CITY OF NEW YORK DEPARTMENT OF CITY PLANNING
INFORMATION TECHNOLOGY DIVISION
GEOGRAPHIC SYSTEMS

FILE DOCUMENTATION 9 July 2009

FILE NAME: LION DIFFERENCES FILE (LDF)

DESCRIPTION:

The LION Differences File (LDF) is a sequential file containing records documenting certain types of changes that have occurred between a particular release of LION and the immediately previous LION release. A new LDF 'edition' is 'published' in conjunction with each new production release of LION[1].

Identifying Segments and Nodes in LION Over Time

Every record in LION represents a segment of a street or non-street feature. Every LION record is uniquely identified by its LION key (the combination of the borough code, face code and sequence number). Each segment represented in LION is uniquely identified by its segment-ID.

Segment-ID's correspond to geographic entities (segments), whereas LION keys correspond to physical LION records. This distinction is more than merely semantic, since there are cases where more than one LION record represents the same segment. Specifically, when a segment lies along a boundary of two boroughs, it is represented in LION by two records, one record for each borough. For example, a segment lying along the Brooklyn-Queens boundary is represented by a Brooklyn record and a Queens record; the two records have the same segment-ID (because they represent the same geographic segment), but they have different LION keys (because they are different LION records). (Note: The two LION records for a borough boundary segment differ not only in their LION keys but also in the contents of certain other fields.)

The endpoints of LION segments are called nodes. Every node in LION is uniquely identified by a node-ID. Every node in LION also has a real-world geodetic location, specified by its spatial coordinates. One of the nodes of every LION segment is designated as the segment's from-node, and the other node is designated as the to-node.

For purposes of the LDF, a segment is considered to derive its existence and retain its identity across LION releases by virtue of the continuing validity of its segment-ID, regardless of what changes may have occurred to the segment (such as a change in the LION key(s) of the record(s) representing the segment, or a change in the node-ID's or the spatial locations of the segment's

[1] Production releases of LION correspond to major production releases of the Geosupport System. The first edition of the LDF was published as part of Geosupport Release 02A. It will document changes in LION between Releases 01B and 02A.

nodes.) Similarly, a node is considered to derive its existence and retain its identity across LION releases by virtue of the continuing validity of its node-ID, regardless of what changes may have occurred to the node (such as a change in its spatial location or in which segments are incident upon it).

In conjunction with the implementation of the LDF, the following LION updating policies will be instituted. If a LION segment is deleted, its segment-ID will be permanently 'retired', that is, it will never again be assigned to any segment. (Note: this is not necessarily true of LION keys.) In this context, segment deletion refers not only to outright physical deletion (such as to reflect a street closure), but also splitting of the segment into other segments or merging of the segment with other segments. In all these cases, the segment-ID that was assigned to the segment prior to its deletion, splitting or merging, will never be re-assigned to any other segment, not even to an 'offspring' segment created by a split or merge. Similarly, if a node is deleted from LION, its node-ID will be permanently 'retired'.

The segment-ID serves as a unique, immutable identifier of a segment over time, and therefore is the appropriate data item to store in user application files as the segment identifier. In contrast, the LION key of a segment is neither immutable nor necessarily unique (in the borough boundary case), and therefore is unsuitable for use as a segment identifier. The changeability of LION keys over time is a consequence of the role that they play in sequencing a feature's segment records. This sequencing enables Geosupport System Function 3S to provide a list of the cross streets along any street in proper geographic sequence. To maintain proper sequencing of a street as new segments are introduced via splitting, merging or extending the street, it is sometimes necessary to change the LION keys of existing segments. On the other hand, a segment-ID is an arbitrary identifying number with no inherent geographic significance, hence can be kept constant.

Structure of the LDF

The LDF has three record types, identified by the Record Type Code. Each edition of the LDF has a single header record (Record Type Code = 'H'). Each edition of the LDF can also have one or more node change records (Record Type Code = 'N') and one or more segment change records (Record Type Code = 'S'). It is possible for an edition of the LDF to have no node change records and/or no segment change records.

The three record types all have a record length of 100 but they have different record layouts. The type 'N' records and the type 'S' records are further differentiated by various types of 'actions', such as add, delete, move etc., identified by an Action Code.

The record layouts of the three record types are below.

Header Record (Type 'H' Record)

Item	Length	From	To	Value/Comment
Record Type	1	1	1	'H'
Filler	4	2	5	
Old LION Release	3	6	8	
Filler	3	9	11	
Date of Old LION Release	6	12	17	MMDDYY format
Filler	5	18	22	
New LION Release	3	23	25	
Filler	3	26	28	
Date of New LION Release	6	29	34	MMDDYY format
Filler	5	35	39	
Record Count	6	40	45	Includes header record
Filler	45	46	90	
Cumulative LDF Record Number	10	91	100	

The fields Date of Old LION Release and Date of New LION Release contain the dates on which those LION releases were deployed in production as VSAM files on the DoITT mainframe. (In general, mainframe LION releases, LDF editions and batch major releases of the Geosupport System are all deployed in production on the DoITT mainframe on the same dates.)

The field Record Count contains the number of records in the given LDF edition, including the header record. The field Cumulative LDF Record Number is discussed later in this document.

Node Change (Type 'N') Records

Item	Length	From	To	Value/Comment
Record Type	1	1	1	'N'
Filler	1	2	2	
Action Code	1	3	3	'A', 'D' or 'M'
Filler	7	4	10	
X-Coordinate	7	11	17	If Action Code is M, these are
Y-Coordinate	7	18	24	the coords. of original location
Filler	7	25	31	
Node-ID	7	32	38	
Filler	2	39	40	
Destination X-Coordinate	7	41	47	Used only for Action Code M
Destination Y-Coordinate	7	48	54	Used only for Action Code M
Filler	36	55	90	
Cumulative LDF Record Number	10	91	100	

The possible action codes for a type N record are as follows:

Action Code	Meaning	Description
A	add node	A node (more precisely, a new node-ID) exists in the new LION release that didn't exist in the old LION release. The fields labeled X-Coordinate and Y-Coordinate contain the spatial coordinates of the new node. The fields labeled Destination X-Coordinate and Destination Y-Coordinate are blank (these fields are used only in Action Code 'M' records).
D	delete node	A node (more precisely, a node-ID) that existed in the old LION release does not exist in the new LION release. The fields labeled X-Coordinate and Y-Coordinate contain the erstwhile spatial coordinates of the deleted node. The fields labeled Destination X-Coordinate and Destination Y-Coordinate are blank.
M	move node	An existing node has been moved spatially, but retains the same node-ID. The fields labeled X-Coordinate and Y-Coordinate contain the old spatial coordinates of the node. The fields labeled Destination X-Coordinate and Destination Y-Coordinate contain the new spatial coordinates of the node.

If a node has been renumbered while remaining at the same spatial location, the LDF represents this by a combination of a delete and an add, i.e., there is an Action Code 'D' node change record for the old node-ID and an Action Code 'A' node change record for the new node-ID.

Segment Change (Type S) Records

Item	Length	From	To	Value/Comment
Record Type	1	1	1	'S'
Filler	1	2	2	
Action Code	1	3	3	'A', 'C', 'D', 'M' or 'S'
Filler	7	4	10	
Items Appurtenant to Old Segment:				
Old Segment-ID	7	11	17	
Old LION Key	10	18	27	
Old From Node-ID	7	28	34	
Old To Node-ID	7	35	41	
Filler	2	42	43	
Items Appurtenant to New Segment:				
New Segment-ID	7	44	50	
New LION Key	10	51	60	
New From Node-ID	7	61	67	
New To Node-ID	7	68	74	
Filler	16	75	90	
Cumulative LDF Record Number	10	91	100	

Segment change records are classified into action types 'A' (add), 'C' (change node(s)), 'D' (delete), 'M' (merge) and 'S' (split). For new segments, changed nodes and segment deletions, there will be a single segment change record representing the added, changed or deleted segment. For merges and splits of segments, there will be one segment change record for each combination of a parent (old) segment and an offspring (new) segment. For example, if an old segment has been split into three new segments, there are three segment change records with Action Code 'S', each record relating the given old segment to one of the three new segments. The possible Action Codes for segment change records are as follows:

Action Code	Meaning	Description
A	add segment	A new segment (more precisely, a new segment-ID) exists. The old segment fields are blank. The new segment fields have values.
C	change node(s) of	At least one of an existing LION segment's nodes (i.e.,

		existing segment node-ID's) has changed. The Old Segment-ID and Old LION Key fields are identical to the New Segment-ID and New LION Key fields, respectively. Note: Mere spatial relocations of nodes with no change of node-ID do not engender records of this type; such cases are represented by node change Action Code 'M' records.
D	delete segment	A Segment-ID that existed in the old LION release does not exist in the new LION release. The old segment fields have values. The new segment fields are blank.
M	merge segments	The specified old segment has been merged with one or more other segments from the old LION release to form the new segment. Both the old and the new segment fields have values. Other segment change Action Code 'M' records represent the merging of each of the other old segments into the given new segment.
S	split segments	The specified old segment has been split into two or more new 'offspring' segments, one of which is the new segment specified in this Action Code 'S' record. Both the old and new segment fields have values. Other segment change Action Code 'S' records represent the splitting of the given old segment into each of its other 'offspring' segments.

Ordering of LDF Records

Within each edition of the LDF, the records are ordered as follows. The highest-level sort key is LDF Record Type. Thus, the single header record occurs first, followed by all node change records, followed by all segment change records. Within each record type, the records are ordered so that related records occur consecutively, as follows.

Node change records are related to each other only in the case when a node has been renumbered (i.e., given a new node-ID) while remaining at the same spatial location. Such a node renumbering is represented in the LDF by a combination of a delete (i.e., by an Action Code 'D' record to delete the old node-ID) and an add (an Action Code 'A' record to add the new node-ID). To insure that the records of such related pairs occur consecutively, the set of all node change records is sorted on the fields labeled X-Coordinate and Y-Coordinate.

Segment change records are related to each other in the case of a segment merge and in the case of a segment split. To insure that the set of records representing a particular merge or split occur consecutively, the segment change records are ordered as follows. The set of all

segment change records is sorted on Action Code. The Action Code 'M' records (the merge records) are further sorted on New Segment-ID followed by Old-Segment-ID. The Action Code 'S' records are sorted on Old Segment-ID followed by New Segment-ID.

The sort ordering of the LDF can be summarized as follows, where indentation signifies a lower-level sort key:

All records are sorted on Record Type.
- Type 'N' records are sorted on X-Coordinate, Y-Coordinate.
- Type 'S' Records are sorted on Action Code.
  - Type 'S' Action Code 'M' Records are sorted on New Segment-ID, Old Segment-ID.
  - Type 'S' Action Code 'S' Records are sorted on Old Segment-ID, New Segment-ID.

Cumulative LDF Record Numbers

All the records (including header records) in all LDF editions collectively will be consecutively numbered, without gaps, in the field Cumulative LDF Record Number in positions 91-100. The Cumulative LDF Record Number will be cumulative across editions of the LDF, so that a given value will uniquely identify one record in one LDF edition. For example, if the first edition of the LDF were to contain 693 records, they would be numbered 1 through 693 consecutively; the records in the next LDF edition would be numbered consecutively beginning with 694, and so forth.

LDF Releases and Data Set Names

LDF's are sequential batch files. For each LION release, two copies of the LDF will be deployed, called the 'resynch' LDF and the 'archive' LDF of the given release. These files will be identical in content and will differ only in their DSN's and in how long they will exist.

The Resynch LDF: On the DoITT mainframe, the resynch LDF will have the same DSN in every release, A030.STREET.LDF. Whenever a new resynch LDF is deployed, it will replace the previous resynch LDF, which will be superseded and deleted. Users should code the full DD statement for the resynch LDF as:

```
//<anyDDname>   DD    DSN=A030.STREET.LDF,DISP=SHR
```

The resynch LDF is intended to be used with automated procedures to be developed by users to update their application files that are keyed to LION geography to reflect topological changes in new LION (and Geosupport System) releases, such as the creation, deletion, merging and subdivision of segments, and concomitant changes to node-ID=s and segment-ID's if these identifiers are stored in the application file. Because of the constancy of the DSN of resynch LDF's across releases, the user need not change the DD statement for the LDF in the JCL for

the resynchronization job each time it is run.

The Archive LDF: The DSN of the archive LDF will vary with each release. On the DoITT mainframe, the DSN of the archive LDF will be of the form A030.STREET.LDF.R<release-ID>. If the user needs to access an archived LDF, the full DD statement for it should be coded as follows:

//<anyDDname> DD DSN=A030.STREET.LDF.R<release-ID>,DISP=SHR

Each archive LDF release contains the changes to LION between the LION release indicated in the archive LDF=s DSN and the immediately preceding LION release. For example, the first archive LDF release, which was deployed with LION Release 02A having the DSN of A030.STREET.LDF.R02A, and it contained the changes between LION Release 01B and LION Release 02A. When deployed, each new archive LDF release will join the set of archive LDF's deployed in all previous releases, which will continue to remain accessible indefinitely.

At data centers other than DoITT, to conform to local file-naming standards, the DSN's of LDF's may differ from those specified above. Users at those data centers should ascertain the proper DSN's from the appropriate local Geosupport System administrator.