

DBFIND

INTRINSIC NUMBER 404

Can be used both when B-Tree index exists on a key item of the master and when B-Tree index does not exist. For a detailed discussion on B-Trees, refer to chapter 11, “B-Tree Indices.”

In the absence of a B-Tree index, or when the B-Tree index exists but the `BTREEMODE1` flag is off, locates related master set key item entry that matches the specified search item value and sets up pointers to the first and last entries of a detail data set chain in preparation for chained access to the data entries which are members of the chain. The path is determined and the chain pointers are located on the basis of a specified search item and its value.

When a B-Tree index exists for the master (explicit), `DBFIND` can be used for B-Tree searches both for the master data set as well as its corresponding detail data sets (implicit). The `dset` parameter determines if the `DBFIND` is for the master or one of its corresponding detail data sets. A B-Tree `DBFIND` using mode 1 on binary items (not X or U) will be treated as non-B-Tree search regardless of the presence of a B-Tree index as well as the `BTREEMODE1` option. To do B-Tree index searches on binary items, use modes 4 or 24 in conjunction with a structured argument.

OPENTURBO vs TurboIMAGE Difference

100% - Mode 10 (Not yet certified by TPI vendors); new Mode 80, Mode 88 and Mode 89 are used for specifying ORACLE search condition in free SQL form.

OPENTURBO Performance Enhancements

Ignore status words 3 through 10, this option triggers subsequent `DGBET(5/6)` in `NO-CURSOR`, `BULK-FETCH` and `PRE-FETCH` mode.

- Mode = 80 (SQL SELECT condition clause – user must provide ‘WHERE’)
- Mode = 88 (SQL SELECT condition clause without ‘WHERE’. Un-restricted condition clause syntax will be supported for Mode = 89, in A.02.01 official release)
- ITEM = an VALID ITEM NAME must be provided (PRIMARY PATH ITEM NAME)
- ARGUMENT contains the WHERE condition clause, but ‘WHERE’ itself is not included.
 - All operators must be UPPER CASE and must be surrounded by BLANKS
 - Allowed operators are =, >, <, >=, <=, <>, LIKE, AND, OR
 - ORDER BY is also supported; the ‘ORDER BY ‘ must be UPPER CASE and must be at the last of the WHERE clause.
 - No special characters are allowed in the STRING literal
 - Use ORACLE column name, not TurboIMAGE field name; OpenTURBO doesn’t parse nor translate them, simply attach it to the WHERE clause
 - Example: **FLIGHT_KEY LIKE ‘20030915%’ OR FLIGHT_KEY LIKE ‘20030914%’ =>** FLIGHT_KEY is the column name, not the TurboIMAGE item name FILGHT-KEY, the FLIGHT_KEY must be an INDEX for table COUPON, using LIKE is much faster than `SUBSTRING()` aggregate function.
 - The entire ARGUMENT must be terminated by a NULL char which is HEX 0, ‘\0’ for C/C++ or X”00” for COBOL.

OPENTURBO Additional Features

Exact, wildcard, range, relational operators search are all supported by default, no need for special settings.

- Mode = 88 (Un-restricted condition clause syntax will be supported for Mode = 89, in A.02.01 official release)
- ITEM = an VALID ITEM NAME must be provided (PRIMARY PATH ITEM NAME)
- ARGUMENT contains the WHERE condition clause, but 'WHERE' itself is not included.
 - All operators must be UPPER CASE and must be surrounded by BLANKS
 - Allowed operators are =, >, <, >=, <=, <>, LIKE, AND, OR
 - ORDER BY is also supported; the 'ORDER BY ' must be UPPER CASE and must be at the last of the WHERE clause.
 - No special characters are allowed in the STRING literal
 - Use ORACLE column name, not TurboIMAGE field name; OpenTURBO doesn't parse nor translate them, simply attach it to the WHERE clause
 - Example: **FLIGHT_KEY LIKE '20030915%' OR FLIGHT_KEY LIKE '20030914%'** => FLIGHT_KEY is the column name, not the TurboIMAGE item name FILGHT-KEY, the FLIGHT_KEY must be an INDEX for table COUPON, using LIKE is much faster than SUBSTRING() aggregate function.
 - The entire ARGUMENT must be terminated by a NULL char which is HEX 0, '\0' for C/C++ or X"00" for COBOL.

Syntax

DBFIND, base, dset, mode, status, item, argument

Parameters

base is the name of the array used as the *base* parameter when opening the database. The first element of the array must contain the base ID returned by DBOPEN. (Refer to DBOPEN for more information about base ID.)

dset is the name of an array containing the left-justified name of the master or the detail data set to be accessed, or is an integer referencing the data set by number. The data set name can be up to 16 characters long. If shorter, it must be terminated by a semicolon or a blank.

mode must be an integer equal to 1, 4, 10, 21, or 24. If your database is enabled for third-party indexing, refer to your vendor's documentation for additional DBFIND modes.

Mode	Method
-------------	---------------

1	Used for both B-Tree or non-B-Tree DBFIND. Perform a B-Tree DBFIND if ALL of the following conditions prevail:
---	--

Item type is X or U (ASCII).

DBFIND is for master and a B-Tree index exists on the key item or DBFIND is for detail and B-Tree index exists on the key item of the master to which the search item has a path.

BTREEMODE1 flag is set ON by DBUTIL or DBCONTROL mode 15.

Argument contains a wildcard character.

The chain count is accurate in halfword 5-6, and status halfwords 7-8 and 9-10 record the last entry in the last chain, and the first entry in the first chain of the super-chain for detail data sets. For master data sets, the status halfwords 5-6 (chain-count for detail) reflect the total number of entries qualified in the master data set. All other fields of the status array return zeroes.

4 For B-Tree searches on numeric as well as ASCII types. For detail data sets, returns accurate chain (super-chain) counts and record numbers of last entry in last chain and first entry in first chain. For masters, returns the total number of qualified entries in status halfword 5-6. Other fields return zeroes. Requires a structured argument format described under argument.

10 Allows you to simulate the current DBFIND mode 1, as in versions prior to B-Trees, as if there were no B-Tree index, even when the item has B-Tree index, BTREEMODE1 is on, and the argument contains a wildcard character. It returns accurate chain count. This is the same as TPI mode 10. Requires a simple argument.

21 Same as mode 1, except it is a faster version and does not return accurate chain count or record numbers of first entry and last entries. Requires a simple argument.

For details, the chain count and status halfwords 7-8 and 9-10 will have 231-1. For master data sets, the status halfwords 5-6 (chain-count for detail) reflect the total number of entries qualified in the master data set. All other fields of the status array return zeroes. A DBFIND mode 21 on a non-ASCII key returns an error.

24 Same as mode 4, except it is a faster version and does not return accurate chain counts and record numbers of last entry and first entry in super-chain. For detail data set, the halfwords 5-6, 7-8, and 9-10 will have 231-1. For master data set, the halfword 5-6 will have 231-1 and 7-8 and 9-10 will have zeroes. The argument is in structured format described under argument.

The third-party indexing (TPI) modes of DBFIND which are NOT supported for B-Trees are: 11, 12, and various TPI DBFIND modes 1nn, 2nn, 3nn, 4nn, and 5nn. If your database is enabled for TPI, refer to your vendor documentation for additional DBFIND modes. The section on DBUTIL in chapter 8 of this book has a brief description of the TPI option.

status is the name of an array of 10 halfwords in which TurboIMAGE/XL returns status information about the procedure. If the procedure executes successfully, the status array contents are:

Element	Contents
1	If the procedure succeeds, the return status is 0. Table 5-11. describes the contents of element 1 when the procedure does not succeed.
2	Zero.

- 3-4 Word current record number set to zero.
- 5-6 Word count of number of entries in chain or qualifier. For modes 21 and 24, $2^{31}-1$.
- 7-8 Word record number of last entry in chain. For modes 21 and 24, $2^{31}-1$ for details. Always zeroes for masters.
- Word record number of first entry in chain. For modes 21 and 24, $2^{31}-1$ for details. Always zeroes for masters.

item is the name of an array containing a left-justified name of the detail data set search item or is an integer referencing the search item number that defines the path containing the desired chain. The specified search item defines the path to which the chain belongs. For a B-Tree DBFIND on a master, it is the key item name or a number. The name can be up to 16 characters long. If shorter, it must be terminated by a semicolon or a blank.

argument contains a value for the key item to be used in calculated access to locate the desired chain head in the master data set for non-B-Tree index searches. This is referred to as a **simple argument**. For B-Tree index searches, there are two formats. One is for mode 1 wildcard searches on X and U types and allows a trailing wildcard character. The argument is scanned for the first occurrence of the wildcard character, for example, an @ character. Subsequent text in the item value is ignored. The other is for modes 4 and 24, which requires a **structured format** as follows:

- | | |
|----------------|---|
| 3-4 | version number. Must be numeric zero, or an error is returned. |
| 5-6 | The size (in bytes) of argument1 (not including these two bytes) for search types =, \llcorner <, \llcorner <=, >, \llcorner >=, @c, PK. |
| 7-8 | The size (in bytes) of argument2 (not including these two bytes) for the search type []. Must be numeric zero for other search types, or an error is returned. |
| 9... 9+n-1 | Argument1. The n bytes of argument data (for example, for an X10 field, $n = 10$). |
| 9+n... 9+n+m-1 | Argument2. For search-type [] only. The m bytes of the second argument's data (for example, for an X10 field, $m = 10$, n must match m). |

If a wildcard character is present in the argument(s), the wildcard will be considered as part of the value for these B-Tree search types: = \llcorner , < \llcorner , <=, > \llcorner , >=, [], or PK on ASCII types.

If your database is enabled for third-party indexing (TPI), refer to your vendor documentation for information on DBFIND arguments. The section on DBUTIL in chapter 8 of this book has a brief description of the TPI option.

Discussion

When the DBFIND is not for a B-Tree index, the current values of chain count, backward pointer, and forward pointer for the detail data set referenced in *dset* are replaced by the corresponding value from the chain head.

A current path number, which is maintained internally, is set to the new path number and the current record number for the data set is set to zero. Refer to chapter 10 for further information about chain heads and internally maintained data set information.

Although a master set entry with the specified key item value exists, the data set chain may be empty.

For B-Tree DBFINDs, there could be multiple qualified chain heads. The chained access could traverse multiple chains, referred to as **super-chain**. When using modes 1 or 4 for B-Tree DBFINDs, the qualified chain heads are retrieved, and the chain count becomes a cumulative chain count of the super-chain. The record number of last entry is obtained from the last entry of the last chain and the record number of the first entry is obtained from the first entry of the first chain. For modes 21 and 24, the qualified chain-heads are retrieved only during DBGETs, and hence, inaccurate chain counts and record numbers of last entry and first entry are returned for DBFIND.

For B-Tree DBFINDs:

- Can be used both for master data set and detail data sets.
- Allows wildcard search, relational operators, as well as range search.
- DBFIND mode 1 will result in a B-Tree wildcard search, if all of the following are true:
 - Item type is X or U (ASCII).
 - DBFIND is for master and a B-Tree index exists on the key item or DBFIND is for detail and B-Tree index exists on the key item of the master to which the search item has a path.
 - BTREEMDODE1 flag is set ON by DBUTIL or DBCONTROL mode 15.
- Argument contains a wildcard character.
- Status array reflects information based on set, mode, and search type. B-Tree searches for modes 1 and 4 for details give super-chain (multiple detail chain) counts and record numbers of the first entry in first detail chain and last entry in last detail chain. For masters, modes 1 and 4 give a count of entries qualified in the master and zeroes for the first and last qualified entries in the master.

NOTE A call to DBOPEN does not open individual data sets. Thus, a call to DBFIND (or DBGET) that accesses a data set for the first time (or after the data set has been closed), must open the data set, as well as the B-Tree index file (if applicable). This causes extra overhead not incurred by subsequent calls to the same data set by DBFIND or DBGET.

Consult appendix A for more information about the conditions for the DBFIND return status values shown in Table 5-12.

Table 5-12. DBFIND Return Status Values

File System, Memory Management, and Transaction Management Failures:	-1	FOPEN failure.
	-3	FREADDIR failure.
	-4	FREADLABEL failure.
	-168	Cannot attach <i>n</i> to MPE XL XM: file system error <i>nn</i> .
	-169	Invalid mode for XM attach options.
	-175	Cannot attach <i>n</i> to MPE XL XM: XM error <i>nn</i> .
	-176	Cannot detach <i>n</i> from MPE XL XM: XM error <i>nn</i> .
	-178	Cannot detach <i>n</i> from MPE XL XM: file system error <i>nn</i> .
	-209	Invalid mode for XM detach options.

Calling Errors:	-11	Bad database reference.
	-21	Bad data set reference.
	-31	Bad mode.
	-51	Bad list length.
	-52	Bad item.
	-222	Only DBXUNDO allowed when a dynamic transaction encounters an error.
	-258	Invalid argument for index.
	-259	Invalid mode for index.
	-260	No previous list of qualified data entries.
	-426	BTE: B-Tree doesn't exist.
	-429	BTE: DBFIND argument version is bad.
	-430	BTE: DBFIND (mode 4/24) argument type is bad.
	-431	BTE: DBFIND (mode 4/24) argument #1 length is bad.
	-432	BTE: Bad wildcard character.
	-433	BTE: DBFIND (mode 4/24) argument #2 length is bad.
	-434	BTE: Data set detail instead of master.
	-436	BTE: Failed to extract data from root file.
	-437	BTE: Failed to convert @c to dbfind.
	-438	BTE: Invalid item number.
	-439	BTE: Conversion error.
-444	BTE: DBFIND on non-key field of master.	
-446	BTE: Argument 2 specified for relop of (</<=</=>=</>).	
-449	BTE: Failed to position index at start of key range.	
Communications Errors:	-102	DSWRITE failure.
	-106	Remote 3000 data inconsistent.
	-107	NS 3000 or DS 3000 system error.
Exceptional Conditions:	-193	DBU control block is full.
	-3nn	Internal error.
	-314	Error in patch information for set.
	17	No master entry.
	63	DBG disabled; potential damage; only DBCLOSE allowed.