

# **SUPERDEX**

## **Fourth-Generation Language Interface**

User Manual

**Version 3.1**

All updates to or derivatives of the SUPERDEX computer software provided herein are copyrighted and may not be copied except for archival purposes, to replace a defective copy, or for program error verification by Licensee. Copyrighted material may not be copied onto any media (e.g. magnetic tape, paper tape, disc memory cartridges, read-only memory, etc.) for any other purposes. The authorization to duplicate copyrighted materials hereunder shall not be construed to grant the Licensee or Licensee's customer the right to use copyrighted SUPERDEX material in any manner other than which is provided in this agreement or otherwise approved in writing by Dr. Wolfgang Matt or Bradmark Technologies.

**(c) 1988 Bradmark Technologies, Inc.**

Released March, 1992

IMAGE, TurboIMAGE, TurboIMAGE/XL, and BRW are trademarks of Hewlett-Packard Company

PowerHouse, QTP, QUICK, and QUIZ are registered trademarks of Cognos Incorporated

Speedware is a registered trademark of Speedware Corporation.

SUPERDEX is a trademarked product name of Bradmark Technologies, Inc. for the SI-IMAGE package developed and implemented by Dr. Wolfgang Matt

## About this manual

In writing this manual, we have assumed that you have working knowledge, although not internal knowledge, of IMAGE and the HP3000, as well as the PowerHouse module or modules you need to interface to SUPERDEX.

All references to IMAGE in this manual and throughout the SUPERDEX package also apply to TurboIMAGE and TurboIMAGE/XL unless otherwise noted.

This manual is arranged in the following format:

**Section 1** provides an *Overview* of the SUPERDEX package, its capabilities, and benefits.

**Section 2** describes how SUPERDEX interfaces to Cognos' QUICK, QUIZ, and QTP, and the various access methods that are available for retrieving entries using SUPERDEX.

**Section 3** describes how SUPERDEX interfaces to Speedware's Speedware, and the access methods available for retrieving entries using SUPERDEX.

**Section 4** describes how the SUPERDEX interface works with Hewlett Packard's BRW package.



# Table of contents

<b>Section 1: Overview</b> .....	<b>1-1</b>
<b>Section 2: PowerHouse</b> .....	<b>2-1</b>
<b>PowerHouse Overview</b> .....	<b>2-2</b>
How the QUICK Interface works.....	2-3
How the QUIZ interface works.....	2-5
How the QTP interface works.....	2-6
Access methods.....	2-7
<b>Configuration</b> .....	<b>2-12</b>
Installation.....	2-13
Data dictionary changes for QUICK interface.....	2-14
KGO file changes for QUICK interface.....	2-16
<b>QUICK interface</b> .....	<b>2-17</b>
Invoking QUICK.....	2-18
Modifying screens for SUPERDEX access.....	2-19
Qualifying entries for retrieval.....	2-21
SIPARMS item.....	2-22
SICOUNT item.....	2-25
Sample QUICK screen.....	2-26
<b>QUIZ interface</b> .....	<b>2-29</b>
Invoking SIQUIZ.....	2-30
Modifying source files for SUPERDEX access.....	2-31
Qualifying entries for retrieval.....	2-33
Commands.....	2-34
SIDEF command.....	2-35
SIPATH command.....	2-37
Other SIQUIZ commands.....	2-40
Notes on QUIZ commands.....	2-42
Sample source files.....	2-43
<b>QTP interface</b> .....	<b>2-45</b>
Invoking SIQTP.....	2-46
<b>Maximum limits</b> .....	<b>2-47</b>
<b>Error and exceptional conditions</b> .....	<b>2-48</b>
<b>Section 3: Speedware</b> .....	<b>3-1</b>
<b>Section 4: BRW</b> .....	<b>4-1</b>



## **Section 1**

# **Overview**

The SUPERDEX Fourth-Generation Language interface was designed and implemented so that users of Fourth-Generation Languages can easily utilize SUPERDEX's lookup and retrieval capabilities.

Currently, this interface is documented for Cognos' PowerHouse package, Speedware Corp.'s Speedware package, and Hewlett Packard's BRW package.





## **Section 2**

# **PowerHouse**

This section discussing the interfaces to Cognos' QUIZ, QUICK, and QTP.

**Chapter 1 PowerHouse Overview**

**Chapter 2 Configuration**

**Chapter 3 QUICK Interface**

**Chapter 4 QUIZ Interface**

**Chapter 5 QTP Interface**

**Chapter 6 PowerHouse Maximum Limits**

**Chapter 7 PowerHouse Error and exception conditions**

## PowerHouse Overview

This chapter overviews the interface to QUICK, QUIZ, and QTP.

The first three parts describe How the QUICK interface works, How the Quiz interface works, and How the QTP interface works.

The last part overviews the various Access Methods available through SUPERDEX's Fourth-Generation Language Interface and PowerHouse.

# How the QUICK interface works

## Installation

The SUPERDEX SL or XL is copied into CURRENT.COGNOS. UDCs (including PHUDC.CURRENT.COGNOS), menus, job streams, etc... must be changed to run QUICK with ;LIB=G.

The QUICK interface requires that a dummy manual master dataset called **SIPROC** and its elements be added to the data dictionary for each database that will be accessed via SUPERDEX (for multiple bases, each SIPROC file is suffixed with a number). For each **SIPROC** that is added, **RECORD** and **ITEM** statements are also added. After changes, a new QSCHMAC is compiled.

The QUICK interface also requires that the QKGO file have an *expression size* of at least 200.

## Adding, updating, and deleting entries

QUICK is run through SUPERDEX's SL or XL, so that entries added, updated, and deleted from QUICK programs also cause the SI-indices to be updated automatically.

## Qualifying and retrieving entries

The QUICK interface works by reinterpreting the *SEQUENTIAL* option of the **GET** verb to perform sorted indexed-sequential SUPERDEX access rather than an *IMAGE* serial read. To accomplish this, only minor additions to QUICK screens are required, and no changes to default statements and procedures generated by QDESIGN are required.

In each QUICK screen, the statement **FILE SIPROC DESIGNER** to declare the dummy SIPROC file as a DESIGNER file and **ACCESS VIA SIPARMS** to define the access method, are added.

Also, a **POSTPATH** procedure is written which accesses the *PATH* designated for sequential access by QDESIGN and sets up the parameters for SUPERDEX access using standard **LET** statements. The dataset to access is defined in the *SIDSET* parameter; the SI-path is defined in the *SIPATH* parameter; and the *SIMODE* parameter specifies how entries should be qualified and whether they should be returned in ascending or descending order. The search value, partial or generic key, range of values, or multiple search values are specified with the desired operators in the *SIARG* parameter.

Because most of SUPERDEX's capabilities are accomplished by embedding multiple values and conditional, relational, and Boolean operators in the search argument, it is normally sufficient to simply vary the value of the *SIARG* and use an *SIMODE* of 1.

Entries are by default returned in ascending sorted sequential order, and may alternately be returned in descending order by altering the *SIMODE*. Other *SIMODE*s may be used to additionally retrieve entries that are alphabetically higher or lower than the *SIARG* value and to perform other specialized functions, such as returning the alphabetically highest or lowest entry in the dataset.

After defining the **SIPARMS**, the POSTPATH procedure calls `GET SIPROC USING SIPARMS OPTIONAL`.

The default QDESIGN-generated **FIND** procedure remains unchanged: SUPERDEX automatically redefines the access performed by the *SEQUENTIAL* option to be an indexed-sequential read via SUPERDEX rather than an IMAGE serial read. After entries are retrieved, the qualifying number of entries is returned in the **SICOUNT** item when *SIMODE* 1 is used.

# How the QUIZ interface works

## Installation

The SUPERDEX SL or XL is copied into CURRENT.COGNOS. UDCs (including PHUDC.CURRENT.COGNOS), menus, job streams, etc. must be changed to run SIQUIZ with an INFO parameter, to define the location of QUIZ, which then automatically invokes QUIZ with ;LIB=G to access the SUPERDEX SL or XL.

A front-end program called **SIQUIZ** is provided in PUB.SUPERDEX. It is run instead of QUIZ, and defines the QUIZ source file to **USE** and interprets two new SUPERDEX QUIZ interface commands which specify the SUPERDEX access to perform.

All QUIZ job streams should be modified to run QUIZ with ;LIB=G.

 The QUIZ interface requires no modifications to the data dictionary.

## Qualifying and retrieving entries

In SUPERDEX's QUIZ interface, entries are qualified and retrieved using the new **SIPATH** command, which specifies the dataset, SI-path, and search argument. It is used in place of or in addition to the **SELECT** command. The **CHOOSE** command is not used to access SI-paths, but continues to work as always on IMAGE search fields.

As in the QUICK interface, the search argument specified (in the SIPATH command) may contain partial and generic key values, ranges, multiple values, and conditional, relational, and Boolean operators.

For cases in which the user is prompted for the search criteria, the prompt is included in the SIPATH command, and the user's response may include multiple values and operators. The **SIDF** command may be used like QUIZ's DEFINE command to define and initialize variables for use in **SIPATH** commands.

For retrievals involving multiple SI-paths, multiple **SIPATH** commands are used. Retrievals involving multiple datasets also require the use of multiple SI-path commands, with each specifying the dataset to access. Entries are unconditionally returned in ascending sorted sequential order.

# How the QTP interface works


## Installation

Installation is the same as for the QUIZ interface.

The SUPERDEX SL or XL is copied into CURRENT.COGNOS. UDCs (including PHUDC.CURRENT.COGNOS), menus, job streams, etc. must be changed to run SIQTP with an INFO parameter, to define the location of QTP, which then automatically invokes QTP with ;LIB=G to access the SUPERDEX SL or XL.

A front-end program called **SIQTP** is provided in PUB.SUPERDEX. It is run instead of QTP, and defines the QTP source file to **USE** and interprets two new SUPERDEX QTP interface commands, which specify the SUPERDEX access to perform.

All QTP job streams should be modified to run QTP with ;LIB=G, even if they do not access SI-PATHS.

 **The QTP interface requires no modifications to the data dictionary.**

## Adding, updating, and deleting entries

QTP is run through SUPERDEX's SL or XL, so that entries added, updated, and deleted by QTP also cause the SI-indices to be updated.

## Qualifying and retrieving entries

The QTP interface uses the same commands and operates identically to the QUIZ interface.

## Access methods

### Multiple keys in master and detail datasets

**QUICK:** A **POSTPATH** procedure is written to set up the parameters for **SUPERDEX** access, including the **SI-path** to access, and **GET SPROC USING SIPARMS OPTIONAL** is called. When the retrieval is done via **GET...SEQUENTIAL** in the **QDESIGN**-generated **FIND** procedure, an indexed-sequential read is performed by **SUPERDEX** rather than a serial read.

**QUIZ and QTP:** The **SELECT** command is replaced by or works with an **SIPATH** command, which defines the **SI-path** to access and the value to search for. This causes **QUIZ** to perform an indexed-sequential read via **SUPERDEX** rather than a serial read. The **CHOOSE** command is still used for access by **IMAGE** search fields but is not used for **SUPERDEX** access.

### Concatenated keys containing multiple fields

**QUICK:** The **SIARG** item in the data dictionary is redefined as necessary to represent the **SI-subkeys** in the concatenated **SI-key**, their lengths, and their data types. The values for each **SI-subkey** are defined as separate fields in the **POSTPATH** procedure.

**QUIZ and QTP:** The **QUIZ** and **QTP** interfaces support only concatenated **SI-keys** that contain **SI-subkeys** of data types **X** or **U**. The value of each **SI-subkey** in the concatenated **SI-key** is assigned to a separate variable, which may be done by hard-coding or prompting using the **SIDEF** command. Then, these values are concatenated together in the **SIPATH** command.

### Sorted sequential retrieval

Entries are unconditionally returned in sorted sequential order for entries qualified in indexed access mode. In relational access mode, an **SI-link** may be specified in the **SIPATH** parameter to enforce sorting order.

**QUICK:** By default, entries that match the specified **SIARG** are returned in ascending order. By adding **2000** to the **SIMODE**, entries are instead returned in descending order. All the entries in a dataset may be read in ascending or descending sorted order by using **SIMODE 1100** or **3100**, respectively.

**QUIZ and QTP:** Entries are unconditionally returned in ascending sorted sequential order. To read all the entries in a dataset in ascending order, specify an **SIARG** of **@**. If a different sort is required, simply access the file serially and add the **SORT** command. This is typically much faster.

## Keyword retrieval

Keyworded SI-paths may be accessed through QUICK, QUIZ, and QTP without restriction.



**Keyword retrieval is available only in the SUPERDEX II package.**

## Generic and partial key retrieval

**QUICK:** Partial key access can be performed in SUPERDEX'd QUICK using two different methods, which are similar to the generic access methods used for KSAM files in QUICK.

The first is to use an *SIMODE* of 1 and specify the partial key value appended with an @ as the *SIARG*, e.g. **HEWL@**. This will locate all entries that match on the significant characters followed by anything.

The second method is to specify the value in the *SIARG* without an @ but vary the *SIMODE* based on the length of the value in the *SIARG*. For example, an *SIARG* containing the partial key **ROLA** would dictate *SIMODE* 102 or -104 (100 plus the number of words or bytes, respectively, in the value).

Generic key retrieval is accomplished by embedding the ? matchcode in the *SIARG*, which holds the place of any alphanumeric character. For example, the *SIARG* **L?TTER** would locate "LETTER" and "LITTER"; by appending an @ (**L?TTER@**), "LETTERMAN", "LITTERBUG", and "LOTTERY" would also be located.

**QUIZ and QTP:** Partial key retrieval is performed by specifying the partial key value appended with an @ in the **SIPATH** command. For generic key retrieval, one or more ? matchcodes are embedded in the search value, each one acting as a place holder for any alphanumeric character.

## Approximate match retrieval

**QUICK:** Approximate match retrieval is performed by using an *SIMODE* that specifies how many characters in the *SIARG* SUPERDEX should match on, which is typically the length of the value specified. If no matching entry exists, the nearest matching entry is returned.

The *SIMODE* also dictates whether the internal SI-pointer in the B-tree should be set before or after the matching or nearest matching entry, and whether entries should be returned in ascending or descending order.

For example, an *SIARG* containing the value **UNITED** would dictate *SIMODE* 1103 or -1106, both of which would cause SUPERDEX to match on the entire value. The *SIMODE* is calculated as 100 plus the number of words or bytes (negated if bytes), plus or minus 1000 to dictate ascending order retrieval. Using these *SIMODE*s, the SI-pointer would be set before the matching or nearest matching entry. Any entry beginning with "UNITED" would be included in the retrieval.



If *SIMODE* 3103 were instead specified (adding 3000 instead of 1000 for descending order), any "UNITED" entry would be excluded, since the SI-pointer is set before the matching entry and entries are being returned in descending order. With an *SIMODE* of 3203 or -3206 (200 plus the number of words or bytes, plus or minus 3000 for descending order retrieval), the SI-pointer would be set after the matching or nearest matching entry, and the "UNITED" entries would now be included.

Note that for compatibility with QUICK, if SUPERDEX does not find a matching entry and returns the nearest matching entry, **ACCESS <> OK** is returned, but the internal SI-pointer is still set. In this case, the program should be written to ignore this error.

**QUIZ and QTP:** Approximate match retrieval is not supported from either QUIZ or QTP.

### Greater-than, less-than, and range retrieval

**QUICK:** These retrievals are performed by embedding special operators in the *SIARG* for *SIMODE* 1 or 10.

Greater-than-or-equal-to retrieval is accomplished by suffixing the search value with the **>=** operator (e.g. **>=1000**), less-than-or-equal-to retrieval uses the **<=** operator as a suffix, and not-equal-to retrieval uses the **<>** operator. The **<>** operator can also appear after another value in the same *SIARG* to perform a Boolean AND NOT operation (e.g. **SUPER@<>SUPERDEX**).

Range retrievals are performed by using the **>=** and **<=** operators in combination. For example, a range search to find all the entries with amounts between 500 and 1000, inclusive, is specified with the *SIARG* **>=500<=1000**.

By default, entries are returned in ascending sorted order, but may be returned in descending order by adding 1000 to the *SIMODE*.

**QUIZ and QTP:** These retrievals are performed by embedding the same operators as described for the QUICK interface in the search criteria. They may either be specified by the user or hard-coded in the program. Entries are unconditionally returned in ascending order. If descending order is desired, add a "SORT item D" statement in the QUIZ or QTP program.

### Grouped retrieval

 grouped retrieval is available only in the SUPERDEX II package.

**QUICK:** Whenever the group is referenced by its SI-path name in the *SIPATH* parameter of *SIPROC*, all SI-keys that form the group are unconditionally searched.

Grouped SI-paths are accessed in the same way as non-grouped SI-paths--the only difference is in the configuration of the SI-path.

There may be some ambiguity in searching by an SI-key in a grouped SI-path whose item length is shorter than the group length and which is therefore padded with spaces. For example, if CITY, an X16, and STATE, an X2, are grouped together with an SI-key length of 8 words (to accommodate CITY), an *SIARG* of CA would find not only all entries in the state of "California" but also those in the cities of "CALABASAS" and "CARLSBAD". To resolve this ambiguity, use *SIMODE* 1 or 10 and pad the *SIARG* with enough trailing spaces to cover the full SI-key length.

**QUIZ and QTP:** Whenever the group is referenced by its SI-path name in the **SIPATH** command, all SI-keys that form the group are unconditionally searched.


### Super-grouped retrieval

 **Super-grouped retrieval is available only in the SUPERDEX II package.**

**QUICK:** To reference a super-grouped SI-path, use the IMAGE master set as the PRIMARY file. Quick will search all paths related to the super-grouped SI-path in the master and detail datasets, but will only display the master entries. To display the detail entries, reference the detail dataset as DETAIL or SECONDARY on the FILE statement.

**QUIZ and QTP:** To reference the master dataset, list it as the first file in the ACCESS statement. To reference the detail datasets, use the "LINK TO" option on the "ACCESS" statement. Whenever the super-group is referenced by its SI-path name in the **SIPATH** command, all SI-keys that form the group are unconditionally searched.

### Relational access: multiple criteria retrieval

 **Relational access retrieval capability using multiple fields, sets, and bases is available only in the SUPERDEX II package.**

**QUICK:** Boolean operations are accomplished by embedding the appropriate operators in the search value in the *SIARG*.


For example, a search for all part descriptions that contain both the words PAPER and CLIP would be specified with an *SIARG* of **[PAPER] [CLIP] &**. To find all invoices that are unpaid or canceled, the *SIARG* would be **[UNPD] [CANC] |**; and to find all entries in California and not Los Angeles, **[CA] [LOS ANGELES] ! &**.

**QUIZ and QTP:** Boolean operations may be specified in either of two methods, based on whether search values are hard-coded in the program or prompted for.

If hard-coded, the same syntax as in the QUICK interface is used.

If the user is instead prompted for the search criteria, the values may be specified using simpler syntax and operators. For example, to perform the search for all part descriptions that contain both the words PAPER and CLIP, the user could specify **PAPER+CLIP**. To find all entries in California and not Los Angeles, the user would specify **CA-LOS ANGELES**, and to find all invoices that are unpaid or canceled, the user would enter **UNPD,CANC**. If necessary, parentheses can be used to enforce a special order of evaluation.

## Relational access: multiple fields, sets, bases


 **Relational access retrieval capability using multiple fields, sets, and bases is available only in the SUPERDEX II package.**

**QUICK:** Multiple GET SIPROC USING SIPARMS OPTIONAL statements are performed in succession, with one GET per SI-path. If performing retrievals against multiple fields or sets within a database, the *SIDSET* and *SIPATH* parameters are varied to specify the desired access. If performing retrievals against multiple bases, SIPROC is suffixed with a number that specifies the base referenced in the data dictionary.

In the preceding example, four GETs would be performed in succession: one against the *PAID-FLAG* SI-path in the ORDER-HEADER set in the ORDERS base (referenced by SIPROC1), another against the *ORDER-TOTAL-AMT* SI-path, another against the *AVG-DAYS-TO-PAY* SI-path in the CUSTOMERS set, and the last against the dataset and SI-path of the same name but in the ORHIST database (referenced by SIPROC2).

**QUIZ and QTP:** Multiple SIPATH commands are used, with one for each SI-path, which specify both the SI-path and dataset to access.

As in QUIZ and QTP, it is required that if entries are retrieved from more than one dataset that the sets be **LINKED TO** and related by IMAGE paths.

 **Retrievals against multiple SIPATHs can only be issued against the database of the primary file.**

## Custom indexing

**QUICK:** The custom SI-path is referenced in the *SIPATH* parameter and the custom SI-key value is specified in the *SIARG* parameter.

**QUIZ and QTP:** The custom SI-path and custom SI-key value are specified in the **SIPATH** command.

# Configuration

This chapter describes the method for installing and configuring the SUPERDEX PowerHouse interface.

Refer to the **SUPERDEX User Manual** for information about installing SUPERDEX, establishing the SI-item and SI-dataset(s), defining SI-paths, and establishing B-trees and SI-indices.

This chapter assumes that you have already loaded SUPERDEX on your system, as described in the separate *SUPERDEX loading instructions*.

The first part discusses the *Installation* procedure for the PowerHouse interface.

Next, the *Data dictionary changes for QUICK interface* is described. If the QUICK interface will not be used, these changes are not required.

The last part describes the *QKGO file changes for QUICK interface*. These changes are also not required, if the QUICK interface will not be used.

# Installation

Once the SUPERDEX installation tape has been loaded, as described in the separate *SUPERDEX loading instructions*, several steps are necessary for completing the SUPERDEX installation.

Some of these steps are one-time operations, while others are required for future creation of users, accounts, programs, etc. The installation procedures may include:

- changes to the COGNOS account
- data dictionary changes for the QUICK interface
- QKGO file changes for the QUICK interface

## Changes to the COGNOS account

Add DS capability to QUIZ and QTP, as described previously.

Copy the SUPERDEX SL or XL into CURRENT.COGNOS.

Also, change the UDCs in PHUDC.CURRENT.COGNOS that run QUICK, QUIZ, and QTP as follows:

```
:RUN QUICK.CURRENT.COGNOS;LIB=G  
:RUN QUIZ.CURRENT.COGNOS;LIB=G  
:RUN QTP.CURRENT.COGNOS;LIB=G
```

Be sure to use these same commands in menus, job streams, etc.

## Data dictionary changes for QUICK interface

The QUICK interface requires a dummy manual master dataset to be added to the data dictionary for each database that will be accessed via SUPERDEX. To do so:

1. Obtain a current QSCHEMA.
2. Add a FILE statement for the dummy manual master dataset SIPROC:

```
FILE SIPROC          ORGANIZATION MASTER      &
  OF base            PASSWORD "password"    &
                   NOCREATE
```

where *base* is the name of the database containing the data entries and *password* is the user class to assign.

If SUPERDEX will be used with multiple databases, define multiple SIPROC files with multiple FILE statements and the OPEN name SIPROC, suffixing each dataset name with an integer starting with 1 (e.g. **SIPROC1**, **SIPROC2**, etc.), for example:

```
FILE SIPROC1        ORGANIZATION MASTER      &
  OF base1          PASSWORD "password"    &
OPEN SIPROC         NOCREATE

FILE SIPROC2        ORGANIZATION MASTER      &
  OF base2          PASSWORD "password"    &
OPEN SIPROC         NOCREATE
```

where *base1* and *base2* represent two separate databases which both contain entries to be accessed by SUPERDEX'd QUICK.

If using ARCHITECT to create the data dictionary and database, the NOCREATE option causes the FILE definition to be excluded altogether. In this case, exclude the NOCREATE keyword and instead specify a low capacity for the SIPROC file(s), as shown:

```
FILE SIPROC          ORGANIZATION MASTER      &
  OF base            PASSWORD "password"    &
                   CAPACITY 1
```

This will cause the SIPROC file to be created as a standalone master dataset with a capacity of 1.

3. Add **ELEMENT** statements for the elements used in **SIPROC**:

```
ELEMENT SIPARMS      X(200)
ELEMENT SIDSET       X(16)
ELEMENT SIPATH       X(34)
ELEMENT SIMODE       9(005)
ELEMENT SIARG        X(148)
ELEMENT SICOUNT      9(010)
```

4. Add a **RECORD** statement for each **SIPROC**:

```
RECORD SIPROC
  ITEM SIPARMS      CHARACTER SIZE 200 KEY
  REDEFINED BY
    ITEM SIDSET     CHARACTER SIZE 16
    ITEM SIPATH     CHARACTER SIZE 34
    ITEM SIMODE     INTEGER SIGNED SIZE 2
    ITEM SIARG      CHARACTER SIZE 148
  END
  ITEM SICOUNT      INTEGER SIGNED SIZE 4
```

**SIARG** may be further redefined to support concatenated SI-keys (which contain more than one SI-subkey), for example:

```
ITEM SIARG          CHARACTER SIZE 148
REDEFINED BY
  ITEM SIARG1       INTEGER SIGNED SIZE 4
  ITEM SIARG2       CHARACTER SIZE 144
END
```

5. Use QDD to compile the QSCHEMA into the QSCHEMAC data dictionary.

## QKGO file changes for QUICK interface

The QUICK interface requires that the *expression size* in the QKGO file be at least **200**. To do so:

1. :SETCATALOG the PHUDC UDC file from the COGNOS account.

2. Invoke the SETQKGO UDC:

**SETQKGO**

3. The HP3000 QKGO Construction and Maintenance Screen is displayed. Specify

**F**

in the ACTION field and press RETURN.

Then, specify the QKGO filename (usually QKGO) and press RETURN.

Then, enter

**3**

in the ACTION field and press RETURN.

4. The Execution-time parameter values screen is displayed. If the Expression size is less than **200**, specify

**3**

in the ACTION field and press RETURN.

Then, specify the value

**200**

and press RETURN. Then, press **f8** to return to the first screen.

5. Back on the HP3000 QKGO Construction and Maintenance Screen specify

**UR**

in the ACTION field and press RETURN. Then, press **f8** to exit.



## QUICK interface

This section discusses SUPERDEX's interface to QUICK. It is assumed that you have:

- loaded the tape according to the separate SUPERDEX loading instructions
- installed the PowerHouse interface as explained in the Installation chapter of the Configuration section
- enhanced the data dictionary and assured that the *expression size* in the QKGO file is at least 200, as specified in the Configuration section

The first part in this chapter explains how to go about Invoking QUICK to access SUPERDEX capabilities.

The next part describes the process of Modifying screens for SUPERDEX access in order to access entries using SUPERDEX.

The third part explains the differences in Qualifying entries for retrieval between the QUICK interface and SUPERDEX's DBFIND intrinsic.

Part four describes the SIPARMS Item, which is used to define the parameters for SUPERDEX access.

Next, the SICOUNT item is discussed, which is used to return the number of qualifying entries.

The last part gives a before and after view of Sample QUICK screens modified for SUPERDEX access.

## Invoking QUICK

After copying the SUPERDEX SL or XL into CURRENT.COGNOS and establishing the necessary configuration in the data dictionary, as explained in the *Configuration/Establishing SI-indices* section, QUICK should be run as follows:

```
:RUN QUICK.CURRENT.COGNOS [ ; INFO="parameter" ] ; LIB=G
```

Where the **INFO="parameter"** is optional.

### Invoking QUICK from ARCHITECT

If invoking QUICK from within ARCHITECT, do not use function 24; instead, include the QUICK UDC in the Toolbox to run QUICK with ;LIB=G.

### Invoking QUICK from QDESIGN

If invoking QUICK from within QDESIGN, do not use the **GO** command, since it invokes QUICK with ;LIB=S rather than ;LIB=G. Instead, use the QUICK UDC by typing

```
:QUICK
```

as shown, including the leading colon.

## Modifying screens for SUPERDEX access

The QUICK interface reinterprets the *SEQUENTIAL* option of the **GET** verb to perform sorted indexed-sequential SUPERDEX access instead of an IMAGE serial read. Only minor additions to QUICK screens are required to accomplish this. No changes to the default statements and procedures generated by QDESIGN are required.

For every QUICK screen that requires SUPERDEX access, the following statements must be added:

```
FILE SIPROC DESIGNER
  ACCESS VIA SIPARMS
```

Also, a POSTPATH procedure must be written which accesses the PATH designated for sequential access and sets up the parameters for SUPERDEX access (**SIPARMS**):

```
PROCEDURE POSTPATH
  BEGIN
    IF PATH = 2
      THEN BEGIN
        LET SIDSET = "CUSTOMERS "
        LET SIPATH = "CUSTOMER-NAME "
        LET SIMODE = 1
        LET SIARG = "ACME MANUFACTURING "
        GET SIPROC USING SIPARMS OPTIONAL
        IF NOT ACCESSOK
          THEN
            ERROR "No records were found matching key values."
          END
        END
      END
```

The **SIPARMS** are defined as follows:

<b>SIDSET</b>	name of the dataset containing entries
<b>SIPATH</b>	name of the SI-path to utilize
<b>SIMODE</b>	defines the method of SUPERDEX access
<b>SIARG</b>	the search argument

The values for **SIPARMS** are documented fully for reference in the *SIPARMS item* chapter in this section.

In the example above, the customer "ACME MANUFACTURING" is being searched for in the CUSTOMERS dataset via the SI-PATH *CUSTOMER-NAME*. Note that a blank space specified as a terminator (alternately, @ could have been used).

In this POSTPATH procedure, **PATH = 2** because sequential access is done via PATH 2, as designated by the default QDESIGN-generated **FIND** procedure (access via the IMAGE master search field is done through PATH 1). When **GET...SEQUENTIAL** is subsequently called for PATH 2 in the FIND procedure, SUPERDEX access is automatically performed instead.

If multiple databases are configured in the data dictionary, specify the name of the SIPROC file that references the corresponding database, for example:

```
GET SIPROC1 USING SIPARMS OPTIONAL
```

The primary file is specified with **OPEN $n$**  to force a separate access path. Use the **SIPROC $n$**  on the **GET** the matches the number on the **OPEN $n$** .

 If you are using version 6.09 and are having problems getting any records to qualify, add the **NOGENERIC** clause to the **GET** statement in the POSTPATH procedure.

## Qualifying entries for retrieval

In QUICK, entries in IMAGE datasets are generally retrieved either by key (using **GET...VIA**) or via a serial read (using **GET...SEQUENTIAL**). Fast access is normally attained only when specifying the full, exactly-matching key value for an entry.

Using SUPERDEX, QUICK is able to rapidly qualify multiple entries based on various criteria, and may be used to refine the selection using various fields, datasets, and databases.

Entries qualified in QUICK may be retrieved in ascending or descending sorted sequential order without the need to sort.

### Differences between the QUICK interface and SUPERDEX's DBFIND intrinsic

Examples of various types of retrievals are documented in the *Programming* section of the **SUPERDEX User Manual**.

The main difference between retrievals using the QUICK interface and SUPERDEX's DBFIND intrinsic is that the terminology is altered. The following table shows the equivalent QUICK interface terminology vs. DBFIND:

<b>QUICK</b>	<b>DBFIND</b>	<b>description</b>
<b>SIDSET</b>	<i>dset</i>	dataset
<b>SIMODE</b>	<i>mode</i>	mode of operation (see below for discussion)
<b>SIPATH</b>	<i>item</i>	SUPERDEX SI-path (optionally followed by <b>SI-LINK</b> )
<b>SIARG</b>	<i>argument</i>	search value

In the QUICK interface, the **TRIM** function is useful for constructing complicated SIARG search values.

### Special SIMODE values

To cause SUPERDEX'ed QUICK to return all the entries that are alphabetically greater-than the *SIARG* rather than just the entries that qualify based on the *SIARG* specified, add **1000** to or subtract 1000 from the *SIMODE* value.

By default, entries are returned by SUPERDEX'ed QUICK in ascending alphabetical order. To instead return entries in descending order, add **2000** to or subtract 2000 from the *SIMODE* value.

To cause SUPERDEX'ed QUICK to return all the entries that are alphabetically less-than the *SIARG*, add or subtract 3000 to or from the *SIMODE* value.

## SIPARMS item

The parameters for SUPERDEX access are defined in the POSTPATH procedure, using standard **LET** statements.

### Parameters

The dataset to access is defined in the *SIDSET* parameter, and the SI-path is defined in the *SIPATH* parameter. The *SIMODE* specifies how entries should be qualified and whether they should be returned in ascending or descending order. The search value, partial or generic key, range of values, or multiple search values are specified with the desired operators in the *SIARG* parameter.

The parameters of **SIPARMS** have the following meaning:

**SIDSET**            The **OPEN** name of the dataset (which may differ from the **FILE** name). It must be enclosed in quotes and, if less than 16 characters, appended with a single space.

**SIPATH**            The name of the SI-path to access. It must be enclosed in quotes and, if less than 16 characters, appended with a single space.

In performing successive **GET SIPROC...**s against multiple datasets, a common item used to logically link the files together (called the *SI-link*) may be additionally specified. It is required that the item assigned as the SI-link be configured as an SI-subkey in a concatenated SI-key; alternately, for SI-paths against a master dataset, it may be the IMAGE search master field.


The SI-link is separated from the SI-path name by a comma, with the combined value terminated by a SPACE or ; as shown:

*SI-path,SI-link,*

If performing a *projection*, which is used to logically link two files that do not contain a common item by reassigning the SI-link, the *SIPATH* parameter takes the form:

*SI-path,new SI-link,*

To locate entries that have been found by previous **GETs** in the active SI-subset, specify a value of **0 ;**.

- SIMode** Same as the *mode* parameter used in SUPERDEX's DBFIND intrinsic, but additional *modes* are also supported.
- SIMODE 0** Resets to sequential access. The *SIPATH* and *SIARG* parameters are ignored and do not need to be specified.
- SIMODE 1** Compatible with IMAGE DBFIND *mode 1*, but also allows multiple values and various operators to be included in the *SIARG* (explained later). If no entry is found that matches the *SIARG*, zero entries are returned.
-  Only SIMODE 1 returns the number of qualifying entries in the SICOUNT item.
- SIMODE 10** Same as *SIMODE 1*, but does not return the number of qualifying entries in the *SICOUNT* item and is therefore more efficient.
- In *SIMODE 1*, not only is the SI-pointer set to the first qualifying SI-index entry, but the B-tree is traversed to locate all qualifying SI-indices in order to set the value of *SICOUNT*. It is more efficient to only set the SI-pointer, so *SIMODE 10* should be used in place of *SIMODE 1* whenever the number of qualifying entries is not required.
- In *SIMODE 10*, the qualifying entry count in *SICOUNT* is always set to 1.
- SIMODE 100** Positions the SI-pointer before the first SI-index in the B-tree (i.e. before the lowest alphabetical entry in the file in ascending sorted sequential order). In this *SIMODE*, the *SIARG* is ignored and may be left blank.
- SIMODE 1nn** Reads *nn* words of the *SIARG* and sets the SI-pointer **before** the first qualifying SI-index entry. If no matching entry exists, zero entries are returned but the SI-pointer is set immediately before the nearest-matching SI-index. If prefixed with a minus sign (-), reads *nn* bytes instead of words.
- SIMODE 200** Positions the SI-pointer after the last SI-index in the B-tree (i.e. after the highest alphabetical entry in the file in descending sorted sequential order). In this *SIMODE*, the *SIARG* is ignored and may be left blank.
- SIMODE 2nn** Reads *nn* words of the *SIARG* and sets the SI-pointer **after** the last qualifying entry. If no matching entry exists, zero entries are returned but the SI-pointer is set immediately after the nearest-matching SI-index. If prefixed with a minus sign (-), reads *nn* bytes instead of words.
- SIMODE n+1000** By default, only entries that match the specified *SIARG* are returned. By adding 1000 to or subtracting 1000 from the *SIMODE*, all entries greater-than or equal-to the *SIARG* will be returned in ascending order.
- SIMODE n+2000** By default, entries are returned in ascending sorted sequential order. By adding 2000 to or subtracting 2000 from the *SIMODE*, entries will instead be returned in descending order.
- SIMODE n+3000** Same as *SIMODE +1000*, but returns entries less-than or equal-to the *SIARG* in descending order.

## SIARG

Value or values to search for. The *SIARG* value used in conjunction with *SIMODEs 1 and 10* may contain:

- an exact SI-key value (or concatenated value)
- a partial SI-key value with an @ appended (the wildcard operator may optionally be redefined from @ to a different character using SIMAINT)
- a generic SI-key value containing one or more embedded ?s (the matchcode operator may optionally be redefined from ? to a different character using SIMAINT)
- a partial SI-key value preceded by either the >=, <=, or <> relational operators
- a range of two or more values with embedded relational operators (e.g. >=A@<=B@)
- an ASCII value prefixed by ==, which causes the value to be converted to binary for comparison
- multiple SI-key values, delimited with square brackets ([ ]), and including one or more of the Boolean operators &, |, or !&, specified in Reverse Polish Notation
- the special operator ^, used for swapping the active and backup SI-subsets
- the special operators / and/or \ used for manipulating the active SI-chain and backup SI-chain
- the special operator \*, used for projection
- the special operator @@, which returns the entry count of a virtual SI-chain many combinations of these constructs



## SICOUNT item

In addition to the input parameters used in SIPARMS, a separate item **SICOUNT** is used as an output parameter and returns the number of entries which were qualified by **GET SIPROC...** if called using *SIMODE 1*.

## Sample QUICK screen

This example shows a simple QUICK program that accesses the manual master EMPLOYEES. At execution, if CUSTOMER-NUMBER (the key) is specified, the IMAGE path is used; if a value for another field is entered, the dataset is read sequentially.

```
SCREEN CUST  
  
FILE CUSTOMERS PRIMARY  
  
FIELD CUSTOMER-NUMBER  
FIELD CUSTOMER-NAME  
FIELD STATE  
FIELD PHONE-PREFIX
```

Below, the **FILE SIPROC DESIGNER** and **ACCESS VIA SIPARMS** statements have been added, as well as a **POSTPATH** procedure to additionally facilitate SUPERDEX access using one of two SI-paths or IMAGE access via a serial read:

```
SCREEN CUST

FILE CUSTOMERS PRIMARY
FILE SIPROC DESIGNER
  ACCESS VIA SIPARMS

FIELD CUSTOMER-NUMBER
FIELD CUSTOMER-NAME
FIELD STATE
FIELD PHONE-PREFIX

PROCEDURE POSTPATH
  BEGIN
    IF PATH = 2
      THEN BEGIN
        PROMPT CUSTOMER-NAME
        IF NOT PROMPTOK
          THEN LET PATH = 0
        ELSE BEGIN
          LET PATH = 2
          LET SIDSET = "CUSTOMERS "
          LET SIPATH = "CUSTOMER-NAME "
          LET SIMODE = 1
          LET SIARG = CUSTOMER-NAME + "@"
          GET SIPROC USING SIPARMS OPTIONAL
          IF NOT ACCESSOK
            THEN
              ERROR "No records with matching CUSTOMER-NAME."
          END
        END
      END

    IF PATH = 0
      THEN BEGIN
        PROMPT STATE
        IF NOT PROMPTOK
          THEN LET PATH = 0
        ELSE BEGIN
          LET PATH = 2
          LET SIDSET = "CUSTOMERS "
          LET SIPATH = "STATE "
          LET SIMODE = -102
          LET SIARG = STATE
          GET SIPROC USING SIPARMS OPTIONAL
          IF NOT ACCESSOK
            THEN
              ERROR "No records with matching STATE."
          END
        END
      END
    END
  END
```

```
IF PATH = 0
  THEN BEGIN
    LET PATH = 2
    LET SIDSET = "CUSTOMERS "
    LET SIMODE = 0
    GET SIPROC USING SIPARMS OPTIONAL
  END
END
```

In the example, SUPERDEX access is permitted by CUSTOMER-NAME and STATE using the SI-paths of the same names.

The screen works as follows:

1. CUSTOMER-NUMBER is prompted for. If specified, the IMAGE path is used.
2. If CUSTOMER-NUMBER is not specified, CUSTOMER-NAME is prompted for. If specified, the **SIPARMS** parameters are set to access the appropriate SI-path, and an @ is appended to the name specified to unconditionally perform a partial-key retrieval.
3. If CUSTOMER-NAME is not specified, STATE is prompted for. The **SIPARMS** parameters establish the correct SI-path and set the *SIMODE* to -102, which specifies that only the first two bytes of the value are recognized.
4. If no value was entered for CUSTOMER-NUMBER, CUSTOMER-NAME, or STATE, a sequential search is performed. For this, the SI-path is reset (by specifying an *SIMODE* of 0), since it may have been set by a previous **GET**.

## QUIZ interface

This chapter discusses SUPERDEX's interface to QUIZ. It is assumed that you have:

- loaded the tape according to the separate SUPERDEX loading instructions
- installed the PowerHouse interface as explained in the Installation chapter of the Configuration section

The first part in this chapter explains how to go about Invoking QUIZ to access SUPERDEX capabilities.

The next part describes the process of Modifying source files for SUPERDEX access in order to access entries using SUPERDEX.

The next part explains the differences in QUALIFYING entries for retrieval between the QUIZ interface and SUPERDEX's DBFIND intrinsic.

The next part overviews the new SIQUIZ Commands, as well as the impact of SIQUIZ on existing QUIZ commands.

The following three parts describe the new SIDEF command, used for defining and initializing SIQUIZ variables, the SIPATH command, used for defining the dataset, SI-Path, and prompt or argument; and other SIQUIZ commands which act like QUIZ commands.

The next part describes the impact on, considerations for, and other NOTES on existing QUIZ commands.

The last part shows before and after Sample source files modified for SUPERDEX access through SIQUIZ.

## Invoking SIQUIZ

The **SIQUIZ** program is run as a front-end to QUIZ. This is done as follows:

```
:RUN SIQUIZ.PUB.SUPERDEX;INFO="parameter[ parameter][ parameter]"
```

where *parameter* is one or more of those listed below, delimited by spaces. Strings other than the following are passed to QUIZ literally.

**AUTO**=*mpefile*

Equivalent to executing the statements:

```
USE mpefile  
GO  
EXIT
```

where *mpefile* is a QUIZ source or program file.

**PROCLOC**=*.group[.account]*

Specifies the group and account in which the QUIZ program resides, normally CURRENT.COGNOS.

**SUSPEND**

Leaves QUIZ in a suspended state rather than terminating it. This copy is reactivated if the USE file does not change. **This parameter should not be used under MPE/XL.**

## Modifying source files for SUPERDEX access

The QUIZ interface uses the new **SIPATH** command to define the parameters for SUPERDEX access: the dataset, SI-path, and search argument. It is used in place of or in addition to the **SELECT** command.

For example, the following QUIZ report:

```
ACCESS CUSTOMERS
SELECT IF CUSTOMER-NAME = "ACME MANUFACTURING"
REPORT CUSTOMER-NUMBER CUSTOMER-NAME
GO
EXIT
```

would be modified for SUPERDEX's QUIZ interface as shown:

```
;SIPATH CUSTOMER-NAME "ACME MANUFACTURING"
ACCESS CUSTOMERS
REPORT CUSTOMER-NUMBER CUSTOMER-NAME
GO
EXIT
```

In the example above, the customer "ACME MANUFACTURING" (defined as the *argument*) is being searched for in the CUSTOMERS *dataset* via the *SI-path* CUSTOMER-NAME.

The parameters defined by the **SIPATH** command are as follows:

<i>dataset</i>	name of dataset containing entries (optional if same set as in <b>ACCESS</b> statement)
<i>SI-path</i>	name of SI-path to utilize
<i>argument</i>	search argument

The values for these parameters, along with the **SIPATH** command is documented fully in the *SIPATH command* chapter at the end of this section.

Additionally, the **SIDEF** command may be used like QUIZ's **DEFINE** command to define and initialize variables for use in **SIPATH** commands.

For example, the following QUIZ report:

```
ACCESS CUSTOMERS
DEFINE CUST-TEMP CHAR*30 = PARM PROMPT "Enter customer name:"
SELECT IF CUSTOMER-NAME = CUST-TEMP
REPORT CUSTOMER-NUMBER CUSTOMER-NAME
GO
EXIT
```

would be modified for SUPERDEX's QUIZ interface as shown:

```
;SIDEF CUST-TEMP PROMPT "Enter customer name:"
;SIPATH CUSTOMERS.CUSTOMER-NAME CUST-TEMP "@"
ACCESS CUSTOMERS
REPORT CUSTOMER-NUMBER CUSTOMER-NAME
GO
EXIT
```

This example will accept the input from the user with the **SIDEF** command. It will then append the entered data with "@" using the **SIPATH** command. This will force a partial lookup, without the user needing to specify the "@".



## Qualifying entries for retrieval

In QUIZ, entries in IMAGE datasets are generally retrieved either by key (using **CHOOSE**) or via a serial read (using **SELECT**). Fast access is normally attained only when using **CHOOSE**, for which the full, exactly-matching key value for an entry must be specified.

Using SUPERDEX, QUIZ is able to rapidly qualify multiple entries based on various criteria, and may be used to refine the selection using various fields, datasets, and databases.

Entries qualified in QUIZ may also be retrieved in ascending sorted sequential order without the need to sort.

### Differences between the QUIZ interface and SUPERDEX's DBFIND intrinsic

Examples of various types of retrievals are documented in the *Programming* section of the SUPERDEX User Manual.

SIPATH Command	DBFIND	Description
<b>DATASET</b>	<i>Dset</i>	Dataset (Optional in QUIZ)
<b>SIPATH</b>	<i>Item</i>	SUPERDEX SIPATH (Optionally followed by SI-link)
<b>PROMPT TEXT</b>	None	Displayed text
<b>ARGUMENT</b>	<i>Argument</i>	Search value (QUIZ hardcoded)
<b>Search Criteria</b>	<i>Argument</i>	Search value (QUIZ reply from prompt)
None	<i>Mode</i>	Mode of operation (Mode 10 used in QUIZ)

**Restrictions:** Relational retrieval using multiple sets can be used to restrict the number of qualifying entries from the primary file, based on search criteria specified for **LINKed** files in the same database. The **LINKed** files are accessed by QUIZ through normal IMAGE access. It is therefore necessary to keep the **SELECT** commands for **LINKed** files.

## Commands

Two new commands, which are similar to QUIZ commands, are provided in SIQUIZ to facilitate SUPERDEX access against SI-paths:

- SIDF** similar to QUIZ's **DEFINE** command, but used only for SUPERDEX access on an SI-path
- SIPATH** similar to QUIZ's **SELECT** command, but used only for SUPERDEX access on an SI-path

Additional SIQUIZ commands work the same in SIQUIZ as they do in QUIZ:

- USE** specifies the QUIZ source file to execute
- GO** invokes the **USEd** QUIZ source file
- EXIT** ends and suspends the program
- QUIT** ends and terminates the program

The new **SIDF** and **SIPATH** commands may affect the use of the following QUIZ commands:

- CHOOSE** may not be used to access SI-paths, only IMAGE paths (use **SIPATH** instead)
- DEFINE** may not be used for SI-paths, only IMAGE paths (use **SIDF** instead)
- SELECT** may be eliminated or modified to perform access not done by SUPERDEX's **SIPATH** command
- SORT** may be eliminated, since entries are returned in sorted order

# SIDEF command

The **SIDEF** command is used to define the contents of a variable for use in an **SIPATH** command. Two different formats are allowed: one to prompt for the value to be assigned to the variable, and the other to instead assign (hard-code) the value.

**Syntax**           **SIDEF** *variable* **PROMPT** "*prompt text*"  
  **R****P****P****R****O****M****P****T**  
  or  
  **SIDEF** *variable* "*expression*"

In the first form above, the value of *prompt text* is displayed on the terminal and the user replies with *search criteria*, which is assigned to *variable*. If the search criteria do not fit on one line, the user may terminate the line with an **&** preceded by a **SPACE**, which causes a continuation prompt to be issued.

In the second form, the hard-coded value of *expression* is assigned to the *variable* as a constant.

The **SIDEF** parameters are defined as follows:

**VARIABLE**           An arbitrary variable name to which the *search criteria* specified by the user or hard-coded *expression* is assigned.

The value of the variable may be dynamically reassigned, for example:

```
SIDEF CUST-NAME PROMPT "Enter customer name:"  
SIDEF CUST-NAME ">=" CUST-NAME "@"
```

would prompt for the value to assign to the variable **CUST-NAME** in the first command, and prefix that value with the **>=** operator and suffix it with the **@** operator (to perform a greater-than-or-equal-to retrieval using a partial key) in the second command.

**PROMPT**               This specifies that the user will supply the search criteria. If **R****P****R****O****M****P****T** is used,  
**R****P****R****O****M****P****T**             a relational access for the search criteria will be enforced.

**PROMPT TEXT**       Issued as a user prompt with the response passed as the *search criteria* and assigned to the *variable*.

**SEARCH  
CRITERIA**

The value or values to search for on the referenced SI-path, in response to the *prompt text*. This is similar to the *argument*, but some constructs are not supported and an easier method is provided for users to specify multiple values for relational access.

Refer to the discussion of search criteria in the *SIPATH command* chapter in this section for more information.

**EXPRESSION**

String constant to assign to the referenced variable. May also consist of multiple string constants, enclosed in quotes, and variables previously defined using the **SIDEF** command.

Can also contain the QUIZ reserve word SYSDATE, with an arithmetic expression(+ number of days or - number of days). The variable would contain a valid date in the YYMMDD ASCII format

When referencing a variable previously defined in a **SIDEF** command, substrings can be specified. For example:

```
;SIDEF TODAY SYSDATE  
;SIDEF YEARMONTH TODAY [1:4]  
;SIDEF MONTH TODAY [3:2]
```

The format for substring designation is [*START POSTION:LENGTH*]



The SIPATH parameters are defined as follows:

- DATASET** Optional parameter. Name of the dataset to access, specified only for retrievals against a dataset other than the primary dataset (the first one specified in the QUIZ ACCESS command).
- SI-PATH** The name of the SI-path to use for the retrieval.
- SI-LINK** Optional parameter. The name of an item that appears in multiple datasets which is used to logically link the sets, or to impose sorting order in relational access mode. The SI-link item must be configured as an SI-subkey in a concatenated SI-key or a search field for an SI-path related to a master dataset.
- PROMPT** This specifies that the user will supply the search criteria. If **R**PROMPT is used, relational access for the search criteria will be enforced.
- R**PROMPT
- PROMPT TEXT** Issued as a user prompt with the response passed as the *search criteria*.
- ARGUMENT** String expression which may include constants and/or variables (created using the **S**IDEF command) concatenated together and delimited by a single space.

The *argument* may contain:

- an exact SI-key value (or concatenated value)
- a partial SI-key value with an @ appended (the wildcard operator may optionally be redefined from @ to a different character using SIMAINT)
- a generic SI-key value containing one or more embedded ?s (the matchcode operator may optionally be redefined from ? to a different character using SIMAINT)
- a partial SI-key value preceded by either the >=, <=, or <> relational operators
- a range of two or more values with embedded relational operators (e.g. >=A@<=B@)
- an ASCII value prefixed by ==, which causes the value to be converted to binary for comparison
- multiple SI-key values, delimited with square brackets ([ ]), and including one or more of the Boolean operators &, |, or !&, specified in Reverse Polish Notation
- the special operator \*, used for projection
- many combinations of these constructs

- SEARCH CRITERIA** The value or values to search for on the referenced SI-path, in response to the *prompt text*. This is similar to the *argument*, but some constructs are not supported and an easier method is provided for users to specify multiple values for relational access.

## Boolean

Boolean operations between multiple values may be specified using the following Boolean operators:

Boolean operators	
+	and
/	or
-	and not

The Boolean operators are specified as separators with no spaces and optionally delimited with parentheses, as shown:


**(A@+B@) , (P@-PR@)**

This locates all the entries in a keyworded SI-path with words that begin with "A" and "B", or that begin with "P" and not "PR".

In this example, the parentheses are required because expressions are normally evaluated from left to right.

## Multiple SI-paths retrieval

For retrievals that require the use of multiple SIPATH commands, the *search criteria* of all except the last SIPATH command must be terminated by a Boolean operator (+, /, or -) specifying how the retrieval by the current SI-path is to be related to the following one.

- Restrictions:** Two additional restrictions exist in the *search criteria* that may be specified:
- for range searches as well as searches that use values that start with the <=, >=, or <> relational operators, an ? embedded in a value is not recognized as an operator but as a regular character
  - values for alphanumeric SI-keys (data types U and X) must be appended with @ when specified with the relational operators <=, >=, or <>
-  For complete restrictions, see Section 4 of the SUPERDEX User Manual.

## Other SIQUIZ commands

### USE command

Once the SIPATH command has been issued (either via the SIQUIZ program or embedded in the source program), SIQUIZ's **USE** command specifies which QUIZ source file should be executed:

```
>USE QUIZ-program
```

where *QUIZ-program* is the name of the QUIZ source file.

Lines in the QUIZ source file may not be more than 72 characters in length. Additionally, nested USE files may not contain SIQUIZ commands--all SIQUIZ commands must be contained in the first USE file.

### GO command

After executing the USE command, the **GO** command invokes the referenced QUIZ source file:

```
>GO
```

This command has no parameters.

### EXIT command

After the QUIZ program has executed, the **EXIT** command may be used to exit and terminate the SIQUIZ program:

```
>EXIT
```

This command may be abbreviated to **EXI**, **EX**, or **E**.



## QUIT command

After the QUIZ program has executed, the **QUIT** command may be used to exit and suspend the SIQUIZ program:

```
>QUIT
```

## Notes on QUIZ commands

### **CHOOSE command**

QUIZ's **CHOOSE** command works only on IMAGE paths and may not be used to access SUPERDEX SI-paths--the **SIPATH** command must be used instead.

### **DEFINE command**

QUIZ's **DEFINE** command works only on IMAGE paths and may not be used to set up definitions for SUPERDEX SI-paths--the **SIDEF** command must be used instead.

### **SELECT command**

The **SELECT** command is either replaced by or modified for SIQUIZ's **SIPATH** command.

If the **SIPATH** command is functionally equivalent to the **SELECT** command, the **SELECT** command may be removed and replaced by the **SIPATH** command; otherwise, the **SELECT** command is ignored.

If the required access cannot be accomplished by one or more **SIPATH** commands, the **SELECT** command should be modified to perform only the additional access needed.

If the required access is against a non-primary (**LINKed TO**) file, the **SELECT** command should be modified to perform this access. An **SIPATH** command against a non-primary file simply de-selects primary entries which do not have at least one qualifying secondary entry.

### **SORT command**

Because entries on an SI-path which are qualified using the **SIPATH** command are automatically returned in ascending sorted order, any corresponding **SORT** command may be eliminated from or replaced by a **SORTED** command in the QUIZ source file. You may continue to use the **SORT** command to print or save the records in descending order.

## Sample source files

This example shows an unmodified QUIZ source file that accesses the manual master EMPLOYEES using the field *LAST-NAME*, which is not a key:

```
ACCESS EMPLOYEES
DEFINE EMPLOYEE-LAST CHARACTER*30 = PARM PROMPT "Enter last name:"
SELECT IF LAST-NAME = EMPLOYEE-LAST
REPORT FIRST-NAME LAST-NAME STATE
GO
EXIT
```

This report is modified for SUPERDEX access in SIQUIZ by replacing the **SELECT** command with an **SIPATH** command that accesses the SI-path named *LAST-NAME-KEY*, prefixed with a semicolon, as the first command in the source file:


```
;SIPATH LAST-NAME-KEY PROMPT "Enter last name:"
ACCESS EMPLOYEES
REPORT FIRST-NAME LAST-NAME STATE
GO
EXIT
```

This example is of a QUIZ program that additionally accesses the HOURS-WORKED dataset:

```
ACCESS EMPLOYEES LINK TO HOURS-WORKED
DEFINE EMPLOYEE-LAST CHARACTER*30 = PARM PROMPT "Enter last name:"
DEFINE WEEK-ENDING DATE = "Enter end day of week:"
SELECT IF LAST-NAME = EMPLOYEE-LAST AND DATE = WEEK-ENDING
REPORT FIRST-NAME LAST-NAME STATE
GO
EXIT
```

Modifying for SQUIZ requires two **SIPATH** commands and a change to the **ACCESS** command. Now, only the **LINKed TO** file is specified by the **ACCESS** command, with the other dataset included in the first **SIPATH** command:

```
;SIPATH EMPLOYEES.LAST-NAME-KEY PROMPT "Enter last name:"  
;SIPATH HOURS-WORKED.WEEK-ENDING PROMPT "Enter end day of week:"  
ACCESS EMPLOYEES  
REPORT FIRST-NAME LAST-NAME STATE  
GO  
EXIT
```

 The user must append an + to the search argument for the first prompt to inform SUPERDEX that an AND operation is to be performed with the second prompt.

## QTP interface

This section discusses SUPERDEX's interface to QTP. It is assumed that you have:

- loaded the tape according to the separate SUPERDEX loading instructions
- installed the PowerHouse interface as explained in the Installation chapter of the Configuration section

The only part in this chapter describes how to go about Invoking SIQTP.

Because the QTP interface operates identically to the QUIZ interface, refer to the QUIZ Interface chapter for details.

## Invoking SIQTP

The **SIQTP** program is run as a front-end to QTP. This is done as follows:

```
:RUN SIQTP.PUB.SUPERDEX;INFO="parameter[ parameter][ parameter]"
```

where *parameter* is one or more of the strings defined in the QUIZ interface section, or passed literally to QTP.

### Operating SIQTP

The SIQTP program is used in the same way as SIQUIZ, as explained in the previous section.

## Maximum limits

The following table identifies SUPERDEX's internal limits when used with the PowerHouse interface. Most limits are not checked, and results when exceeded are unpredictable.

### SUPERDEX PowerHouse interface maximum limits

<i>Facility</i>	<i>Maximum limit</i>
Datasets per database with SI-paths	198
SI-paths per database - average SI-index length <30 words	400
SI-paths per database - average SI-index length 99 words	270
SI-paths per dataset - database BLOCKMAX 512 words	22
SI-paths per dataset - database BLOCKMAX 1024 words	53
SI-key length	127 words
SI-subkey length	63 words
SI-indices per SI-dataset - average SI-index length 10 words	102,400,000
SI-indices per SI-dataset - average SI-index length 30 words	68,266,000
SI-indices per SI-dataset - average SI-index length 99 words	20,686,000
Keywords per simple SI-key or per first concatenated SI-subkey	16
Words excluded from keywording - keyword length 4 words	18,000

## Error and exceptional conditions

### SIQUIZ and SIQTP errors and warnings

The **SIQUIZ and SIQTP errors and warnings** table lists the various error messages that could be issued by the SIQUIZ or SIQTP programs, their meanings, and their corrective actions.

#### SIQUIZ and SIQTP errors and warnings

Messages prefixed with **\*E\*** are designated as errors, while those prefixed with **\*W\*** are warnings.

<i>Type</i>	<i>Description</i>
<i>Message</i>	<b>*W* COMPILED REPORT, PLEASE ENTER SOURCE FILE</b>
<i>Meaning</i>	A compiled report was USEed and no SIPATH statement was specified in SIQUIZ/SIQTP.
<i>Action</i>	Because SIPATH statements embedded in source files are dropped when they are compiled, either specify an SIPATH statement in SIQUIZ/SIQTP or USE a source file rather than a compiled report.
<i>Message</i>	<b>*W* COMPILED REPORT, UNABLE TO SPECIFY SIPATH</b>
<i>Meaning</i>	A compiled report file was specified with the AUTO option.
<i>Action</i>	Specify a source file instead.
<i>Message</i>	<b>*E* FILE XXXXXXXX NOT FOUND</b>
<i>Meaning</i>	The specified source file does not exist.
<i>Action</i>	Make sure entered correctly. Qualify with group and account if necessary
<i>Message</i>	<b>*E* MISSING VARIABLE</b>
<i>Meaning</i>	An SIPATH command references a variable that was not already initialized by an SDEF command.
<i>Action</i>	Variables must first be initialized by the SDEF command.
<i>Message</i>	<b>*W* NO PATH DEFINED</b>
<i>Meaning</i>	The source file does not contain an SIPATH statement, and no SIPATH statement was issued in SIQUIZ/SIQTP.
<i>Action</i>	First specify an SIPATH statement either in the source file or in SIQUIZ/SIQTP.



**SIQUIZ and SIQTP errors and warnings (cont'd)**

<i>Type</i>	<i>Description</i>
<i>Message</i>	<b>*E* NO REPORT TO RUN</b>
<i>Meaning</i>	GO was entered without a previous USE statement.
<i>Action</i>	First specify the file to USE.
<i>Message</i>	<b>*W* NO SIPATH STATEMENT FOUND IN XXXXXXXX</b>
<i>Meaning</i>	The source file USEed does not contain an SIPATH statement, and no SIPATH statement was issued in SIQUIZ/SIQTP.
<i>Action</i>	If the correct file name was specified, enter the appropriate SIPATH statement in SIQUIZ/SIQTP and GO.
<i>Message</i>	<b>*E* QTP CANNOT BE RUN</b>
<i>Meaning</i>	SIQTP cannot create the QTP process, typically because <b>PROCLOC=</b> has not been specified.
<i>Action</i>	Correct the condition that caused the error and rerun SIQTP.
<i>Message</i>	<b>*E* QUIZ CANNOT BE RUN</b>
<i>Meaning</i>	SIQUIZ cannot create the QUIZ process, typically because <b>PROCLOC=</b> has not been specified.
<i>Action</i>	Correct the condition that caused the error and rerun SIQUIZ.
<i>Message</i>	<b>*E* TOO MANY SIDEFs</b>
<i>Meaning</i>	More than 16 SIDEF statements have been entered.
<i>Action</i>	Do not exceed the maximum limit.
<i>Message</i>	<b>*E* UNKNOWN STATEMENT</b>
<i>Meaning</i>	An unrecognized statement has been specified.
<i>Action</i>	Specify only valid SIQUIZ/SIQTP statements; check your spelling.



## **Section 3**

# **Speedware**

This section discussing the interfaces to Speedware Corp.'s Speedware product.

**Chapter 1 Speedware Overview**

**Chapter 2 Configuration**

**Chapter 3 Interface**

**Chapter 4 Speedware Maximum Limits**



## Section 4

# BRW interface

### Installation

REXEC.PUB.SYS (under MPE/V) or BRWEXEC.PUB.SYS (under MPE/XL) is copied to a group/account where the SUPERDEX SL or XL resides and run with ;LIB=G or ;LIB=P. Menus, job streams, etc... must be changed accordingly.

### Configuration

A special pseudo-dataset called **SIPROC** is used to provide SUPERDEX access from within BRW. It is defined with a key that specifies the required parameters for SUPERDEX access.

An oddity in the interface is that BRW unfortunately disallows special characters to be included in search arguments (even if they are included in data values) and therefore SUPERDEX operators are rejected. For this reason, it is necessary to place the search arguments in a separate file to be accessed by BRW rather than specifying them in BRW directly. We recommend using a file named **ARGFILE** for this purpose.


The required parameters for SUPERDEX are set up as items within BRW. Define the following items with the attributes shown:

<i>Item</i>	<i>Source Table</i>	<i>Type</i>	<i>Display Size</i>
SIARG	ARGFILE	S	148
SIARG	SIPROC	S	148
SICOUNT	SIPROC	I	9
SIDSET	SIPROC	S	16
SIMODE	SIPROC	I	4
SIPARMS	SIPROC	S	200
SIPATH	SIPROC	S	34

If BRW will be used to access concatenated SI-keys, the **SIARG** item should be redefined as required (i.e. SIARG1, SIARG2, etc.) to represent the SI-subkeys that comprise the concatenated SI-key.

Set up the file that contains the search argument, the SIPROC pseudo-dataset, and the dataset that contains the data entries with the attributes shown:

<i>Source Table</i>	<i>Location</i>	<i>Access Sequence</i>	<i>Key</i>
ARGFILE	ARGFILE	1	SIPARMS
SIPROC	<i>base</i>	2	
<i>dataset</i>	<i>base</i>	3	

 No key is defined for the dataset that contains the entries, which normally implies a serial read. This will automatically be redefined as an indexed-sequential read along an SI-path by SUPERDEX, and therefore a key should never be specified for any set which will be accessed by SUPERDEX.

### Qualifying and retrieving entries

To qualify and retrieve entries, assign the following values to the items defined for SUPERDEX:

SIARG	DBFIND search <i>argument</i>
SIDSET	dataset that contains the entries
SIMODE	DBFIND <i>mode</i>
SIPATH	SI-path name

The following item is an output parameter:

SICOUNT	number of qualifying entries
---------	------------------------------

The tables and order of access defined causes BRW to access the **ARGFILE** file and retrieve the search argument, then access the **SIPROC** pseudo-dataset, which establishes the SUPERDEX parameters to use, and then perform an indexed-sequential search of the dataset using SUPERDEX rather than a sequential read. The number of qualifying entries is returned in the **SICOUNT** item.

If BRW does not find any entries through SUPERDEX, it performs a sequential read because there is no current path defined. For this reason, it is recommended that an **SIMODE** other than 1 be used, which will prevent the sequential read.

## INDEX

### A

ACCESS $\leftrightarrow$ OK .....	2-9
Access methods .....	2-7
Approximate match retrieval	
QTP .....	2-9
QUICK .....	2-8
QUIZ .....	2-9
ARCHITECT	
QUICK .....	2-18
Ascending order	
QTP .....	2-7
QUICK .....	2-7, 2-21
QUIZ .....	2-7
AUTO= .....	2-30

### B

BRW Interface .....	4-1
---------------------	-----

### C

Concatenated keys .....	2-7
Custom indexing	
QTP .....	2-11
QUICK .....	2-11
QUIZ .....	2-11

### D

Definition	
SIARG .....	2-24
SIDSET .....	2-22
SIMODE .....	2-23
SIPATH .....	2-22
Descending order	
QTP .....	2-7, 2-42
QUICK .....	2-7, 2-21
QUIZ .....	2-7, 2-42

### E

Errors	
SIQTP .....	2-48
SIQUIZ .....	2-48
Example	
QUICK .....	2-26

### F

FILE	
SIPROC .....	2-3
FILE statement	
QUICK .....	2-19

### G

Generic retrieval	
QTP .....	2-8
QUICK .....	2-8
QUIZ .....	2-8
GET statement	
QUICK .....	2-21
Greater-than retrieval	
QTP .....	2-9
QUICK .....	2-9
QUIZ .....	2-9
Grouped retrieval	
QTP .....	2-10
QUICK .....	2-9
QUIZ .....	2-10
grouping	
mismatched SI-key lengths .....	2-10
operation .....	2-10

### I

INFO	
QUICK .....	2-18
QUIZ .....	2-30
Invoking QUICK .....	2-18
Invoking QUIZ .....	2-30

### K

Keyword retrieval .....	2-8
-------------------------	-----

### L

Less-than retrieval	
QTP .....	2-9
QUICK .....	2-9
QUIZ .....	2-9
LIB=G	
QTP .....	2-6
QUICK .....	2-3
QUIZ .....	2-5

<b>M</b>	
Maximum limits.....	2-47
Modifications	
QDD.....	2-19
Screens.....	2-19
Multiple Criteria retrieval	
QUICK.....	2-10
Multiple criteria retrieval	
QTP.....	2-10
QUIZ.....	2-10
Multiple databases	
QUICK.....	2-20
Multiple fields, sets, bases retrieval	
QTP.....	2-11
QUICK.....	2-11
QUIZ.....	2-11

<b>O</b>	
OPENn.....	2-20

<b>P</b>	
Partial retrieval	
QTP.....	2-8
QUICK.....	2-8
QUIZ.....	2-8
POSTPATH.....	2-3, 2-7
QUICK.....	2-19
Projection	
QUICK.....	2-22
PROMPT.....	2-38

<b>Q</b>	
QDD	
Modifications.....	2-19
QDESIGN	
Invoking QUICK.....	2-18
QKGO	
Expression size.....	2-3
QTP.....	2-45
Access methods.....	2-7
Approximate match retrieval.....	2-9
Ascending order.....	2-7
Concatenated keys.....	2-7
Custom indexing.....	2-11
Descending order.....	2-7
Generic retrieval.....	2-8
Greater-than retrieval.....	2-9
Grouped retrieval.....	2-10
Less-than retrieval.....	2-9
LIB=G.....	2-6
Partial retrieval.....	2-8

Range retrieval.....	2-9
Relational access	
Multiple criteria retrieval.....	2-10
Multiple fields, sets, bases retrieval.....	2-11
SIPATH.....	2-7
Super-grouped retrieval.....	2-10
QTP interface	
overview.....	2-6
Qualifying entries	
QUICK.....	2-21
QUIZ.....	2-33
QUICK	
ACCESS ⇔ OK.....	2-9
Access methods.....	2-7
Approximate match retrieval.....	2-8
ARCHITECT.....	2-18
Ascending order.....	2-7, 2-21
Concatenated keys.....	2-7
Custom indexing.....	2-11
Descending order.....	2-7, 2-21
Example screen.....	2-26
FILE statement.....	2-19
Generic retrieval.....	2-8
GET statement.....	2-21
Greater-than retrieval.....	2-9
Grouped retrieval.....	2-9
INFO.....	2-18
Invoking.....	2-18
QDESIGN.....	2-18
Less-than retrieval.....	2-9
Multiple databases.....	2-20
Partial retrieval.....	2-8
POSTPATH.....	2-7, 2-19
Projection.....	2-22
Qualifying entries.....	2-21
Range retrieval.....	2-9
Relational access	
Multiple Criteria.....	2-10
Multiple fields, sets, bases retrieval.....	2-11
Screen modifications.....	2-19
SI-Link.....	2-22
SICOUNT.....	2-25
SIMODE.....	2-21
SIPARMS.....	2-19, 2-22
SIPROC.....	2-7
SL or XL.....	2-18
Super-grouped retrieval.....	2-10
TRIM function.....	2-21
QUICK interface	
overview.....	2-3
Quick interface.....	2-17



## Q (Continued)

QUIZ .....	2-42
Access methods .....	2-7
Approximate match retrieval .....	2-9
Ascending order .....	2-7
CHOOSE .....	2-42
Concatenated keys .....	2-7
Custom indexing .....	2-11
DEFINE .....	2-42
Descending order .....	2-7
Examples .....	2-43
Generic retrieval .....	2-8
Greater-than retrieval .....	2-9
Grouped retrieval .....	2-10
INFO .....	2-30
AUTO= .....	2-30
Less-than retrieval .....	2-9
LIB=G .....	2-5
Modifying source .....	2-31
Partial retrieval .....	2-8
Qualifying entries .....	2-33
Range retrieval .....	2-9
Relational access	
Multiple criteria retrieval .....	2-10
Multiple fields, sets, bases retrieval .....	2-11
SELECT .....	2-42
SIDEF .....	2-5, 2-35
Expression .....	2-36
Prompt text .....	2-35
Search Criteria .....	2-36
Variable .....	2-35
SIPATH .....	2-5, 2-7, 2-37
Argument .....	2-38
Boolean operators .....	2-39
Dataset .....	2-38
Multiple path retrieval .....	2-39
Prompt text .....	2-38
Search criteria .....	2-38
SI-Link .....	2-38
SI-Path .....	2-38
SORT .....	2-42
Super-grouped retrieval .....	2-10
QUIZ interface	
overview .....	2-5

## R

Range retrieval	
QTP .....	2-9
QUICK .....	2-9
QUIZ .....	2-9
Relational access	
Multiple criteria retrieval	
QTP .....	2-10
QUIZ .....	2-10
Multiple fields, sets, bases retrieval	
QTP .....	2-11
QUICK .....	2-11
QUIZ .....	2-11
Retrieval	
Sorted .....	2-7
RPPROMPT .....	2-38

## S

Screen modifications	
QUICK .....	2-19
SI-Link	
QUICK .....	2-22
SI-link .....	2-7
SIARG	
Definition .....	2-24
SICOUNT .....	2-4
QUICK .....	2-25
SIDEF .....	2-5
Expression .....	2-36
Prompt text .....	2-35
QUIZ .....	2-35
Search Criteria .....	2-36
Variable .....	2-35
SIDEF;Substrings .....	2-36
SIDSET	
Definition .....	2-22
SIMODE	
Definition .....	2-23
Mode 0 .....	2-23
Mode 1 .....	2-23
Mode 10 .....	2-23
Mode 100 .....	2-23
Mode 1nn .....	2-23
Mode 200 .....	2-23
Mode 2nn .....	2-23
Mode n+1000 .....	2-23
Mode n+2000 .....	2-23
Mode n+3000 .....	2-23
QUICK .....	2-21

## S (Continued)

SIPARMS .....	2-3
QUICK .....	2-19, 2-22
SIPATH .....	2-5, 2-7
Argument .....	2-38
Boolean operators .....	2-39
Dataset .....	2-38
Definition .....	2-22
Multiple path retrieval .....	2-39
Prompt text .....	2-38
QUIZ .....	2-37
Search criteria .....	2-38
SI-Link .....	2-38
SI-Path .....	2-38
SIPROC .....	2-3, 2-7
SIPROCh .....	2-20
SIQTP .....	2-6, 2-46
Errors and warnings .....	2-48
INFO= .....	2-6
Invoking .....	2-46
SIQUIZ .....	2-5, 2-40
Errors and warnings .....	2-48
EXIT .....	2-40
INFO= .....	2-5
Invoking .....	2-30
QUIT .....	2-41
USE .....	2-40
Sorted retrieval .....	2-7
Super-grouped retrieval	
QTP .....	2-10
QUICK .....	2-10
QUIZ .....	2-10
SUPERDEX II .....	2-8, 2-10, 2-11
SYSDATE .....	2-36

## T

TRIM function	
QUICK .....	2-21

## W

Warnings	
SIQTP .....	2-48
SIQUIZ .....	2-48