

Immediate Response enters checkpoint mode from immediate mode when it is searching for data and reaches the maximum your system administrator has set for pages of data or calls to the database. When Immediate Response puts you in checkpoint mode, it displays a message telling you whether it reached the maximum number of pages or calls. The maximum value is stated.

You can continue displaying data or stop the display. When Immediate Response reaches the end of the database or you enter STOP, the program returns to immediate mode.

You can enter the following commands in checkpoint mode:

Command	Function
GO	Continues the display of data until Immediate Response reaches the end of the database or another maximum.
QUIT	Terminates the session and returns to the Main Menu.
STOP	Ends the display of data and returns to immediate mode.

Mode Switching

When you first start using Immediate Response you might have some confusion about what mode you are in. That is, you might enter a command and get an unexpected response. You will quickly learn the prompts for the modes and the commands to switch between modes. Figure 5-9 shows the prompts for the Immediate Response modes.

Mode	Prompt
System	?:
Immediate	IMMED:
Checkpoint	CKP:

Figure 5-9 The Prompts for Each Mode

Note: From any mode, entering the QUIT command terminates Immediate Response and returns to the Main Menu.

Figure 5-10 shows you how to switch between modes. In the left column is the mode you are in, the middle column is the command or action taken, and the right column is the mode you end up in.

Starting Mode	Action	Ending Mode
Immediate	SYSTEM command	System
	END IMMED command	System
	Any immediate mode command	Immediate
	Reach a display maximum	Checkpoint
Checkpoint	GO command	Checkpoint or the mode from which you started
	STOP command	Immediate
System	Any system mode command	System

Figure 5-10 Mode Switching Options

Displaying

This section describes how the DISPLAY command extracts data from a database and displays the information on the terminal in a specified display format.

- DISPLAY also gives you the capability of doing simple calculations on fields being reported or displayed.
- Reports generated by the DISPLAY command are referred to as "displays."
- You use the DISPLAY command only in Immediate Response queries. You use the REPORT command in all other types of queries.

DISPLAY in Immediate Response and in the Editor

The DISPLAY statement formats the reports that display on your terminal.

- You can use the DISPLAY statement in the immediate mode for displays that return to your terminal.
- You can also enter DISPLAY statements in the Editor panel (Option 6 on the Main Menu) and save the statements in the foreground library. For a discussion of the Editor panel, see Chapter 3, *General Concepts*.

With the DISPLAY statement you can display fields, give your reports titles, summarize fields (TOTAL, COUNT, CUM, AVG), and sort (ORDER) the contents of fields.

The following sections contain:

- The syntax (the way you write a statement so that Immediate Response can recognize it) of the DISPLAY command and how to use it.
- Examples of how to write and use the DISPLAY command.
- Information about other commands and keywords that help you build your Immediate Response displays in this section.

Commands Associated with DISPLAY

The DISPLAY command instructs Immediate Response to display data. In immediate mode, the DISPLAY command processes your query and displays the results at the terminal.

Note: For the formal syntax of the DISPLAY command, see *DISPLAY Command*.

The following lists the commands associated with the DISPLAY command.

Command	Meaning
DISPLAY command	Creates a display.
END QUERY command	Terminates the query.
QUERY command	Starts a query and specifies a database.

Figure 5-11 Immediate Response Commands Associated with Displaying

Keywords Associated with the DISPLAY Command

The following lists the keywords associated with the DISPLAY command. The list contains keywords in positional order.

Keywords	Meaning
ITEM keyword	Tells Immediate Response what is coming next.
TITLE keyword	Creates a title on the display.
AVG keyword	Computes and displays the average value.
COUNT keyword	Computes and displays the number of values.
CUM keyword	Computes and displays the sum of values.
TOTAL keyword	Computes and displays a controlled sum.
BY keyword	Keyword used with the summary commands.
GRAND keyword	Keyword used to get grand summaries.
ORDER keyword	Keyword used to alter the sequence of fields in a display.
DESC keyword	Changes ORDER sort sequence.
IF keyword	Keyword used to make the display conditional.

Figure 5-12 Keywords Associated with the DISPLAY Command

The section Immediate Response Modes provides an overview of Immediate Response modes.

The section A Sample Immediate Response Session explains a simple Immediate Response session and gives examples of how to use the immediate mode.

A Basic Display with Title

You generate a display of data by writing a query, which is a series of statements. These command statements instruct Immediate Response in selecting the data to display. Each statement tells Immediate Response what to do — what data to select, what calculations to make, and what data to display.

For example, the query in Figure 5-13, when entered from the immediate mode, produces the display in Figure 5-14.

Query

```
QUERY CUSTOMER
DISPLAY ITEM CUSTNO CUSTNAME,
        TITLE 'PRESENT CUSTOMERS'
```

Figure 5-13 Immediate Mode Query

The QUERY statement tells Immediate Response what database to use. The DISPLAY statement lists the data fields to be displayed and the TITLE keyword operand contains the text for the title of the display.

Output

```
                        PRESENT CUSTOMERS

CUSTNO                       CUSTNAME

00001           TULSA TIMES
00013           ESSEX ENTERPRISES
00028           AMERICAN WIRE AND RING SERVICE
00043           BANK OF THE PEOPLE
00048           STONEWELL STATE SYSTEMS
00089           GOING PLACES INSURANCE COMPANY
00115           SUNNY GRAPE GROWERS

NR06** END OF DISPLAY, DB CALL COUNT=      8, PAGE COUNT=    1
IMMED:
```

Figure 5-14 An Immediate Response Display

Default Formatting

Immediate Response inserts up to ten blank spaces between fields in the display. This makes the display easier to read.

- When there are more fields than will fit across the terminal screen, Immediate Response shrinks the amount of space between fields down as far as one space to fit as many fields as possible on a single line.
- If there are still too many fields than fit, Immediate Response “wraps” each line and continues it on the next.

Messages

The message line at the end of the display is an information line provided by Immediate Response to indicate that you are at the end of your display.

The message also gives you information on how many calls your display generated against the database and how the pages are displayed. For an explanation of this message, see the section Display Message Line.

Trying Different Displays

You can use Immediate Response queries to experiment. The DISPLAY command, which produced Figure 5-14, disappeared after processing, leaving only the QUERY command. You can type another DISPLAY command, and try another display format.

For example, the entry in Figure 5-15 produces the display in Figure 5-16.

Query

```
IMMED:
QUERY CUSTOMER
DISPLAY ITEM CUSTNAME CUSTPH,
        TITLE 'NEW CUSTOMER' 'PHONE NUMBERS'
```

Figure 5-15 Using the DISPLAY Command TITLE Keyword to Produce a Two Line Title

You can examine different fields, and try different formats until you find the appropriate one. Figure 5-16 shows multiple title lines on your display.

Output

```
                          NEW CUSTOMER
                          PHONE NUMBERS

CUSTNAME                                    CUSTPH

TULSA TIMES                               8185551620
ESSEX ENTERPRISES                         2335556043
AMERICAN WIRE AND RING SERVICE            0305550101
BANK OF THE PEOPLE                        8325554695
STONEWELL STATE SYSTEMS                   4345557001
GOING PLACES INSURANCE COMPANY            7385559123
SUNNY GRAPE GROWERS                       2345553966

NR06** END OF DISPLAY, DB CALL COUNT= 8, PAGE COUNT= 1
IMMED:
```

Figure 5-16 Immediate Response Display with Two Title Lines

The following sections describe query commands.

QUERY Command

QUERY DATABASE database name

The QUERY command initiates a query.

DATABASE	Optional keyword.
database name	Name of the database to be accessed. (See the section LISTLIB Command to determine database name.)

Make the QUERY command the first statement in any query. See the example in Figure 5-17.

END QUERY Command

END QUERY

The END QUERY command terminates:

- A query
- A SELECT group

Note, in immediate mode, the END QUERY command clears all statements in the immediate work area. For more information about the immediate work area, see the section Immediate Response Modes.

See an example of END QUERY in Figure 5-17.

DISPLAY Command

DISPLAY ITEM fieldname(s) TITLE 'text',

{ AVG / COUNT / CUM / TOTAL } fieldname(s) BY fieldname(s),

GRAND,
ORDER fieldname(s) DESC fieldname(s),
IF logical expression

The DISPLAY command, when entered in immediate mode, specifies that data is to be routed to the terminal.

Note: You can also print a display using the Batch Simulator (see your VISION:Inform system administrator).

The DISPLAY command can calculate and display summaries (AVG, COUNT, CUM, and TOTAL), and title the display.

ITEM Indicates that literal data and names of fields to be output follow it. ITEM is optional, if you place the field names or values immediately following the DISPLAY command.

fieldname(s) The field names or values to be output by the DISPLAY command.

- Fields appear on the display in the order, in which, you list them on the statement.
- The field name can be a database field, a temporary field, an arithmetic expression, a summary function, or a constant.
- Enclose literal constants in single quotation marks

TITLE Specifies that the display has one or two title lines. The title appears at the top of each screen.

'text' Character text of each title line. Enclose each character string in single quotation marks.

- To produce two title lines, enclose each title line in single quotation marks.
- Make the length of a title from 1 to the number of characters that will display on the screen.
- Two single quotation marks represent apostrophes.

ORDER fieldname(s) Indicates the fields to be sorted.

DESC fieldname(s) Indicates that the fields are to be sorted in descending order. Since ascending is the default, you can omit this parameter to sort in ascending order.

{ AVG | COUNT | CUM | TOTAL } Specifies the type of summary to be taken.

fieldname(s) Specifies the database fields or temporary fields to be summarized.

BY fieldname(s) Specifies the level of summary. A summary generates whenever the value of the specified (control break) field changes.

GRAND Generates a grand summary for all summarized fields.

IF logical expression Specifies a logical expression that determines what is displayed.

Figure 5-17 is an example of an entire query and its display.

Query

```
IMMED:
QUERY CUSTOMER
DISPLAY ITEM CUSTNAME CUSTNO,
TITLE 'PRESENT CUSTOMERS'
```

Output

```
                        PRESENT CUSTOMERS

                     CUSTNAME              CUSTNO

TULSA TIMES                                 00001
ESSEX ENTERPRISES                           00013
AMERICAN WIRE AND RING SERVICE              00028
BANK OF THE PEOPLE                          00043
STONEWELL STATE SYSTEMS                     00048
GOING PLACES INSURANCE COMPANY              00089
SUNNY GRAPE GROWERS                         00115

NR06** END OF DISPLAY, DB CALL COUNT=   8, PAGE COUNT=     1
IMMED:
```

Figure 5-17 Example of an Immediate Response Query and the Display it Produces

In addition to the option to title your display, the DISPLAY statement provides other options.

- You can sort the data named in the DISPLAY statement in either ascending or descending order.
- You can summarize the data by:
  - Averaging.
  - Counting the valid occurrences.
  - Generating the cumulative total of all valid occurrences of specified fields.
  - Generating grand totals of all valid occurrences of all the fields in the DISPLAY command ITEM parameter.
- You can set up selection criteria for the data to be displayed.

Using DISPLAY Command Parameters

The following sections show examples featuring selected parameters of the DISPLAY command. A description of each parameter explains how it affects your displays. Sample queries and displays illustrate each parameter.

Sorting Data Prior to Displaying

You can display your data in a sequence different from the sequence it is stored in the database. Use the DISPLAY command ORDER keyword to sort the data alphabetically or numerically in ascending or descending order.

DISPLAY Command ORDER Keyword

DISPLAY ITEM fieldname(s) ORDER fieldname(s) DESC fieldname(s)

ITEM fieldname(s)

Specifies the fields to be displayed. For details, see the section DISPLAY Command.

ORDER

- A required keyword that indicates data is to be sorted.
- Each field that appears in the ORDER list, does not need to appear in the ITEM keyword list.
- Each field name in the ORDER list can be a database field or a temporary field.
- Each database field must be in a segment already selected by the query.
- Each field is grouped automatically.
- A field that appears after a summary keyword such as TOTAL, AVG, CUM, and COUNT cannot be in the ORDER list. That is, do not put a summarized field name after the ORDER keyword.

fieldname(s)

Specifies the field or fields to be sorted.

- The first field you specify is the primary sort field, the second is the secondary sort field, and so forth.
- If you specify fields after the ORDER keyword, but not after the DESC keywords, they are sorted in the default ascending order.

For more information on multiple sort fields, see the section Ordering Fields for Summaries.

DESC

Is an optional keyword that you use with the ORDER keyword. Use it only if you want a field specified after the ORDER keyword to be sorted in descending order (higher to lower order values).

fieldname(s) Sorts the fields that appear after the ORDER keyword in descending order.

Fields that follow the ORDER keyword, but which are not specified after the DESC keyword, are sorted in ascending order.

Note:

- This example shows that a DISPLAY command can be carried over from one line to the next. If the statement you want to enter is too long for one line, you can continue a statement on the next line by putting a comma at the end of the line and pressing the Enter key. Immediate Response responds with CONT:, waiting for you to continue the statement.
- After the TITLE keyword there are two sets of title text, both enclosed by single quotation marks and separated by a blank, which display two lines of title text.

Figure 5-18 is an example of a query using ORDER and its resulting display. Note that duplicate information (CUSTNAME) does not display.

Query

```
IMMED:
QUERY CUSTOMER
DISPLAY ITEM CUSTNAME ORDERNO,
CONT:
ORDER CUSTNAME,
CONT:
TITLE 'CUSTOMER ORDERS' 'SORTED BY CUSTOMER NAME'
```

Output

```
                        CUSTOMER ORDERS
                    SORTED BY CUSTOMER NAME

            CUSTNAME                        ORDERNO

AMERICAN WIRE AND RING SERVICE               11211
BANK OF THE PEOPLE                           23468
ESSEX ENTERPRISES                            09541
                                             03619
                                             01142
GOING PLACES INSURANCE COMPANY               13842
                                             03921
                                             14821
STONEWELL STATE SYSTEMS                      06781
                                             06573
SUNNY GRAPE GROWERS                          01391
                                             00284
TULSA TIMES                                  08432
                                             07321

NR06** END OF DISPLAY, DB CALL COUNT=     29,PAGE COUNT=    1
IMMED:
```

Figure 5-18 A Display Using the ORDER Keyword

Sorting Data Using Two Fields

You can sort using more than one field. The first field in the ORDER keyword is the primary sort field, the next is the secondary sort field, and so on. The data is sorted by the first field, then within that field by the second field, and so forth. Figure 5-19 is an example of this technique of organizing your data for display.

Note:

- This example shows customer numbers (CUSTNO) as the primary sort field. The order number (ORDERNO) then sorts within each customer number.
- Field names listed after the ORDER keyword need not be display items.

Query

```
IMMED:
QUERY CUSTOMER
DISPLAY CUSTNO CUSTNAME ORDERNO,
  ORDER CUSTNO ORDERNO,
  TITLE 'CUSTOMERS SORTED ON' 'CUSTOMER NUMBER'
```

Output

```
                          CUSTOMERS SORTED ON
                            CUSTOMER NUMBER

CUSTNO                    CUSTNAME                      ORDERNO

00001          TULSA TIMES                               07321
               TULSA TIMES                               08432
00013          ESSEX ENTERPRISES                         01142
               ESSEX ENTERPRISES                         03619
               ESSEX ENTERPRISES                         09541
00028          AMERICAN WIRE AND RING SERVICE            11211
00043          BANK OF THE PEOPLE                        23468
00048          STONEWELL STATE SYSTEMS                   06573
               STONEWELL STATE SYSTEMS                   06781
00089          GOING PLACES INSURANCE COMPANY            03921
               GOING PLACES INSURANCE COMPANY            13842
               GOING PLACES INSURANCE COMPANY            14821
00115          SUNNY GRAPE GROWERS                       00284
               SUNNY GRAPE GROWERS                       01391

NR06** END OF DISPLAY, DB CALL COUNT=     29, PAGE COUNT=     1
IMMED:
```

Figure 5-19 A Display Using the ORDER Keyword to Sort Two Fields

Displaying Summary Information

Immediate Response has a set of summary commands that you use to generate the average, the count, the cumulative total of the values in a field, and grand totals. These same summaries are available as built-in summary functions (see the section Selection Criteria for Displays).

Summary Keywords

You tell Immediate Response to calculate a summary with one of the following summary keywords:

AVG	Displays the average value of all valid occurrences of the field referenced.
COUNT	Displays the count of all valid occurrences.
CUM	Displays the cumulative total of all valid occurrences.

TOTAL Displays the total of all valid occurrences.

You can use these summary keywords with or without the DISPLAY command (all DISPLAY keywords are available). That is to say, you can use the summary keyword (AVG, COUNT, CUM, and TOTAL) as stand-alone commands or as parameters to the DISPLAY command.

For example:

```
TOTAL INAMT BY CUSTNO,
      TITLE 'TOTAL INVOICE BY',
      'CUSTOMER NUMBER'
```

DISPLAY Command Summary Keywords

DISPLAY ITEM fieldname(s) TITLE 'text',

{ AVG / COUNT / CUM / TOTAL } fieldname(s) BY fieldname(s) GRAND

ITEM fieldname Specifies the fields to be displayed. For details, see the section DISPLAY Command.

{ AVG / COUNT / CUM / TOTAL } Specifies the type of summarization.

fieldname(s) The database fields or temporary fields to be summarized.

BY fieldname(s) Calculates the summary for each group of identical values in each field listed after the BY keyword.

If you omit the BY fieldname or if you specify GRAND, grand summaries are produced.

A single BY keyword is used per DISPLAY command, which specifies the summary control for all summary keywords within the display.

GRAND Generate grand summaries for the fields being summarized.

Ordering Fields for Summaries

A parameter BY fieldname displays in the syntax of the summary keywords. This tells Immediate Response to generate a summary for a field.

For example, in a display of customers with a summary of items shipped for each order number, you first need an arrangement of the order numbers so that all occurrences of the same order number stay together. This ordering insures that a true summary of the quantity shipped for that particular order number is represented in the total.

Query

```
IMMED:
QUERY CUSTOMER
DISPLAY CUSTNO ORDERNO INVNO QTYSHIP,
ORDER CUSTNO ORDERNO,
TOTAL QTYSHIP BY ORDERNO,
TITLE 'ORDERING FOR SUMMARIES'
```

Output

```
                  ORDERING FOR SUMMARIES

CUSTNO          ORDERNO         INVNO            QTYSHIP

00001           07321           I1648                 20
                                I1648                 20
                                I1648                 50
QTYSHIP  TOTAL BY ORDERNO  07321  IS              90
                08432           I1627                  6
QTYSHIP  TOTAL BY ORDERNO  08432  IS               6
00013           01142           I6781                  5
                                I6781                  5
                                I6781                  5
QTYSHIP  TOTAL BY ORDERNO  01142  IS              15
                03619           I1728                 25
                                I1728                 25
                                I1728                 20
QTYSHIP  TOTAL BY ORDERNO  03619  IS              70
                09541           I3334                  5
QTYSHIP  TOTAL BY ORDERNO  09541  IS               5

NR06** END OF DISPLAY, DB CALL COUNT=    93, PAGE COUNT=    1
IMMED:
```

Figure 5-20 Ordering Fields For Summarization Using Summary Keywords

Notice in the ORDER statement that CUSTNO is listed first, then ORDERNO. The database is stored in customer number order, but the order numbers are not necessarily stored in sorted order. To generate a correct total, sort the order numbers together within customer. To avoid changing the order of customer numbers, sort them first, then sort the order numbers.

ORDER is very important to use when you are summarizing data BY a field for which you do not know the stored order in the database.

Summary Example — TOTAL

Figure 5-21 produces a total for the quantity of back orders for each order number.

Query

```
IMMED:
DISPLAY CUSTNO ORDERNO ITEMORD ITEMNAME ITMPRICE QTYBKORD,
ORDER CUSTNO ORDERNO ITEMORD DESC CUSTNO,
TOTAL QTYBKORD BY ORDERNO,
TITLE 'BACK ORDERED ITEMS' 'FOR EACH ORDER'
```

Output

```
                        BACK ORDERED ITEMS
                          FOR EACH ORDER

CUSTNO      ORDERNO     ITEMORD      ITEMNAME      ITMPRICE   QTYBKORD

00115       00284       041          M4 PR FORM           3          0
                        042          M4 OUTPUT FORM       3          0
                        046          M4 BASIC REQ FORM    3          0
QTYBKORD TOTAL BY ORDERNO  00284   IS              0
            01391       017          M4 REF MANUAL       28          0
                        019          M4 OPS GUIDE-OS     22          1
QTYBKORD TOTAL BY ORDERNO  01391   IS              1
00089       03921       G46          T4 UTL SYS MANUAL    7          0
QTYBKORD TOTAL BY ORDERNO  03921   IS              0
            13842       H18          T4-202 CLASS       100          0
QTYBKORD TOTAL BY ORDERNO  13842   IS              0
            14821       G43          T4 CON/FAC MANUAL   15          0
QTYBKORD TOTAL BY ORDERNO  14821   IS              0

NR06** END OF DISPLAY, DB CALL COUNT=     72, PAGE COUNT=    1
IMMED:
```

Figure 5-21 A Summary Example (TOTAL)

Summary Example — COUNT

Figure 5-22 counts the items ordered by a customer from a sales person. A reorder produces a display sorted primarily on sales personnel and how many items each sales person sold.

Note that you cannot count a field that is at the same level as a field in the BY list.

For example:

```
DISPLAY ORPERSON ORDERNO CUSTNAME,
COUNT ORPERSON BY ORDERNO
```

produces the diagnostic message:

```
VQ21** SUMMARY ITEMS MUST BE SUBORDINATE TO ''BY'' ITEMS
```

because, in this case, ORPERSON and ORDERNO are on the same level.

Query

```
IMMED:
QUERY CUSTOMER
DISPLAY CUSTNAME ORPERSON ORDERNO ITEMORD ITMPRICE,
ORDER ORPERSON CUSTNAME ORDERNO,
COUNT ITEMORD BY ORPERSON,
TITLE 'SALES PERSONNEL' 'CUSTOMER LIST'
```

Output

```
                          SALES PERSONNEL
                           CUSTOMER LIST

            CUSTNAME             ORPERSON ORDERNO ITEMORD  ITMPRICE

GOING PLACES INSURANCE COMPANY    AJC     03921    G46             7
ITEMORD  COUNT BY ORPERSON AJC IS                 1
BANK OF THE PEOPLE                ANS     23468    017            28
GOING PLACES INSURANCE COMPANY            14821    G43            15
ITEMORD  COUNT BY ORPERSON ANS IS                 2
SUNNY GRAPE GROWERS               ART     01391    017            28
                                                   019            22
ITEMORD  COUNT BY ORPERSON ART IS                 2
STONEWELL STATE SYSTEMS           BCC     06781    046             3
                                                   247           100
                                                   673             6
                                                   674           100
ITEMORD  COUNT BY ORPERSON BCC IS                 4
NR06** END OF DISPLAY, DB CALL COUNT=     72, PAGE COUNT=    1
IMMED:
```

Figure 5-22 A Summary Example (COUNT)

Summary Examples — AVG and CUM

Figure 5-23 is an example of the CUM and the AVG commands. The item quantity ordered field is averaged by customer name, and the invoice amount being accumulated, and a cumulative amount displays when each customer changes. Notice the BY keyword governs when both summaries are calculated and displayed.

Query

```
IMMED:
QUERY CUSTOMER
DISPLAY CUSTNAME CUSTPH ITMQTYOR INAMT,
AVG ITMQTYOR,
CUM INAMT BY CUSTNAME,
TITLE 'CUSTOMER CONTACTS' 'FOR NEW PRODUCTS'
```

Output

```
                                CUSTOMER CONTACTS
                                FOR NEW PRODUCTS

CUSTNAME                              CUSTPH           ITMQTYOR       INAMT

TULSA TIMES                           8185551620             20       270.00
                                                             20        -
                                                             50        -
                                                             20        -
                                                              6        36.00
INAMT    CUM   BY CUSTNAME TULSA TIMES      IS        306.00
ITMQTYOR AVG   BY CUSTNAME TULSA TIMES      IS            23
ESSEX ENTERPRISES                     2335556043              5       105.00
                                                              5        -
                                                              5        -
                                                             25       210.00
                                                             25        -
                                                             20        -
                                                              5       140.00
                                                              2        -
INAMT    CUM   BY CUSTNAME ESSEX ENTERPRISE IS       761.00
ITMQTYOR AVG   BY CUSTNAME ESSEX ENTERPRISE IS           11
AMERICAN WIRE AND RING SERVICE        0305550101              1        22.00
INAMT    CUM   BY CUSTNAME AMERICAN WIRE AN IS       783.00
ITMQTYOR AVG   BY CUSTNAME AMERICAN WIRE AN IS            1
BANK OF THE PEOPLE                    8325554695              3        84.00
INAMT    CUM   BY CUSTNAME BANK OF THE PEOP IS       867.00
ITMQTYOR AVG   BY CUSTNAME BANK OF THE PEOP IS            3

NR06** END OF DISPLAY, DB CALL COUNT=     72, PAGE COUNT=    1
IMMED:
```

Figure 5-23 A Summary Example (AVG and CUM)

Note: A grand summary displays at the end of the display, because you entered the GRAND keyword at the end of the DISPLAY command.

Figure 5-24 produces totals for invoice amount and freight cost by order number and customer name.

Query

```
IMMED:
QUERY CUSTOMER
DISPLAY CUSTNAME ORDERNO SHIPPER FRTCOST INAMT,
ORDER CUSTNAME ORDERNO SHIPPER,
TOTAL FRTCOST INAMT BY ORDERNO CUSTNAME GRAND,
TITLE 'GRAND SUMMARIES' 'FOR SHIPPING COST'
```

Output

```
                                  GRAND SUMMARIES
                                  FOR SHIPPING COST

          CUSTNAME                ORDERNO  SHIPPER   FRTCOST        INAMT
AMERICAN WIRE AND RING SERVICE    11211      KLM          2.00        22.00
FRTCOST  TOTAL BY ORDERNO   11211              IS             2.00
INAMT    TOTAL BY ORDERNO   11211              IS            22.00
INAMT    TOTAL BY CUSTNAME AMERICAN WIRE AN   IS            22.00
FRTCOST  TOTAL BY CUSTNAME AMERICAN WIRE AN   IS             2.00
BANK OF THE PEOPLE              23468     GAE            4.00        84.00
FRTCOST  TOTAL BY ORDERNO   23468              IS             4.00
INAMT    TOTAL BY ORDERNO   23468              IS            84.00
INAMT    TOTAL BY CUSTNAME BANK OF THE PEOP   IS            84.00
FRTCOST  TOTAL BY CUSTNAME BANK OF THE PEOP   IS             4.00
ESSEX ENTERPRISES               01142     BIF           10.00        105.00
FRTCOST  TOTAL BY ORDERNO   01142              IS            10.00
INAMT    TOTAL BY ORDERNO   01142              IS           105.00
                                03619     CEC            3.00        210.00
FRTCOST  TOTAL BY ORDERNO   03619              IS             3.00
INAMT    TOTAL BY ORDERNO   03619              IS           210.00
                                09541     CEB            2.00        140.00
FRTCOST  TOTAL BY ORDERNO   09541              IS             2.00
INAMT    TOTAL BY ORDERNO   09541              IS           140.00
INAMT    TOTAL BY CUSTNAME ESSEX ENTERPRISE   IS           455.00
FRTCOST  TOTAL BY CUSTNAME ESSEX ENTERPRISE   IS            15.00
GOING PLACES INSURANCE COMPANY  03921    -             -            -
                                13842    -             -            -
                                14821   EES           25.00         75.00
FRTCOST  TOTAL BY ORDERNO   14821              IS            25.00
INAMT    TOTAL BY ORDERNO   14821              IS            75.00
INAMT    TOTAL BY CUSTNAME GOING PLACES INS   IS            75.00
FRTCOST  TOTAL BY CUSTNAME GOING PLACES INS   IS            25.00
FRTCOST  GRAND TOTAL IS           46.00
INAMT    GRAND TOTAL IS          636.00

NR06** END OF DISPLAY, DB CALL COUNT=     72, PAGE COUNT=    1
IMMED:
```

Figure 5-24 A Summary Example With A GRAND Summary

Built-in Summary Functions

You can perform summaries on fields within structured database records using built-in summary functions.

- The built-in summary function is not applicable when processing records that are not structured (see Chapter 7, *Advanced Topics*).
- After performing a built-in summary, you can use the result for comparison, further calculation, or as output on a display.

The built-in summary functions are AVG, COUNT, MIN, MAX, and TOTAL. Although these functions are similar to the summary keywords available as part of the DISPLAY command, the syntax, output format, and level specification are all different. The major difference is that built-in summaries are computed for each report record output, whereas the DISPLAY summaries apply to the entire display.

DISPLAY Command Built-In Summary Function Syntax

DISPLAY ITEM fieldname(s),

{ AVG | COUNT | MAX | MIN | TOTAL } (fieldname$_a$ BY fieldname$_b$)

ITEM fieldname(s) Specifies the fields to be displayed. For details, see the section Introducing Immediate Response Commands.

{ AVG | COUNT | MAX | MIN | TOTAL } Specifies what type of summary to be performed.

- **AVG** Computes the average of all valid occurrences of fieldname$_a$ subordinate to fieldname$_b$.
- **COUNT** Counts the number of all valid occurrences of fieldname$_a$ subordinate to fieldname$_b$.
- **MIN** Determines the minimum of all valid occurrences of fieldname$_a$ subordinate to fieldname$_b$.
- **MAX** Determines the maximum of all valid occurrences of fieldname$_a$ subordinate to fieldname$_b$.
- **TOTAL** Computes the sum of all valid occurrences of fieldname$_a$ subordinate to fieldname$_b$.

fieldname(s)$_a$ Is a field in a lower level segment that you want to summarize.

BY fieldname(s)$_b$ Specifies the controlling field for the summaries (all occurrences of fieldname$_a$ are summarized within each occurrence of fieldname$_b$).

If you omit the BY fieldname, the default produces summaries when all occurrences of the immediate parent segment are exhausted.

Note: It is the presence of the parentheses that differentiates the built-in summary functions from the summary keywords of the DISPLAY command. The left parenthesis follows a keyword (AVG, COUNT, MAX, MIN, or TOTAL) with or without intervening spaces.

Syntax

Built-in summary syntax requires that the field on which the summary is to be performed be enclosed in parentheses following the function requested; for example, AVG(INAMT). Or, to change the level of summary: AVG(INAMT BY CUSTNO).

Display Format

When specified for display on a DISPLAY command, built-in summaries print in columns like data fields and given column headings that reflect the summary and field being summarized.

Level Specification

With built-in summaries, the level of summary is the direct parent unless specified otherwise with the BY keyword.

If you specify any selection statements, data outside the selection context are not available, either for built-in summaries, or for displaying.

What Built-In Summaries Do

The built-in summary functions are AVG, COUNT, MAX, MIN, and TOTAL.

AVG(fieldname)	Computes the average of all valid occurrences of the specified item for the context specified in the query.
COUNT(fieldname)	Counts the number of valid occurrences of the specified item within the context specified in the query.
MAX(fieldname)	Determines the maximum valid value of the specified item within the context specified in the query.
MIN(fieldname)	Determines the minimum valid value of the specified item within the context specified in the query.
TOTAL(fieldname)	Computes the sum of all valid occurrences of the specified item for the context specified in the query.

Use Figure 5-25 for the explanations of the built-in functions the remainder of this section. This information is the contents of two records from the CUSTOMER database.

LEVEL 1 (root) LEVEL 2 LEVEL 3

	CUSTNO	CUSTNAME	ORDERNO	ORDRDATE	ORDRCMPLT		ITM-PRICE	ITM-QTYOR
RECORD 1	00013	ESSEX ENTERPRISES	01142	010606	Y	ITEMORD		
						803	8	5
						857	8	5
						858	5	5
			03619	010817	Y			
						045	3	25
						113	3	25
						114	3	20
			09541	011002	N			
						017	28	5
	00048	STONEWELL STATE				019	22	2
			06573	010101	Y			
						G40	23	10
						G43	15	5
RECORD 2			06781	010710	Y			
						046	3	20
						247	100	1
						673	6	3
						674	100	1

Figure 5-25 Two Records in the CUSTOMER Database

Displaying Built-In Summaries

Figure 5-26 shows the query and output for a count of order numbers for each customer.

Query

```
IMMED:
QUERY DATABASE CUSTOMER
DISPLAY CUSTNO CUSTNAME COUNT(ORDERNO),
TITLE 'BUILT-IN FUNCTIONS'
```

Output

```
                          BUILT-IN FUNCTIONS

CUSTNO                     CUSTNAME              CNT(ORDERNO)

00001           TULSA TIMES                                 2
00013           ESSEX ENTERPRISES                           3
00028           AMERICAN WIRE AND RING SERVICE              1
00043           BANK OF THE PEOPLE                          1
00048           STONEWELL STATE SYSTEMS                     2
00089           GOING PLACES INSURANCE COMPANY              3
00115           SUNNY GRAPE GROWERS                         2

NR06** END OF DISPLAY, DB CALL COUNT=     29, PAGE COUNT=    1
IMMED:
```

Figure 5-26 Reporting Built-In Summaries

Looping Considerations When Using Built-In Summaries

You may only specify fields from lower level segments for built-in summaries. Because of this, a looping process occurs. Lower level segments and looping are discussed in detail in Chapter 7, *Advanced Topics*.

If you specify a built-in summary function, the summarization computes for each occurrence of the summarized field's parent segment. For example, if you specify the field for a built-in summary function in level 3, the summarization occurs for each of the level 2 segment occurrences in its path. Figure 5-27 illustrates this process.

The built-in summary specification is the first reference to the segment in which the field to be summarized resides. For example, if the built-in summary field is in level 3, all fields referenced in the query or SELECT group within the query prior to the built-in summary function is in the root or level 2.

The following examples help clarify the effect that looping has on built-in summaries.

In Figure 5-27, the query calculates a total item price for each order, of each customer. This total is taken because item price is in level 3 of the database, and the ORDER segment is its parent at level 2. The CUSTNO field is in the root segment of the database.

Query

```
IMMED:
QUERY CUSTOMER
DISPLAY CUSTNO TOTAL(ITMPRICE),
TITLE 'BUILT-IN FUNCTIONS'
```

Output

```
      BUILT-IN FUNCTIONS

CUSTNO          TOT(ITMPRICE)

00001                    12
                          6
00013                    21
                          9
                         50
00028                    22
00043                    28
00048                    38
                        209
00089                     7
                        100
                         15
00115                     9
                         50

NR06** END OF DISPLAY, DB CALL COUNT=    72, PAGE COUNT=    1
IMMED:
```

Figure 5-27 Totals Calculated with Built-In Summaries

In Figure 5-28, the specification of BY CUSTNO computes the summarization at the root level. Contrast this query with Figure 5-27, which uses the default summary level.

Query

```
IMMED:
QUERY CUSTOMER
DISPLAY CUSTNO TOTAL(ITMPRICE BY CUSTNO),
TITLE 'CHANGING SUMMARY LEVELS'
```

Output

```
                    CHANGING SUMMARY LEVELS

CUSTNO          TOT(ITMPRICE)

00001                    18
00013                    80
00028                    22
00043                    28
00048                   247
00089                   122
00115                    59
NR06** END OF DISPLAY, DB CALL COUNT=    44, PAGE COUNT=    1
```

Figure 5-28 Changing Summary Levels in Built-In Summaries

Built-In Summaries for Data Selection

You use built-in summaries in IF expressions for data selection. The summary is calculated first and then the comparison is made. For example, if:

```
IF AVG(QUANTITY) GT 10
```

is the expression, then the QUANTITY field is averaged and the result is compared to 10.

Figure 5-29 is an example of using a built-in summary for data selection. Notice that only those orders with an average quantity greater than 5 display.

Query

```
IMMED:
QUERY DATABASE CUSTOMER
DISPLAY ORDERNO ORDRDATE AVG(ITMQTYOR),
IF AVG(ITMQTYOR) GT 5,
TITLE 'BUILT-IN FUNCTIONS'
```

Output

```
                BUILT-IN FUNCTIONS

ORDERNO           ORDRDATE           AVG(ITMQTYOR)

07321             010228                    27
08432             010801                     6
03619             010817                    23
06573             010101                     7
06781             010710                     6
00284             010712                    20

NR06** END OF DISPLAY, DB CALL COUNT=    58, PAGE COUNT=    1
IMMED:
```

Figure 5-29 Using a Built-In Summary in an IF Expression

Selection Criteria for Displays

The DISPLAY command IF keyword specifies the selection criteria for displaying data. Use the IF keyword to indicate the conditions for which the display occurs.

DISPLAY Command IF Keyword

DISPLAY ITEM fieldname(s),
IF field conditional-operator field

ITEM fieldname(s) Specifies the fields to be displayed. For details, see the section DISPLAY Command.

IF Specify the required keyword.

field Specify one of the following:

- a database field
- an arithmetic expression
- a constant value enclosed in single quotation marks
- a temporary field.

conditional-operator Specify one of the six relational operators (See Chapter 6, *Data Selection and Field Processing*).

field Specify one of the following:

- a database field
- an arithmetic expression
- a constant value enclosed in single quotation marks
- a temporary field.

DISPLAY Command with Single Condition

Figure 5-30 illustrates the use of a single condition on the DISPLAY command.

Query

```
IMMED:
QUERY DATABASE CUSTOMER
DISPLAY ITEM CUSTNAME CUSTNO,
IF CUSTNO GT 100,
TITLE 'CONDITIONAL PHRASE' 'ON DISPLAY'
```

Output

```
                    CONDITIONAL PHRASE
                       ON DISPLAY

           CUSTNAME                       CUSTNO

SUNNY GRAPE GROWERS                        00115

NR06** END OF DISPLAY, DB CALL COUNT=    8, PAGE COUNT=    1
IMMED:
```

Figure 5-30 A Conditional Display

Figure 5-30 only displays customers with customer numbers greater than 100.

DISPLAY Command with a Logical Operator

Conditional operators are the standard logical operators (equal, greater than, less than, greater than or equal, less than or equal, not equal). You can use more complex logical operations with the AND and OR connectors.

Figure 5-31 shows how to connect two conditional phrases with the AND connector. The same rules hold true for evaluating ANDs and ORs. For more information on the evaluation of logical connectors, see Chapter 6, *Data Selection and Field Processing*.

Query

```
IMMED:
QUERY CUSTOMER
DISPLAY CUSTNAME CUSTNO ORDERNO,
IF (CUSTNO GT 50) AND (ORDERNO LT 1000),
TITLE 'CONDITIONAL PHRASES' 'CONNECTED BY LOGICAL OPERATOR'
```

Output

```
                    CONDITIONAL PHRASES
                 CONNECTED BY LOGICAL OPERATOR

CUSTNAME                      CUSTNO          ORDERNO

SUNNY GRAPE GROWERS            00115            00284

NR06** END OF DISPLAY, DB CALL COUNT=    29, PAGE COUNT=    1
IMMED:
```

Figure 5-31 Conditions Connected With a Logical Operator

Figure 5-31 displays the customer name, customer number, and order number for customers with customer numbers greater than 50 and order numbers less than 1000.

Commenting Your Queries

You can place comments on any query line. You can add them to the end of an existing query line or they can be the entire line.

Immediate Response recognizes a comment because it starts with a semi-colon (;). You can save comments with queries created on the Editor panel, but you cannot use comments in the immediate mode.

Comment Syntax

;comments Specifies that the semicolon (;) and what follows are not processed as query statements. You can place the semicolon on the statement after a command, or you can make it the first non-blank character on the statement.

Do not use comments in immediate mode.

Examples:

```
; THIS QUERY HAS A COMMENT FIRST
QUERY CUSTOMER;NO SPACE BEFORE COMMENT
DISPLAY CUSTNO CUSTNAME , ; CONTINUATION COMMENT
  ORDER CUSTNO
END QUERY ; THIS QUERY IS OVER
```

Finding Your Data

Before you can write a query to display data on your terminal, you must know the name of the database in which the data is stored. You must also know the field names to tell Immediate Response what fields to display.

Use the LISTLIB and the GLOSSARY commands to find the names of available databases and fields within the databases.

LISTLIB Command

Note:

- You can obtain glossary information from the DATAVIEW panel, FIELDS panel, and FIELDS DETAIL panel in the Full Screen Editor.
- For additional information, see the section *Displaying Database Information in the Editor Panel*.

LISTLIB { **DATABASE** / **QUERY** }

The LISTLIB command displays all the names in the foreground library by category, in alphabetical order. You need the database name when you build a query and when you use the GLOSSARY command.

{ **DATABASE** / **QUERY** } Specifies the display of all file definitions or queries and statements stored in the foreground library.

Query

```
IMMED:
LISTLIB DATABASE
```

Output

```
DATAVIEW   TYPE       CREATED

BADFIN     NON-DLI    01166
CUSTITEM   LDV        01166
CUSTOMER   DLI        01166
FINANCE    NON-DLI    01166
ITEM       DLI        01166
```

Figure 5-32 Example of LISTLIB DATABASE

You can also use LISTLIB to list the queries you stored in your foreground library. For example:

```
LISTLIB QUERY
```

This statement displays a list of all query names stored in the foreground library. If you enter this command without a keyword, all items available to your user ID in the foreground library are listed.

Displaying Field Names

Once you know the Immediate Response database name, which you obtained by performing a LISTLIB command, you can do a GLOSSARY command to show the field names.

- Use GLOSSARY to display information such as field length and whether the database contains alphabetic data or numeric data.
- The GLOSSARY command displays alternate names and field descriptions, if the system administrator defined them for the selected database.

GLOSSARY Command

GLOSSARY DATABASE name { **SEGMENT segname** / **FIELD fieldname** }

The GLOSSARY command displays glossary information. If you do not specify either the SEGMENT or the FIELD keyword, the entire database is listed.

DATABASE name — Specifies the database name. The keyword is optional if you enter the name of the database immediately after the GLOSSARY command.

SEGMENT segname — Displays the segment information. You can leave out the SEGMENT keyword if the segment name immediately follows the database name. SEGMENT is the default.

FIELD fieldname — Displays the field information You must specify FIELD to obtain glossary information on a field.

Note: In the GLOSSARY command, SEGMENT and FIELD are mutually exclusive.

Examples:

To list all the segments and fields in the database, enter:

```
GLOSSARY DATABASE name
```

To find all fields associated with a particular segment of the database, enter:

```
GLOSSARY name SEGMENT segname
```

This lists all fields in the segment segname.

Listing Database Information

Figure 5-33 shows the GLOSSARY command and the output for the ORDER segment of the CUSTOMER database.

Command

```
IMMED:
glossary customer segment order
```

Output

```
CUSTOMER - IMS DATABASE FILE  DATE CREATED - 01.167   TIME - 15.49.125

NAME       TYPE    START  LENGTH  DEC OUT-LEN  KEY  SEGNAME    PARENT

ORDCMPLT   CHAR       29       1           10         ORDER    CUSTOMER
ORDDUDAT   CHAR       22       6            8         ORDER    CUSTOMER
ORDERNO    CHAR        1       5            7     1   ORDER    CUSTOMER
ORDINVGN   CHAR       28       1           10         ORDER    CUSTOMER
ORDPONUM   CHAR       17       5            8         ORDER    CUSTOMER
ORDRDATE   CHAR        8       6            8         ORDER    CUSTOMER
ORPERSON   CHAR       14       3            8         ORDER    CUSTOMER
SEG20FIL   CHAR       30      10           10         ORDER    CUSTOMER
***END OF DEFINITION***
```

Figure 5-33 A SEGMENT Glossary Example

Listing Field Information

You can also look at the information for one particular field in the database. The GLOSSARY command syntax is:

GLOSSARY name FIELD fieldname

Figure 5-34 shows the GLOSSARY command with the FIELD parameter, which displays the characteristics of the field CUSTNAME.

```
?:
glossary customer  field custname
     NAME   TYPE    START  LENGTH  DEC OUT-LEN  KEY  SEGNAME    PARENT
 CUSTNAME   CHAR        6      30           30    S   CUSTOMER  CUSTOMER
```

Figure 5-34 A FIELD Glossary Example

The GLOSSARY command determines the characteristics of the database, what segment each field is in, what level each segment is on, the type and length of each field, even the date and time of creation of the database (see Figure 5-33).

Using the LISTLIB and GLOSSARY commands together, you can find any field in any available database and determine its characteristics.

Running a Query

In system mode, you can append the statements in a saved query to the query statements in the immediate work area and execute them together using the RUN command.

RUN Command

RUN ITEM name TYPE { **QUERY** / **STMTS** }

The RUN command specifies that a query (or collection of statements) and the statements currently in the immediate work area are to be combined and submitted for immediate work area.

ITEM name Specifies the query (or collection of statements) to append to the immediate work area. ITEM is an optional keyword.

TYPE { **QUERY** / **STMTS** } Specifies whether the item is a query or a collection of statements. If you omit this keyword, TYPE QUERY is assumed.

Note: Diagnostic messages appear if the statements in the immediate work area and the statements from the saved query are not compatible.

The statements stored in the foreground library are not affected, but the statements in the immediate work area are affected according to the following rules.

- If there is a QUERY command in the stored statements, the preceding immediate mode statements are deleted.

  The new QUERY statement causes Immediate Response to act as though an END QUERY command has been coded at the end of the immediate mode statements. The QUERY statement starts a new query.

- If a non-display command (such as LET or SELECT) is in the stored statements, it is appended to the immediate mode statements after execution (it will be in the immediate work area).

Using the RUN Command

To effectively use the RUN command, you need to know and understand the implications of being in system mode and immediate mode and how to move from one mode to another. For information about the two modes and how to change modes, see Chapter 6, *Data Selection and Field Processing*.

This RUN command is a system mode command.

```
RUN ITEM TEST TYPE STMTS
```

Immediate Response treats the statements stored in the foreground library as if they are in the immediate work area. If your entire query is in the foreground library, be careful that there are no statements in the immediate work area.

Immediate Response takes the statements in the foreground library and appends them to any statements in the immediate work area. (You can use the LIST command to find out what is in the immediate work area.)

When you store the Immediate Response query in the foreground library and enter the RUN command, you keep Immediate Response from deleting your DISPLAY command. You entire query stays in the foreground library after Immediate Response processes your query.

Follow the same guidelines that you normally follow to look at your display.

- After you page through your display, you are still be in system mode.
- If the display is correct, you can save it in the foreground library to use again.
- If you have a problem, or the display is not what you want, change the statements stored in the foreground library or in the immediate work area.

Paging Through Displays

After Immediate Response processes your query, and your display is ready to view, an indicator appears at the bottom of your screen. Displays can be longer than the screen of the terminal.

If you must page forward through a multi-screen display, the technique varies depending on the environment in which Immediate Response operates. Appendix A, *Paging and Viewing* describes the indicators and paging techniques for the various environments.

Display Message Line

The final line of your display is an Immediate Response message line indicating the end of your display. The message also provides information about the number of database calls it took to create your display and how many pages displayed.

The VISION:Inform system administrator can set up your user ID profile with two entries that limit the number of calls that a display produces and the number of pages that can be brought back to the terminal.

- If your query reaches one of the maximums, Immediate Response puts you in checkpoint mode, tells you the maximum value reached, and waits for you to stop the display or continue it.
  - To stop the display and return to immediate mode, enter STOP.

  - To continue, enter GO. Immediate Response continues to search for and display data until it reaches the end of the database or another maximum.
  - To stop the display and exit the Immediate Response panel, enter QUIT.

- If you reach the maximum number of database calls or pages, you did not construct an efficient query.
  - Check your selection criteria to see if you can improve your query.
  - Check the volume of data on the database. If you are processing against a complex database, more calls than usual are necessary to process your query.

Since Immediate Response processes online, large displays can have a negative impact on resources. It is advisable to be resource efficient, and to generate large reports with the Batch Simulator, or standard VISION:Bridge query processing.

Chapter

6

Data Selection and Field Processing

This chapter discusses selecting only the fields you need and processing that data. Topics include:

- Description and use of various data selection commands (SELECT and IF).
- Selection techniques using expressions, conditional phrases, temporary fields, partial fielding, and automatic table lookup.
- Description and use of data selection keywords (ALL, ANY, HAS, HAVE, and TO).

This chapter presents data selection in the framework of specific selection needs and techniques, not as lists of commands. There are many examples to demonstrate the various data selection commands and techniques. All the data selection commands are described fully and summarized; however, the same or similar command formats appear in more than one example.

What is Data Selection?

Data selection is the process of telling VISION:Bridge exactly what data you need. Often, when you use VISION:Bridge, you are looking for specific data. The data you seek can be very specific, such as the birth date of a single employee, or it could be as general as listing the names of all the customers with which your company does business.

Very often you want a simple report that gives you data from a small part of a database. At other times, you want to produce many reports based upon repeated reference to the same data in a database. Using data selection, you can refine your queries so that you get only the data you need.

The essence of data selection is the conditional phrase. Conditional phrases provide VISION:Bridge with the criteria that point to the exact data you need. This occurs by limiting the range of alternatives available.

Here is an example of what is meant by limiting alternatives with a conditional phrase. Suppose you meet a friend on the street one day, and since you have not seen each other for some time, you suggest that you both get together later. If you say, “I'll meet you on 2nd Street at eight o'clock”, two things are very clear to your friend, where to meet and at what time.

"On 2nd Street" and "at eight o'clock" form a conditional phrase. They supply the criteria that limit the range of alternatives available. In this example, the criteria are very specific and they leave little room for misunderstanding: eight o'clock on 2nd Street. Your data selection can be as specific or as general as you like. VISION:Bridge provides you with a great deal of flexibility.

Sometimes, it is easier to obtain the data you need by default. That is, you can tell VISION:Bridge what data you do not want; VISION:Bridge then provides you with everything else. In this way, you can select an entire database, a single data record, or even a single field value from a single record.

Conditional Phrases

A conditional phrase consists of the word IF followed by one or more expressions.

Note: See the section *Expressions*.

IF expression$_1$ expression$_2$ ... expression$_n$

You never use conditional phrases alone except for the IF command. You attach them to various VISION:Bridge commands, like this:

```
REPORT INAMT IF CUSTNO EQ 00001
SELECT IF FIELDA LT 5
LET TEMPFLD = CUSTNO IF MONTH EQ 'MARCH'
SELECT CUSTNO IF ORDCMPLT = 'N'
```

The VISION:Bridge language is an active language in that it is built around commands. Each command tells VISION:Bridge to do something. Commands by themselves are direct orders that are carried out under all conditions. If you say:

```
REPORT INAMT
```

VISION:Bridge reports every value for INAMT that is in the database. As you can imagine, this is a rather crude approach to data selection. You get the data you need, but you are also likely to get data you do not need.

Conditional phrases add precision to the language. They let you qualify commands, so that you select only the data you want. Thus,

```
REPORT INAMT IF CUSTNO EQ 00048
```

produces all values of INAMT only for the customer whose number is 00048.

You might imagine that VISION:Bridge retrieves the information you need directly from the database; in this case, the values of INAMT for customer 00048. The actual process is:

1. VISION:Bridge looks at the value of CUSTNO for the first data record in the database and checks to see if its value is 00048 (such that the conditional phrase is true).
2. If it is, VISION:Bridge executes the REPORT command, lists all the values of INAMT for that record, and moves to the next data record, looking for another value of CUSTNO equal to 00048.
3. If the value of CUSTNO is not equal to 00048, VISION:Bridge does not execute the REPORT command, and moves to the next record, looking for another value of CUSTNO equal to 00048.

True — False Testing

Expressions are essentially true—false tests. You supply the conditions and VISION:Bridge tests the data against the conditions you supply.

- For example, you tell VISION:Bridge to do something, such as report or display some fields if the conditional phrase you supply is true.
- If the conditions you specify are not true, no action is taken and VISION:Bridge moves to the next record or next segment occurrence and applies the conditional test again.

The exception is the IF/Else construct. For details, see the section IF/ELSE Statements.

Expressions

Expressions, which make up conditional phrases, are defined as:

$operand_1$ operator $operand_2$

There are three kinds of expressions:

- Relational expressions (See the section Relational Expressions.)
- Logical expressions (See the section Logical Expressions.)
- Arithmetic expressions (See the section Arithmetic Expressions.)

It is a important to understand what each expression does and how they combine to form complex selection statements.

Relational Expressions

A relational expression is part of a conditional phrase. It has two operands connected by a relational operator. The operands in a relational expression can be data fields, constants, or arithmetic expressions.

operand$_1$ relational-operator operand$_2$

Relational operators set up conditions based upon the items they connect. For example, if you say that one item is greater than or less than another item, you are indicating how they relate to one another.

The VISION:Bridge relational operators are:

Relational operators	Meaning
EQ or =	Equal to
NE or ¬=	Not equal to
LT or <	Less than
GT or >	Greater than
LE or <=	Less than or equal to
GE or >=	Greater than or equal to
TO	Designates a range of values

When you write relational expressions, you can use either the letters (EQ) or the symbol (=). VISION:Bridge can interpret either one.

You use these relational operators to specify the relationships between any two field values, regardless of whether they are numeric or alphabetic.

Numeric Condition

If you want to report all the customers whose customer number is greater than 00058, you write a REPORT statement like this:

```
REPORT CUSTNO  IF  CUSTNO  GT  00058
                   ~~~~~~~~~~~~~~~~
                   Relational expression
```

This REPORT statement reports all customer numbers whose value is greater than 00058. Any customer whose number is less than 00058 does not appear on the report. Neither does any customer whose number is 00058. GT is the relational operator in the REPORT statement.

Alphanumeric Condition

The following REPORT statement contains a relational expression that tests for an alphanumeric value. It reports the customer name and number of each customer whose order is complete.

```
REPORT CUSTNO CUSTNAME IF ORDCMPLT = 'Y'
```

In the CUSTOMER database, there is a field called ORDCMPLT (for "order complete").

- If a customer's order is complete, the value of the field is Y.
- If it is not complete, the value is N.

Enclosing Literal Values in Quotation Marks

When testing for alphanumeric values, enclose the literal value you are testing in single quotation marks. The character string must not exceed 15 characters in length. For example:

```
REPORT CUSTNO IF CUSTNAME EQ 'TULSA TIMES'
```

This example reports the customer number for each customer whose name is TULSA TIMES.

Partial Fielding

Sometimes, the value you need exceeds 15 characters. In that case, use the partial field function. For information on how to use partial fielding, see the section Partial Field Processing.

Using a Range of Values

A relational expression can also specify a range of values by using the relational operator TO. The keyword TO is a special type of relational operator that designates a range of values.

```
REPORT CUSTNAME IF CUSTNO EQ 48 TO 1000
```

is interpreted to mean:

```
REPORT CUSTNAME IF CUSTNO GE 48,
        AND CUSTNO LE 1000
```

These two REPORT statements are equivalent. The word AND is a logical operator and is discussed next.

Notice that the leading zeros of CUSTNO are not necessary. VISION:Bridge right aligns numeric fields.

Logical Expressions

Logical expressions consist of one or more relational expressions.

relational-expression$_1$ logical-operator relational-expression$_2$

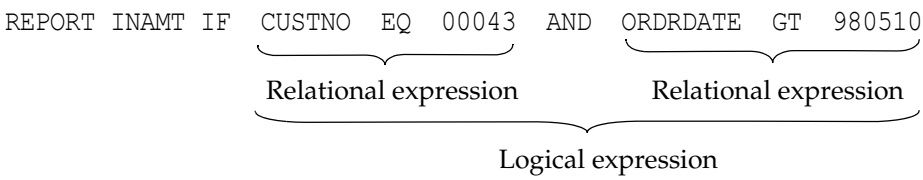
You connect multiple relational expressions by AND or OR. The words AND and OR are logical operators.

- Like the relational expressions that are used to build them, logical expressions can be true or false.
- Unlike relational expressions, however, a logical expression can be true even though some of the relational expressions contained in it are false.

This is due to the meanings of AND and OR.

- When using AND to connect relational expressions, each of the individual relational expressions must be true for the logical expression to be true.
- When using OR to connect relational expressions, at least one of the relational expressions must be true for the logical expression to be true.

AND Example



Here you see two relational expressions connected by AND. Because of the AND, values of INAMT are reported only when both relational expressions are true. That is, when the customer number is 00043 and the order date is greater than 980510.

This date is in the format YYMMDD, where YY is year (98), MM is month (05 or May), and DD is the day of the month (10). For example, January 1, 1998 is 980101.

In this example, the REPORT command executes only when the conditions, in this case, a single logical expression, are satisfied.

OR Example

By simply replacing the AND with an OR, the meaning of the expression completely changes, and as a result, the report you get is different.

```
REPORT INAMT IF  CUSTNO  EQ  00043   OR  ORDATE  GT  980510
```

Relational expression Relational expression

Logical expression

In this example, every value of INAMT is reported for customer number 00043, including every order date greater than 980510, regardless of the customer number.

When using the OR, the command executes each time either of the relational expressions is true. The results of both statements are shown in Figure 6-1.

Logical Expressions Using a List

You can also specify logical OR expressions in list forms, as shown in the following example:

```
IF CUSTNO EQ 00043 OR CUSTNO EQ 00076 ,
   OR CUSTNO EQ 00097 OR CUSTNO EQ 00175
```

is equivalent to the following list form:

```
IF CUSTNO EQ (00043, 00076, 00097, 00175)
```

Query

```
QUERY CUSTOMER
REPORT INAMT IF CUSTNO EQ 00043,
 AND ORDRDATE GT '980510'
  TITLE 'USING THE LOGICAL CONNECTOR ''AND'''
 END REPORT
  REPORT INAMT IF CUSTNO EQ 00043,
   OR ORDRDATE GT '980510'
  TITLE 'USING THE LOGICAL CONNECTOR ''OR'''
 END REPORT
END QUERY
```

Output

```
May 10, 2001       USING THE LOGICAL CONNECTOR 'AND'       PAGE 1
------------------------------------------------------------------
  INVOICE
  AMOUNT
------------------------------------------------------------------
    84.00
May 10, 2001       USING THE LOGICAL CONNECTOR 'OR'        PAGE 1
------------------------------------------------------------------
  INVOICE
  AMOUNT
------------------------------------------------------------------
    36.00
   105.00
   210.00
   140.00
    84.00
   260.00
    18.00
        -
    75.00
   180.00
    56.00
```

Figure 6-1 Using the AND and OR Logical Operators

Sometimes you must use both logical connectors in the same statement. In this case, AND evaluates before OR. For example:

```
CUSTNO EQ 00046 OR CUSTNO EQ 00050 AND INAMT GT 500
```

This expression first.

You can use parentheses change the order of the evaluation. The logical expression within the parentheses is evaluated first. For example:

```
(CUSTNO EQ 00046 OR CUSTNO EQ 00050) AND INAMT GT 500
```

This expression first.

Notice the difference in the meaning of these two statements. For additional discussion of parentheses, see the section Arithmetic Expressions.

NOT Operator

You can negate any logical expression by using the word NOT in front of it. This operator can be very useful. It is also very powerful since it changes the meanings of all the relational operators, as well as changing the meaning of the logical operators AND or OR.

For more information on the use of NOT, see the section NOT Operator.

Arithmetic Expressions

You can manipulate the data in some way before you report it, or you can select data based upon the results of a calculation performed on the data. Arithmetic processing is one way to manipulate data.

Note:

- Built-in summary functions (AVG, COUNT, MAX, MIN, and TOTAL) operate at the record level.
- Summary commands (AVG, COUNT, CUM, MAX, MIN, PCT, RATIO, and TOTAL) operate at the group or entire file level.

There are three ways to perform arithmetic processing:

- Using summary commands discussed in the section A Sample VISION:Bridge Session. These commands generate automatic totals and subtotals that are displayed on reports. Summary totals cannot, however, be used in data selection.
- Using VISION:Bridge built-in summary functions. You use values generated by built-in summaries in data selection statements. They are discussed in the section A Sample VISION:Bridge Query.
- Performing arithmetic calculations in conditional phrases in DISPLAY, EXTRACT, IF, LET, REPORT, and SELECT statements. This section describes this type of arithmetic processing.

Note:

- In a VISION:Bridge query, the IF statement can stand alone.
- In Immediate Response, it is always used as a keyword on a DISPLAY statement.

An arithmetic expression contains arithmetic operators and the operands:

operand1 arithmetic-operator operand2

- The operands can be constants (an actual number that does not change throughout the query), or data field names (representing a value that can change).
- Arithmetic operators are symbols that represent the four basic arithmetic operations.

The VISION:Bridge arithmetic operators are:

Arithmetic operators	Meaning
*	Multiplication
/	Division
+	Addition
-	Subtraction

The + and - symbols can also be leading signs to indicate positive and negative numbers.

Hierarchy of Operation

The operations within an arithmetic expression execute according to a specific hierarchy, basically from left to right.

Multiplication and division occur before addition and subtraction.

Writing Fractions

Fractions in arithmetic expressions must be expressed on one line using parentheses. For example:

$$\frac{A + 3 * C}{B - 6}$$

are written as:

```
(A + 3 * C) / (B - 6)
```

Notice the use of parentheses in this expression.

Using Parentheses

Parentheses clarify the order of operations to VISION:Bridge. Always use parentheses in pairs as a left "(" and a right ")". The number of left and right parentheses must always be the same.

- Operations enclosed in parentheses are calculated first.
- If you nest sets of parentheses, the innermost set is calculated first.
- Use parentheses at any time for clarification.

Notice the difference when the previous example is evaluated with parentheses and without them.

Without Parentheses	With Parentheses
12 + 3 * 9 / 9 - 6	(12 + 3 * 9) / (9 - 6)
12 + 27 / 9 - 6	(12 + 27) / (9 - 6)
12 + 3 - 6	(39) / (3)
Answer = 9	Answer = 13

The answers are different.

- Without parentheses, VISION:Bridge follows its default order of evaluation. Working from left to right, VISION:Bridge performs multiplication followed by division, and then addition followed by subtraction.
- With parentheses, the defaults are overridden. VISION:Bridge still performs multiplication before division and addition before subtraction, except now these operations occur within the parentheses first. Only after completely processing what is inside the parentheses, does VISION:Bridge turn to evaluating the outside, in this case, dividing 39 by 3.

Nesting Parentheses

This example shows how to nest parentheses, and combine them to form more complex hierarchies of evaluation. Assume the following values:

```
SALARY    =  1000
DEDUCT    =   200
SALES     =   300
COMMRATE  =   .10
REORDER   =   100
```

Now, replace the values in the following arithmetic expression, and evaluate it.

```
LET NET  =  SALARY + ((SALES + REORDER) * COMMRATE) - DEDUCT
```

The LET command performs arithmetic operations, and saves the result in a database field or a temporary field (see the section Temporary Fields).

A temporary field is a valid data field that is created in a query, and exists only for the duration of the query that creates it. It can be reported and used in expressions, just like any database field. Here, the value of NET is set to be the result of the calculation to the right of the equal sign.

```
LET NET = SALARY + ((SALES + REORDER) * COMMRATE ) - DEDUCT
LET NET = 1000 + ((300 + 100) * .10) - 200
LET NET = 1000 + (400 * .10) - 200
LET NET = 1000 + (40) - 200
LET NET = 1040 - 200
LET NET = 840
```

You can see by the order in which the terms are evaluated, that the inner set of parentheses forced VISION:Bridge to add the values of SALES and REORDER first, before multiplying by COMMRATE.

Notice, now, what happens when you remove the inner set of parentheses.

```
LET NET = SALARY + (SALES + REORDER * COMMRATE) - DEDUCT
LET NET = 1000 + (300 + 100 * .10) - 200
LET NET = 1000 + (300 + 10) - 200
```

```
LET NET = 1000 + 310 - 200
LET NET = 1310 - 200
LET NET = 1110
```

This is quite a difference. Since multiplication occurs before addition, VISION:Bridge multiplies REORDER by COMMRATE before adding the result of this calculation to SALES. Then, that value is added to SALARY and the value of DEDUCT is subtracted. Now, if VISION:Bridge merely evaluated expressions from left to right, watch what happens.

```
LET NET = SALARY + SALES + REORDER * COMMRATE - DEDUCT
LET NET = 1000 + 300 + 100 * .10 - 200
LET NET = 1300 + 100 * .10 - 200
LET NET = 1400 * .10 - 200
LET NET = 140 - 200
LET NET = -60
```

The result is quite different from the previous two examples.

Using Parentheses to Control the Order of Processing

VISION:Bridge does not evaluate strictly from left to right. By using parentheses like this:

```
LET NET = (((SALARY + SALES) + REORDER) * COMMRATE) - DEDUCT
```

you can force processing of the expression from left to right, overriding the multiplication / division / addition / subtraction rules entirely. Remember, what is inside the parentheses evaluates before what is outside. Parentheses are used in pairs — a left parenthesis and a right parenthesis.

Expression Summary

Relational expressions

```
IF      VALUE        GE       a * b +5
```

left hand value — relational operator — right hand value

The expression evaluates as true or false. Statements execute only if the expression is true.

Relational Operators

EQ or =	Equal to
LT or <	Less than
LE or <=	Less than or equal to
NE or ¬=	Not equal to
GT or >	Greater than
GE or >=	Greater than or equal to
TO	Connects two values to specify a range

Logical Expressions

```
IF     a GT b       OR      SALARY + 10 = 1500
```

left hand value — relational operator — arithmetic expression

Logical operators connect relational expressions. The logical expression evaluates as true or false. Statements only execute when the logical expression is true.

Logical Operators

AND	Both conditions must be true.
OR	At least one condition must be true.
NOT	Condition must be false.

Arithmetic expressions

```
a * b + (c - d) / 3.9
```

Arithmetic Operators

*	Multiplication	Hierarchy of operation (order of precedence)
/	Division	
+	Addition	
-	Subtraction	

Perform arithmetic operations only on numeric data. You can perform arithmetic operations on numeric characters that you define as alphanumeric, but you must be sure that the characters are numeric.

Hierarchy of operations

- Expressions evaluate from left to right.
- Parentheses are cleared first.
- Arithmetic Operators: Multiplication and division before addition and subtraction.
- Relational Operators: NOT before AND and OR.
- Logical Operators: AND before OR.

Building Conditional Phrases

Now you will learn how to combine relational, logical, and arithmetic expressions to form complex conditional phrases.

Example 1.

```
REPORT CUSTNO INAMT/12,
       IF INAMT + FRTCOST  GT 300  AND ORDCMPLT EQ ' N'
```

Arithmetic expression

Arithmetic expression

Relational expression

Relational expression

Conditional phrase

The first arithmetic expression (INAMT/12) is not part of the conditional phrase. It is on the left of the IF, and it tells VISION:Bridge to print the result of an arithmetic expression.

If the conditions specified to the right of the IF are true, VISION:Bridge executes the REPORT statement and prints the result of dividing INAMT by 12.

The conditional phrase is built from two relational expressions (IF INAMT + FRTCOST GT 300) and (ORDCMPLT EQ ' N') connected by the logical operator AND.

- The first relational expression also contains an arithmetic expression (INAMT + FRTCOST). To evaluate the first relational expression, VISION:Bridge adds the field values for INAMT and FRTCOST, and then tests the value of the result to see if it is greater than 300.
- The second relational expression (ORDCMPLT EQ ' N') tests for a field value for ORDCMPLT that is equal to the letter N. Since the letter N is an alphanumeric constant, it is enclosed in single quotation marks.

Finally, VISION:Bridge evaluates the results of the two relational expressions jointly. Since they are connected by an AND, both expressions must be true for the conditional phrase to be true. The values of CUSTNO and INAMT/12 are reported only if INAMT + FRTCOST is greater than 300 and ORDCMPLT is equal to the letter N.

Following the rules for evaluating expressions, VISION:Bridge evaluates this conditional phrase from left to right as if parentheses are included in the phrase:

```
REPORT CUSTNO INAMT/12,
       IF (INAMT + FRTCOST GT 300) AND (ORDCMPLT EQ 'N')
```

Example 2.

This example is much more complex than the previous one. Suppose that your company has a new policy that says freight costs will not exceed one percent of the invoice amount where there are quantities of five or more of any item. This policy is retroactive to January 1, 2001.

- You want to find customers who have been overcharged for shipping costs between January and December and have not yet paid for the merchandise. For these people, you will send a corrected invoice.
- You do not want to see those customers who have items backordered and have not paid. These people will be billed properly when the merchandise is shipped.

Therefore, we want to find and report the names of the customers that ordered items in quantities of five or more between January and December, 2001, have not yet paid for the orders, and whose freight cost exceeds one percent of the invoice amount.

Very often this is how your queries are first stated — as a word problem that lists the selection criteria to be applied to the database. The you translate the request into the VISION:Bridge language. The first step is to determine the field names you need.

You want to report CUSTOMERS	CUSTNO
Who ordered items in QUANTITIES of five or more	ITMQTYOR
Between January and December 1998	ORDRDATE
And have not PAID	PAIDDATE
And whose FREIGHT COSTS	FRTCOST
Exceed one percent of INVOICE AMOUNT	NAMT
Unless they have ITEMS BACKORDERED	QTYBKORD

The following shows how to specify the values of these fields that satisfy the request.

Connector	Field	Condition
	ITMQTYOR	GE 5
AND	ORDRDATE	GE 010101 AND LE 011231
AND	PAIDDATE	EQ ' '
AND	FRTCOST	GT (INAMT * .01)
AND	QTYBKORD	EQ 0

Now, the conditional phrase is almost written. Insert the appropriate logical connectors to string the conditions together into one conditional phrase.

Here is the conditional phrase that selects the data as specified in the statement of the problem.

```
REPORT  CUSTNO CUSTNAME FRTCOST INAMT,
        IF ITMQTYOR GE 5 AND,
           (ORDRDATE GE 010101 AND ORDRDATE LE 011231) AND,
            PAIDDATE EQ '     ' AND (FRTCOST GT (INAMT * .01)),
            AND QTYBKORD EQ 0
```

Figure 6-2 shows the entire query and the report it produces. VISION:Bridge evaluates the conditional phrase from left to right, evaluating what is inside the parentheses before evaluating what is outside. As you can see, the selection criteria were much more difficult to say than they were to write using VISION:Bridge.

Query

```
QUERY CUSTOMER
REPORT CUSTNO CUSTNAME INVNO INAMT FRTCOST,
       IF ITMQTYOR GE 5 AND,
          (ORDRDATE GE 010101 AND ORDRDATE LE 011231) AND,
          (FRTCOST GT(INAMT * .01)) AND PAIDDATE EQ '      '
          AND QTYBKORD EQ 0
TITLE 'COMPLEX CONDITIONAL PHRASE'
END REPORT
END QUERY
```

Output

```
NOV 19, 2001            COMPLEX CONDITIONAL PHRASE           PAGE 1
-------------------------------------------------------------------------
CUSTOMER          CUSTOMER            INVOICE   INVOICE   FREIGHT
NUMBER              NAME              NUMBER    AMOUNT    COST
-------------------------------------------------------------------------
00001     TULSA TIMES                 I1627       36.00     2.00
00048     STONEWELL STATE SYSTEMS     I8206       18.00     1.00
```

Figure 6-2 Complex Conditional Selection

NOT Operator

The NOT operator negates a logical expression, where a logical expression is two relational expressions tied together with AND or OR. In Figure 6-3 the report lists all customer numbers, except 00043 and 00115.

Query

```
QUERY CUSTOMER
REPORT CUSTNO INAMT,
IF NOT (CUSTNO EQ '00043' OR CUSTNO EQ '00115')
TITLE 'USING THE ''NOT'' KEYWORD'
END REPORT
END QUERY
```

Output

```
NOV 03,2001          USING THE 'NOT' KEYWORD           PAGE 1

CUSTOMER    INVOICE
NUMBER      AMOUNT

00001      270.00
00001       36.00
00013      105.00
00013      210.00
00013      140.00
00028       22.00
00048      305.00
00048      260.00
00048       18.00
00089           -
00089           -
00089       75.00
```

Figure 6-3 Using the NOT Operator

In this case, you use the NOT operator to code the two instances when you do not want VISION:Bridge to display information. This saves you from coding each customer number in a relational expression using the EQ operator to obtain the same report as in Figure 6-3.

You can code most queries more than one way to yield the same results. Use the commands and operators that make sense to you.

No Data Selected

It is possible for you to code a query that produces a report that has no data. That is, your report prints with a title, date, column headings, and so on, but only dashes print where you think data should be.

With Immediate Response, the following message can display:

```
NR07** NO DATA SELECTED, DB CALL COUNT = ##
```

This tells you that there is no data in the database that satisfies your selection criteria.

- If your conditional phrase was accurate, this might be the right output. Sometimes, no data can satisfy the selection criteria you specify.
- If, however, you should have received data, check your conditional phrase carefully. It is easy to specify criteria that can never be satisfied.

Using AND Instead of OR

The most common mistake is using AND instead of OR. For example,

```
REPORT ORDERNO INAMT,
       IF CUSTNO EQ 00013 AND CUSTNO EQ 00048
```

is a simple REPORT statement asking to see orders for customers 00013 and 00048.

Figure 6-4 shows the report this statement produces. As you see, no data was selected, because CUSTNO cannot be two values at the same time. It can be 00013 or 00048, but it can never be both 00013 and 00048. As a result, VISION:Inform could not find a case that satisfied the conditions specified.

```
NOV 09, 2001               NO DATA SELECTED                PAGE 1
-------------------------------------------------------------------
ORDER    INVOICE
NUMBER   AMOUNT
-------------------------------------------------------------------
    -         -
```

Figure 6-4 Criteria Cannot be Met

This is a common way to select no data from a database. More complex conditional phrases can do the same thing, but are more difficult to debug.

If you think you should be receiving data, but you are not, check to see if there are ANDs where ORs should be. The opposite problem, receiving more data than you want could be the result of the opposite condition, using ORs where ANDs should be.

Finally, be careful when using the NOT operator — it changes the meaning of expressions. For a detailed explanation of the NOT operator, see the section NOT Operator.

Selecting Data

There are several commands you can use to extract precisely the data you want. One is the SELECT command.

The SELECT command restricts the range of the database that can be accessed by the statements within its control. It establishes a limited context — to look at the database from a specific point of view.

The SELECT command can:

- Define a range by placing conditional phrases in the SELECT statement or in the statements within its control.
- Specify selection criteria that can span many REPORT groups. The context established by a SELECT command remains in effect until VISION:Bridge encounters an END SELECT command. For example, if you use:

```
SELECT SEGMENT ITEMORD IF ITEMORD EQ 'G46'
 REPORT CUSTNAME
 END REPORT
 REPORT ORDERNO ORDRDATE
 END REPORT
END SELECT
```

  you receive two reports. The data reported on both is restricted to cases where VISION:Bridge finds an occurrence of the ITEMORD field in the ITEMORD segment equal to G46. In other words, this query says, if anyone ordered a G46, print the customer's name on one report and the order number and date on another report.
- Establish the summary level for VISION:Bridge built-in summaries. (See the section A Sample VISION:Bridge Query).

  Do not confuse built-in summaries with the VISION:Bridge summary commands (see the section A Sample VISION:Bridge Session), or the Immediate Response summary keywords (in the section Displaying Summary Information).

  Built-in summaries are computed within data records. You can use built-in summaries in reports and in data selection statements. However, since they are computed within a record, as opposed to report summaries that are computed across all records, they are sensitive to looping, which is discussed in Chapter 7, *Advanced Topics*.

Establishing a SELECT Group

Always use the SELECT command with an END SELECT command to form a SELECT group. A SELECT group begins with a SELECT command, includes a variety of processing statements, and ends with an END SELECT command. The data context established by a SELECT command remains in effect until VISION:Bridge finds an END SELECT.

The SELECT defines the first cut at the database. The statements within a SELECT group can only access the data specified in the SELECT statement. This eliminates the need to continually redefine the data that should be processed and also makes efficient use of system resources.

Conditional IF phrases added to the SELECT statement, or to statements within the group, further refine the data that can be accessed. While additional statements can further limit the data that is reported, they cannot override the limits set by a SELECT command.

- In Immediate Response, the END SELECT is the only way to release control of a SELECT command in a query.
- In VISION:Bridge, you can use either an END SELECT command to release control of a SELECT command within a query, or use an END QUERY command to release control of the SELECT command at the end of the query.

SELECT Command

SELECT FIELD name(s) SEGMENT name(s), IF logical expression

The SELECT command indicates that access to the information in the database is being restricted. All the statements between this and the related END SELECT command are under the control of its terms.

FIELD and SEGMENT are both optional. You can use both in the same SELECT statement. If neither is listed, the database is entered at the root level by default.

FIELD name(s)	One or more field names indicating the level at which the database is entered. The view of the database is the same as that accessed by the related segment level. Use of the keyword FIELD is optional if a field name appears first.
SEGMENT name(s)	One or more segment names indicating the level at which the database is entered.
IF logical expression	Optional conditional phrase that further limits the range of data that can be accessed.
END SELECT	The last statement in the SELECT group. Marks the end of control of a SELECT command.

You can nest SELECT groups (see the section Nested SELECT Groups) where each inner group is under control of any outer groups.

You can list SELECT groups sequentially (see the section Sequential SELECT Groups) where each group must relinquish control before the next one begins.

SELECT statements can also contain only a conditional phrase. In these cases, VISION:Bridge and Immediate Response automatically access the root segment.

You cannot use a SELECT command in a VISION:Bridge IF group (see the section IF/ELSE Statements).

Using the SELECT Statement

This section contains SELECT statement examples:

- Without conditions
- With a condition
- Reporting fields not in the SELECT segment

SELECT Statement Without Conditions

Figure 6-5 demonstrates the effect of the SELECT statement. The SELECT statement instructs VISION:Bridge to enter the database at the ORDER segment. ORDER is a level 2 segment that is directly subordinate to the root segment, CUSTOMER. Customers have many orders. Each occurrence of the ORDER segment contains information about a different order.

The SELECT statement lists no conditions, so all occurrences of the segment are available. The use of the REPORT statement with no field names tells VISION:Bridge to list all the data in the ORDER segment.

Query

```
QUERY CUSTOMER
 SELECT SEGMENT ORDER
  REPORT
   TITLE 'SELECT SEGMENT W/OUT QUALIFICATION'
  END REPORT
END QUERY
```

Output

```
NOV 09,2001        SELECT SEGMENT W/OUT QUALIFICATION       PAGE 1
-------------------------------------------------------------------
ORDER   ORDER   SALES   ORDER   ORDER     ORDER   ORDER
NUMBER  DATE    PERSON  P.O.    DUE DATE  INVOICE COMPLETE?
-------------------------------------------------------------------
07321   010228   RGH    A0004  010312       Y       N
08432   010801   JKW    A0002  011101       Y       Y
01142   010606   BOW    B4201  010706       Y       Y
03619   010817   GAE    X3401  010917       Y       Y
09541   011002   BIT    E6590  011130       Y       N
11211   010316   JMS    X7658  010430       Y       Y
23468   010820   ANS    M1345  010830       Y       Y
06573   010101   SMM    A4801  010201       Y       Y
06781   010710   BCC    A4802  010801       Y       Y
03921   010417   AJC    C8901  010510       N       N
13842   010610   JEE    C8902  010730       N       N
14821   010815   ANS    C8903  010930       Y       Y
00284   010712   PEG    A1151  010730       Y       Y
01391   010525   ART    B5566  010625       Y       N
```

Figure 6-5 Reporting a Segment Without Qualification

SELECT Statement With Conditions

Figure 6-6 is slightly different than Figure 6-5. The SELECT statement contains a conditional phrase. It tells VISION:Bridge to select the ORDER segment only if the order date is greater than 8/17/01 (010817).

Again, the REPORT statement lists no fields. Whenever an occurrence of the ORDER segment is found where the value of ORDRDATE is greater than 010817, VISION:Bridge prints the entire segment.

Only two segment occurrences met this conditional test. A look at the values of the field labeled order date verifies that both of the orders selected have order dates that meet the condition specified.

Once you specify a conditional phrase in a SELECT statement, you are restricted from accessing data that does not meet the conditions. This restriction remains in effect until an END SELECT or END QUERY is encountered. This is demonstrated in Figure 6-7.

Query

```
QUERY CUSTOMER
 SELECT SEGMENT ORDER IF ORDRDATE GT 010817
 REPORT
  TITLE 'SELECT SEGMENT WITH QUALIFICATION'
 END REPORT
END QUERY
```

Output

```
NOV 09, 2001        SELECT SEGMENT WITH QUALIFICATION         PAGE 1
-------------------------------------------------------------------------
ORDER   ORDER   SALES   ORDER   ORDER     ORDER     ORDER
NUMBER  DATE    PERSON  P.O.    DUE DATE  INVOICE   COMPLETE?
-------------------------------------------------------------------------
09541   011002  BIT     E6590   011130       Y         N
23468   010820  ANS     M1345   010830       Y         Y
```

Figure 6-6 Qualifying a Report

Reporting Fields not in the SELECT Statement Segment

Using the same SELECT statement from Figure 6-6, a REPORT statement is added that requests data from a higher level segment.

In Figure 6-7, the conditional phrase on the SELECT statement remains, but several fields are added to the REPORT statement. Specifically, this example demonstrates that you are not restricted to reporting from the segment in the SELECT statement.

You can report any fields from any segment in the database, but the values selected for output are limited to the viewpoint established by the SELECT statement. In this case, the output is limited to occurrences of the ORDER segment where the order date is greater than 010817.

Thus, the values of CUSTNO and CUSTNAME, although they are higher in the database structure than the ORDER segment, are limited to customers who have orders that meet the conditions specified.

Query

```
QUERY CUSTOMER
 SELECT SEGMENT ORDER IF ORDRDATE GT 010817
  REPORT CUSTNO CUSTNAME ORDERNO ORDRDATE
   TITLE 'SELECT SEGMENT WITH QUALIFICATION'
   TITLE 'REPORT FROM HIGHER LEVEL SEGMENT'
 END REPORT
END QUERY
```

Output

```
NOV 09, 2001          SELECT SEGMENT WITH QUALIFICATION          PAGE 1
                      REPORT FROM HIGHER LEVEL SEGMENT
-------------------------------------------------------------------------
CUSTOMER              CUSTOMER              ORDER    ORDER
NUMBER                  NAME                NUMBER   DATE
-------------------------------------------------------------------------
00013       ESSEX ENTERPRISES               09541    011002
00043       BANK OF THE PEOPLE              23468    010820
```

Figure 6-7 Reporting Fields from a Higher Level Segment

Different Selection Methods

You could have achieved the same results, without the SELECT statement, by adding a conditional phrase to the REPORT statement.

```
REPORT CUSTNO CUSTNAME ORDERNO ORDRDATE,
       IF ORDRDATE GT 010817
```

is equivalent to:

```
SELECT SEGMENT ORDER IF ORDRDATE GT 010817
REPORT CUSTNO CUSTNAME ORDERNO ORDRDATE
```

Both forms have their uses.

- To produce a single report you can choose the REPORT statement and a conditional phrase. It is a shorter statement and easier to see what is happening just by looking at the statement.
- To produce a number of reports based on the same selection criteria, use SELECT. The SELECT criteria remains in effect until an END SELECT is found. Remember, you can have multiple REPORT groups inside a SELECT group.

Using Selection Criteria in a SELECT Group

You can also further refine the selection criteria within a SELECT group. Figure 6-8 and Figure 6-9 shows a query and two reports.

- The first report is based on the conditional phrase in the SELECT statement.
- The second report has additional selection criteria in the REPORT statement.

First, select all the orders that are incomplete. That is, find all the occurrences where ORDCMPLT EQ 'N'. Do this with the SELECT statement. The first report lists the customer numbers, customer names, and order numbers for orders that are incomplete.

When orders are not shipped while awaiting backordered items, the second report can list the backordered items for each incomplete order. Use the GROUP statement to make the reports easier to read.

Query

```
QUERY CUSTOMER
 SELECT SEGMENT ORDER IF ORDCMPLT EQ 'N'
  REPORT CUSTNO CUSTNAME ORDERNO QTYBKORD
   GROUP BY CUSTNO CUSTNAME
   TITLE 'MULTIPLE REPORTS IN A SELECT GROUP'
   TITLE 'NO SELECTION ON REPORT STATEMENT'
  END REPORT
  REPORT CUSTNO ORDERNO ITMQTYOR ITEMNAME QTYBKORD,
  IF QTYBKORD NE 0
   GROUP BY CUSTNO ORDERNO
   TITLE 'MULTIPLE REPORTS IN A SELECT GROUP'
   TITLE 'CONDITIONAL PHRASE ON REPORT STATEMENT'
  END REPORT
 END SELECT
END QUERY
```

Output

```
NOV 09, 2001          MULTIPLE REPORTS IN A SELECT GROUP       PAGE 1
                       NO SELECTION ON REPORT STATEMENT

CUSTOMER             CUSTOMER              ORDER    BACKORD
NUMBER                NAME                NUMBER   QUANTITY

00001    TULSA TIMES                       07321
                                           07321
                                           07321
                                           07321            20
00013    ESSEX ENTERPRISES                 09541
                                           09541             2
00089    GOING PLACES INSURANCE COMPANY    03921
                                           13842
00115    SUNNY GRAPE GROWERS               01391
                                           01391             1

NOV 09, 2001        MULTIPLE REPORTS IN A SELECT GROUP         PAGE 1
                  CONDITIONAL PHRASE ON REPORT STATEMENT

CUSTOMER  ORDER    QUANTITY           ITEM           BACKORD
 NUMBER   NUMBER   ORDERED            NAME           QUANTITY

00001    07321           20   M4 RUN CNTL. FORM            20
00013    09541            2   M4 OPS GUIDE-OS               2
00115    01391            1   M4 OPS GUIDE-OS               1
```

Figure 6-8 Multiple Reports in Select Group

Looping under SELECT control is always from the top down.

- If levels 1 and 2 are both accessed, the information at level 1 is processed once for each occurrence of level 2.
- If levels 2, 3, and 4 are accessed, the information from level 2 is processed once for each occurrence of level 3 and each occurrence of level 4.

You can also use additional SELECT statements to further refine the data context. SELECT groups can follow one another (sequential SELECT groups), or they can be nested inside one another (nested SELECT groups).

Sequential SELECT Groups

With sequential SELECT groups, the influence of one SELECT group explicitly ends by an END SELECT command before encountering the next SELECT command.

```
        QUERY DATABASE

      ⎧ SELECT. . .
A     ⎪ . . .
      ⎪ . . .
      ⎩ END SELECT

      ⎧ SELECT. . .
      ⎪ . . .
B     ⎪ . . .
      ⎩ END SELECT

        END QUERY
```

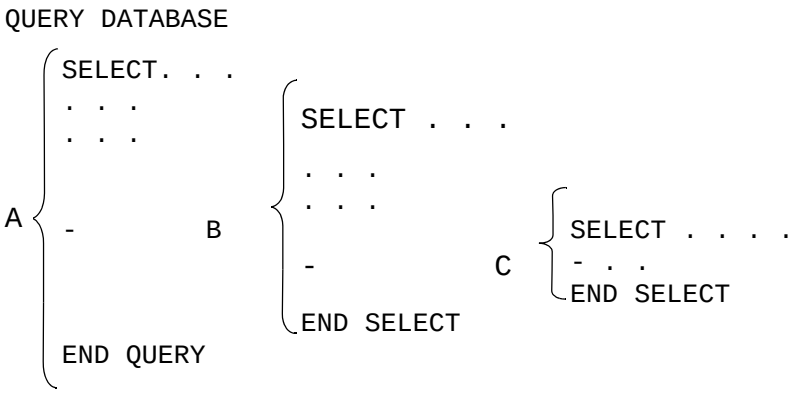
This has several implications:

- Processing loops started by a SELECT command stop at the END SELECT command. Therefore, loops do not remain in effect from one SELECT group to the next.
- The selection criteria do not extend across SELECT groups. Sequential SELECT groups are completely independent of one another.

An example of sequential SELECT groups follows in Figure 6-9 and Figure 6-10.

Nested SELECT Groups

A nested SELECT group is one that falls inside an existing SELECT group in a query. Nested SELECT groups are interdependent. The inner group is under control of the outer group and provides an additional level of data selection.



In the example above, SELECT group B is under the control of SELECT group A. Group C is under the control of both groups A and B. A processing effect started in group A, for example, continues through to the END SELECT for group A.

Figure 6-9, Figure 6-10, and Figure 6-11 show a query with three SELECT groups. Two of them are sequential and two are nested. The second SELECT group is nested inside the first, and further refines the data context established by the first SELECT group.

Note: Statements need not be indented; however, it adds clarity and makes it easier to understand.

The third SELECT group, which appears after the END SELECT commands of the first two, is completely independent of the first two groups. The data context established for the third report is limited only by the conditions specified in the third SELECT statement.

Query

```
QUERY CUSTOMER
 SELECT SEGMENT ORDER IF ORDCMPLT EQ 'N'
  REPORT CUSTNO CUSTNAME ORDERNO QTYBKORD ORDCMPLT
   GROUP BY CUSTNO CUSTNAME
   TITLE 'FIRST SELECT GROUP'
  END REPORT
   SELECT SEGMENT ITEMORD IF QTYBKORD NE 0
    REPORT CUSTNO ORDERNO ITMQTYOR,
     ITEMNAME QTYBKORD ORDCMPLT
     GROUP BY CUSTNO ORDERNO
     TITLE 'A NESTED SELECT GROUP'
    END REPORT
   END SELECT
 END SELECT
 SELECT SEGMENT ORDER IF ORDCMPLT EQ 'Y'
  REPORT CUSTNO CUSTNAME ORDERNO QTYBKORD ORDCMPLT
   GROUP BY CUSTNO CUSTNAME
   TITLE 'A SEQUENTIAL SELECT GROUP'
  END REPORT
 END SELECT
END QUERY
```

First REPORT group

Second REPORT group

Third REPORT group

Figure 6-9 Using Both Sequential and Nested SELECT Groups (Page 1 of 3)

Output

```
NOV 09, 2001                FIRST SELECT GROUP                 PAGE 1

CUSTOMER             CUSTOMER           ORDER   BACKORD     ORDER
NUMBER                NAME              NUMBER  QUANTITY  COMPLETE?

00001    TULSA TIMES                    07321                 N
                                        07321                 N
                                        07321                 N
                                        07321          20     N
00013    ESSEX ENTERPRISES              09541                 N
                                        09541           2     N
00089    GOING PLACES INSURANCE COMPANY 03921                 N
                                        13842                 N
00115    SUNNY GRAPE GROWERS            01391                 N
                                        01391           1     N

NOV 09, 2001              A NESTED SELECT GROUP                PAGE 1

CUSTOMER  ORDER    QUANTITY          ITEM          BACKORD    ORDER
 NUMBER   NUMBER   ORDERED           NAME          QUANTITY COMPLETE?

00001    07321           20  M4 RUN CNTL. FORM         20     N
00013    09541            2  M4 OPS GUIDE-OS            2     N
00115    01391            1  M4 OPS GUIDE-OS            1     N
```

First report

Second report

Figure 6-10 Using Both Sequential and Nested SELECT Groups (Page 2 of 3)

```
NOV 09, 2001              A SEQUENTIAL SELECT GROUP              PAGE 1
-------------------------------------------------------------------------
CUSTOMER              CUSTOMER                ORDER   BACKORD    ORDER
 NUMBER                 NAME                  NUMBER  QUANTITY COMPLETE?
-------------------------------------------------------------------------
00001    TULSA TIMES                          08432                 Y
00013    ESSEX ENTERPRISES                    01142                 Y
                                              01142                 Y
                                              01142                 Y
                                              03619                 Y
                                              03619                 Y
                                              03619                 Y
00028    AMERICAN WIRE AND RING SERVICE       11211                 Y
00043    BANK OF THE PEOPLE                   23468                 Y
00048    STONEWELL STATE SYSTEMS              06573                 Y
                                              06573                 Y
                                              06781                 Y
                                              06781                 Y
                                              06781                 Y
                                              06781                 Y
00089    GOING PLACES INSURANCE COMPANY       14821                 Y
```

Third report

Figure 6-11 Using Both Sequential and Nested SELECT Groups (Page 3 of 3)

Each SELECT group has one REPORT statement that produces a report.

The first REPORT statement in Figure 6-9 reports data if the value of the ORDCMPLT field is equal to N. With the exception of the ORDCMPLT field being reported, the first report is identical to the first report in Figure 6-8 since the SELECT statements in the two examples are the same.

The first REPORT group is followed by another SELECT statement that says SELECT IF QTYBKORD NE 0. Since no END SELECT is encountered, the second SELECT group operates within the context established by the first SELECT group and is said to be nested under the first SELECT. Only if ORDCMPLT is equal to N does VISION:Bridge look to see if *any* occurrence of QTYBKORD is not equal to 0. The important word is "any."

Since there are multiple occurrences of the ITEMORD segment (each order can have many items), there are multiple occurrences of the field QTYBKORD, which is in the ITEMORD segment.

> Does IF QTYBKORD NE 0 mean if any occurrence of QTYBKORD not equal to 0, or does it mean if every occurrence of QTYBKORD not equal 0?

The SELECT statement means that if any occurrence of QTYBKORD is not equal to 0, then the entire segment is available for reporting. Compare the second report in Figure 6-8 and the second report in Figure 6-10, and you can see what this means on the report.

The second report in Figure 6-8 used a single SELECT statement, and then specified IF QTYBKORD NE 0, on the REPORT statement. Here are the two sets of selection criteria.

```
SELECT IF ORDCMPLT EQ 'N'
REPORT...IF QTYBKORD NE 0
```

Figure 6-9 Selection

```
SELECT IF ORDCMPLT EQ 'N'
SELECT IF QTYBKORD NE 0
       REPORT...
```

Figure 6-8 Selection

In Figure 6-8, the first report printed every occurrence of QTYBKORD if any occurrence was not equal to 0.

In Figure 6-8, the second report only reported occurrences of QTYBKORD that did not equal to 0. Although it reported more data than was wanted, the second report in Figure 6-10 did report exactly as requested.

Finally, the third report in Figure 6-11 is based on a SELECT statement that falls outside of the influence of the first two SELECT statements. The two END SELECT commands, one each for the first and second SELECT statements, releases control to the third SELECT statement. They no longer have influence over the data selection. The third report confirms this. This time data was to be selected if ORDCMPLT is equal to Y. Note that only records where ORDCMPLT is equal to Y are reported.

There is no limit to the number of SELECT statements you can use in a query. Remember that the effect of a SELECT continues until an END SELECT command is encountered.

IF/ELSE Statements

This section is relevant to VISION:Bridge only. See the section Selection Criteria for Displays for equivalent information using Immediate Response.

The IF Group

The IF command starts a conditional statement that can stand alone. The conditions specified in an IF command remain in effect until an END IF command is encountered. An IF command must always have an associated END IF command. All the statements between an IF command and an END IF command form what is known as an IF group.

IF Command vs. Conditional Phrase

The IF command indicates that the statements under its control execute only if the conditions specified in the IF statement are true.

Whereas, a conditional phrase attached to a REPORT or SELECT statement, instructs VISION:Bridge to execute the command if the conditional phrase is true, the IF command says if the conditional phrase is true, then execute all the command statements that follow until an END IF is encountered. END IF releases the influence of the IF command.

- Additional query statements can follow an IF group.
- A query can have more than one IF group.

ELSE Command

By removing the conditional phrase from individual commands and letting it stand alone, VISION:Bridge can perform alternate actions when the IF statement is not true.

The ELSE command, used in an IF group, can perform an alternate set of instructions, if the conditions specified on the logical expression in the IF statement are false. Alternate commands, such as REPORT commands, can follow the ELSE command.

The ELSE command is optional in an IF group.

Exception Reports

The IF/ELSE combination is very useful for creating exception reports. In effect, IF/ELSE divides the database into two independent groups of data.

- The IF command limits the data available to that which satisfies the logical expression.
- All the data that does not satisfy the logical expression on the IF statement is made available for processing following the ELSE command.

For example, imagine a personnel file with data about all the employees in a company. To produce two reports, one for female employees and one for male employees, write a query like this:

```
QUERY PERSONNEL
 IF SEX = 'F'
  REPORT NAME SSNUMBER
  END REPORT
 ELSE
  REPORT NAME SSNUMBER
  END REPORT
 END IF
END QUERY
```

The data that fails to meet the IF SEX = 'F' condition is under control of the ELSE portion of the IF statement.

- This means all records where SEX = 'M' appear in the second report.
- If there are any values of SEX other than M or F, they are also included in the second report.

You can use IF/ELSE to search for invalid data in a database. If you specify all the valid values of a field on the IF statement, the ELSE command provides access to all the records containing values other than those specified on the IF statement.

IF Groups (IF/ELSE Groups)

An IF group begins with an IF statement, includes a number of processing statements, and ends with an END IF command.

- The IF group can contain an ELSE command.
- IF groups can contain all statements, except the QUERY and SELECT statements.

IF/ELSE Commands

IF EXPR logical expression
ELSE

An IF statement always begins with the IF command followed by a logical expression.

- The logical expression in the IF statement can contain more than one condition.
- The IF command indicates that the statement under its control executes only if the logical expression is true.

EXPR logical expression	Contains the conditions that must be met for the expression to be true. EXPR is optional if a logical expression appears immediately after IF.
ELSE	The optional ELSE command indicates that an alternate set of instructions executes, if the logical expression is false.
END IF	Marks the end of an IF or IF/ELSE group.

Using the IF Command

Figure 6-12 shows a report that uses only an IF command. Notice the END IF command. The IF command to the END IF command forms an IF group. The query requests data for those orders that have invoices. The field ORDINVGN contains a Y if the order has an invoice and an N if it does not.

ORDINVGN is a field in the level 2 ORDER segment. Imagine a two step selection process.

- Only those records, where at least one occurrence of the ORDINVGN field is equal to Y, are available for processing.
- Within those records, only the occurrence of the ORDER segment where ORDINVGN is equal to Y are available for reporting.

As seen in Figure 6-12, every order number reported shows ORDINVGN to be a Y. Compare this with Figure 6-13 where ELSE was added to the same query.

Query

```
QUERY CUSTOMER
 IF ORDINVGN = 'Y'
 REPORT CUSTNO ORDERNO ORDINVGN
   GROUP BY CUSTNO ORDERNO
   TITLE 'SINGLE IF STATEMENT'
  END REPORT
 END IF
END QUERY
```

Output

```
NOV 09, 2001                SINGLE IF STATEMENT            PAGE 1
------------------------------------------------------------------------
CUSTOMER  ORDER    ORDER
 NUMBER   NUMBER  INVOICE
------------------------------------------------------------------------
00001    07321      Y
         08432      Y
00013    01142      Y
         03619      Y
         09541      Y
00028    11211      Y
00043    23468      Y
00048    06573      Y
         06781      Y
00089    14821      Y
00115    00284      Y
         01391      Y
```

Figure 6-12 Using a Single IF Statement

The following query uses an IF group, and produces the report in Figure 6-13. The END IF command identifies the end of the IF group to VISION:Bridge. The query statements that follow the END IF are not under the control of the IF statement.

- As seen on the first report, all the orders listed show ORDINVGN to be a Y.
- The second report is based on the ELSE criteria. All orders reported have an ORDINVGN value of N.

Query

```
QUERY CUSTOMER
 IF ORDINVGN = 'Y'
  REPORT CUSTNO ORDERNO ORDINVGN
   GROUP BY CUSTNO ORDERNO
   TITLE 'PART 1 -- THE IF STATEMENT'
  END REPORT
 ELSE
  REPORT CUSTNO ORDERNO ORDINVGN
   GROUP BY CUSTNO ORDERNO
   TITLE 'PART 2 -- THE ELSE STATEMENT'
  END REPORT
 END IF
END QUERY
```

Output

```
NOV 09, 2001             PART 1 --  THE IF STATEMENT           PAGE 1

CUSTOMER  ORDER    ORDER
 NUMBER   NUMBER  INVOICE

00001    07321      Y
         08432      Y
00013    01142      Y
         03619      Y
         09541      Y
00028    11211      Y
00043    23468      Y
00048    06573      Y
         06781      Y
00089    14821      Y
00115    00284      Y
         01391      Y

NOV 09, 2001           PART 2 -- THE ELSE STATEMENT           PAGE 1

CUSTOMER  ORDER    ORDER
 NUMBER   NUMBER  INVOICE

00089    03921      N
         13842      N
```

Figure 6-13 Using the IF/ELSE Statement (Page 1 of 2)

Multiple Reports in One IF Group

You can have more than one report within an IF group. Figure 6-14 shows an IF statement that requests data if CUSTNO = 00013. Within the context of the IF statement, there are requests for two reports.

- The first specifies orders and items for those orders that are complete (ORDCMPLT = 'Y').
- The second report specifies order numbers and items for those orders that are not complete (ORDCMPLT = 'N').

Both of these reports fall under the condition that CUSTNO = 00013.

Query

```
QUERY CUSTOMER
 IF CUSTNO = '00013'
  REPORT CUSTNO ORDERNO ITEMORD ORDCMPLT,
   IF ORDCMPLT = 'Y'
   GROUP BY CUSTNO ORDERNO
   TITLE 'WITHIN AN IF GROUP -- FIRST REPORT'
   END REPORT
  REPORT CUSTNO ORDERNO ITEMORD ORDCMPLT,
   IF ORDCMPLT = 'N'
   GROUP BY CUSTNO ORDERNO
   TITLE 'WITHIN AN IF GROUP -- SECOND REPORT'
  END REPORT
 END IF
END QUERY
```

Output

```
NOV 09,2001        WITHIN AN IF GROUP -- FIRST REPORT        PAGE 1

CUSTOMER  ORDER    ITEMS     ORDER
 NUMBER   NUMBER  ORDERED  COMPLETE?

00013     01142   803          Y
                  857          Y
                  858          Y
          03619   045          Y
                  113          Y
                  114          Y

NOV 09, 2001       WITHIN AN IF GROUP -- SECOND REPORT       PAGE 1

CUSTOMER  ORDER    ITEMS     ORDER
 NUMBER   NUMBER  ORDERED  COMPLETE?

00013    09541   017          N
                 019          N
```

Figure 6-14 Multiple Reports Within a Single IF Group

You could write this query using nested IFs. See Figure 6-15. Instead of adding a conditional phrase to each REPORT statement, you could specify IF ORDCMPLT = 'Y' on an IF statement and use the ELSE to access all the orders where ORDCMPLT is not equal to Y. Assume those orders where ORDCMPLT = N.

Figure 6-15 shows a more complex report. It uses nested IFs and two IF/ELSE groups to produce three reports. Conditions established by an IF statement remain in effect until an END IF is encountered. Therefore, the first IF at line 2 remains in effect until line 19. The second IF, at line 3 remains in effect until line 13.

This is the general structure of the query in Figure 6-15:

```
IF
  IF
    REPORT
  ELSE
    REPORT
  END IF
ELSE
  REPORT
END IF
```

Query

```
1       QUERY CUSTOMER
2        IF ORDINVGN = 'Y'
3          IF ORDCMPLT = 'Y'
4           REPORT CUSTNO ORDERNO ORDINVGN ORDCMPLT
5            GROUP BY CUSTNO ORDERNO
6           TITLE 'PART 1 -- IF STATEMENT -- 1ST REPORT'
7          END REPORT
8          ELSE
9          REPORT CUSTNO ORDERNO ORDINVGN ORDCMPLT
10          GROUP BY CUSTNO ORDERNO
11          TITLE 'PART 2 -- IF STATEMENT -- 2ND REPORT'
12         END REPORT
13         END IF
14       ELSE
15        REPORT CUSTNO ORDERNO ORDINVGN ORDCMPLT
16         GROUP BY CUSTNO ORDERNO
17         TITLE 'PART 3 -- ELSE STATEMENT -- 3RD REPORT'
18        END REPORT
19       END IF
20      END QUERY
```

Figure 6-15 Nested IFs and Multiple IF Groups (Page 1 of 2)

In Figure 6-15, the first REPORT statement selects customer records where both ORDINVGN (Order Invoice Generated) and ORDCMPLT (Order Complete) are Y.

In Figure 6-15, the ELSE at line 8 is under the influence of the second IF statement and causes VISION:Bridge to select those records where ORDCMPLT is anything other than Y.

As a result, the second report in Figure 6-16 shows that all occurrences of ORDINVGN are Y and all occurrences of ORDCMPLT are N.

In Figure 6-15, the END IF at line 13 releases the control of the second IF statement so the ELSE immediately following at line 14 refers to the first IF statement. In Figure 6-16, the report lists only those orders where ORDINVGN are not Y.

Output

```
NOV 09, 2001        PART 1 -- IF STATEMENT -- 1ST REPORT      PAGE 1
-------------------------------------------------------------------------
CUSTOMER  ORDER    ORDER     ORDER
 NUMBER   NUMBER  INVOICE  COMPLETE?
-------------------------------------------------------------------------
00001    08432      Y         Y
00013    01142      Y         Y
         03619      Y         Y
00028    11211      Y         Y
00043    23468      Y         Y
00048    06573      Y         Y
         06781      Y         Y
00089    14821      Y         Y
00115    00284      Y         Y

NOV 09, 2001      PART 2 -- IF STATEMENT -- 2ND REPORT       PAGE 1
-------------------------------------------------------------------------
CUSTOMER  ORDER    ORDER     ORDER
 NUMBER   NUMBER  INVOICE  COMPLETE?
-------------------------------------------------------------------------
00001    07321      Y         N
00013    09541      Y         N
00115    01391      Y         N

NOV 09, 2001       PART 3 -- ELSE STATEMENT -- 3RD REPORT     PAGE 1
-------------------------------------------------------------------------
CUSTOMER  ORDER    ORDER     ORDER
 NUMBER   NUMBER  INVOICE  COMPLETE?
-------------------------------------------------------------------------
00089    03921      N         N
         13842      N         N
```

Figure 6-16 Nested IFs and Multiple IF Groups (Page 2 of 2)

Temporary Fields

Temporary fields are data fields that you create. They only exist for the query that creates them.

- You can use them like any database field.
- You can report temporary fields, use them in calculations, and use them as a basis for data selection.
- You can also specify the values that the temporary fields contain and you can change their values during the execution of a query.

Creating Temporary Fields

There are two VISION:Bridge commands that you use to create temporary fields. One is called the SET command (see the section Using SET with LET) and the other is called the LET command (see the section Using LET).

Defining Explicit Temporary Fields

The SET command creates an explicitly defined temporary field. That means you explicitly specify the attributes of the field (such as field type, field length, and the number of decimal places) right in the SET statement at the time the field is created.

A SET statement can assign an initial value to a temporary field, but it cannot change the value after it has been assigned.

Defining Implicit Temporary Fields

The LET command can change the values of temporary fields. This is the main use for the LET command.

- With the LET command, you can have temporary fields take on the values of database fields or hold and report the results of calculations.
- The LET command creates an implicit temporary field. This means that VISION:Bridge determines the attributes of the field based upon the characteristics of the initial value assigned to the field and subsequent operations where the field is the result and not by any attributes that you specify.

Using SET with LET

You can use SET and LET together to first explicitly define a temporary field and then to manipulate that field in some way. See the section Temporary Field Examples, which shows a number of examples that use both SET and LET.

SET creates temporary fields with explicitly defined attributes. That is, you specify the length, precision, field type, and initial value of the temporary field you want to create. This field is initialized only once, at the beginning of your query.

- You instruct VISION:Bridge to set up a temporary field by specifying a field name and the attributes you want the field to have.
- You can reference this temporary field or use it in other VISION:Bridge statements.
- The temporary field is only valid for the query in which it is defined.

Once you initialize a field with a SET command, you can change the values of that field with a LET. The precision will be maintained. However, you cannot use SET to redefine or respecify a temporary field previously created with another SET or LET.

With SET, you can define a temporary field as being a character, zoned decimal, packed decimal, or fixed binary field. Character type fields can be up to 255 characters long.

The maximum length and precision of numeric temporary fields are summarized in Figure 6-17.

Field Type	Length (in bytes)	Maximum Number of Digits (content)	Maximum Decimal Places	Maximum Value (see note below)
ZONED	1-15	Equal to length of the field	Equal to the length value up to a maximum of 9	All digits equal to 9
PACKED	15 (8 for Immediate Response)	(2 * length)-1	Equal to the maximum number of digits up to a maximum of 9	All digits equal to 9
FIXED	1 2 3 4	3 5 7 10	3 5 7 9	127 32,767 8,388,607 2,147,483,647

Figure 6-17 Value Limits for Numeric Type Temporary Fields in SET Statements

Note: The maximum initial value is 15 digits, including one decimal digit, or 8 digits without a decimal digit, but not greater than maximum content.

SET Command

SET FIELD name TYPE field type,
LENGTH integer DECIMALS integer,
VALUE initial value

The SET command indicates the assignment of a temporary field and specifies the attributes you want.

FIELD name Specifies the name of the temporary field being created. The name can be from 1 to 8 characters with the first character being alphabetic and the remainder alphanumeric. A field name is required.

TYPE field type Specifies the type of field being created.

CHAR — character field.
ZONED — zoned decimal field.
PACKED — packed decimal field.
FIXED — fixed-point binary field.

TYPE CHAR is the default if no TYPE is specified.

LENGTH integer

Specifies the number of bytes for the field being created. Each field type has an acceptable range for length and a default length.

Type	Length (bytes)	Default (bytes)	
CHAR	1-255	16	
ZONED	1-15	15	
PACKED	1-15	8	(1-8 for Immediate Response)
FIXED	1-4	4	

DECIMALS integer

Specifies the number of decimal places assigned to a numeric temporary field.

- Enter a value from 1 to the maximum specified in Figure 6-17.
- If you do not specify a decimal value, zero decimal places are assumed.

VALUE initial value

Assigns an initial value to the temporary field.

- Although field TYPE CHAR can be up to 255 characters in length, the initial value must be 15 characters or less and must be enclosed in single quotation marks.
- If you do not specify a value, the default initial value is set to blanks.

Initial value of fields of TYPE ZONED, PACKED, or FIXED must be unsigned with or without a decimal point.

Decimal places exceeding the DECIMALS specification are truncated.

- If you do not specify DECIMALS, regardless of whether VALUE contains a decimal point, the maximum number of digits is limited to 8.
- If you do not specify DECIMALS, the default initial value is set to zero.

Using SET

You can use the SET command at any point outside of a REPORT group to define (and initialize) a temporary field.

The following are examples of valid and invalid SET statements.

Valid SET Statements

Example 1. Character Field

```
SET FIELD ABC TYPE CHAR LENGTH 16 VALUE 'TODAY'
```

This example creates a character field 16 bytes long with an initial value of TODAY. Enclose the initial value for a character field in single quotation marks.

Example 2. Packed Decimal Field

```
SET XYZ TYPE PACKED LENGTH 4 DECIMALS 2 VALUE 32.15
```

Field XYZ is a packed decimal field 4 bytes long with 2 decimal places. As seen in Figure 6-17 a 4-byte packed decimal field may be up to 7 digits long. The initial value of XYZ is set to 32.15. This value remains in effect for the entire query unless you explicitly change it with a LET.

Example 3. Fixed Point Binary Field

```
SET X1 TYPE FIXED DECIMALS 2 VALUE 1234.567
```

Field X1 is defined as a fixed point binary field. Since you did not specify a length value, VISION:Bridge assigns the default value for fixed point fields at 4. Figure 6-17 shows that a 4-byte fixed point binary field can be up to 10 digits in length. However, since the decimals specification value is 2, the initial value of 1234.567 will be truncated to 1234.56.

Example 4. Packed Decimal Field

```
SET Y1 TYPE PACKED
```

This statement creates a packed decimal field with no initial value. Since you did not provide additional specifications, VISION:Bridge uses the default value of 8 bytes for the length and 0 for the number of decimal places.

Example 5. Truncated Character Field

```
SET TEMP TYPE CHAR LENGTH 2 VALUE '123'
```

This statement is valid, but the initial value of the field is truncated to 12, because the length of the field is set to two.

Invalid SET Statements

The VISION:Bridge syntax checker issues diagnostic messages for these invalid SET statements:

Example 1. Number of decimal places exceeded

```
SET TEMP1 TYPE PACKED LENGTH 3 DECIMALS 6
```

The maximum number of decimal places for a 3-byte packed field is 5.

Example 2. Initial value too large

```
SET TEMP2 TYPE FIXED LENGTH 2 VALUE 40000
```

The initial value for the 2-byte fixed point field TEMP2 exceeds the maximum value of 32,767.

Example 3. Value too large for zoned field

```
SET TEMP3 TYPE ZONED LENGTH 6 DECIMALS 2 VALUE 31426
```

In this example, the initial value of TEMP3 is too large. With zoned fields, the maximum number of digits is equal to the length of the field. Two digits have been accounted for by the DECIMALS specification, which leaves only four digits for the integer portion of the field.

Example 4. Value too large for packed field

```
SET TEMP4 TYPE PACKED LENGTH 4 DECIMALS 3 VALUE 21536.20
```

Similar to the above example, the initial value of TEMP4 is too large to fit in the defined space. With packed decimal fields, the maximum number of digits is twice the length of the field minus one digit for the sign of the value; in this case seven. Having specified three decimal places leaves only four digits for the integer portion of the field value.

Example 5. Default type with value too large

```
SET AA VALUE 'ABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

Since you did not specify TYPE, VISION:Bridge assigns the default value of CHAR with the default length of 16. The initial value of field AA exceeds the maximum default length. Although you can explicitly define character type fields to be up to 255 characters in length, the maximum initial value is 15 characters. Thus, had you explicitly initialized field AA to a length greater than the default value of 16, the initial value specified would still exceed the VISION:Bridge limit of 15 characters.

Using LET

Use LET to assign a value to a field.

- The field can be a field from the database, or a temporary field created by this LET, or an earlier LET or SET.
- The value assigned can be a constant, the contents of another field, or the results of an arithmetic expression.
- If the field is not already defined, you instruct VISION:Bridge to set up a temporary field by using the name you selected for it on the LET statement.

Using SET, Then LET

To accumulate values in a temporary field as with a counter, first create the temporary field with a SET and then use LET to update it.

Using LET in Batched Queries

Be careful when using LET in batched queries. VISION:Bridge does not allow queries that change a master file field value or a flag field value with a LET to be batched, because that assigned value is propagated for every other query in that batch. Therefore, if you are batching queries, it is good practice to first assign the master file field to a temporary field and then perform manipulations, calculations, and so on, using the temporary field rather than the actual database field.

LET Command

LET FIELD name = value IF logical expression

The LET command assigns a value to a field. If the field has not been previously defined, a temporary field is created.

FIELD name Specify the name of a new or existing field that is assigned a value.

If the field name is not in the database or was not created by an earlier LET or SET, a temporary field with this name is created.

- It has the attributes of the field or value to the right of the equal sign.
- The attributes can change depending on subsequent usage of the temporary field.

The name of a temporary field can have from one to eight characters; the first must be alphabetic and the remainder can be alphanumeric.

The use of the FIELD keyword is optional.

An existing field from the database can be assigned a new value. This value is valid only during this query. The value stored in the database under that field name remains unchanged.

= value Value to be assigned to the field on the left. This value can have the contents of any existing field, a constant, or the results of an arithmetic calculation. You must enter the equal sign (=).

When using grouping characters ($1,000) such as a comma or $, or assigning a field to be a character constant (such as `LET FIELDA = 'ABC'`), you must enclose the value in single quotation marks:

```
LET FIELDA = '1,000.99'.
```

IF logical expression Specifies the conditions under which the LET executes.

LET Example

```
LET FIELD SALARY = SALYTD
```

This LET instructs VISION:Bridge to assign to a field named SALARY the same value that SALYTD has at this point. Each time this LET executes, the current value of SALYTD is assigned to SALARY.

If you have not already defined the field named SALARY, VISION:Bridge sets up a temporary field named SALARY in addition to the assignment function. This temporary field is created as a numeric field of the same length as SALYTD.

```
LET SALARY = SALYTD
REPORT EMPNAME SALARY
```

In this example, the REPORT statement displays EMPNAME and SALARY for each employee. If SALARY is a temporary field, it has no preassigned column heading. You can assign one using the ITEM statement, as illustrated in Figure 6-18.

Temporary Field Examples

Figure 6-18 compares the LET and SET doing the same task. In this example, a temporary field is created to report a 10 percent discount on customer invoice amounts (INAMT * .90). This is done twice, once with a LET and once with a SET followed by a LET.

Query

```
QUERY CUSTOMER
LET LETAMT = INAMT * .90
SET SETAMT TYPE PACKED LENGTH 4 DECIMALS 2
LET SETAMT = INAMT * .90
 REPORT CUSTNO INVNO INAMT LETAMT SETAMT
  ITEM LETAMT HEADING 'TEMP FIELD' 'LET STATEMENT'
  ITEM SETAMT HEADING 'TEMP FIELD' 'SET STATEMENT'
  GROUP BY CUSTNO INVNO
  TITLE 'SET VERSUS LET'
  TITLE 'TWO TYPES OF TEMP FIELDS'
 END REPORT
END QUERY
```

Figure 6-18 Defining Temporary Fields (Page 1 of 2)

The first LET statement (LET LETAMT = INAMT * .90) creates an implicit temporary field.

The attributes of an implicit temporary field are determined by:

- The attributes of the fields used to assign a value to it.
- The mathematical operations, if any, that determine the value of the field. Here, LETAMT has attributes based on both the attributes of INAMT (4 bytes packed with 2 decimal places) and the operation performed on INAMT to create the value of LETAMT.

Since the operation is multiplication, VISION:Bridge adds the number of decimal places of both multipliers, 2 for INAMT and 2 for .90, and creates LETAMT with 4 decimal places. When multiplying two numbers, the number of decimal places can never be more than the sum of the decimal places of the two numbers being multiplied.

The SET statement (SET SETAMT TYPE PACKED LENGTH 4 DECIMALS 2) creates an explicitly defined temporary field. SETAMT is defined as a packed field, 4 bytes long with 2 decimal places.

The second LET statement (LET SETAMT = INAMT * .90) assigns the value of SETAMT as the value of INAMT * .90.

Output

```
NOV 09, 2001                    SET VERSUS LET                    PAGE 1
                              TWO TYPES OF TEMP FIELDS
-------------------------------------------------------------------------
CUSTOMER  INVOICE    INVOICE      TEMP FIELD       TEMP FIELD
 NUMBER   NUMBER     AMOUNT     LET STATEMENT   SET STATEMENT
-------------------------------------------------------------------------
00001     I1648       270.00         243.0000        243.00
          I1627        36.00          32.4000         32.40
00013     I6781       105.00          94.5000         94.50
          I1728       210.00         189.0000        189.00
          I3334       140.00         126.0000        126.00
00028     I7781        22.00          19.8000         19.80
00043     I9162        84.00          75.6000         75.60
00048     I1545       305.00         274.5000        274.50
          I8137       260.00         234.0000        234.00
          I8206        18.00          16.2000         16.20

00089        -            -                *               *
                          -                *               *

          I1473        75.00          67.5000         67.50
00115     I7630       180.00         162.0000        162.00
          I6932        56.00          50.4000         50.40
```

Figure 6-19 Defining Temporary Fields (Page 2 of 2)

Notice the output for customer 00089. The dashes and asterisks are due to the way VISION:Bridge processes hierarchical data structures.

To perform the calculation on INAMT, VISION:Bridge loops through the database hierarchy until it gets down to an occurrence of INAMT. Since invoices are subordinate to orders, VISION:Bridge processes occurrences of the ORDER segment to get to the SHIPINV segment.

VISION:Bridge does not know if an occurrence of INAMT exists until it gets there. As Figure 6-20 shows customer 00089 has three orders in the database, but only one of those orders has an invoice.

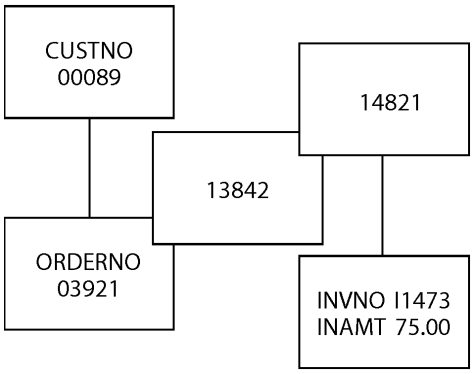


Figure 6-20 A Customer Database Record

VISION:Bridge must process each occurrence of the ORDER segment to determine if an occurrence of the SHIPINV segment exists. For the first and second occurrence of ORDERNO, there is no INAMT, so VISION:Bridge prints the dash. The dash (-) indicates that there are no occurrences of a field.

The LET statements then execute, but since the INAMT field does not exist, the result is invalid. The asterisk (*) prints whenever the result of a computation is invalid.

A third type of invalid field (not shown) is a plus (+). It displays when the length of the field to be reported is larger than the output edit area. For more information on invalid fields, see the section Invalid Fields.

Partial Field Processing

Sometimes you will want to do data selection based on just a part of a data field value, or if you want to report just part of a data field value. This can be done on character type fields with partial field processing.

Partial field processing lets you specify the exact parts of a character field value that you want VISION:Bridge to access.

For example, assume you want to select all the customers from a certain telephone area code. To do this, look at the first three positions of the phone number field, and select records based on the value of the first three digits only.

To use partial fielding, you specify:

- The name of the field.
- Where you want VISION:Bridge to start looking in the field value.
- How many characters to use.

Partial Field (PF) Function

PF (field name, starting position, length)

Note: You can use only one partial field expression in a relational expression.

The PF function specifies part of a character field. You can use partial fielding in place of field names in logical expressions, LET statements, REPORT statements, and DISPLAY statements.

field name Specifies the character field. You will use part of this field.

starting position Specifies the starting position of the partial field in the character field. Use an integer constant from 1 to 99.

length Specifies the number of characters to use.

- Use an integer constant from 1 to 99.
- If you omit the length, all characters from the start position up to and including the last character are used.

Using the Partial Field Function

Figure 6-21 shows partial fielding used in a relational expression on an IF statement to select orders that have item names beginning with M4.

With the expression IF PF(ITEMNAME,1,2) EQ 'M4', VISION:Bridge inspects just the first two characters of every occurrence of ITEMNAME. If the first two characters are M4, the occurrence of ITEMNAME is selected for output.

Query

```
QUERY CUSTOMER
 IF PF(ITEMNAME,1,2) EQ 'M4'
  REPORT CUSTNO ORDERNO ITEMORD ITEMNAME
   GROUP BY CUSTNO ORDERNO
   TITLE 'PARTIAL FIELD FUNCTION'
   TITLE 'ORDERS W/ M4 ITEMS ON THEM'
  END REPORT
 END IF
END QUERY
```

Figure 6-21 Selection Based on Partial Fielding (Query)

Output

```
NOV 09, 2001              PARTIAL FIELD FUNCTION            PAGE 1
                         ORDERS W/ M4 ITEMS ON THEM
-----------------------------------------------------------------------
CUSTOMER  ORDER    ITEMS          ITEM
 NUMBER   NUMBER  ORDERED         NAME
-----------------------------------------------------------------------
00001    07321   041       M4 PR FORM
                 042       M4 OUTPUT FORM
                 046       M4 BASIC REQ. FORM
                 115       M4 RUN CNTL. FORM
         08432   673       M4 DIAGNOSTIC MSG.

00013    03619   045       M4 TABLE DEF FORM
                 113       M4 FILE DEF FORM
                 114       M4 TRAN DEF FORM
         09541   017       M4 REF MANUAL
                 019       M4 OPS GUIDE-OS

00028    11211   021       M4 OPS GU IDE-DOS

00043    23468   017       M4 REF MANUAL

00048    06781   046       M4 BASIC REQ FORM
                 247       M4-405 CLASS
                 673       M4 DIAGNOSTIC MSG
                 674       M4-604 CLASS
```

Figure 6-22 Selection Based on Partial Fielding (Report)

You can report the results of partial fielding as shown by the query in Figure 6-23. In this example, we are reporting the month that each order is due (ORDDUDAT). The third and fourth characters of the order due date represents the month of the year. Do this by specifying PF (ORDDUDAT,3,2) on the REPORT statement.

Query

```
QUERY CUSTOMER
 REPORT CUSTNO ORDERNO PF(ORDDUDAT,3,2)
  GROUP BY CUSTNO ORDERNO
   TITLE 'REPORTING A PARTIAL FIELD'
 END REPORT
LET MONTHDUE = PF(ORDDUDAT,3,2)
 IF MONTHDUE LE '12'
  REPORT MONTHDUE CUSTNO ORDERNO
   ORDER BY MONTHDUE
   GROUP BY MONTHDUE CUSTNO
   TITLE 'REPORTING A PARTIAL FIELD -- PART 2'
   TITLE 'USING PARTIAL FIELDING AND TEMP FIELDS'
  END REPORT
 ELSE
  REPORT MONTHDUE CUSTNO ORDERNO
   ORDER BY MONTHDUE
   GROUP BY MONTHDUE CUSTNO
   TITLE 'REPORTING PARTIAL FIELDS -- PART3'
   TITLE 'THE EXCEPTION REPORT'
  END REPORT
END QUERY
```

Figure 6-23 Using Partial Fielding and Temporary Fields

The second REPORT statement in Figure 6-23 and the corresponding second report in Figure 6-24 shows how to use partial fielding with LET statements to create new fields.

The LET statement (LET MONTHDUE = PF(ORDDUDAT,3,2)) creates a temporary field called MONTHDUE and assigns to it the value resulting from partial fielding the ORDDUDAT field.

The MONTHDUE field contains values from 01 to 12, corresponding to the twelve months of the year. The MONTHDUE field is reported and the GROUP statement orders the report by the value of MONTHDUE and CUSTNO.

- If the value of MONTHDUE is greater than 12, we would not want that printed on this report, but we would certainly want to know about it, since it would be an error in the data. This is one reason why you might use an IF/ELSE group.
- If the partial field result of MONTHDUE is between 01 and 12, the second report group executes.
- If the value is anything else, the value prints when the third report group executes. As the dashes in third report in Figure 6-24 indicate, there are no values of MONTHDUE greater than 12. The dash (-) indicates that there are no occurrences of a field.

Output

```
NOV 09, 2001              REPORTING PARTIAL FIELD              PAGE 1
------------------------------------------------------------------------
CUSTOMER  ORDER
 NUMBER   NUMBER
------------------------------------------------------------------------
00001    07321   03
         08432   11
00013    01142   07
         03619   09
         09541   11
00028    11211   04
00043    23468   08
00048    06573   02
         06781   08
00089    03921   05
         13842   07
         14821   09
00115    00284   07
         01391   06

NOV 09, 2001     REPORTING A PARTIAL FIELD -- PART 2           PAGE 1
               USING PARTIAL FIEL DING AND TEMP FIELDS
------------------------------------------------------------------------
MONTHDUE  CUSTOMER  ORDER
           NUMBER   NUMBER
------------------------------------------------------------------------
02       00048    06573
03       00001    07321
04       00028    11211
05       00089    03921
06       00115    01391
07       00013    01142
         00089    13842
         00115    00284
08       00043    23468
         00048    06781
09       00013    03619
         00089    14821
11       00001    08432
         00013    09541

NOV 09, 2001        REPORTING PARTIAL FIELDS -- PART 3        PAGE 1
                          THE EXCEPTION REPORT
------------------------------------------------------------------------
MONTHDUE  CUSTOMER  ORDER
           NUMBER   NUMBER
------------------------------------------------------------------------
   -          -       -
```

Figure 6-24 Using Partial Fielding and Temporary Fields

Automatic Table Lookup

The automatic table lookup feature is not available with Immediate Response.

Lookup tables contain information that correspond to database field values in some way.

- They save space in databases.
- They store a short field value in the database, but when you reference that short field, VISION:Bridge goes to the table and returns a different, usually longer field value, that corresponds to the field value in the database.

In the CUSTOMER database, for example, a field called INSTCODE represents the installation location code. It is one character long. Each value from 1 to 9 represents a different region of the country.

When you define the CUSTOMER database, you create a table that links each value of INSTCODE (the argument) to the value of INSTLOC (the result), a region in the country. These two fields are linked together in a table that automatically translates the one character code to the appropriate region.

- Figure 6-25 shows this sample customer table.
- Figure 6-26 shows a report that references both INSTCODE and INSTLOC. As you can see, the values reported from INSTLOC are more useful for the report and the automatic table lookup saves space in the database.

In the table, suppose you have two columns and nine rows.

- The first column relates to the database INSTCODE field (the argument). It is a one-character field with a value from 1 to 9.
- The second column relates the database field INSTLOC (the result). It is a 16-character field and contains the actual region location name.

When your query requests the INSTCODE and INSTLOC fields, an automatic lookup occurs and the query retrieves the requested data.

```
1    NORTH REGION
2    SOUTH REGION
3    EAST REGION
4    WEST REGION
5    CENTRAL REGION
6    NORTHEAST REGION
7    SOUTHEAST REGION
8    NORTHWEST REGION
9    SOUTHWEST REGION
```

Figure 6-25 Sample Customer Table

For more information on automatic table lookup, consult your system administrator.

Query

```
QUERY CUSTOMER
REPORT CUSTNO INSTNO INSTCODE INSTLOC,
IF CUSTNO LT 00030
GROUP BY CUSTNO
END REPORT
END QUERY
```

Output

```
NOV 11, 2001                          PAGE 1
-----------------------------------------------------------------
CUSTOMER  INSTNO  INSTCODE        INSTLOC
 NUMBER
-----------------------------------------------------------------
00001     0106      8       NORTHWEST REGION
          0347      8       NORTHWEST REGION
00013     0329      9       SOUTHWEST REGION
          5186      8       NORTHWEST REGION
00028     1601      5       CENTRAL REGION
          2702      9       SOUTHWEST REGION
```

Figure 6-26 Using Automatic Table Lookup

ANY and ALL Keywords

Use ANY and ALL in conditional phrases to select or report data where the selection criteria are based on lower level segments. Their use is very simple and their meanings are quite natural.

Using the ANY Keyword

ANY says that the related statements execute for all occurrences of the specified fields if the conditions in the logical expression are satisfied at least once in the current record. For example, use the ANY keyword in a conditional phrase such as:

```
REPORT CUSTNO ORDERNO IF ANY ORDCMPLT EQ 'Y'
```

This ANY keyword tells VISION:Bridge to execute the REPORT statement if the logical expression is true at least once. The report lists the customer numbers and order numbers for those customers where any of the orders for that customer is complete. Now, any given customer could have hundreds of orders. If any of those orders is complete, VISION:Bridge reports the customer number plus all the order numbers for that customer.

Using the ALL Keyword

ALL says that the related statements execute for every occurrence only if the specific conditions are satisfied each time the logical expression executes.

```
REPORT CUSTNO ORDERNO IF ALL ORDCMPLT EQ 'Y'
```

This query statement prints the customer number and order number only if every occurrence of ORDCMPLT for every order for a given customer is equal to Y. This is because the REPORT statement only executes for this record, if the logical expression is true for every occurrence of ORDCMPLT.

ANY logical expression
ALL logical expression

ANY Indicates the related statements execute for all occurrences of the specified fields if the test condition is satisfied at least once for the current record.

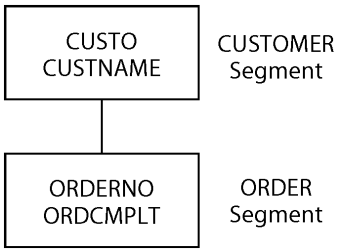
ANY implies multiple relational expressions connected by OR.

ALL Indicates the related statements execute for all occurrences only when the test conditions are satisfied each time the logical expression executes in the current record.

ALL implies multiple relational expressions connected by AND.

Using the ANY and ALL Keywords

The following samples clarify the use of ANY and ALL.



Look at the first two segments in the CUSTOMER database again.

The CUSTOMER segment is the root segment and the ORDER segment is a level 2 segment directly under the CUSTOMER segment. Therefore, it is possible to have many orders for each customer.

To see the customers (root segment) and orders (level 2 segment) where ORDCMPLT equals Y, use any of the following:

```
REPORT CUSTNO ORDERNO ORDCMPLT,
       IF ORDCMPLT EQ 'Y'
```

or,

```
REPORT CUSTNO ORDERNO ORDCMPLT,
       IF ANY ORDCMPLT EQ 'Y'
```

or,

```
REPORT CUSTNO ORDERNO ORDCMPLT,
       IF ALL ORDCMPLT EQ 'Y'
```

All three statements reference the same database fields in essentially the same way, but request different data because of the use of ANY and ALL.

Figure 6-27 shows the difference that the keywords ANY and ALL make.

Output

```
NOV 19, 2001              SIMPLE SELECTION
                      WITHOUT  THE ANY/ALL IDENTIFIER
-------------------------------------------------------------------
CUSTOMER  ORDER     ORDER
 NUMBER   NUMBER  COMPLETE?
-------------------------------------------------------------------
00001    08432       Y
00013    01142       Y
         03619       Y
00028    11211       Y
00043    23468       Y
00048    06573       Y
         06781       Y
00089    14821       Y
00115    00284       Y

NOV 19, 2001                  THE ANY IDENTIFIER
-------------------------------------------------------------------
CUSTOMER  ORDER     ORDER
 NUMBER   NUMBER  COMPLETE?
-------------------------------------------------------------------
00001    07321       N
         08432       Y
00013    01142       Y
         03619       Y
         09541       N
00028    11211       Y
00043    23468       Y
00048    06573       Y
         06781       Y
00089    03921       N
         13842       N
         14821       Y
00115    00284       Y
         01391       N

NOV 19, 2001                  THE ALL IDENTIFIER
-------------------------------------------------------------------
CUSTOMER  ORDER     ORDER
 NUMBER   NUMBER  COMPLETE?
-------------------------------------------------------------------
00028    11211       Y
00043    23468       Y
00048    06573       Y
         06781       Y
```

Figure 6-27 The Difference Between ANY and ALL Keywords

The first REPORT statement uses neither ANY nor ALL.

```
REPORT CUSTNO ORDERNO ORDCMPLT,
       IF ORDCMPLT EQ 'Y'
```

The corresponding first report shows only the customers with at least some complete orders are selected, and then only those orders where ORDCMPLT EQ 'Y' are selected for printing.

The second REPORT statement specifies ANY ORDCMPLT EQ 'Y'.

```
REPORT CUSTNO ORDERNO ORDCMPLT,
       IF ANY ORDCMPLT EQ 'Y'
```

The corresponding second report shows many of the orders show a value of N for ORDCMPLT yet, they too, print. Remember, ANY makes all occurrences available for reporting if any occurrence meets the criteria you specify. Therefore, since at least one occurrence of ORDCMPLT meets the criteria, the REPORT statement executes for every occurrence of the ORDER segment.

The third REPORT statement specifies ALL ORDCMPLT EQ 'Y'.

```
REPORT CUSTNO ORDERNO ORDCMPLT,
       IF ALL ORDCMPLT EQ 'Y'
```

The corresponding third report shows only customers available for reporting are those customers where every occurrence of ORDCMPLT is Y.

If you use ANY and ALL, all occurrences under the lowest common parent of segments mentioned in the logical expression are tested.

HAVE and HAS Keywords

HAVE and HAS are synonymous keywords to specify the range of the ANY and ALL operators. They are interchangeable.

Combined with ANY and ALL, HAVE and HAS increase the power of the VISION:Bridge language by letting you apply selection criteria at a specified level in the database. You specify the level.

VISION:Bridge “locks” on a particular segment and loops within that segment and down the path of the database. See Figure 6-28 and Figure 6-29 to contrast queries with and without the HAVE/HAS keywords.

HAVE logical expression
HAS logical expression

Specifies the range of the ANY and ALL operators. HAVE and HAS are synonymous. “Locks” on a database segment and applies selection criteria while looping within that segment.

Using the HAVE and HAS Keywords

For example, to report customers with orders that ship items in quantities greater than 10, you can use the following statement:

```
REPORT CUSTNO ORDERNO INVNO ITEMNO QTYSHIP,
       IF QTYSHIP GT 10
```

This produces a report like that shown in Figure 6-28. It lists the data for the fields specified, but only those occurrences where QTYSHIP is greater than 10. If there are other items on the order with quantities less than or equal to 10, they are not reported.

Output

```
NOV 19, 2001                SIMPLE SELECTION
-------------------------------------------------------------------------
CUSTOMER  ORDER    INVOICE    ITEMS     QUANTITY
 NUMBER   NUMBER   NUMBER    SHIPPED    SHIPPED
-------------------------------------------------------------------------
00001     07321    I1648     041              20
                   I1648     042              20
                   I1648     046              50

00013     03619    I1728     045              25
                   I1728     113              25
                   I1728     114              20

00048     06781    I8137     046              20

00115     00284    I7630     041              20
                   I7630     042              20
                   I7630     046              20
```

Figure 6-28 Simple Selection

Suppose, however, you want to see all the orders and items for those customers with any quantities shipped greater than 10. You want to loop through the data structure at the QTYSHIP level (the SHIPINV segment at level 4) while "locked" on the CUSTNO field in the root segment. Use the HAS keyword:

```
REPORT CUSTNO ORDERNO INVNO ITEMNO QTYSHIP,
       IF CUSTOMER HAS ANY QTYSHIP GT 10
```

This REPORT statement requests the fields for those customers where any occurrence of QTYSHIP is greater than 10. This produces the report in Figure 6-29.

Output

```
NOV 19, 2001          USING THE HAS AND ANY KEYWORDS
-------------------------------------------------------------------------
CUSTOMER  ORDER    INVOICE    ITEMS     QUANTITY
 NUMBER   NUMBER   NUMBER    SHIPPED    SHIPPED
-------------------------------------------------------------------------
00001     07321    I1648    041              20
                   I1648    042              20
                   I1648    046              50
          08432    I1627    673               6

00013     01142    I6781    803               5
                   I6781    857               5
                   I6781    858               5
          03619    I1728    045              25
                   I1728    113              25
                   I1728    114              20
          09541    I3334    017               5

00048     06573    I1545    G40              10
                   I1545    G43               5
          06781    I8137    046              20
                   I8137    247               1
                   I8137    674               1
                   I8206    673               3

00115     00284    I7630    041              20
                   I7630    042              20
                   I7630    046              20
          01391    I6932    017               2
```

Figure 6-29 Using the HAS and ANY Keywords

The language differs slightly from the typical IF statement, because you must specify the name of the segment (level in the database) where you want the summarization to occur. Here, you want locking or summarizing on the root segment, so use the format:

```
IF CUSTOMER HAS ANY...
```

Customers, orders, and invoices are reported if any QTYSHIP is greater than 10. For customer 00048, only one occurrence of QTYSHIP is greater than 10, but all the orders are reported.

The next level of selection is to report only those customers and orders where the orders have occurrences of QTYSHIP greater than 10. In other words, you want to lock on each ORDER (rather than each customer) while looping through the occurrences of QTYSHIP. This is a REPORT statement to do just that:

```
REPORT CUSTNO ORDERNO INVNO ITEMSHIP QTYSHIP,
       IF N'ORDER' HAS ANY QTYSHIP GT 10
```

Note: An N precedes the segment ORDER which is enclosed in single quotation marks. This is because ORDER is a reserved keyword. Using a name in a quoted string preceded by an N tells VISION:Bridge to process the string as a dataname and not a command.

The statement, IF N'ORDER' HAS ANY... directs VISION:Bridge to lock on each ORDER and loop through the occurrences of QTYSHIP.

The results of this query display in Figure 6-30. There are fewer occurrences than in the previous example. Only those orders with at least one occurrence of QTYSHIP greater than 10 appear.

Output

```
NOV 19, 2001          USING THE HAS AND ANY KEYWORDS
                             ON THE ORDER SEGMENT
-------------------------------------------------------------------
CUSTOMER  ORDER   INVOICE   ITEMS    QUANTITY
 NUMBER   NUMBER  NUMBER   SHIPPED   SHIPPED
-------------------------------------------------------------------
00001     07321   I1648    041             20
                  I1648    042             20
                  I1648    046             50

00013     03619   I1728    045             25
                  I1728    113             25
                  I1728    114             20

00048     06781   I8137    046             20
                  I8137    247              1
                  I8137    674              1
                  I8206    673              3
00115     00284   I7630    041             20
                  I7630    042             20
                  I7630    046             20
```

Figure 6-30 Using the HAS and ANY Keywords on a Lower Level Segment

Invalid Fields

When the system encounters a field with inappropriate data, it flags the field as invalid. This invalid property is propagated and other result fields, which are dependent upon the invalid fields, also are flagged as invalid. The field's contents, however, are not destroyed or changed in any way.

A field can become invalid for any of the following reasons:

- Division by zero was attempted.
- Field overflow (the result of a calculation is too large to fit in the field).
- A value is too large for the field to which it is being moved.
- The type of the data is incorrect for the operation.
- The data or field cannot be found.

A special character defines the invalid field, and prints when you attempt to display an invalid field. (The characters can vary from one installation to another.)

*	This field is invalid in the database or as a result of computation during query processing.
-	This field does not exist in the record.
+	This field is too large to fit in the output column provided, or data was found to be invalid at printing time.

Chapter

7

Advanced Topics

Database Structures Concepts

You use VISION:Bridge to examine and manipulate data. Data is stored in the computer in units of information called fields. Fields are pieces of information you want to access. Related fields form a record.

Records

A record is a set of related data that the computer treats as a single unit. The types of data stored in fields in an employee's payroll record, for example, include the employee's name, number, salary, department, job title, taxes paid, and so on.

Files

Records are stored together in files. A file is a collection of related records. For example, payroll data for all employees is stored in a payroll file. This file consists of payroll records for every employee. When you need payroll information, you tell the computer to search the payroll file.

Fields

Usually, you need specific data from a file or from a record. For example, employee names or order numbers. This data is organized into fields, the smallest unit of data that you can reference. Each field has a name, called the field name, and one or more values (see Figure 7-1).

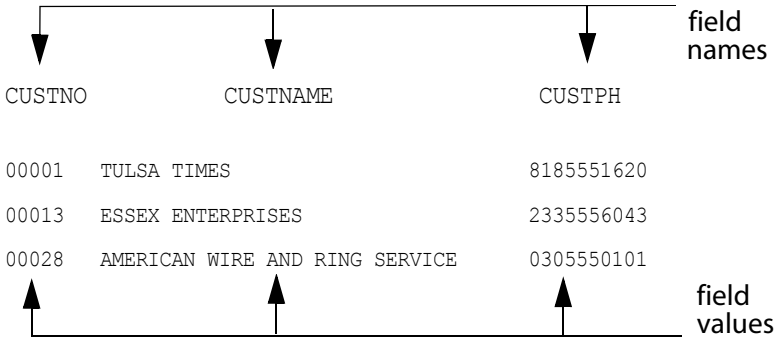


Figure 7-1 Fields Names and Fields Values

Flat Organization

There is a lot of repeated information in these fields. If one customer has three orders, for example, every field is repeated three times (see Figure 7-2). This data organization is called flat. It is a very simple way to organize data. Field values are the basic unit of information. Fields are organized into records, and records are organized into files.

Hierarchical Organization

Another, more economical way to organize data is in a hierarchical structure, much like the structure of a company.

In a company, there is a president, a few vice-presidents, managers, and many workers. Starting with the president, there is a one-to-many relationship — one president to many vice-presidents, one vice-president to many managers, and so on.

Hierarchical files model this one-to-many relationship, which is often found in data. One customer can have many orders and each order can have many items. In Figure 7-2, one customer (TULSA TIMES) has three orders (10004, 10011, and 10016). Figure 7-3 is a diagram of the data in Figure 7-2 turned into a hierarchy.

Segments

When you introduce hierarchic structure to a file, you introduce a new structural element called the segment. A segment is a set of related fields.

For example CUSTNO, CUSTNAME, and CUSTPH occur one time for each customer. They belong together and comprise a segment called CUSTOMER (see Figure 7-3).

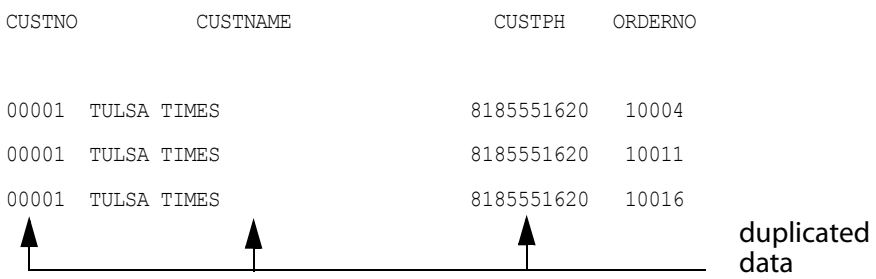


Figure 7-2 Duplicate Field Values

In every segment there is one or more fields that occur only once, fields with unique values that distinguish one segment from another.

A customer usually has only one name, one customer number, and one phone number. If this information was not unique for each customer, you would never know who owned the information you were looking for and you would never be able to find the data you need.

These fields occupy a very special position in the hierarchy. They sit right at the top. Together they form a segment called the root segment.

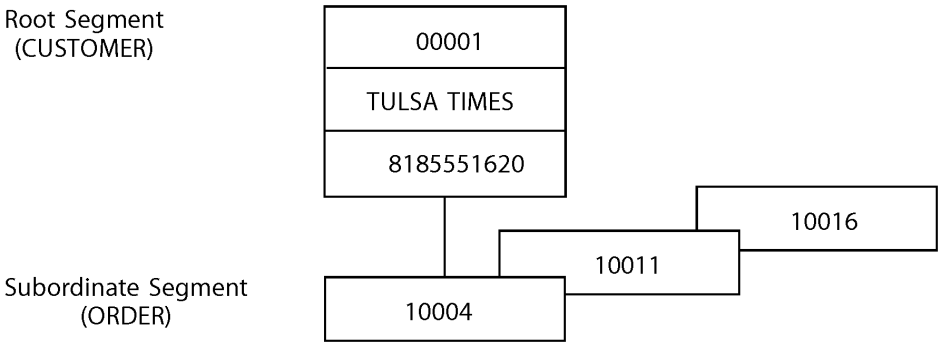


Figure 7-3 A Hierarchical Structure

The root segment in Figure 7-3 consists of the customer number (CUSTNO), the customer name (CUSTNAME), and the customer phone number (CUSTPH). All other segments are dependent on the root segment. The root segment is the parent and the segments underneath it are the children. Every segment, except the root segment is a child of another segment.

All the fields in a segment belong to a segment type. A unique number identifies each segment type (see Figure 7-5).

The CUSTOMER and ORDER segments in Figure 7-3 are segment types. Segment types give form to the data and define the hierarchy of the file.

There is an occurrence of the segment ORDER whenever there is data for the segment. In Figure 7-3, there are three segment occurrences of the segment type ORDER, because in this case there are three orders: 10004, 10011, and 10016. There can be one or many segment occurrences of a segment type, depending on the data stored in the database.

File vs. Database

Like a file, a database is a collection of related records. But, unlike a file, a database stores data in such a way that unnecessary redundancy is eliminated (the hierarchical structure described is one way of doing this). A database can also serve many users at once; this is often not true of files.

You need to know three things to access the data in a database:

- The name of the database.
- The names of the segments in the database.
- The names of the fields in the segments.

Figure 7-4 is a partial diagram of the database being used, the CUSTOMER database. There are six segments in the database (CUSTOMER, ORDER, SHIPINV, ITEMSHIP, ITEMORD, and INSTALL). Each segment has one or more fields. The ORDER segment, for example, has two fields, the order number (ORDERNO) and the order date (ORDRDATE).

You must know these names before you can access the data. (For a complete diagram of the CUSTOMER database, see Figure 7-7.)

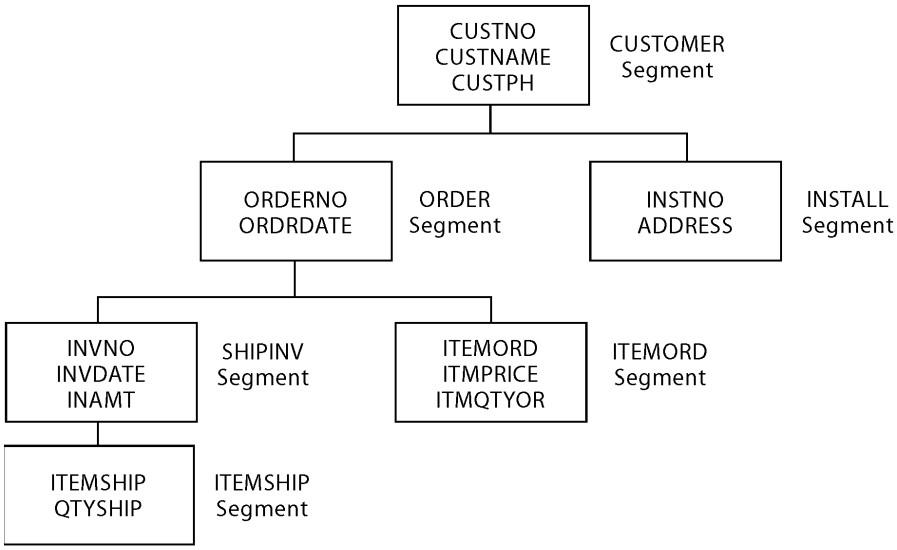


Figure 7-4 The CUSTOMER Database

File Organization

You can organize files of data in several ways. Two ways are the flat organization and the hierarchical organization. A flat file repeats data fields as many times as is necessary; a hierarchical field does not repeat data at all. However, you must know the hierarchy if you want to access the data.

Segment Numbers and Level Numbers

You define the structure of the hierarchy by numbering the levels and the segments in the hierarchy. As shown in Figure 7-5, you identify the segments of a hierarchical database by segment number and level number.

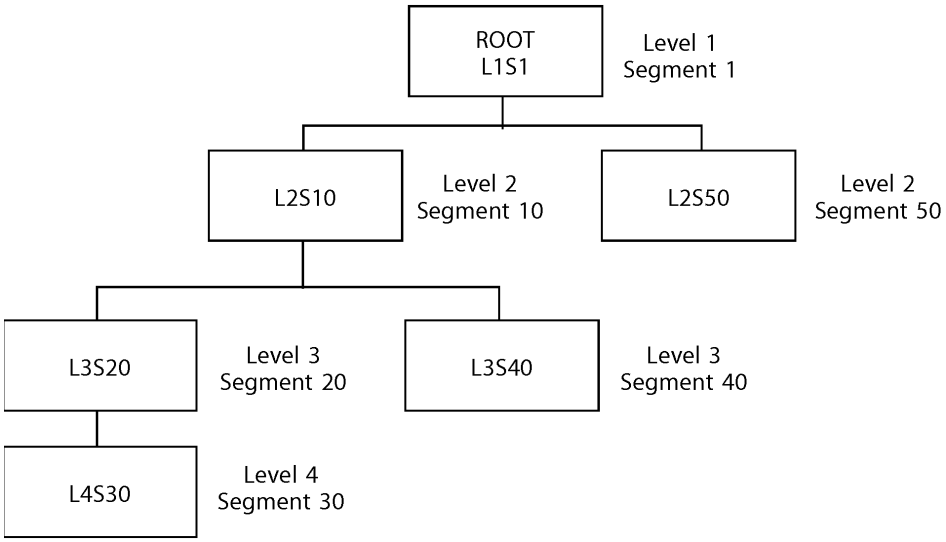


Figure 7-5 Organization of a Hierarchical File

In VISION:Bridge, there is a facility to establish relationships among databases or files that are not physically related. With this facility, known as the logical data view, the user reports data from several files as if they were working with a single file. For more information on logical data views, see your system administrator.

Defining Files

Data can be ambiguous to a computer. The same piece of data can represent a character string, a numeric value, or an instruction, depending on how the computer looks at it. Data can be organized into fields, records, and flat or hierarchical files. You need to describe the data organization so the computer can look at the data with the same structure.

File Definitions

You describe the data organization with file definitions. A file definition is a set of instructions that tells the computer how to organize the data in your files. File definitions do the following:

- Tell VISION:Bridge where data fields begin and end.
- Associate a field name with the location of the field.
- Tell VISION:Bridge what fields go together to form segments, and specify the structural relationships between segments.

A file definition tells VISION:Bridge all about the fields in a file. For instance, it tells the computer whether the field contains a number or a string of characters. It also describes where the field begins and ends.

For example, the file definition for a CUSTOMER file might state that you start with the first character in a record, count five characters, and name the field CUSTNO (for customer number). Then, count the next thirty characters and call this second field CUSTNAME (for customer name). The file definition continues to do this until every field is described for the CUSTOMER file.

Field Names

A field name can contain a maximum of eight characters — an alphabetic character followed by up to seven alphanumeric characters.

- Field names cannot duplicate system defined names (reserved keywords), such as commands, keywords, built-in function names, or operators.
- You can find the list of restricted keywords in the Appendix D, *Reference Summary*.
- VISION:Bridge supports Alternate Names for fields. The system administrator defines Alternate Names. They are up to 30 characters in length, and can be used in place of the 8-character field name. The database GLOSSARY will list both Primary and Alternate Names for a field when both are available.

Accommodating Pre-existing Field Names

To facilitate the use of pre-existing field names that do not meet these requirements, you enclose non-conforming names in single quotation marks and precede them with the letter N. Quotation marks within a name are represented by two consecutive quotation marks. Examples:

N ‘707-SAL’

N ‘FIELD’

N ‘O’’CLOCK’

Segment Numbering

File definitions also tell VISION:Bridge which fields go together into segments, and how the different segments relate to each other. You define the file structure with a numbering scheme that assigns a number to each segment and each level in the hierarchy.

You always number segments from top to bottom and left to right. The root segment is level 1, segment 1. Figure 7-5 shows a correct numbering scheme for a hierarchical database. Notice the pattern in which the segment numbers increase.

By using segment numbers and level numbers, it becomes easy for the computer to model the file structure. Although the data is stored in the computer as a sequential list, the translation provided by the file definition directs the computer to process the data as if it were stored in a hierarchical manner.

The file definition directs VISION:Bridge to go from this physical structure (how the data is actually stored):

```
00001TULSA TIMES818555162010004
      F95F91G4110011F95F99
```

to the logical structure shown in Figure 7-6 (how VISION:Bridge looks at the data for processing).

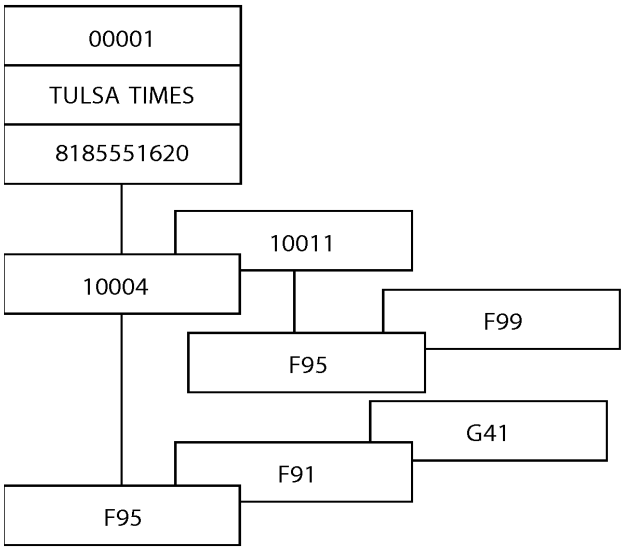


Figure 7-6 Logical Structure of the Data

Structured File Processing

VISION:Bridge automatically processes structured files by looping. The first reference to a lower level segment starts a loop on the occurrences of that segment type.

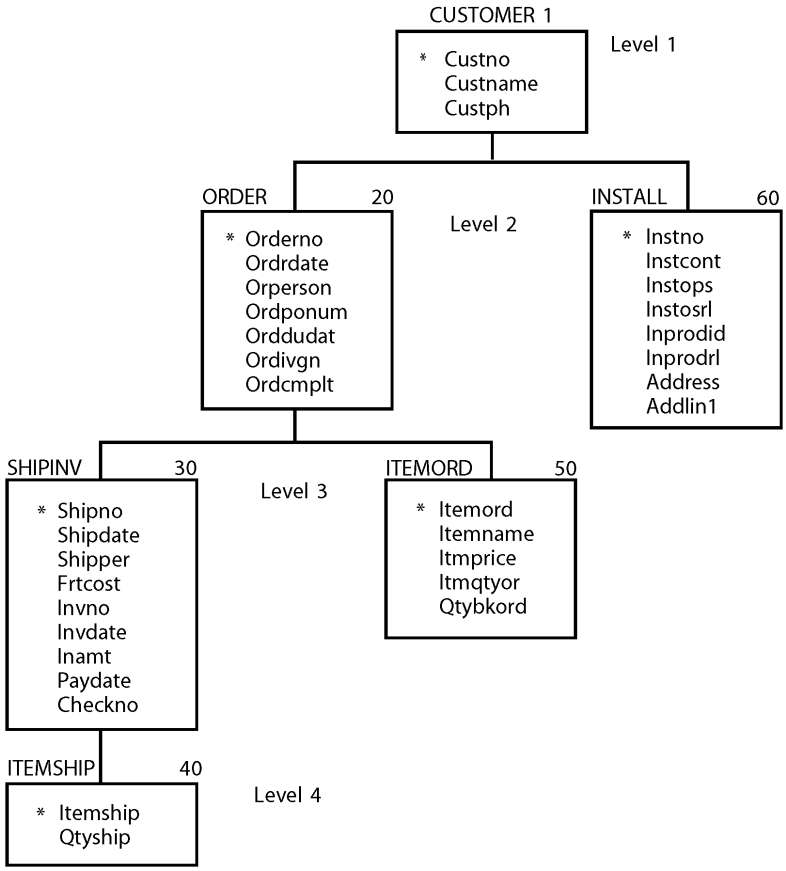


Figure 7-7 Diagram of the CUSTOMER Database

The diagram in Figure 7-7 shows the structure of the records on the CUSTOMER database.

- A root segment (CUSTOMER) and five subordinate segments are organized over four levels.
- Each segment past the root has one or more occurrences. The segment name and number are listed above the box for that segment.
- The fields that make up each segment are listed in the boxes.
- Asterisks indicate segment key fields.

The sample record in Figure 7-8 illustrates different types of looping. It is composed of data in the CUSTOMER, ORDER, ITEMORD, and INSTALL segments only.

- Customer A1 has orders B1, B2, and B3.
- Customer A1 also has installation D1.
- Order B1 has items C1 and C2.
- Order B2 has no items.
- Order B3 has item C3.

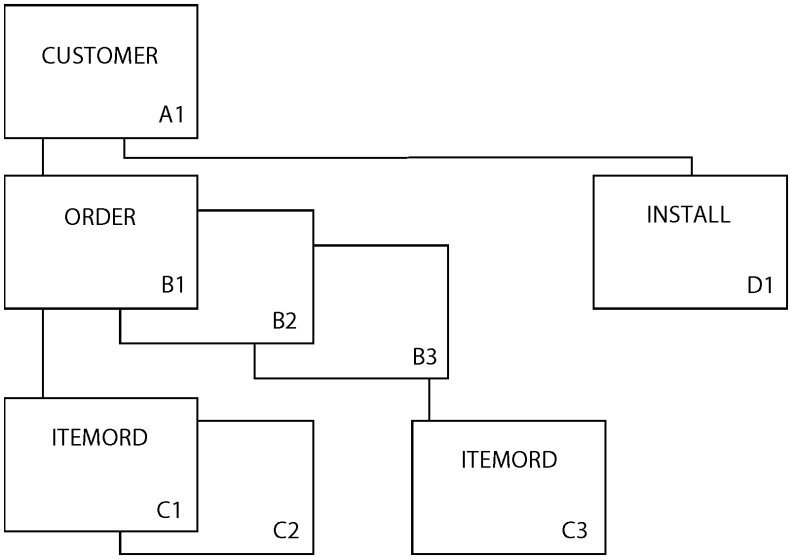


Figure 7-8 Customer A1 and Related Information

Basic Looping

You initiate basic loops by a reference to any one lower level segment type.

In Figure 7-9, reference to field ORDERNO starts a loop on the ORDER segment. See Figure 7-8 for the record structure.

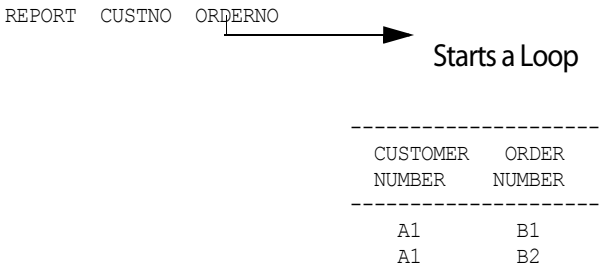


Figure 7-9 A Simple Loop

VISION:Bridge prints reports, repeating A1 for each occurrence of the ORDER segment. The only difference in an Immediate Response display is that there is no repeated data.

In VISION:Bridge, you control printing of repetitive data due to looping with the GROUP statement. See Chapter 4, *Using VISION:Bridge*.

Single-Path Nested Loop

A nested loop references more than one lower level segment, in the same path.

For example, in Figure 7-10, the reference to ORDERNO starts the first loop; the reference to ITEMORD starts a second loop nested within the first. The dash indicates that there was no occurrence of that segment.

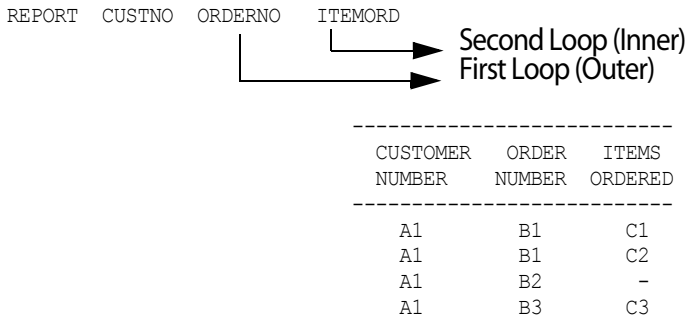


Figure 7-10 Nested Loop

Each inner loop completes before the next occurrence of the outer loop processes.

In Figure 7-11 the order of the fields starting the loops is reversed. This makes no difference in the data that is output to the report.

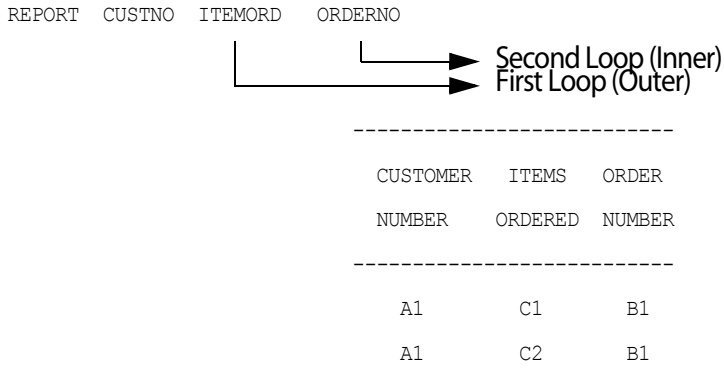


Figure 7-11 A Simple Nested Loop in Inverse Order

Parallel Looping

Figure 7-12 shows the data in two paths of a sample customer file record. At the second level of the customer file, there are two different segment types. The only thing the ORDER segment and the INSTALL segment have in common is that they both contain information about the customer.

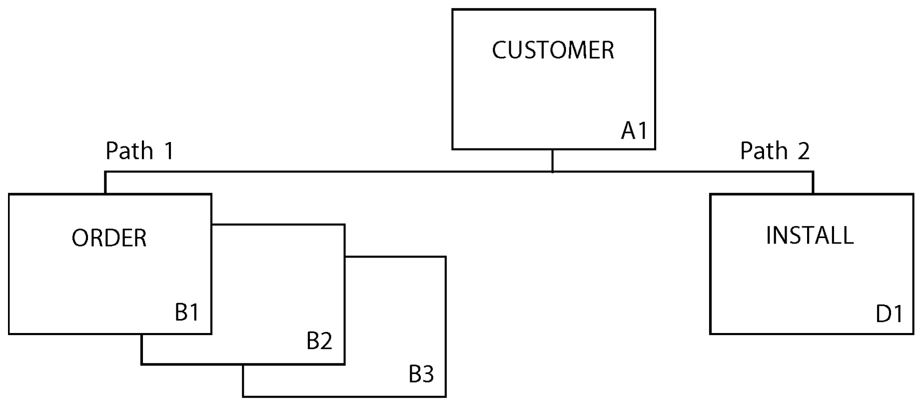


Figure 7-12 Two Paths in the Customer Record

If you make references to segments in different paths, a loop starts across both paths.

In Figure 7-13, a reference to the ORDER and INSTALL segments processes and prints the first occurrence of each, then the second occurrence of each, and so on until all occurrences of both are exhausted. If one segment is exhausted before the other, dashes print in the columns for the fields of that segment type.

Given the data shown in Figure 7-12, the reference to the ORDERNO and INSTNO fields produced the result shown in Figure 7-13. The occurrences of the INSTALL segment are exhausted before all occurrences of the ORDER segment are processed, printing dashes on the report.

```
REPORT  CUSTNO  ORDERNO   INSTNO
```

Starts a Parallel Loop

```
-------------------------------
  CUSTOMER  ORDER  INSTALLATION
  NUMBER    NUMBER    NUMBER
-------------------------------
    A1        B1        D1
    A1        B1         -
    A1        B2         -
```

Figure 7-13 A Parallel Loop

Looping Rules

- The first reference to a segment beyond the root starts a loop through the occurrences of that segment. The reference can be to the name of the segment or to a field within the segment. For an exception to this rule for the LET statement, see the section Looping with the LET Statement.
- When you reference different segments in the same path, you initiate nested looping.
- Reference to segments in different paths initiates parallel looping. The two segments loop through simultaneously, an occurrence of each per loop.
- You cannot start more than one loop on the same segment.
- Loops continue until you satisfy the selection criteria or until you process all data.

Looping Examples

This section contains different types of looping examples. In each case, the query is shown with the output. There is also a diagram to show the processing of an individual record.

In the examples, loops are started on the REPORT statement. However, you can start loops in any statement that references a field name or segment name of a lower level segment.

Immediate Response DISPLAY statements produce the same data. The only difference is that Immediate Response does not display repeated data. You can control this in VISION:Bridge with the GROUP command.

Basic Looping Examples

Query

```
QUERY CUSTOMER
REPORT CUSTNO ORDERNO
TITLE 'BASIC LOOP'
END REPORT
END QUERY
```

Output

```
NOV 23, 2001            BASIC LOOP            PAGE 1
-----------------------------------------------------

  CUSTOMER  ORDER
   NUMBER   NUMBER
-----------------------------------------------------

     00001    07321
     00001    08432
     00013    01142
     00013    03619
     00013    09541
     00028    11211
     00043    23468
     00048    06573
     00048    06781
```

Figure 7-14 A Basic Loop

Figure 7-14 shows an actual query and the output it produces. A loop was initiated on the ORDER segment by the reference to the ORDERNO field. You can eliminate the repetition of the customer number with the GROUP statement.

The processing of the shaded area is illustrated in Figure 7-15.

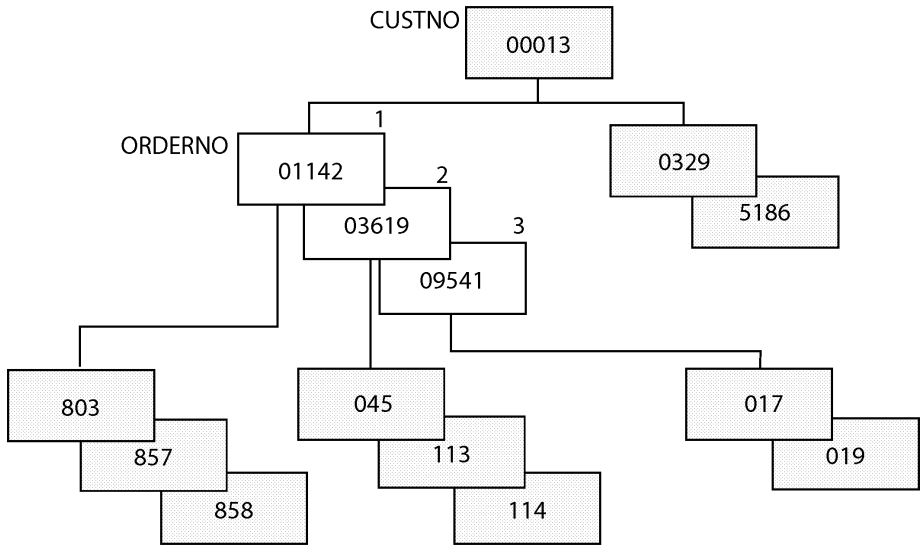


Figure 7-15 Basic Loop Processing

Figure 7-15 illustrates the sequence in which the occurrences of the ORDER segments for CUSTNO 00013 are processed. The result of the looping is seen in the non-shaded area, which corresponds to the shaded area of Figure 7-14. The numbers outside the boxes indicate the sequence in which the segment occurrences are processed.

Single-Path Nested Looping Examples

Query

```
QUERY CUSTOMER
REPORT CUSTNO ORDERNO ITEMORD
TITLE 'NESTED LOOP'
END REPORT
END QUERY
```

Output

```
NOV 23, 2001          NESTED LOOP           PAGE 1
-----------------------------------------------------
  CUSTOMER  ORDER    ITEMS
   NUMBER   NUMBER  ORDERED
-----------------------------------------------------
   00001    07321     041
   00001    07321     042
   00001    07321     046
   00001    07321     115
   00001    08432     673
   00013    01142     803
   00013    01142     857
   00013    01142     858
   00013    03619     045
   00013    03619     113
   00013    03619     114
   00013    09541     017
   00013    09541     019
   00028    11211     021
   00043    23468     017
   00048    06573     G40
   00048    06781     G43
   00048    06781     046
   00048    06781     247
```

Figure 7-16 A Single-Path Nested Loop

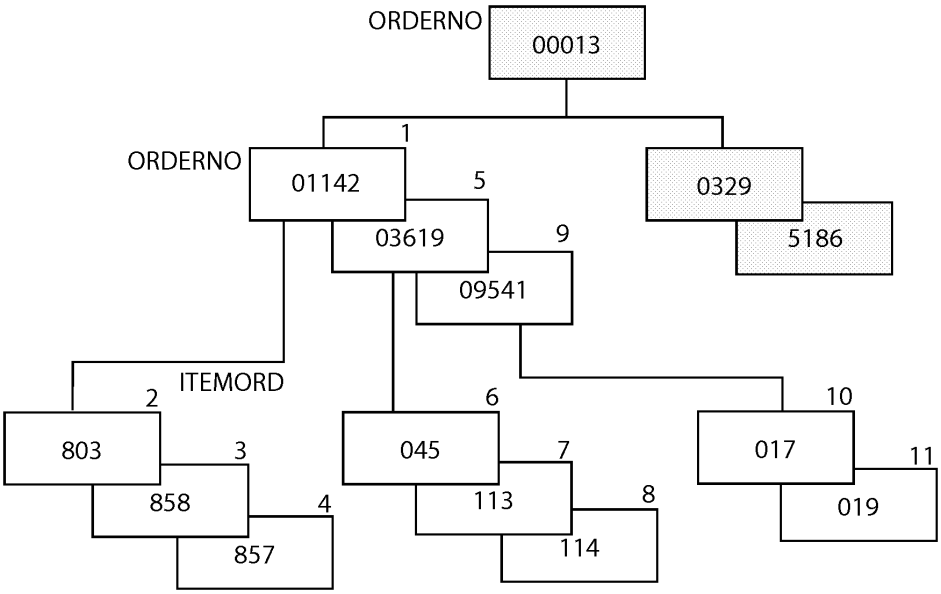


Figure 7-17 Single-Path Nested Loop Processing

Figure 7-17 illustrates the order in which you access segment occurrences for customer 00013, which is highlighted in the query in Figure 7-16. All ITEM segment occurrences print for an individual ORDER segment before processing of the next occurrence of the ORDER segment.

Multi-Path Parallel Looping Examples

Query

```
QUERY CUSTOMER
REPORT CUSTNO ORDERNO INSTNO
TITLE 'PARALLEL LOOP'
END REPORT
END QUERY
```

Output

```
NOV 23, 2001          PARALLEL LOOP          PAGE 1
-------------------------------------------------------
  CUSTOMER  ORDER    INSTALLATION
   NUMBER   NUMBER      NUMBER
-------------------------------------------------------
   00001     07321       0106
   00001     08432       0374
   00013     01142       0329
   00013     03619       5186
   00013     09541          -
   00028     11211       1601
   00028         -       2702
   00043     23468       4523
   00048     06573       3921
   00048     06781          -
   00089     03921       0154
   00089     13842          -
   00089     14821          -
   00115     00284       0188
   00115     01391          -
```

Figure 7-18 A Multi-Path Parallel Loop

A parallel loop starts in Figure 7-18 when you enter the ORDERNO and INSTNO fields on the REPORT statement. Since these fields reside in different level 2 segments, the segments loop through simultaneously, one occurrence of each per loop.

Figure 7-19 shows looping for the customer in the shaded area in Figure 7-18.

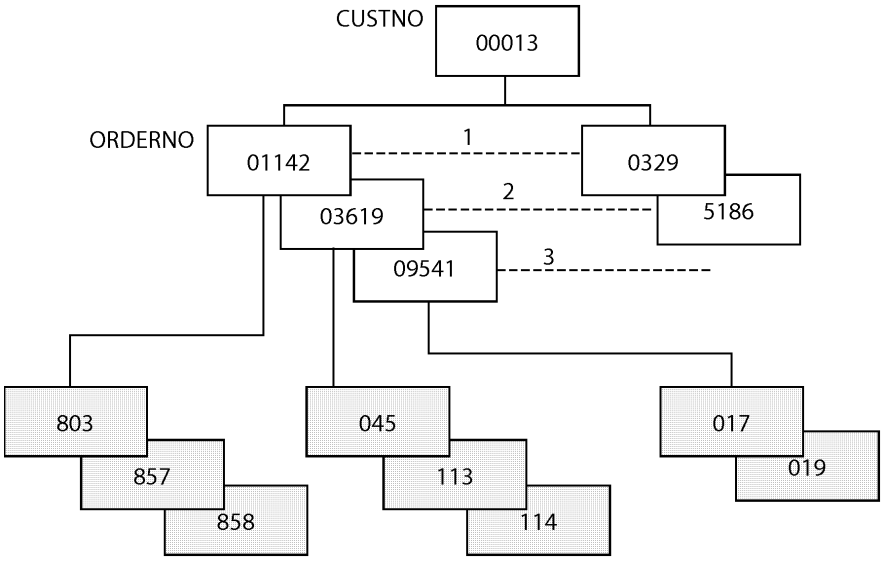


Figure 7-19 Multi-Path Parallel Loop Processing

Figure 7-19 shows the presentation of segment occurrences in each iteration of the parallel looping. There were more occurrences of the ORDER segment than of the INSTALL segment. As a result, (as seen in the boxed area of Figure 7-18) a dash appears on the report in place of the missing occurrence.

Starting and Ending Loops

Looping through data in structured files is described in the preceding section. This section focuses on the different methods of starting and stopping looping functions.

Looping means that a certain portion of the data in the database is available for processing, depending on what fields you reference in your query statements. Individual occurrences of the data fields you requested are returned to your remaining query statements one at a time, in the order in which they reside on the database.

In VISION:Bridge

In VISION:Bridge, you define the data you are requesting on one of the following statements: SELECT, REPORT, or IF.

- You tell VISION:Bridge to no longer restrict processing to the portion of the database you define, with these statements: END SELECT, END REPORT, or END IF.
- You can further limit the SELECT and REPORT statements to specific occurrences of the data fields they reference by adding an IF expression.

In Immediate Response

In Immediate Response, you define the data you are requesting on one of these statements: SELECT, DISPLAY, AVG, COUNT, CUM, or TOTAL.

- You tell Immediate Response to no longer restrict processing to the portion of the database you define with the END SELECT.
- You can further limit the SELECT and DISPLAY statements to specific occurrences of the data fields they reference by adding an IF expression.

For example, use SELECT CUSTNO ORDERNO when you want to process each of the order numbers (ORDERNO) and the associated customer numbers (CUSTNO) one by one with the remaining statements in your query, or until the system encounters an END SELECT.

If you only want to process those customer numbers and order numbers that have at least one item number of G41 associated with them, you can add an IF expression to the SELECT statement as follows:

```
SELECT CUSTNO ORDERNO IF ITEMNO EQ G41.
```

The IF expression processes first and, when true, the ORDERNO and CUSTNO data processes.

IF Expression vs. IF Statement

The IF expression, as part of a SELECT, REPORT, DISPLAY, or LET statement, functions in a different manner than the IF statement.

- The IF statement makes data available to the query when true.
- The IF expression, on another statement, only limits the data already available, on the true condition.

The LET can also start the looping process, but has a different purpose and has no corresponding END LET. You can find a discussion of LET in the section Looping with the LET Statement.

Figure 7-20 summarizes the commands you use to start and end loops. Figure 7-20 also illustrates the conditions under which these statements are valid.

Type of Command Starting Loop	Type of Command Ending Loop	Conditions
REPORT	END REPORT	always
SELECT	END SELECT	always
IF	END IF	always
LET	END QUERY	if LET is not in a SELECT or IF group
LET	END SELECT	if LET is in a SELECT group
LET	END IF	if LET is in an IF group

Figure 7-20 Starting and Ending Loops

Looping with the LET Statement

With VISION:Bridge, the general rule is that looping begins with the first reference to a lower level field. This is not the case with the LET statement. You initiate looping according to the matrix shown in Figure 7-21 with the LET statement.

	LET A	=B	IF C	Loop On
Case 1	Database Field	Any type	Any type	A
Case 2	Temp	Database Field	Any type	B
Case 3	Temp	Temp or Constant	Database Field	C

Figure 7-21 Looping with the LET Statement

In Figure 7-21, the LET statement starts the looping function with the first reference to a database field in a lower level segment. However, a subsequent reference to another database field that resides in another lower level segment does not start an additional loop of the occurrences of that segment. This is only true with the LET statement.

Optimizing Database Access For Immediate Response

Immediate Response automatically optimizes the search of database records for DL/I and VSAM (KSDS) files.

- Immediate Response performs optimization on DL/I files by constructing qualified SSAs (Segment Search Arguments).
- When Immediate Response processes VSAM files, it performs optimization by constructing ranges of record keys.

No segment is read twice. If segments are needed, memory optimized processing is used. (See the VISION:Inform *System Administrator Guide* for information on memory optimized processing.) The examples, in sections DL/I File Optimization (Immediate Response) and VSAM File Optimization (Immediate Response), are built around the CUSTOMER database.

DL/I File Optimization (Immediate Response)

To optimize DL/I files, Immediate Response constructs SSAs as follows:

All logical expressions, down to, but not including, a LET statement, are candidates for optimization. For example:

```
QUERY CUSTOMER
SELECT ORDERNO IF ITMQTYOR > 100
DISPLAY ITEMORD IF ITMPRICE > 25
```

shows that both `ITMQTYOR > 100` and `ITMPRICE > 25` are candidates for optimization.

In the following example:

```
QUERY CUSTOMER
SELECT ORDERNO IF IMTQTYOR > 100
LET ITMPRICE = ITMPRICE * .80 IF ITMQTYOR > 100
DISPLAY ITEMORD IF ITMPRICE > 25
```

only `ITMQTYOR > 100` is a candidate for optimization.

Immediate Response generates qualified SSAs using segment key fields and search fields.

Constructing Logical Expressions

Construct the logical expressions as a group of conditions connected by AND/OR (other connectors like NOT, HAS, ALL, and ANY are not considered for optimization). Include parentheses as part of the logical expression.

fieldname operator literal list

fieldname Specify a key or search field or a generic key or search field (partial field from position 1).

operator Can be =, ¬=, >, <, > =, or < =.

literal list Specify one or more characters or numeric literals. You can also specify range (literal TO literal). You cannot use summary functions and arithmetic expressions for optimization.

If there are more than eight sets of relational conditions for a given segment, there is a split of SSAs. For example:

```
DISPLAY CUSTNO IF CUSTNO = 1 2 3 4 5 6 7 8 9
```

generates two qualified SSAs. One contains the conditions 1 2 3 4 5 6 7 8 and the second contains condition 9 (due to the DL/I limitation of eight sets in one SSA).

When you specify qualification on sibling segments, you generate a set of qualified SSAs for each path. For example:

```
IF ITMQTYOR > 100 AND SHIPPER = 'JKW'
```

generates two separate SSAs, one for `ITMQTY > 100` and one for `SHIPPER = 'JKW'`.

The search applies the first SSA, and from that position on, the second SSA is applied immediately. Only records that satisfy both criteria are selected.

For this selection criteria:

```
IF ITMQTYOR > 100 OR SHIPPER = 'JKW'
```

Immediate Response generates two qualified SSAs, one for `ITMQTYOR> 100` and one for `SHIPPER = 'JKW'`. The search applies the first SSA, and a complete new search begins for the second qualification. Records are selected depending on their position in the database.

Note that:

```
IF CUSTNO > 200 OR ITMQTYOR > 100
```

is treated as an OR condition on a sibling segment. You can have up to two conditions connected by OR in sibling optimization. This limit does not apply to AND conditions. You can connect conditions by a combination of AND and ORs.

Note: You cannot optimize HDAM files.

Once you select a record, use a minimal call strategy in the intra-record processing. As a general rule, segments that are uniquely keyed, are only on one active path, and are not referenced in the query, do not cause a DL/I call.

VSAM File Optimization (Immediate Response)

To optimize VSAM files, Immediate Response constructs key ranges of records for processing. Immediate Response generates ranges using the root key only. All logical expressions down to a LET statement are candidates for optimization (see the section DL/I File Optimization (Immediate Response)).

Constructing Logical Expressions

Construct the logical expression that derives the key ranges as a group of conditions connected by AND/OR (other connectors like HAS, ANY, ALL, and NOT are not considered for optimization). You can include parentheses as part of the logical expression. Structure each condition as:

keyfield operator literal list

keyfield Is the VSAM data set key or generic key.

operator Is one of the following: =, ¬=, >, <, > =, or < =.

literal list Is one or more characters or numeric literals. You can also specify range (literal TO literal). You cannot use summary functions and arithmetic expressions for optimization.

Generate key ranges from the compound logical expressions in order to optimize VSAM access to records. The optimization mechanism positions on each range and sequentially obtains records from that position until you exhaust the range. This method applies to each of the ranges that are derived from the logical expression.

Note: You cannot optimize RBA and ESDS file types.

Optimizing Database Access for VISION:Bridge

The OPTMODE parameter in the PARMBLK determines the database optimization method used by VISION:Inform. (PARMBLK is discussed in the *VISION:Inform Installation Guide* for your environment.) There are three methods of optimization:

- Full optimization at all database levels (dynamic tuning).
- Optimization at the root level only (dynamic tuning).
- No optimization (static tuning).

See your VISION:Inform system administrator for the current setting of the OPTMODE parameter in PARMBLK.

Full Optimization (VISION:Bridge)

In an optimized search method, VISION:Inform completely bypasses the records not needed. This reduces processing time and more efficiently accesses the database.

VISION:Inform uses the full optimization method if OPTMODE=1 in the PARMBLK and your query meets the following requirements:

- The query must contain a "global" conditional phrase. The conditions must pertain to all the statements in the query; the query cannot include IF and ELSE statements.
- The "global" conditional phrase must be in a SELECT IF, REPORT IF, or EXTRACT IF statement.
- The query can contain multiple SELECT statements and only one REPORT or EXTRACT statement.
- The SELECT, REPORT, and EXTRACT statement containing the "global" conditional phrase must not be preceded by LET or END statements.
- The conditional phrase must name the key fields or database search fields that are character (not numeric) for IMS databases.
  - Any field can be named for DB2 tables.
  - For VSAM and sequential files, only the root segment key (which must be character and less than 16 bytes) can be named.
  - Partial fielding of the named field is only allowed for DB2 tables and the relational operators can only be EQ or NE.
- The relational operators EQ, LT, LE, GT, and GE can be used in the conditional phrase; NE can only be used with DB2 tables and IMS databases.
- For IMS databases, either the logical connector AND or the logical connector OR can be used in the conditional phrase; combinations of both ANDs and ORs cannot be used together.

- Compound conditional phrases (a series ANDed or ORed together with a level of nesting) must have all of the simple conditional phrases within the compound optimizable. See the next section for the definition of optimizable conditional phrases.
- Compound conditional phrases are broken down into simple compound phrases. With IMS databases there can only be up to eight simple phrases in a query or batch of queries.
- If the database is HDAM, optimization is done on lower level segments only (not on the root).
- If any query in a batch does not have a "global" conditional phrase for a segment, that segment will not be optimized for that batch.

If your query does not meet these requirements or the batch of queries has conflicting requirements, VISION:Inform uses no optimization (OPTMODE=3).

Optimizable Conditional Phrases

The following five simple conditional phrases are the only types of phrases that are "optimizable" in the global conditional phrase. All other types of simple conditional phrases are considered non-optimizable.

In the following examples, RL represents a relational operator such as EQ or GE.

Example 1. FIELD RL CONSTANT or CONSTANT RL FIELD

Example 2. FIELD RL FIELD (where both fields are in the same segment)

Example 3. FIELD RL SERIES (For example, FIELD = 1, 3, 7, 11)

Example 4. FIELD RL RANGE (For example, FIELD = 10 to 50)

Example 5. PF(FIELD) RL CONSTANT or CONSTANT RL PF(FIELD)

File Types	Allowable Conditional Phrases
DB2 tables	Examples 1 through 5
IMS Databases	Examples 1 through 4
VSAM and sequential files	Examples 1 and 4

Optimization at the Root Level (VISION:Bridge)

In an optimized search method, VISION:Inform completely bypasses the records not needed. This reduces processing time and accesses the database more efficiently.

VISION:Inform optimizes database access at the root level if OPTMODE=2 in the PARMBLK and your query meets the following requirements:

- The query must contain a "global" conditional phrase. The conditions must pertain to all the statements in the query. You cannot include an ELSE statement.
- "Global" conditional phrases must be in SELECT IF, REPORT IF, EXTRACT IF, or IF statements.
- The conditional phrase must name the key field of the root segment of the database. Partial fielding is not allowed.
- The conditional value must be a literal enclosed in quotation marks, regardless of the field type of the root segment key.
- The relational operators EQ, LT, LE, GT, and GE can be used in the conditional phrase; NE cannot be used.
- The queried database cannot be HDAM.

If your query does not meet these requirements, VISION:Inform uses no optimization (equivalent to OPTMODE=3).

No Optimization (VISION:Bridge)

When the OPTMODE parameter of PARMBLK is set to 3, VISION:Inform does not optimize database access. This is known as static tuning. In the case of static tuning, you must create your own WHERE clauses and SSAs as appropriate, because this method of optimization does not generate qualification of any kind. WHERE clauses could be included in the relational definition and SSA would be included using preselection requests cataloged with the database definition as an logical data view (LDV).

Examples of Forcing Access Optimization

The following examples cause VISION:Inform to optimize database access.

Note: Example 1 optimizes for both OPTMODE=1 and OPTMODE=2.

Example 1.

```
QUERY DATABASE CUSTOMER

REPORT ORDERNO IF CUSTNO = '00115'

END REPORT

END QUERY
```

The optimized search strategy in this example contains a global conditional phrase in the REPORT statement immediately following the QUERY statement. Only information for customer 00115 will be accessed by the query. CUSTNO is the key field in the root segment and it is on the left side of the conditional phrase.

Example 2.

Note: Example 2 optimizes only for OPTMODE=2, due to the use of the IF command.

```
QUERY DATABASE CUSTOMER

IF CUSTNO = '00048' AND INSTNO = '23468'

REPORT ...

END REPORT

END QUERY
```

The optimized search strategy in this example contains a global conditional phrase in the IF statement immediately following the QUERY statement. All segments for customer 00048 will be accessed using the optimized search strategy. The conditional phrase also contains the AND logical operator and a test for installation 23468. The segments that apply to this customer will be searched using the optimized strategy to locate the segments related to the 23468 installation.

Example 3.

The query in this example does not qualify for access optimization, because the conditional phrase is preceded by a LET statement:

```
10 QUERY DATABASE CUSTOMER

20 LET QTYBKORD = QTYBKORD + 100

30 REPORT ITEMORD QTYBKORD IF CUSTNO = '00115'

...

...

...
```

It processes with no optimization.

Optimization vs. Access Strategy (VISION:Bridge)

It is important to understand the difference between VISION:Inform's optimization and the general access strategy used for any database.

Optimization

VISION:Inform's optimization consists of generating predicates to be used in:

- WHERE clauses of SQL SELECT statements, or
- segment search arguments (SSAs) for DLI calls, or
- start/end search values for VSAM files,

thereby qualifying the access method READ operation for efficiency. VISION:Inform's optimization does not control when an actual READ is performed — but does add qualification, if possible, to the READ when it does occur.

Access Strategy

Access strategy, which is independent of the access method, determines when a READ of a file occurs.

When determining the access strategy, VISION:Inform must take into account the possibility of multiple queries being batched together in a single pass of the database. VISION:Inform cannot process a database in such a fashion as to skip a READ of record for one query that may need to be made available to another query in the same batch.

You can affect when the READ occurs by the structure of a query.

Example 1.

The following example demonstrates how you can effect the access strategy. The example assumes a DB2 logical data view with two DB2 tables joined together. If the query is structured with all qualification on the REPORT statement, it would appear as follows:

```
QUERY DB2LDV
REPORT TABLE1-FIELD TABLE2-FIELD IF TABLE1-FIELD = 'A' AND,
                                    TABLE2-FIELD = 'B'
END REPORT
END QUERY
```

In the above format, READs will always occur for TABLE1 and TABLE2 fields, prior to any comparisons being made on either TABLE1 or TABLE2.

If the query is re-structured in one of the ways as shown in the following examples, fewer READs will occur on TABLE2.

Example 2.

```
QUERY DB2LDV
SELECT TABLE1-SEGMENT IF TABLE1-FIELD = 'A'
SELECT TABLE2-SEGMENT IF TABLE2-FIELD = 'B'
REPORT TABLE1-FIELD TABLE2-FIELD
END REPORT
END QUERY
```

or

```
QUERY DB2LDV
SELECT TABLE1-SEGMENT IF TABLE1-FIELD = 'A'
REPORT TABLE1-FIELD TABLE2-FIELD IF TABLE2-FIELD = 'B'
END REPORT
END QUERY
```

In Example 2, READs will be done on the TABLE1 field, then the TABLE1 comparison is made. If, and only if, the TABLE1 comparison is TRUE, the TABLE2 field will be read and the TABLE2 comparison made.

Appendix

A

Paging and Viewing

Paging Commands in IMS

When the result displayed by VISION:Bridge (such as from a GLOSSARY command) does not fit on the screen, PA1 displays in the lower left corner of the screen. Press the PA1 key to clear the screen and display subsequent lines of output.

Viewing Reports in IMS

Use the following procedure for viewing VISION:Bridge reports in the IMS environment that exists in a non-response mode. Use this procedure so that IMS can queue messages.

To retrieve a report from the communication file:

1. Enter the appropriate VIEW command.

   VISION:Bridge displays a screen with a header page that summarizes the processing of the query and displays PA1 in the lower left corner.
2. Press PA1 on your terminal to view the next page of your output.
3. Continue paging through your report using PA1 until you reach the end of the output.

   On the last page of the report, PA2 displays at the bottom of the screen.
4. Press PA2 on your terminal to return to the system prompt.

When viewing reports under IMS, you can also specify which page of the report output to display by using the equal (=) prompt at the bottom right corner of the screen.

1. Following the equal sign, enter a number that corresponds to the output page number you want to display (for example, 1 is the header page, 2 is the first page of the report).
2. After the equal sign, type a plus (+) or minus (-) sign followed by a number to indicate the number of output pages to go forward or backward.
3. Finally, type an L after the equal sign to jump to the last page of output.
4. To return to the system prompt, press PA2 from the last page of output.

Viewing Displays in IMS (Immediate Response)

Use the following procedure for viewing Immediate Response displays in the IMS environment that exist in a non-response mode. Use this procedure so that IMS can queue messages.

After Immediate Response processes your query and the display is ready to view, PA1 appears in the lower left corner of the screen.

1. Press PA1 to clear the screen and display the results.

   If the display is larger than will fit on the screen, PA1 appears again.
2. Continue pressing PA1 until you view all pages of the display.

   Immediate Response displays a message indicating the end of the display and prompts with PA1 again if there is another display.

   When there are no more displays, Immediate Response returns to the mode from which you began.

Paging Commands in CICS

When the result displayed by VISION:Bridge (such as from a GLOSSARY command) does not fit on the screen, a BMS paging command displays in the lower left corner (it is usually P/N). (See your VISION:Inform system administrator for the BMS paging command used at your installation.)

Press the Enter key to clear the screen and display subsequent lines of output.

Viewing Reports in CICS

To retrieve a report from the communication file:

1. Enter the appropriate VIEW command. The header page of your report displays with the BMS paging command in the lower left corner. (See your VISION:Inform system administrator for the BMS paging command used at your installation.)
2. Use the following (default) commands to page through reports by typing them in the lower left corner of the screen.

P/N	display next page
P/+n	n pages forward
P/-n	n pages backward
P/C	display current page
P/P	display previous page
P/L	display last page
P/n	displays page number n

3. When the last page of the report displays, P/1 appears in the lower left corner. Press Enter to return to the first report page.

T/C	Terminates the current report and displays the first page of the second report. If there are no more reports, the screen clears, except for P/N in the lower left corner. Press Enter to return to the system mode of the Command Input session.
T/A	Terminates all reports. BMS responds with a message that the reports have been purged. Press Clear to return to the Command Input session.

Viewing Displays in CICS (Immediate Response)

After Immediate Response processes your query, and the display is ready for viewing, P/N appears in the lower left corner of the screen. (See your VISION:Inform system administrator for the BMS paging command used at your installation.)

1. Press Enter to clear the screen and display the results.
2. If the display is larger than fits on one screen, P/N appears again. Continue pressing Enter until you view all pages of the display.

   Immediate Response displays a message indicating the end of the display and prompts with P/N again if there is another display. When there are no more displays, Immediate Response returns to the mode from which you began.

Viewing Alternate Output Format Reports

See Locating Alternate Output Format Report Files on page 4-32 for an explanation of viewing reports that are in an alternate output format.

Appendix

B Glossary of the CUSTOMER Database

```
CUSTOMER - IMS DATABASE FILE  DATE CREATED -  01.111    TIME - 12.25.125
   NAME     TYPE   START   LENGTH    DEC   OUT-LEN   KEY   SEGNAME   PARENT
 ADDLIN1    CHAR      22       25              25            INSTALL   CUSTOMER
           FIRST LINE OF THE ADDRESS
 ADDLIN2    CHAR      47       25              25            INSTALL   CUSTOMER
           SECOND LINE OF THE ADDRESS
 ADDRESS    CHAR      22       50              50            INSTALL   CUSTOMER
           FIRST & SECOND LINES OF THE ADDRESS
 CHECKNO    CHAR      76        7               7            SHIPINV   ORDER
           CHECK_NUMBER
           THE PRE-PRINTED CHECK NUMBER
 CUSTNAME   CHAR       6       30              30       S    CUSTOMER  CUSTOMER
           CUSTOMER_NAME
           CUSTOMER/COMPANY NAME
 CUSTNO     CHAR       1        5               6       1    CUSTOMER  CUSTOMER
           CUSTOMER_NUMBER
           A FIVE-DIGIT CUSTOMER NUMBER
 CUSTPH     CHAR      36       10              10            CUSTOMER  CUSTOMER
           PHONE_NUMBER
           CUSTOMER PHONE NUMBER INCLUDING AREA CODE
 FRTCOST    PACKD     23        4      2       10            SHIPINV   ORDER
           FREIGHT_COST
           COST OF SHIPMENT
 INAMT      PACKD     54        4      2       10            SHIPINV   ORDER
           INVOICE_AMOUNT
           AMOUNT OF THE INVOICE
 INPRODID   CHAR      18        2              8             INSTALL   CUSTOMER
 INPRODRL   CHAR      20        2              8             INSTALL   CUSTOMER
 INSTCODE   CHAR      22        1              1             INSTALL   CUSTOMER
 INSTLOC    CHAR      23       16             16             INSTALL   CUSTOMER
 INSTCONT   CHAR       5        3              8             INSTALL   CUSTOMER
 INSTNO     CHAR       1        4              6       1     INSTALL   CUSTOMER
           INSTALLATION_NUMBER
           A FOUR-DIGIT INSTALLATION NUMBER
 INSTOPS    CHAR       8        8              8             INSTALL   CUSTOMER
 INSTOSRL   CHAR      16        2              8             INSTALL   CUSTOMER
 INVDATE    CHAR      48        6              7             SHIPINV   ORDER
           INVOICE_DATE
 INVNO      CHAR      41        5              5             SHIPINV   ORDER
           INVOICE_NUMBER
 ITEMNAME   CHAR       8       20             20             ITEMORD   ORDER
           ITEM_NAME
 ITEMORD    CHAR       1        7              7       1     ITEMORD   ORDER
           ITEM_ORDERED
 ITEMSHIP   CHAR       1        7              8       1     ITEMSHIP  SHIPINV
           ITEM_SHIPPED
 ITMPRICE   ZONED     28        6      0       8             ITEMORD   ORDER
           ITEM_PRICE
 ITMQTYOR   ZONED     34        7      0      10             ITEMORD   ORDER
```

```
           QUANTITY_ORDERED
ORDCMPLT    CHAR      29       1            10            ORDER     CUSTOMER
           IS THE ORDER COMPLETE? (Y OR N)
ORDDUDAT    CHAR      22       6             8            ORDER     CUSTOMER
           DATE THE ORDER IS DUE
ORDERNO     CHAR       1       5             7       1    ORDER     CUSTOMER
           ORDER_NUMBER
           THE FIVE-DIGIT ORDER NUMBER
ORDINVGN    CHAR      28       1            10            ORDER     CUSTOMER
ORDPONUM    CHAR      17       5             8            ORDER     CUSTOMER
           PURCHASE_ORDER_NUMBER
ORDRDATE    CHAR       8       6             8            ORDER     CUSTOMER
ORDER_DATE
ORPERSON    CHAR      14       3             8            ORDER     CUSTOMER
           ORDER_PERSON
           INITIALS OF PERSON WRITING THE ORDER
PAIDDATE    CHAR      83       6             8            SHIPINV   ORDER
           DATE_PAID
           DATE THE ORDER WAS PAID (MMDDYY)
PAYDATE     CHAR      70       6             7            ORDER     ORDER
           PAYMENT_DUE_DATE
           DAY THE PAYMENT IS DUE (MMDDYY)
QTYBKORD    ZONED     41       7     0      10            ITEMORD   ORDER
           BACK_ORDERS
           QUANTITY BACKORDERED
QTYSHIP     ZONED      8       7     0      10            ITEMSHIP  SHIPINV
           QUANTITY_SHIPPED
SEG2OFIL    CHAR      30      10            10            ORDER     CUSTOMER
           UNUSED
SEG3OFIL    CHAR      89      10            10            SHIPINV   ORDER
           UNUSED
SEG4OFIL    CHAR      15      10            10            ITEMSHIP  SHIPINV
           UNUSED
SEG5OFIL    CHAR      48      10            10            ITEMORD   ORDER
           UNUSED
SEG6OFIL    CHAR      72      10            10            INSTALL   CUSTOMER
           UNUSED
SHIPDATE    CHAR       5       6             8            SHIPINV   ORDER
           DATE_SHIPPED
SHIPNO      CHAR       1       4             6       1    SHIPINV   ORDER
           SHIPPING_NUMBER
           FOUR_DIGIT SHIPPING NUMBER
SHIPPER     CHAR      11       3             7            SHIPINV   ORDER
           SHIPPING_PERSON
           INITIALS OF PERSON WHO SHIPPED THE ORDER
***END OF DEFINITION***
```

Appendix

C

Examples of Editing Numeric Data

Picture	Value of Data	Edited Result	Explanation
999.99+	+003.45	003.45+	No suppression of leading zeros. Trailing sign.
ZZZ.99+	+003.45	3.45+	Suppression of leading zeros. Trailing sign.
$ZZZ.99+	+003.45	$ 3.45+	Fixed currency symbol. Suppression of leading zeros. Trailing sign.
$$$.99+	+003.45	$3.45+	Floating currency symbol. Suppression of leading zeros. Trailing sign.
999.99	000.00	000.00	No suppression of leading zeros.
ZZZ.99	000.00	.00	Suppression of leading zeros to decimal point.
ZZ9.99	000.00	0.00	Suppression of leading zeros to digit just left of decimal point.
,*.99	00123.45	***123.45	Check protection up to decimal point. Grouping character replacement by protection symbol if adjacent to leading zeros.
,*.**	00000.00	*****.**	Check protection through entire field. Replaces all positions, except the decimal point on a zeros field.
$++++.99	023.45	$+23.45	Fixed currency symbol. Floating sign.

Picture	Value of Data	Edited Result	Explanation
$++++.99	123.45	$+123.45	Same as above. Note: You need to allow an extra position when using a floating symbol in case there are no leading zeros.
$+++.99	123.45	‘‘not editable"	No room for floating symbol.
+$$$$.99	023.45	+ $23.45	Fixed sign. Floating $.
ZZZ.ZZ	000.00		Suppression of all leading zeros results in a zero value edited as all blanks.
$$$.$$	000.00		Same as ZZZ.ZZ above.
$$$.99	000.00	$.00	Floating currency symbol. Suppression of leading zeros to decimal point.
-99.9	-12.3	-12.3	Leading sign of negative value.
-99.9	+12.3	12.3	Sign suppressed if positive value.
.9	003.45	3.4	Floating sign if negative. Leading zeros suppressed.
ZZ,ZZZ.99	12345.6	12,345.60	Suppression of leading zeros. Grouping character left of decimal point. Extension of decimal places.
ZZ,ZZZ.999,90	00045.6789	45.678,90	Suppression of leading zeros includes imbedded or adjacent grouping characters. Grouping character left and right of decimal point. Extension of decimal place.
999.99	12.3	012.30	Extension of leading zeros and decimal places.

Appendix

D Reference Summary

The book is an alphabetically arranged list of the commands and descriptions used to create queries in the VISION:Inform® queries in both VISION:Bridge™ and Immediate Response formats.

- Each section describes the syntax of a command statement and includes a description of the command, its parameters, examples, and references to other commands and manuals.
- Each section title indicates whether the command is applicable to VISION:Bridge, Immediate Response, or both.

This appendix was previously published as a separate guide, VISION:Inform *Reference Summary*.

Syntax for a Command Statement

A statement starts with a command and can be followed by one or more parameters. Parameters can be optional or required. Parameters are composed of one or more keywords or keywords followed by keyword operands.

The general syntax for a statement is:

$$\text{COMMAND (KEYWORD}_1\text{ (operand}_1\text{))} \quad \left[\left\{\begin{array}{l}\underline{\text{KEYWORD}_2} \\ \text{KEYWORD}_3\ \text{operand}_3\end{array}\right\}\right] \ldots$$

Text, which follows the syntax, describes each element in a command statement.

Conventions

Note: Parameters can also be referred to as identifiers, modifiers, or options.

- Brackets [ ] indicate an optional parameter.
- Parameters enclosed in braces { } represent options from which you can select one or more.
- Entries printed in **UPPERCASE** are system commands, parameters, or keywords.
- Entries printed in **lowercase** are keyword operands (informational entries, such as field names) that you provide.
- Many keywords are optional if their parameters are entered in the order specified in the format. The description indicates whether or not a parameter is optional.
- Underlined entries are defaults.
- Separate commands, parameters, and operands from one another by one or more blanks.

  You can use a comma in place of, or in addition to, a blank to improve readability. Commas are optional and are not processed or interpreted, unless one is the last character on a line to indicate continuation.
- You can continue statements that do not fit on one line onto subsequent lines by terminating each line, except the last, with a comma.

Invalid Field Characters

A special character, defining the type of invalid field, prints when you attempt to report an invalid field. Your installation can change these characters at the time of installation.

* This field is invalid as a result of a computation during query processing.

– This field does not exist on the record.

\+ This field is too large to fit in the output column provided, or data was invalid at printing time.

Reserved Keywords

A data name is the name of a database segment, a database field, or a temporary field. A data name contains a maximum of eight characters – an alphabetic character followed by up to seven alphanumeric characters.

Do not specify field names that are the same as system-defined names (reserved keywords), such as commands, parameters, built-in function names, or operators.

The list of restricted keywords follows:

ACTIVE	ALL	ALTSORT	AND	ANY
AVG	AWAITING	BATCHING	BE00	BLKLNSUP
BORDER	BS00	BT00	BY	CALL
CALLS	CE00	CLASS	CMSSTR	COMMAND
CONTROL	COPY	COUNT	CUM	DATABASE
DATAVIEW	DATE	DATEFMT	DATEPOS	DB00
DB2	DBCALL	DBUGABND	DBUGSKIP	DECIMALS
DEFAULT	DEFINE	DELETE	DESC	DISABLE
DISABLED	DISPLAY	EDIT	ELSE	EMPTYFLD
ENABLE	END	ENQDEQ	ENTRY	EOF
EQ	EQUAL SIGN (=)	EQUATE	ERRORS	ESTAE
EXCLUDE	EXPR	EXTRACT	EXTREL	FDHELP
FIELD	FILE	FILEID	FILRFDMP	FORMAT
FROM	GE	GLOSSARY	GO	GRAND
GRANDSUM	GROUP	GT	HAS	HAVE
HEADING	HEADINGS	HEIGHT	HELD	ID
IDPREF	IF	IM00	IMMED	INCLUDE
INSERT	INTSTR	INTVL	IRABEND	IRCHKOUT
IRINFO	IRNODESC	IRSPOOL	IRTRACE	ISDATE
ITEM	JULANX	JULIAN	KEYWORD	LABELS
LE	LENGTH	LET	LG00	LILIAN
LINE	LINENOS	LINES	LIST	LISTLIB
LNKTRC	LOGON	LONG	LT	M4INPT
M4LIST	M4REPO	MAINT	MASK	MAX
MAXITEMS	MAXLINES	MAXQRY	MAXTIME	MESSAGEQ
MIN	MINQRY	MODE31	NAME	NE
NEWLINE	NEWPAGE	NOCHECK	NONPRINT	NOPURGE

NOT	OF	OPTION	OPTIONS	OR
ORDER	OUTFMT	OVERRIDE	PAGE	PAGEGRP
PAGEPOS	PAGES	PARM	PASSWORD	PCHECK
PCT	PF	PICTURE	PLAN	PRDDUMP
PRINT	PRINT8	PRIVATE	PROFILE	PROHIBIT
PROMPT	PSBNAME	PSTATUS	PURGE	QCHECK
QE00	QRYKYDMP	QS00	QSTATUS	QT00
QTIME	QUERY	QUIT	RATIO	READY
RENUMBER	REPORT	RESTRICT	RNUMBER	ROUTE
RUN	SAVE	SEGMENT	SEGTBDMP	SELECT
SET	SETTING	SOURCE	SPACES	START
STATUS	STMT	STOP	STORAGE	SUBFILES
SUBMIT	SUBTITLE	SUMMARY	SWITCH1	SWITCH2
SYSDATE	SYSTEM	TARGET	TERM	TEXT
TIME	TITLE	TITLEPOS	TO	TODAY
TODAYX	TOTAL	TQRYNAME	TRBOIMS	TRBOSQL
TRBOSTEN	TURBO	TYPE	USERID	USERS
USING	VALUE	VIEW	WIDTH	WITH
WRAPNO				

To use pre-existing field names that do not meet these requirements, enclose non-conforming names in single quotation marks and precede them with the letter N. Quotation marks within a name are represented by two consecutive quotation marks.

Examples:

```
N'707-SAL'
N'FIELD'
N'O''CLOCK'
```

Arithmetic Expression (VISION:Bridge and Immediate Response)

operand operator operand operator . . .

Represents a computation that produces a value.

operand — Is a constant, a field name (except a variable length character field), or a built-in function.

operator — Is + (addition), – (subtraction), * (multiplication), or / (division).

- You can use + or – as the leading sign of an operand.
- Expressions are normally evaluated from left to right.
- Multiplication and division operators are processed before addition and subtraction.
- You can use parentheses to override the normal processing order. The innermost set of parenthetical terms will be evaluated first.

Examples:

```
COST/AMOUNT
1.15*BASECST
(*COST+OVERG)*QUANTY
```

AVG (VISION:Bridge)

AVG ITEM fieldnames BY fieldname

Reports the average value of all valid occurrences of the specified fields.

ITEM fieldnames — Specifies the fields to be averaged.

- You can omit ITEM if the field name appears immediately after AVG.
- Make the fields subordinate to any field you specify in BY fieldname.
- List the fields on the REPORT statement, but *not* on the GROUP statement.

BY fieldname — Specifies the controlling field for the averages. The average is reported every time this field changes.

- If you do not specify BY fieldname, the average is computed for the entire report.
- You must also list this field on a GROUP statement.

Examples:

```
AVG ITEM INAMT BY CUSTNO
AVG INAMT
AVG ITMPRICE
```

AVG (Immediate Response)

```
AVG    ITEM fieldnames BY fieldnames GRAND, TITLE 'text',
       ORDER fieldnames,
       DESC fieldnames,
       IF logical expression
```

Displays the average value of all valid occurrences of the specified fields.

ITEM fieldnames — Specifies the fields to be averaged.

- You can omit ITEM if the field name appears immediately after AVG.
- Make the fields subordinate to any fields you specify in BY fieldname.

BY fieldnames — Specifies the fields to be used as control break fields for the averages. The average is listed every time this field changes. If you do not specify BY fieldname, grand summaries are produced.

GRAND — Specifies that grand summaries are to be produced.

TITLE 'text' — Specifies one or two title lines for each page of the display or printout.

- The text appearing between the single quotation marks is the title.
- To produce two title lines, enclose each title line in single quotation marks.

ORDER fieldnames	Specifies the fields by which to sort the output data. The first field name is the first sort field, the second field name is the second sort field, and so on. The following rules apply to the field names used with ORDER: ■ Each field name can be either a database field or a temporary field already defined in the query. ■ Each database field must be in a database segment already selected by the query. ■ Each field name is grouped (duplicate values do not display). ■ Do not place a summarized field name in the ORDER list.
DESC fieldnames	Specifies the field names in the ORDER parameter that are to be sorted in descending order.
IF logical expression	Specifies the conditions under which the average is taken. See the section Logical Expression (VISION:Bridge and Immediate Response).

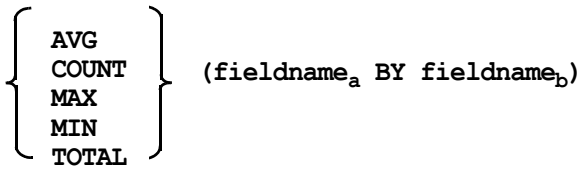
Examples:

```
AVG ITEM INAMT BY CUSTNO
AVG INAMT TITLE 'INVOICE AMOUNT (AVERAGE)'
AVG INAMT BY CUSTNO IF INAMT GT 100
```

Built-In Summary Functions (VISION:Bridge and Immediate Response)

Note:

- Built-in summary functions operate at the record level.
- Summary commands (AVG, COUNT, CUM, MAX, MIN, PCT, RATIO, and TOTAL) operate at the group or entire file level.



Performs summaries on fields within structured database records.

AVG	Computes the average of all valid occurrences of the specified field.
COUNT	Counts the number of valid occurrences of the specified field.
MAX	Determines the maximum valid value of the specified field.
MIN	Determines the minimum valid value of the specified field.
TOTAL	Computes the sum of all valid occurrences of the specified field.
$fieldname_a$	Is the field for which the summary is to be taken.
BY $fieldname_b$	Specifies the controlling field for the summaries (all occurrences of $fieldname_a$ are summarized within each occurrence of $fieldname_b$). If you do not specify BY $fieldname_b$, the immediate parent of $fieldname_a$ is used as the controlling field. ■ Make $fieldname_a$ subordinate to $fieldname_b$. ■ When specified for output on a report or display, built-in summaries are printed in columns like data fields. ■ Neither the segment containing $fieldname_a$ nor any segment on the path between $fieldname_a$ or $fieldname_b$ can be in a segment that has selection criteria on it that would inhibit the processing of all occurrences.

Examples:

```
REPORT CUSTNAME COUNT(ORDERNO)
LET TEMPA = TOTAL(INAMT BY CUSTNO)
DISPLAY CUSTNO IF COUNT(ITEMORD BY ORDERNO) > 4
```

CALL (VISION:Bridge)

Note: Errors can occur if the size and field types of the parameters passed between VISION:Bridge and the user written routine do not agree.

`CALL NAME user-routine USING parameters`

Tells VISION:Bridge to execute a user written program that is stored outside of VISION:Bridge.

NAME user-routine	Provides VISION:Bridge with the name of the routine to be called. Specify a name up to eight characters long, beginning with an alphabetic character, and followed by alphanumeric characters.
USING parameters	Provides the input and output fields that the routine needs for execution. ■ These are positional parameters you must list in the order that the user routine expects. ■ They can be fixed-length database fields, temporary fields created with LET or SET statements, numeric constants, or character constants. ■ You can specify numeric constants up to 15 digits. ■ Make character constants 31 characters or less and enclose them in single quotation marks.

Example:

```
CALL CONVERT USING SCORE METERS
```

Comments (VISION:Bridge)

`;comments`

Specifies that the semicolon (`;`) and what follows are not processed as query statements. You can place the semicolon on the statement after a command, or you can make it the first non-blank character on the statement. Do not use comments in Immediate Response immediate mode.

Examples:

```
QUERY CUSTOMER  ;MY CURRENT FILE
; THIS REPORT SHOULD BE RUN EVERY MONTH
TITLE 'SALES FOR MAY'; CHANGE MONTH EACH RUN
```

COUNT (VISION:Bridge)

COUNT ITEM fieldnames BY fieldname

Reports the count of all valid occurrences of the specified fields.

ITEM fieldnames — Specifies the fields to be counted.

- You can omit ITEM if the field name appears immediately after COUNT.
- List these fields on the REPORT statement, but *not* on the GROUP statement.

BY fieldname — Specifies the controlling field for the counts. The count is listed every time this field changes.

- If you do not specify BY fieldname, the count is computed for the entire report.
- You must also list this field on a GROUP statement.

Examples:

```
COUNT ORDERNO BY CUSTNO
COUNT CUSTNO
```

COUNT (Immediate Response)

```
COUNT    ITEM fieldnames BY fieldnames,
         GRAND,
         TITLE 'text' ORDER fieldnames DESC fieldnames,
         IF logical expression
```

Displays the count of all valid occurrences of the specified fields.

ITEM fieldnames — Specifies the fields to be counted.

- You can omit ITEM if the field name appears immediately after COUNT.
- Make the fields subordinate to any fields you specify in BY fieldname.

BY fieldnames	Specifies the fields to be used as control break fields for the counts. The count is listed every time this field changes. If you do not specify BY fieldname, grand summaries are produced.
GRAND	Specifies that grand summaries are to be produced.
TITLE 'text'	Specifies one or two title lines for each page of the display or printout. ■ The text appearing between the single quotation marks is the title. ■ To produce two title lines, enclose each title line in single quotation marks.
ORDER fieldnames	Specifies the fields by which to sort the output data. The first field name is the first sort field, the second field name is the second sort field, and so on. The following rules apply to the field names used with ORDER: ■ Each field name is either a database field or a temporary field already defined in the query. ■ Each field can be in the ITEM list. ■ Each database field must be in a database segment already selected by the query. ■ Each field name is grouped (duplicate values do not display). ■ Do not put a summarized field name in the ORDER list.
DESC fieldnames	Specifies the field names to sort in descending order in the ORDER parameter.
IF logical expression	Specifies the conditions under which the count is taken. See the section Logical Expression (VISION:Bridge and Immediate Response).

Examples:

```
COUNT ORDERNO BY CUSTNO,
      TITLE 'COUNT OF ORDERS BY CUSTOMERS'
COUNT INSTNO TITLE 'NUMBER OF INSTALLATIONS'
```

CUM (VISION:Bridge)

CUM ITEM fieldnames BY fieldname

Reports the cumulative total of all valid occurrences of the specified fields.

ITEM fieldnames	Specifies the fields selected for cumulative totaling. ■ You can omit ITEM if the field name appears immediately after CUM. ■ List these fields on the REPORT statement, but *not* on the GROUP statement.
BY fieldname	Specifies the controlling field for the cumulative totals. The cumulative total is listed every time this field changes. If you do not specify BY fieldname, the summary is computed for the entire report. List this field on a GROUP statement.

Examples:

```
CUM ITEM INAMT BY CUSTNO
CUM INAMT
```

CUM (Immediate Response)

```
CUM    ITEM fieldnames BY fieldnames GRAND, TITLE 'text',
       ORDER fieldnames,
       DESC fieldnames,
       IF logical expression
```

Displays the cumulative total of all valid occurrences of the specified fields.

ITEM fieldnames	Specifies the fields selected for cumulative totaling. ■ You can omit ITEM if the field name appears immediately after CUM. ■ Make the fields subordinate to any specified fields in BY fieldname.

BY fieldnames	Specifies the fields used as control break fields for the cumulative totals. The cumulative total is listed every time this field changes. If you do not specify BY fieldname, grand summaries are produced.
GRAND	Requests grand summaries.
TITLE 'text'	Specifies one or two title lines for each page of the display or printout. ■ The text appearing between the single quotation marks is the title. ■ To produce two title lines, enclose each title line in single quotation marks.
ORDER fieldnames	Specifies the fields by which the output data is sorted. The first field name is the first sort field, the second field name is the second sort field, and so on. The following rules apply to the field names used with ORDER: ■ Each field name is either a database field or a temporary field already defined in the query. ■ Each field can be in the ITEM list. ■ Each database field must be in a database segment already selected by the query. ■ Each field name is grouped (duplicate values do not display). ■ Do not put a summarized field name in the ORDER list.
DESC fieldnames	Specifies the field names to sort in descending order in the ORDER parameter.
IF logical expression	Specifies the conditions under which the cumulative totals are taken. See the section Logical Expression (VISION:Bridge and Immediate Response).

Examples:

```
CUM ITEM INAMT BY CUSTNO ORDER BY CUSTNO
CUM INAMT IF CUSTNO > 11000
```

DISPLAY (Immediate Response)

```
DISPLAY   ITEM fieldnames TITLE 'text',
          { AVG   }
          { COUNT }
          { CUM   }  fieldnames BY fieldnames,
          { TOTAL }
          GRAND,
          ORDER fieldnames DESC fieldnames,
          IF logical expression
```

Specifies the data to be displayed, the title of the display, summaries, groupings, and the logical conditions under which the command executes.

ITEM fieldnames	Specifies the fields (including arithmetic expressions, summary functions, or constants) to be displayed. The fields display in the order they are listed. ■ You can omit the keyword ITEM if you specify the field names immediately after the DISPLAY command. ■ Specify the ITEM keyword only once in the statement. ■ Enclose constants in single quotation marks.
TITLE 'text'	Specifies one or two title lines for each page of the display or printout. ■ The text appearing between the single quotation marks is the title. ■ To produce two title lines, enclose each title line in single quotation marks (see example).

Note: If you specify ORDER, place any summarized field in the lowest hierarchical level of all the fields specified in the ITEM list.

{ AVG COUNT CUM TOTAL } fieldnames BY fieldnames	Specifies the type of summary taken on the fields. A summary is listed whenever a control break occurs in the fields after the BY keyword. If you do not specify BY fieldname, grand summaries are produced.
GRAND	Requests grand summaries.

ORDER fieldnames	Specifies the fields by which the output data is sorted. The first field name is the first sort field, the second field name is the second sort field, and so on. Apply the following rules to the field names used with ORDER: ■ Each field name is either a database field or a temporary field already defined in the query. ■ Each field can be in the ITEM list. ■ Each database field must be in a database segment already selected by the query. ■ Each field name is grouped (duplicate values do not display). ■ Do not put a summarized field name in the ORDER list.
DESC fieldnames	Specifies the ORDER parameter field names to sort in descending order.
IF logical expression	Specifies the conditions under which the display occurs. See the section Logical Expression (VISION:Bridge and Immediate Response).

Examples:

```
DISPLAY EMPNO IF SALARY>30000 AND HISTRYR=80,
        ORDER EMPNO DESC EMPNO

DISPLAY EMPNO SEX CARTYPE SCHOOL,
        COUNT(DEGREE BY EMPNO) IF EMPNO>'80000',
        COUNT CARTYPE BY EMPNO GRAND,
        TITLE 'EMPLOYEE TYPE ''8'' ON',
        'LIST OF CAR AND SCHOOL'

DISPLAY EMPNO EMPNAME
```

END IF (VISION:Bridge)

END IF

Terminates the conditional criteria set up by the preceding IF statement. END IF marks the end of an IF/ELSE group.

END IMMED (Immediate Response)

END IMMED

Clears the Immediate Response immediate work area and switches from Immediate Response immediate mode to system mode (?:).

Immediate Response has three modes: immediate mode, system mode, and checkpoint mode. For additional information on modes and the immediate work area, see the *VISION:Inform User Guide* (previously titled *VISION:Bridge for VISION:Inform User's Guide*).

END QUERY (VISION:Bridge and Immediate Response)

END QUERY

Marks the end of a query group. In Immediate Response immediate mode (IMMED:), END QUERY terminates a query and clears all statements from the immediate work area.

END REPORT (VISION:Bridge)

END REPORT

Terminates the report specifications set up by the preceding REPORT statement. END REPORT marks the end of a report group.

END SELECT (VISION:Bridge and Immediate Response)

END SELECT

Terminates the selection criteria set up by the SELECT statement. END SELECT marks the end of a SELECT group.

In Immediate Response immediate mode (IMMED:), END SELECT terminates the criteria set up by the preceding SELECT statement. The SELECT statement, the END SELECT statement, and any statements between are cleared from the immediate work area.

EXTRACT (VISION:Bridge)

EXTRACT NAME subfile name ITEM fieldnames DEFINE IF expression

Extracts data fields and outputs a subfile.

NAME subfile name	Enter the DD name of the subfile to be extracted. Make this name correspond to a subfile name in your profile.
ITEM fieldnames	Enter the fields to copy to the subfile. The subfile fields appear in the same order as you enter them on the EXTRACT statement.
DEFINE	Specifies that file definition statements are written to a predetermined data file. This is an optional parameter. Check with your system administrator before using it.
IF expression	Specifies the conditions underwhich the EXTRACT command is executed. See the section Logical Expression (VISION:Bridge and Immediate Response).

Examples:

```
EXTRACT SUB08 EMPNO SALARY,
        DEFINE IF SALARY GT 20000
```

FORMAT (VISION:Bridge)

```
FORMAT    WIDTH value HEIGHT value,
          DATEPOS value PAGEPOS value TITLEPOS value,
          LABELS value HEADINGS value DATEFMT value,
          LINES number BORDER 'x' PAGEGRP value,
          OUTFMT value
```

Specifies the report output format by overriding the defaults.

WIDTH value	Enter a number from 79 to the installation maximum (within the limit of the output device).
HEIGHT value	Enter the number of lines to appear on each page. You can specify from 2 to 255 lines within the limit of the page size.
DATEPOS value	Enter one of the following values:

	UL	Upper left corner (default)
	UR	Upper right corner
	UM	Upper middle of page
	LL	Lower left corner
	LR	Lower right corner
	LM	Lower middle of page
	NO	Omit date completely

PAGEPOS value — Same choices as DATEPOS. UR is the default.

TITLEPOS value — Enter TOP or BOTTOM to specify title centering. TOP is the default.

LABELS value — Enter one of the following values:

SPACE	Reserve 14 spaces at the left margin for printing labels (default).
NOSPACE	Print labels, but begin column data at the leftmost position. Labels can be partially or totally overlaid.
SUPPRESS	Do not print summary labels. Begin columnar data at the leftmost position.

HEADINGS value — Enter NO to suppress the printing of column headings, or YES to print column headings. YES is the default.

DATEFMT value — Enter one of the following keywords to use the current date:

DATE	MMM DD, YYYY format (Default)
TODAY	MM/DD/YY format
ISDATE	YYYY-MM-DD format
JULIAN	YY.DDD format

To produce a specific date (not the current date), enter the date in the form MMDDYY.

LINES number — Enter a number from 0 to 8 for the number of blank lines to print between detail lines. 0 is the default.

BORDER 'x'
: Enter any character in place of x, including one blank. This character prints above and below the column headings instead of the installation default.

PAGEGRP value
: Enter NO to suppress the automatic printing of subtitles at the top of each page. Subtitles print only when they change value.

OUTFMT value
: An optional parameter that indicates the desired output format of the report for the Report Group. Enter one of the following values:

  STD
  : Indicates that the output is to be in standard report format, which is the default.

  HTML
  : Indicates that the output is to be a report in HTML (HyperText Markup Language).

  TAB
  : Indicates that the output is to be data in tab-delimited format.

  COMMA
  : Indicates that the output is to be data in comma-delimited format (also known as Comma Separated Variable, or CSV).

  PLAIN
  : Indicates that the output is to be a report in plain text format.

  The OUTFMT keyword of the FORMAT statement applies only to the report for the Report Group that contains it. Output formats specified on the FORMAT statement can be overridden when the query is submitted, using the Submit Panel or SUBMIT command.

Example:

```
FORMAT WIDTH 80 DATEPOS UR PAGEPOS LL,
       TITLEPOS BOTTOM
```

GLOSSARY (VISION:Bridge and Immediate Response)

GLOSSARY DATABASE name { SEGMENT segname / FIELD fieldname }

In the system mode (?:), the GLOSSARY command displays information about the databases available to the user. SEGMENT and FIELD are mutually exclusive. The default is the entire database.

DATABASE name	Specifies the database name. You can leave out the keyword DATABASE if you enter the database name immediately after GLOSSARY.
SEGMENT segname	Displays the segment name. You can leave out the keyword SEGMENT if the segment name follows the database name. SEGMENT is the default.
FIELD fieldname	Displays the field name. You must specify FIELD to obtain glossary information on a single field.

Examples:

```
GLOSSARY DATABASE CUSTOMER
GLOSSARY CUSTOMER SEGMENT INSTALL
GLOSSARY DATABASE CUSTOMER FIELD INSTNO
GLOSSARY CUSTOMER INSTALL
```

GO (Immediate Response)

GO

Tells Immediate Response to continue with the display. The message ENTER "GO" TO CONTINUE OR "STOP" TO STOP is issued in checkpoint mode each time the output page limit or database call limit is reached. The limits are specified in the user's profile.

GROUP (VISION:Bridge)

```
GROUP BY names   { SUBTITLE }
                 { NEWPAGE  }
```

Suppresses printing of repetitive data in the fields listed.

BY	Optional keyword.
names	Group the names of the fields, in order of significance. Variable length fields cannot be grouped.
SUBTITLE	Prints a subtitle with each change in value of the grouping field. List SUBTITLE fields in the REPORT statement.
NEWPAGE	Starts a new page with each change in value of the grouping field. List NEWPAGE fields in the REPORT statement.

Examples:

```
GROUP BY CUSTNO SUBTITLE
GROUP CUSTNO ORDERNO
```

IF (ELSE) (VISION:Bridge)

```
IF EXPR logical expression
ELSE
END IF
```

Indicates that the statement under its control executes only if the logical expression
is true.

EXPR logical expression	Contains the conditions that must be met for the expression to be true. EXPR is optional. See the section Logical Expression (VISION:Bridge and Immediate Response).
ELSE	The optional ELSE command indicates than an alternate set of instructions executes if the logical expression is false.
END IF	Command to mark the end of an IF/ELSE group.

Examples:

```
IF CUSTNAME EQ 'TULSA TIMES'
   LET STAR = '*'
 ELSE
   LET STAR = '***'
```

IMMED (Immediate Response)

```
IMMED
```

In Immediate Response, the IMMED command switches from the system mode (?:) to the immediate mode.

ITEM (VISION:Bridge)

```
ITEM   ENTRY names SPACES value PICTURE pattern,
       HEADING 'text' NEWLINE NONPRINT,
       CSVQUOTE CSVTRUNC
```

Overrides the default editing specifications for reports.

ENTRY names — Enter the names of the fields to edit.

- If you place the field names immediately after the ITEM command, the word ENTRY is optional.
- These names are for the one or more output fields manipulated by the ITEM statement.
- List each field in only one ITEM statement.
- Also list these fields on the REPORT statement.
- If the REPORT statement contains multiple references to a given field, the ITEM statement applies to the first reference only.

SPACES value — Enter a number from 0 to 99 indicating how many spaces to insert before each data item prints on the report. The default value is 2 spaces.

PICTURE pattern — Enter a character-by-character representation of how the output data from the specified field will look. This representation overrides any output edit specifications in the file definition.

- No edit picture can exceed 30 characters.
- There are two edit picture formats: one for numeric data, and one for character data.
- The keyword PICTURE is required.

For more information, see the *VISION:Inform User Guide* (previously titled *VISION:Bridge for VISION:Inform User's Guide*).

HEADING 'text'	Enter character constants (enclosed in single quotation marks) that define column headings to be printed for the associated field. ■ Use character constants to create headings for temporary fields defined with the LET or SET statement. ■ You can define up to two lines of heading, with each line containing from 1 to 14 characters.
NEWLINE	Prints the associated data item on a new line. If you place several field names on the ITEM statement, each prints on a new line.
NONPRINT	Inhibits printing of the associated data item. If several field names are on the ITEM statement, none are printed.
CSVQUOTE	Specify CSVQUOTE only for a character field, when the output format for the report will be specified as Comma-delimited or Tab-delimited. This indicator tells the background processor to always enclose character fields within quotes (""), regardless of whether or not it contains an embedded comma (,).
CSVTRUNC	Specify CSVTRUNC only for a numeric field, when the output format for the report will be specified as Comma-delimited or Tab-delimited. This indicator tells the background processor to truncate trailing zeros to the right of the decimal point of the numeric field up to the decimal point.

Examples:

```
ITEM ENTRY PONUMBER PICTURE 'P.O. BOX ' XXXXX
ITEM CUSTNAME SPACES 7
ITEM ENTRY ORDRDATE PICTURE XX'/'XX'/'XX
```

LET (VISION:Bridge and Immediate Response)

Note: If the temporary field (field identifier) does not exist, it will be created, regardless of the result of the logical expression.

```
LET    FIELD fieldname = expression  IF logical expression
```

Assigns a value to a field. If the field has not been previously defined, a temporary field is created. See the section SET (VISION:Bridge and Immediate Response).

FIELD fieldname	Is the name of the database field, temporary field, or partial field. See the section PF (Partial Field Function) (VISION:Bridge and Immediate Response).
= expression	Specifies the value assigned to the field. ■ It can be a field, part of a field, a constant, or a built-in function. See the sections PF (Partial Field Function) (VISION:Bridge and Immediate Response) and Built-In Summary Functions (VISION:Bridge and Immediate Response). ■ It can be an arithmetic expression. See the section Arithmetic Expression (VISION:Bridge and Immediate Response). ■ If a temporary field is referenced, it must have been previously defined.
IF logical expression	Specifies the conditions under which the LET statement is executed. See the section Logical Expression (VISION:Bridge and Immediate Response).

Examples:

```
LET TEMP = INAMT*1.1
LET SALMAX = 100000
LET FIELD VALID = INSTNO IF CUSTNO > 10000
```

LIST (VISION:Bridge and Immediate Response)

LIST

In the Immediate Response immediate mode (IMMED:), LIST displays all the statements currently in the immediate work area.

```
LIST ITEM name TYPE { QUERY }
                    { STMTS }
```

In system mode (?:), LIST displays a collection of statements stored in the foreground library.

ITEM	name	Is the name of the statements that were previously saved in the foreground library. ITEM is a required keyword.
TYPE	{ QUERY } { STMTS }	Specifies whether the item is a query (QUERY) or a collection of statements (STMTS). If you do not specify this keyword, TYPE QUERY is the default.

Examples:

```
LIST ITEM AUDIT TYPE STMTS
LIST ITEM NAMES
```

LISTLIB (VISION:Bridge and Immediate Response)

Note: If you do not specify a keyword (DATABASE or QUERY), all names display.

```
LISTLIB { DATABASE }
        { QUERY    }
```

In the system mode (?:), LISTLIB displays names in the foreground library by category, in alphabetical order.

DATABASE	Displays all file definition names (databases and logical data views) accessible to the current user ID.
QUERY	Displays the names of all queries and collections of statements accessible to the current user ID.

Examples:

```
LISTLIB DATABASE
LISTLIB
```

Logical Expression (VISION:Bridge and Immediate Response)

logical expression

By a true or false decision, a logical expression determines whether or not the statement executes.

- In VISION:Bridge, you can use logical expressions in conjunction with the following statements: IF(ELSE), LET, REPORT, SELECT, and EXTRACT.
- In Immediate Response, you can use expressions with the following statements: AVG, COUNT, CUM, DISPLAY, LET, SELECT, and TOTAL. For more information, see the *VISION:Inform User Guide* (previously titled *VISION:Bridge for VISION:Inform User's Guide*).

Simple Form (Relational Expression)

operand operator operand

Where:

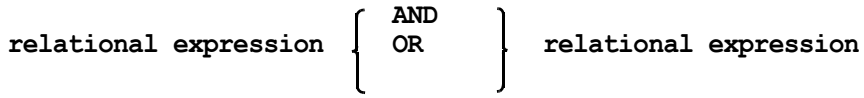
operand	Can be a constant, field name, built-in summary function, or arithmetic expression.	
operator	LT or <	Less than
	LE or <=	Less than or equal to
	EQ or =	Equal
	GT or >	Greater than
	GE or >=	Greater than or equal to
	NE or ¬=	Not equal

Examples:

```
CUSTNO LT 10010
INAMT GE 1.1*TEMPA
```

Compound Form (Multiple Relational Expressions)

Note: You can connect two or more relational expressions by AND or OR.



Examples:

```
CUSTNO = 10010 AND INAMT GT 500
INAMT <= 250 OR QTYSHIP < 3 OR ITMQTYOR < 3
```

Selective Operators — ALL and ANY

```
{ ALL }
{ ANY } logical expression
```

If you use ALL or ANY:

- All occurrences under the common parent of the segments mentioned in the logical expression are tested.
- At least one of the fields in the logical expression must be on a database path not already selected.
- For ANY, one of the occurrences must be true for the logical expression to be true.
- For ALL, all of the occurrences must be true for the logical expression to be true.

Example:

```
ANY QTYSHIP LT 3
```

Selective Operator — NOT

NOT logical expression

The NOT operator reverses the evaluation of the logical expression.

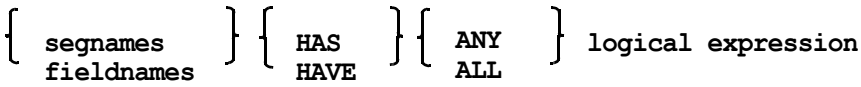
```
NOT (A=B AND A=C)
```

Both relational expressions must be true for the logical expression to fail. The whole expression is true if the expression in parentheses is false. In other words, the NOT operator reverses the evaluation of true or false for the final outcome of the expression.

```
NOT (NUMBER = 1 OR NUMBER = 5)
```

This example uses the NOT operator to exclude a few values that are not wanted for processing instead of including all of the values that are wanted. The example is true for all values of NUMBER, except 1 or 5.

HAS Qualifier

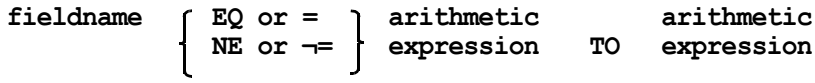


Explicitly specifies the range of the ANY and ALL operators, the normal range of which is the immediate parent. You can specify any higher parent.

Examples:

```
CUSTOMER HAS ANY INAMT GT 435.00
CUSTOMER HAS NOT ALL INAMT LT 435.00
ORDERNO HAS ANY INAMT GT 500
CUSTNO ORDERNO HAVE ALL COUNT(ORDERNO) GT 1
```

Range of Values



Specifies an inclusive range of values.

Example:

```
INAMT = 100 TO 200
```

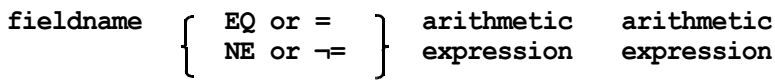
is equivalent to:

```
INAMT >= 100 AND INAMT <= 200
```

Specific Values

Note:

- EQ or = implies OR (equal to at least one).
- NE or ¬= implies AND (not equal to any).



Specifies a choice among several values (a range of values can also be used). You can use (optional) commas for clarity.

Example 1

```
INAMT = 100, 125, 130, 140
```

is equivalent to:

```
INAMT = 100 OR INAMT = 125 OR INAMT = 130 OR INAMT = 140
```

Example 2

```
INAMT NE 100 125 130
```

is equivalent to:

```
INAMT NE 100 AND INAMT NE 125 AND INAMT NE 130
```

MAX (VISION:Bridge)

MAX ITEM fieldnames BY fieldname

Reports the maximum valid value of the specified fields.

ITEM fieldnames	Specifies the data fields selected for summarization. ■ You can enter more than one field. ■ List these fields on the REPORT statement, but not on the GROUP statement. ■ ITEM is an optional keyword.
BY fieldname	Specifies the controlling field for the summary. The summary is reported every time this field changes. ■ List this field on the GROUP statement. ■ If you omit BY fieldname, the summary is computed for the entire report.

Examples:

```
MAX ITEM ITMPRICE ITMQTYOR BY ORDERNO
MAX INAMT
```

MIN (VISION:Bridge)

MIN ITEM fieldnames BY fieldname

Reports the minimum valid value of the specified fields.

ITEM	Specifies the data fields selected for summarization. ■ You can enter more than one field. ■ List these fields on the REPORT statement, but not on the GROUP statement. ■ ITEM is an optional keyword.
BY fieldname	Specifies the controlling field for the summary. The summary is reported every time this field changes. ■ List this field on the GROUP statement. ■ If you omit BY fieldname, the summary is computed for the entire report.

Examples:

```
MIN ITEM INAMT BY CUSTNO
MIN INAMT
```

ORDER (VISION:Bridge)

ORDER BY names DESC names

Outputs data in ascending or descending sequence, which can be different from the way it is stored on the database.

BY	Optional keyword if the sorted fields are listed directly after the ORDER command.
names	Sorted fields listed in order of sort significance.
DESC	Indicates that the named fields sort in descending sequence. If you omit this keyword, the fields that follow sort in ascending sequence.
names	Fields sort in descending sequence. Verify that these fields also appear in BY names.

Examples:

```
ORDER BY CUSTNO ORDERNO
ORDER BY CUSTNAME
ORDER BY SHIPDATE DESC SHIPDATE
```

PCT (VISION:Bridge)

Note: If you use the same field as the numerator in both the PCT and RATIO statements in a single report, then both statements must have the same denominator field name.

PCT ITEM names OF name BY name

Computes a percentage of one field to another.

ITEM names	The names of the numerator fields in the percent calculation. ■ List these fields on the REPORT statement, but not on the GROUP statement. ■ Do not use the same field as numerator in multiple PCT statements. You can use temporary fields of equal value to accomplish this. ■ ITEM is an optional keyword.

OF name	Indicates the name of the denominator field for the percentage calculation. ■ List this field on the REPORT statement, but not on the GROUP statement. ■ OF is a required keyword on the PCT statement.
BY name	The name of a field from the GROUP statement to indicate the level at which summaries are to be taken. If you omit the BY parameter and name, the entire report is summarized.

Example:

```
PCT FRTCOST BY ITMPRICE
```

PF (Partial Field Function) (VISION:Bridge and Immediate Response)

PF(fieldname,start,length)

Specifies part of a character field. You can use partial fielding in place of field names in logical expressions, LET statements, REPORT statements, and DISPLAY statements.

fieldname	Specifies the character field. You will use part of this field.
start	Specifies the starting position in the character field. Use an integer constant from 1 to 99.
length	Specifies the number of characters used. ■ Use an integer constant from 1 to 99. ■ If you omit the length, all characters from the start position up to and including the last character are used.

Example:

```
PF(INSTNO,3,2)
LET TEMPA = PF(ORDERNO,1,2)
```

PSTATUS (VISION:Bridge)

```
PSTATUS { NAME background processors }
        { ALL                        }
```

In the system mode (?:), PSTATUS displays the status of one or more Background Processors.

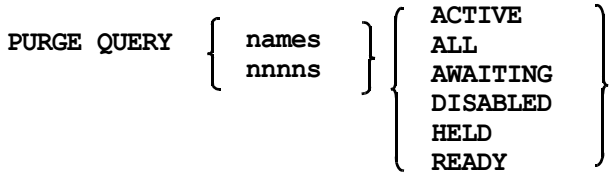
The keywords, ALL and NAME, are mutually exclusive. If you do not specify either, the status of all currently active Background Processors displays.

NAME background processors	Specifies the names of active Background Processors to be listed. If you do not specify a Background Processor name, all active Background Processors are listed.
ALL	Displays all Background Processors that have been executed with the current communication file - both those that are currently active, and those that have run previously but are not currently active.

Examples:

```
PSTATUS
PSTATUS NAME PAYPROC
PSTATUS ALL
```

PURGE (VISION:Bridge)



In the system mode (?:), PURGE deletes queries that are currently executing, waiting for execution, ready for viewing, held due to data view unavailability, or disabled due to an execution problem.

QUERY { names / nnnns }	Specifies the names or 4-digit task numbers of the queries to be deleted. If you do not specify this parameter, all queries with the specified status are deleted.
ACTIVE	Purges the active (executing) query (or queries).
ALL	Deletes all queries (AWAITING, ACTIVE, READY, HELD, or DISABLED). ■ When you specify ALL with QUERY name, the command deletes all queries of that name regardless of status. ■ If you do not specify ALL, only the first query with that name is deleted.
AWAITING	Purges the queries waiting for execution.
DISABLED	Purges disabled queries.
HELD	Purges queries being held.
READY	Purges queries ready for viewing.

Examples:

```
PURGE QUERY B C D AWAITING
```

(Deletes B, C, and D, if they are awaiting execution.)

```
PURGE ALL
```

(Deletes all of the user's queries.)

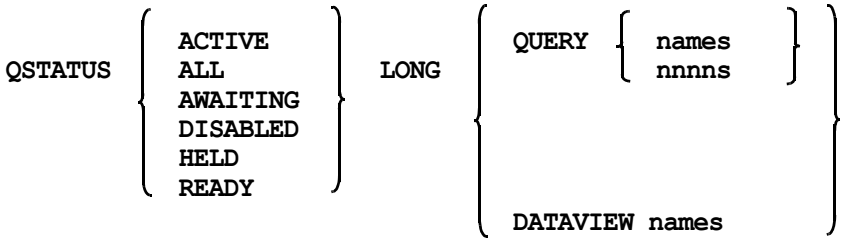
```
PURGE QUERY XYZ ALL
```

(Deletes all queries named XYZ regardless of status.)

```
PURGE ACTIVE
```

(Deletes all active queries.)

QSTATUS (VISION:Bridge)



In the system mode (?:), QSTATUS displays the status of queries submitted by this user ID.

ACTIVE	Lists queries that are currently being processed.
ALL	Lists all the queries for the user ID regardless of status. The keyword ALL is mutually exclusive with the keywords AWAITING, ACTIVE, READY, HELD, and DISABLED.
AWAITING	Lists queries that have not been selected for processing by a Background Processor.
DISABLED	Lists queries that have been disabled due to a problem encountered during execution.
HELD	Lists queries with HELD status because a required data view was unavailable.
READY	Lists queries that are completed and ready for viewing.
LONG	Displays a 2-line format, which includes the submit date and time, end date and time, and Background Processor name.

QUERY	{ names / nnnns }	Specifies the names or 4-digit task numbers of the queries to be listed. Mutually exclusive with the DATAVIEW keyword.
DATAVIEW names		Specifies the specific name or names of the data views for which queries are to be listed. Mutually exclusive with the QUERY keyword.

Examples:

```
QSTATUS AWAITING
QSTATUS QUERY ABCD
QSTATUS DATAVIEW CUSTOMER
```

QUERY (VISION:Bridge and Immediate Response)

QUERY DATABASE name

Initiates a query and specifies the database or logical data view to be used.

DATABASE	Optional keyword.
name	Name of the database or logical data view to be accessed.
END QUERY	Marks the end of a query group.

Examples:

```
QUERY DATABASE CUSTOMER
QUERY EXPXFD
```

QUIT (VISION:Bridge and Immediate Response)

QUIT

Terminates the Command Input or Immediate Response session and returns you to the Main Menu.

RATIO (VISION:Bridge)

Note: If you use the same field as the numerator in both the PCT and RATIO statements in a single report, then use the same denominator field name in both statements.

`RATIO ITEM names TO name BY name`

Reports a ratio of one field value to another.

ITEM names — The numerator fields in the ratio.

- Place these field names on the REPORT statement, but not on the GROUP statement.
- Do not use the same field as the numerator in multiple RATIO statements. You create temporary fields of equal value to accomplish this.
- ITEM is an optional keyword.

TO name — The denominator field for the ratio.

- List this field on the REPORT statement, but not on the GROUP statement.
- TO is a required keyword.

BY name — The name of a field from the GROUP statement to indicate the level at which summaries are to be taken. If you omit the BY keyword and name, the entire report is summarized.

Example:

```
RATIO FRTCOST TO INAMT
```

REPORT (VISION:Bridge)

```
REPORT    ITEM names IF logical expression,
          SUMMARY GRANDSUM EMPTYFLD value BLKLNSUP
```

Specifies what data outputs as a printed report or terminal display.

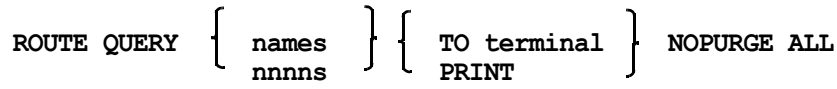
ITEM — Indicates literal data, built-in summaries, and names of fields to be output. The keyword ITEM is optional.

names	The field names, built-in summaries, or literal data output on the report. ■ Enclose literal data in single quotation marks. ■ Fields appear on the report in the sequence they are listed on this statement.
IF	See the section IF (ELSE) (VISION:Bridge).
logical expression	See the section Logical Expression (VISION:Bridge and Immediate Response).
SUMMARY	Suppresses printing of all detail lines and prints only summaries for those fields specified in AVG, COUNT, CUM, MIN, MAX, TOTAL, PCT, and RATIO statements.
GRANDSUM	Requests automatic grand summaries for the output report. You must specify at least one summary BY for the report.
EMPTYFLD value	Enter INCLUDE to include zero value numeric fields and blank character fields in calculating summaries. Enter EXCLUDE to exclude them from summary calculations.
BLKLNSUP	Suppress blank lines, such as those generated by the GROUP and ORDER statements.
END REPORT	This command indicates the end of a report group.

Examples:

```
REPORT CUSTNAME ORDERNO INAMT
REPORT ITEM CUSTNO IF INAMT GT 100
REPORT 'CUSTOMER NUMBER' CUSTNO
```

ROUTE (VISION:Bridge)



In the system mode (?:), ROUTE directs a query (or queries) to a specific destination.

QUERY { names / nnnns }

Specifies the names or numbers of the queries to be routed; QUERY is optional if the order shown is preserved.

{ TO terminal / PRINT }

TO terminal specifies the destination (logical terminal).

- If you omit TO, the originating terminal is assumed.
- PRINT specifies the system printer as the destination.
- TO and PRINT are mutually exclusive.

NOPURGE — Do not purge queries from the communication file after routing. If you omit NOPURGE, queries are purged from the communication file after routing.

ALL — Specifies that all queries be routed to the specified destination.

Examples:

```
ROUTE C D TO LT0001
ROUTE QUERY B PRINT NOPURGE
```

RUN (Immediate Response)

Note: *Statements in the* Immediate Response *immediate work area can be affected and new error messages produced by the combination of statements.*

RUN ITEM name
TYPE { **QUERY** / **STMTS** }

In system mode (?:), RUN specifies that a stored query (or collection of statements) and the statements currently in the Immediate Response immediate work area are combined and submitted for Immediate Response processing. Stored statements are appended to the IMMED statements.

ITEM name	Specifies the name of the query (QUERY) or collection of statements (STMTS) appended to the Immediate Response work area. You can omit ITEM, if you preserve the order shown.
TYPE { QUERY / STMTS }	Specifies the item as a query (QUERY) or collection of statements (STMTS). If you omit the keyword TYPE, TYPE QUERY is assumed.

Example:

```
RUN ITEM AUDIT TYPE QUERY
```

SELECT (VISION:Bridge and Immediate Response)

```
SELECT   FIELD names SEGMENT names,
         IF logical expression
```

Indicates selective access to the information in the database. See the section END SELECT (VISION:Bridge and Immediate Response).

FIELD names	Specify one or more field names indicating the level at which the database is entered. ■ The view is the same as that from the related segment level. ■ The keyword FIELD is optional if a field name appears first.
SEGMENT names	Specify one or more segment names indicating the level at which the database is entered. FIELD and SEGMENT specifications are optional. ■ You can use both in the same SELECT statement. ■ If neither is listed, the database is entered at the root level.
IF logical expression	You can specify an optional conditional phrase that further limits the range of data accessed. See the section Logical Expression (VISION:Bridge and Immediate Response)

Examples:

```
SELECT FIELD CUSTNO IF INAMT LT 500
SELECT SEGMENT ORDER
SELECT IF INAMT GT 1000
```

SET (VISION:Bridge and Immediate Response)

```
SET   FIELD field name TYPE  { CHAR   }
                             { FIXED  }
                             { PACKED }
                             { ZONED  }

      LENGTH integer DECIMALS integer

      VALUE initial-value
```

SET defines a temporary field and its attributes.

FIELD fieldname — Defines the name of the temporary field (one to eight characters; the first must be alphabetic).
The keyword FIELD is optional if a field name immediately follows SET. A field name is required.

TYPE { CHAR | FIXED | PACKED | ZONED } — Specifies the type of the temporary field: character (CHAR, the default), fixed-point binary (FIXED), packed decimal (PACKED), or zoned decimal (ZONED). The keyword TYPE is optional.

LENGTH integer — Specifies the length (in bytes) of the temporary field:

Type	Length (bytes)	Default
CHAR	1 - 255	16
FIXED	1 - 4	4
PACKED (VISION:Bridge)	1 - 15	8
PACKED (Immediate Response)	1 - 8	8
ZONED	1 - 15	15

DECIMALS integer — Specifies the number of decimal places assigned to the temporary field (numeric fields only). You can specify the values listed in the table starting on the next page. The default is zero.

Note: Decimal places exceeding the DECIMALS specification are truncated.

VALUE initial-value		Specifies the initial value assigned to the temporary field. **Note:** If VALUE does not specify a decimal point, regardless of whether DECIMALS is specified or not, the maximum number of whole number digits is limited to eight.
	CHAR	Enclose up to 15 characters in single quotation marks. The default is blanks.
	{ FIXED PACKED ZONED }	Specify an unsigned number that can have a decimal point. The default is zero.

Examples:

```
SET FIELD TEMPA TYPE CHAR LENGTH 5 VALUE 'TODAY'
SET FIELD TEMPB PACKED,
    LENGTH 4 DECIMALS 2 VALUE 32.15
SET FIELD TEMPC TYPE PACKED
```

The following table shows the SET statement value limits for numeric type fields.

Field Type	Length (in bytes)	Maximum Number of Digits	Maximum Decimal Places	Maximum Value
ZONED	1-15	Equal to the length of the field	Equal to the value up to a maximum of 9	All digits equal to 9
PACKED (VISION: Product)	1-15	(2*length) – 1	Equal to the maximum number of digits up to a maximum of 9	All digits equal to 9
PACKED (Immediate Response)	1-8	(2*length) – 1	Equal to the maximum number of digits up to a maximum of 8	All digits equal to 9
FIXED	1 2 3 4	3 5 7 10	3 5 7 9	127 32,767 8,388,607 2,147,483,647

STOP (Immediate Response)

STOP

Discontinues the Immediate Response display and returns to the previous mode. The message ENTER "GO" TO CONTINUE OR "STOP" TO STOP is issued in checkpoint mode each time the output page limit or database call limit is reached. The limits are specified in the user's profile.

SYSTEM (Immediate Response)

SYSTEM

Switches to the system mode (?:) from the Immediate Response immediate mode (IMMED:). This command does not affect the immediate work area. See the section END IMMED (Immediate Response).

TITLE (VISION:Bridge)

TITLE LINE 'text'

Specifies one to nine title lines for the report.

LINE	Optional keyword.
'text'	Character text of the title lines. Enclose each line of text in single quotation marks. Represent apostrophes by two single quotation marks.

Examples:

Statement	Output
`TITLE 'ITEMS ORDERED'`	`ITEMS ORDERED`
`TITLE 'ITEMS ORDERED''IN JANUARY'`	`ITEMS ORDERED` `IN JANUARY`
`TITLE 'ITEMS ORDERED'` `TITLE 'IN JANUARY'`	`ITEMS ORDERED IN JANUARY`

TOTAL (VISION:Bridge)

TOTAL ITEM fieldnames BY fieldname

TOTAL reports the total of all valid occurrences of the specified fields, at specified control breaks.

ITEM fieldnames — Specifies the name of the fields selected for totaling.

- You can omit ITEM if the field name appears immediately after TOTAL.
- List these fields on the REPORT statement, but not on the GROUP statement.

BY fieldname — Specifies the controlling field for the totals.

- The total is listed every time this field value changes.
- If you do not specify BY fieldname, the total is calculated for the entire report.
- You must also list this field on a GROUP statement.

Examples:

```
TOTAL ITEM INAMT BY CUSTNO
TOTAL INAMT
```

TOTAL (Immediate Response)

```
TOTAL   ITEM fieldnames BY fieldnames,
        GRAND,
        TITLE 'text' ORDER fieldnames DESC fieldnames,
        IF logical expression
```

TOTAL displays the grand total of all valid occurrences of the specified fields.

ITEM fieldnames — Specifies the name of the fields selected for totaling.

- You can omit ITEM if a field name appears immediately after TOTAL.
- Make the fields subordinate to any fields you specify in BY fieldname.

BY fieldnames	Specifies the name of the fields selected as control break fields for the totals. The total is listed every time this field changes. If you do not specify BY fieldname, grand summaries are produced.
GRAND	Specifies that grand summaries are to be produced.
TITLE 'text'	Specifies one or two title lines for each page of the display or printout. ■ The text appearing between the single quotation marks is the title. ■ To produce two title lines, enclose each title line in single quotation marks.
ORDER fieldnames	Specifies the field or fields by which the output data is sorted. The first field name is the first sort field, the second field name is the second sort field, and so on. The following rules apply to the field names used with ORDER: ■ Each field name can be either a database field or a temporary field already defined in the query. ■ Each field can be in the ITEM list. ■ Each database field must be in a database segment already selected by the query. ■ Each field name is grouped (duplicate values do not display). ■ Do not put a summarized field name in the ORDER list.
DESC fieldnames	Specifies the field names in the ORDER list to be sorted in descending order.
IF logical expression	Specifies the conditions under which the totals are taken. See the section Logical Expression (VISION:Bridge and Immediate Response).

Examples:

```
TOTAL ITEM INAMT BY CUSTNO
TOTAL INAMT IF CUSTNO > 11000
```

VIEW (VISION:Bridge)

```
VIEW QUERY { names   } { PURGE } { ALL }
           { numbers }
```

In system mode (?:), VIEW requests a query or queries for viewing.

QUERY	{ names / numbers }	Specifies the names or assigned numbers of the queries that produced the reports. ■ You obtain the assigned numbers with the QSTATUS command. ■ You can specify multiple queries for viewing.
PURGE		Specifies that reports are to be purged after viewing. If you do not specify PURGE, the output remains in the communication file after viewing.
ALL		Displays all available reports on your screen, one at a time. When you specify ALL with QUERY name, all reports with that name display on your screen, one at a time.

Examples:

```
VIEW QUERY AUDIT PURGE
VIEW ALL
VIEW 2443 3323
VIEW AUDIT ALL
```

Contacting Computer Associates

For technical assistance with this product, contact Computer Associates Technical Support on the Internet at esupport.ca.com. Technical support is available 24 hours a day, 7 days a week.

Index

Symbols

Numerics

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W