

TPF Database Facility



Glossary

Release 1

Contents

Glossary	1
A – Glossary Entries	1
B – Glossary Entries	2
C – Glossary Entries	2
D – Glossary Entries	3
E – Glossary Entries	4
F – Glossary Entries	4
G – Glossary Entries	4
H – Glossary Entries	4
I – Glossary Entries	5
K – Glossary Entries	5
L – Glossary Entries	5
M – Glossary Entries	5
N – Glossary Entries	6
O – Glossary Entries	6
P – Glossary Entries	6
R – Glossary Entries	7
S – Glossary Entries	7
T – Glossary Entries	8
U – Glossary Entries	8
V – Glossary Entries	8
W – Glossary Entries	8

Glossary

This glossary defines terms commonly used in the TPF Database Facility (TPPDF) product information that are not common to data processing in general, or data processing terms common to other systems that are uniquely defined by the TPDFDF product. Some of these terms may have other meanings in other contexts or in other data processing systems. Commonly defined data processing terms are not usually included here; however, some such terms that are widely used in the TPDFDF product, and critical to its understanding, are included. See the *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994 for a definition of other commonly defined terms. In addition, see the glossaries provided with the TPF and ALCS libraries for terms that are common to the TPF system, the ALCS environment, and the TPDFDF product.

The following cross-references are used in this glossary:

Contrast with.

This refers to a term that has an opposed or substantively different meaning.

Synonym for.

This indicates that the term has the same meaning as a preferred term, which is defined in its proper place in the glossary.

Synonymous with.

This is a backward reference from a defined term to all other terms that have the same meaning.

See. This refers to a more commonly used term or to multiple-word terms in which the term appears.

See also.

This refers to terms that have a related, but not synonymous, meaning.

Deprecated term for.

This indicates that the term should not be used. It refers to a preferred term, which is defined in its proper place in the glossary.

A – Glossary Entries

active keys. The keys currently being used by the TPDFDF product to select logical records (LRECs) in a subfile. Keys remain active until new keys are specified by a TPDFDF macro or function, or until the subfile is closed.

add current file. A file that has subfiles in which a limited number of overflow blocks can be added. If a logical record (LREC) is added that causes the limit to be exceeded, the oldest LREC (or LRECs, if necessary) are deleted to make room. An add current file is indicated by bit 2 of symbol &SW00OP1 in the DSECT for the file. The maximum number of overflow blocks is indicated by symbol &SW00NOC. See also *add current processing*.

add current processing. The steps completed by the TPDFDF product when a logical record (LREC) is added, using the DBADD macro or dfadd function, to an add current file. See also *add current file*.

Airline Control System (ALCS). A transaction processing platform providing high performance, capacity, and availability, that runs specialized transaction processing applications. The ALCS environment runs application programs written for a TPF platform on an MVS system with little or no modifications. This includes application programs that are written using TPDFDF macros and functions. Synonymous with *TPF/MVS*.

ALCS. Airline Control System.

algorithm. In the TPDFDF product, an addressing mechanism that locates the prime block of a subfile. The database administrator specifies the algorithm type in the DSECT macro. See also *algorithm argument*.

algorithm argument. A value specified in TPFDF application programs to indicate the subfile that will be affected by a TPFDF macro or function. The algorithm argument can be a character string or number, depending on the type of algorithm defined in the DSECT for the file. An algorithm argument is also used to indicate a subfile in an intermediate index file or detail file. See also *algorithm*.

application program. A program written for or by a user that applies to the user's work. In a TPFDF environment, any program that requests a TPFDF service using a macro or function.

B – Glossary Entries

B+Tree index. A file structure where all data is contained in data blocks and a B+Tree indexes into the data blocks. The B+Tree, which consists of nodes, is maintained internally by the TPFDF product. The nodes consist of one root node, internal nodes, and leaf nodes. The root and internal nodes point to other nodes, and the leaf nodes point to data blocks. The B+Tree is dynamically balanced so that the B+Tree is of uniform depth. See also *root node*, *internal node*, and *leaf node*.

basic indexing. A method of creating TPFDF structures that include index and detail files.

block. A physical storage area used by the TPFDF product that contains a header, logical records (LRECs), and an optional trailer. (A block is the TPFDF term for a TPF or ALCS record.)

block header. The standard header in data blocks (records) that includes information such as the file ID, the record code check (RCC), data control, program name, and chaining address fields. The block header for TPFDF blocks includes all fields in the standard TPF and ALCS header plus additional fields used only for the TPFDF product.

block index support. A method that the TPFDF product uses to identify which logical records (LRECs) are contained in overflow blocks of a subfile. Block index support allows faster access to LRECs in overflow blocks. This support only works if the LRECs are organized UP (ascending) or DOWN (descending) in each subfile.

block trailer. An optional area located at the end of each TPFDF block that contains information about the block.

C – Glossary Entries

capture/restore utility, information and statistics environment (CRUISE). The overall process of validation, capture, and recovery based on file information stored in the TPFDF database definition (DBDEF).

capture/restore utility, information and statistics environment (CRUISE) commands. The ZFCRU commands that perform CRUISE functions.

capture/restore utility, information and statistics environment (CRUISE) function. The process that performs the steps specified by a parameter table.

central descriptor table. Deprecated term for *database definition (DBDEF) table*.

chain chasing. The phase during recoup when defined record chains are read to determine pool usage.

chaining. The method the TPFDF product uses to link overflow blocks to a prime block in a subfile.

checkpoint. In detach mode, a process in which all the blocks currently in storage are written to DASD. See also *detach mode*.

child node. In a B+Tree index, a node that originates from, and is pointed to by, a higher-level parent node. See *parent node* and *sibling node*.

CRUISE. Capture/restore utility, information and statistics environment.

current LREC. The logical record (LREC) that is being referenced from field SW00REC in the SW00SR slot for the subfile. The current LREC is used by TPFDF macros and functions while processing the subfile. For example, the current LREC can be used as the starting point for locating another LREC while processing a DBRED macro or dfred function.

D – Glossary Entries

data block. A term used with B+Tree indexing that refers to any block in a B+Tree file that is not being used as a node block.

data collection. In the TPFDF product, a facility that collects system activity data used to analyze TPFDF performance.

data identifier (DID). A 2-byte field that identifies the userLREC part of an extended logical record (LREC).

data level independence (DLI). In the TPFDF product, a term used to indicate that all data levels holding blocks before a macro or function are preserved across the call.

database administrator. The person who is typically responsible for designing TPFDF files. The responsibilities of the administrator can include coding DSECT macros and database definitions (DBDEFs).

database definition (DBDEF) macro. A macro that generates database definition (DBDEF) tables each of which defines the major characteristics of a TPFDF file. If the file is an index or detail file, the DBDEF macro defines relations between this file and others in the index structure. There is one DBDEF macro statement for each file in the system.

database definition (DBDEF) table. A table that is located in main storage for each TPFDF file ID. Each table contains information that defines the characteristics of the file. The information used to define a DBDEF table consists of:

- Application file descriptors (DSECTs)
- Default TPFDF values
- Parameters written in the DBDEF macro statement.

This table ensures a centralized capability for maintaining TPFDF attributes by providing the TPFDF product with the file characteristics and accessing information required to process a file.

database definition (DBDEF) index table. A table containing one entry for each TPFDF file ID. Each item contains the address of a DBDEF table that defines the characteristics of that file.

database interface block (DBIFB). A TPFDF work block containing information about each open subfile. The information about each subfile is contained in a SW00SR slot.

DBDEF macro. Database definition macro.

DBDEF table. Database definition table.

DBDEF index table. Database definition index table.

DBIFB. Database interface block.

DDA. Distributed Data Access.

default key. A key defined in a DBDEF macro statement that is used to determine the placement of logical records (LRECs) in a file as they are added to a file, and to select LRECs as they are read from a file.

default-key key list. A key list that is used to determine which key set in a file's database definition (DBDEF) should be used to select logical records from the database.

detac mode. When a program opens a subfile in detac mode, the TPFDF product retains all blocks that it reads or writes in main storage. The TPFDF product only writes modified blocks to DASD when the application program checkpoints the subfile or closes the subfile without using the ABORT parameter. See also *checkpoint*.

detail file. The lowest-level file in an indexed structure. The simplest type of indexing consists of a single top-level index file referencing a detail file. Relationships between index files and detail files are defined by the database administrator (by coding instructions in the DBDEF macro). See also *index file*, *intermediate index file*, *top-level index file*.

DID. Data identifier.

Distributed Data Access (DDA) • hold

Distributed Data Access (DDA). See *TPFDF Distributed Data Access (TPFDF/DDA)*.

DLI. Data level independence.

DSECT. Data definition (assembler) macro. In the TPFDF product, each DSECT macro defines the characteristics of a single file. The database administrator codes a DSECT macro to define the layout of LRECs, block sizes, file processing options, algorithm to be used, and other details of a file.

E – Glossary Entries

extended logical record (LREC). A logical record that contains control fields (maintained by the TPFDF product), a userLREC, and, optionally, a number of subLRECs. An extended LREC provides a flexible way of holding data because each subLREC and the userLREC is a variable length.

extended LREC. Extended logical record.

F – Glossary Entries

fast link. The TPFDF mechanism for transferring control between high-access TPFDF routines.

fast-link table. A TPFDF table containing the main storage addresses of all TPFDF fast-link programs.

fast-link segment. A segment that contains high-access TPFDF routines.

field. A single data element in an LREC; for example, a date, name, or address.

file. Physically, a file comprises a collection of blocks, all having the same file ID. Logically, a file consists of LRECs contained in one or more subfiles. A TPFDF file is defined by a DSECT macro.

file ID. A 2-byte identifier in a block (record). It has the same value for all blocks (records) in a file. The file ID is included in the block header. Synonymous with *record ID*.

fixed file. A TPFDF file where the prime blocks of each subfile are contained in TPF or ALCS fixed file records. Typically, fixed files are permanently assigned to a specific function or file type. Contrast with *pool file*.

fixed-length logical record (LREC). A logical record (LREC) where all fields in the record have a fixed length. See also *variable-length logical record (LREC)* and *extended logical record (LREC)*.

fixed-length LREC. Fixed-length logical record.

fullfile. An operation that allows TPFDF processing to access more than one subfile in a file.

G – Glossary Entries

global modification. A TPFDF operation that allows more than one logical record (LREC) to be modified in a file. A selection key list is used to select the LRECs to be modified, and a modification key list defines how the records are modified. Global modification is invoked using the DBMOD macro or dfmod function with the ALL parameter specified. See also *selection key list*, *modification key list*, and *key list*.

H – Glossary Entries

header. Synonym for *block header*.

hold. A facility that allows multiple entry control blocks (ECBs) to share data and serialize their access to the data. When an application program opens a TPFDF subfile, the program can request that the TPFDF product hold the prime block. If a prime block is held, no other ECB can access logical records (LRECs) in the subfile until the application program releases it or closes the subfile.

I – Glossary Entries

interleaved file. A file in which the prime blocks are grouped together into interleaves. Each interleave has the same number of subfiles. The prime blocks of the different interleaves are interspersed together on DASD. For example, if there are three interleaves (A, B, C), and each contain four prime blocks, the prime blocks occur in the sequence: A1, B1, C1; A2, B2, C2; A3, B3, C3; A4, B4, C4. You can easily increase the number of prime blocks in each interleave by increasing the file size. You cannot easily increase the number of interleaves.

intermediate index file. An index file that is between a top-level index file (highest) and a detail file (lowest). See also *detail file*, *index file*, *top-level index file*.

index key. The field in an index logical record (LREC) that is used to identify the subfile being referenced. The index key is used as an algorithm argument on TPFDF macros and functions. See also *index LREC*.

index LREC. A variable-length or extended logical record (LREC) that references a subfile in a detail file or an intermediate index file. See also *index key*.

index file. A file in an index structure that contains keys and references to other files. See also *detail file*, *intermediate index file*, *top-level index file*.

internal node. Any node that is not a root node or a leaf node. An internal node only contains pointers to other nodes.

K – Glossary Entries

key list. A structure defined by the SW01SR DSECT that contains the displacement and length of logical record (LREC) key fields, a comparison operator, and values to compare the fields against. The key list is used to locate records that match the specified criteria. A key list provides the same function as the KEYn parameters. However, only key lists can be used for default keys on read operations, Boolean operators with keys, and global modification of LRECs. See also *default-key key list*, *modification key list*, *selection key list*, and *sort/merge key list*.

keys. Synonym for *LREC keys*.

L – Glossary Entries

leaf. Synonym for *leaf node*.

leaf block. Synonym for *leaf node*.

leaf node. The lowest-level node. It contains pointers to the data blocks. Synonymous with *leaf* and *leaf block*.

logical record (LREC). The smallest unit of data that a TPFDF program normally accesses (reads, adds, or deletes). An LREC contains several fields, one of which must be the LREC ID. The TPFDF product supports three types of LRECs: fixed-length, variable-length, and extended LRECs.

LREC. Logical record.

LREC ID. A 1-byte field identifying the logical record type. The LREC ID (equated to a value in the file DSECT) is also called the primary key of the LREC.

LREC keys. The TPFDF product allows searches for logical records (LRECs) by comparing program-provided search arguments against specified data fields in the logical record. The search arguments are called search keys. The data fields in the logical record are called logical record (LREC) keys.

M – Glossary Entries

modification key list. A key list that is used to globally modify LRECs in a file. This key list contains the rules for modification, including the displacement and length of the field to be modified, the location of the modification value, and the modification operation.

multiple reference check (MRC) • primary key

multiple reference check (MRC). In the TPFDF product, a method that modifies CRUISE capture processing so that detail files referenced from more than one index file are chain chased only once.

MRC. Multiple reference check.

N – Glossary Entries

NAB. Next available byte.

next available byte (NAB). A 2-byte value contained in the header of each TPFDF block. This value indicates the next byte in the block available for placement of a new logical record.

node. A block containing pointers to other nodes or to data blocks. Synonymous with *node block*.

node block. Synonym for *node*.

O – Glossary Entries

ordinal number. The relative position of a prime block in a file. Ordinal numbers start at zero.

organization. In the TPFDF product, the ascending (UP) or descending (DOWN) sequence of logical records (LRECs) in a subfile. If a subfile is organized, the TPFDF product can access the LRECs without necessarily reading the whole subfile. The database administrator specifies how a file is to be organized by coding the DBDEF macro instructions that define the file.

overflow block. Any block in a subfile that is not a prime block. A subfile can contain zero or more overflow blocks, which are always pool records. See also *prime block* and *pool file*.

P – Glossary Entries

packing. A TPFDF operation that reduces the number of blocks used to hold logical records (LRECs) contained in a subfile. A pack operation is initiated by the TPFDF product when any LREC has been deleted and the number of LRECs in any block falls below a threshold defined in the DSECT macro or DBDEF macro for the file. The number of LRECs placed in each block is also controlled by the DSECT or DBDEF macro. A pack operation can also be initiated by an application program when the subfile is closed, or by the ZUDFM or ZFCRU commands.

parameter table. The source of parameter values that define how a capture/restore utility, information and statistics environment (CRUISE) function is processed.

parametric recoup. The attribute of the file recoup program that allows the program to be activated and controlled by parameter lists.

parent node. In a B*Tree index, a node that produces and points to a lower-level child node. See also *child node* and *sibling node*.

partitioned file. A file where the prime blocks are grouped together into different partitions. Each partition has the same number of subfiles. The prime blocks of the different partitions follow each other sequentially on DASD. For example, if there are three partitions (A, B, C), and each contains four prime blocks, the prime blocks occur in the sequence: A1, A2, A3, A4; B1, B2, B3, B4; C1, C2, C3, C4. You can easily increase the number of partitions by increasing the size of the file. You cannot easily increase the number of prime blocks.

path. A means of referencing a detail file or intermediate index file from an index file. Each different reference is identified by a path number (0, 1, 2, ...) defined by the database administrator. See also *update path* and *read-only path*.

pool file. A TPFDF file where the prime block is contained in a TPF or ALCS pool record. Typically, pool files are used for detail files or intermediate index files. Contrast with *fixed file*.

primary key. Synonym for *LREC ID*.

prime block. The first block in the chain of blocks that comprise a subfile. All other blocks are referred to as overflow blocks. The subfile can contain zero or more overflow blocks. The prime block is either a fixed file or pool record. See also *fixed file*, *pool file*, and *overflow block*.

pushdown chaining. A file in which records are always added to the prime block of the subfile. If there is not enough room, the logical records (LRECs) in the prime block are moved to a new overflow block, the prime block is initialized, and the LREC added. A pushdown chaining file is indicated by bit 3 of symbol &SW00OP1 in the DSECT of the file.

R – Glossary Entries

read-only default key. A default key that can *only* be used to select records. It is not used for add operations.

read-only path. In an index structure, a path where the index key is only partially described. There can be more than one logical record (LREC) in the detail file that matches the index key, so the path cannot be used to update an index structure. Contrast with *update path*.

record ID. Synonym for *file ID*.

recoup. A real-time database validation routine that runs online in an ALCS environment or TPF system. The primary function of recoup is to identify long-term pool file blocks that are not in use. The system can then dispense these blocks again. The TPFDF product includes extensions to TPF and ALCS recoup that allow pool records in TPFDF files to be identified using database definitions (DBDEFs).

root. Synonym for *root node*.

root block. Synonym for *root node*.

root node. The highest level node. If the root node is the only node in the B+Tree index structure, it contains pointers to data blocks; otherwise, it contains pointers to other nodes. Synonymous with *root* and *root block*.

R-type file. A real-time file; that is, any type of application data file (whether stored in fixed or pool prime blocks), excluding W-type files.

S – Glossary Entries

SAPR. Deprecated term for *parametric recoup*.

SAT. Search argument table.

search argument table (SAT). A table that contains the selection criteria used during a selective restore.

selection key list. A key list that allows the TPFDF product to search for logical records by comparing program-provided search arguments against specified data fields in the logical record.

sibling node. In a B+Tree index, one of two or more child nodes that share the same parent node. See also *child node* and *parent node*.

sort/merge key list. A key list that determines how LRECs are sorted into the output file on a sort or merge operation.

standard header. Synonym for *block header*.

subfile. A logical subdivision of a file. A subfile consists of one prime block and, optionally, one or more overflow blocks.

subLREC. An item of data, normally consisting of several fields, that is contained in an extended LREC. A program can add, delete, or modify subLRECs in any extended LREC.

SW00SR slot. An area in the database interface block (DBIFB) that contains control information and work space relating to an open subfile.

T – Glossary Entries

technical LREC (TLREC). A logical record (LREC) used with block indexing or B*Tree indexing to locate data in a subfile. These LRECs are maintained by the TPFDF product and are not directly accessed from application programs.

TLREC. Technical LREC.

top-level index file. The highest file in an index structure. The top-level index file can reference a detail file (lowest level) or an intermediate index file. See also *detail file*, *intermediate index file*, *index file*.

TPF. Transaction Processing Facility.

TPFDF Distributed Data Access (TPFDF/DDA). A feature of the TPFDF product that propagates data from a TPFDF hierarchical database to a DB2 relational database.

TPFDF recoup. An extension to TPF recoup phase 1 that allows pool records in TPFDF files to be identified using database definitions (DBDEFs).

TPF/MVS. Synonym for *Airline Control System (ALCS)*.

traditional (P-type) file. A file that does not contain logical records (LRECs), and was created using TPF or ALCS file access macros or functions (for example, a FINDC or FILEC macro). Traditional files can be defined to the TPFDF product as P-type files using a DSECT macro and a DBDEF macro, which allows certain TPFDF macros, functions, and utilities to be used with the file.

trailer. Synonym for *block trailer*.

Transaction Processing Facility (TPF). A stand-alone, real-time operating system designed for transaction-driven applications requiring speed, reliability, and data currency.

T-type file. A file that is a temporary logical record (LREC) in a W-type file.

U – Glossary Entries

update path. In an index structure, a path where the index key is fully defined, unlike a read-only path that uses partial keys. This is the only path that can be used when adding or updating indexes. Contrast with *read-only path*.

userLREC. Part of an extended logical record (LREC) that is used to hold user data. The last field in the user LREC can be a variable length.

V – Glossary Entries

variable-length logical record (LREC). A logical record (LREC) where one field is variable length. All other fields are fixed length. See also *fixed-length logical record (LREC)* and *extended logical record (LREC)*.

variable-length LREC. Variable-length logical record.

W – Glossary Entries

W-type file. A work file that only lasts the life of the entry control block (ECB).