Laser-Scan Ltd.

STRUCTURE - Acceptance Tests

Issue 1.5 (mod) - 02-Oct-1992

Copyright (C) 2019 Laser-Scan Ltd
Science Park, Milton Road, Cambridge, England CB4 4FY tel: (0223) 420414

"STRUCTURE Acceptance"   Category: Acceptance Tests

Issue 1.0	Paul Hardy	24-Jul-1987
Issue 1.1	Paul Hardy	9-Aug-1987
Issue 1.2	Paul Hardy	20-Aug-1987
Issue 1.3	Andrew Morris	14-Sep-1987
Issue 1.4	Paul Hardy	29-Jun-1988
Issue 1.5	Tim Adams	10-Feb-1989
Issue 1.5 (mod)	K M Sutherland	02-Oct-1992

CONTENTS

1 Introduction

This document describes the acceptance test procedure for the Laser-Scan STRUCTURE (IFF structuring package) which is part of the Laser-Scan LAMPS automated mapping system. It assumes that the user is familiar with digital cartography, with the STRUCTURE modules, with LAMPS, with the LITES2 map editor, and with the VMS operating system. See the "STRUCTURE User Guide", and the "STRUCTURE Reference Manual" for further information on STRUCTURE.

The relevant data files and command files for the acceptance tests are supplied by Laser-Scan on installation of the package.

Note that Laser-Scan reserve the right to make minor modifications to this acceptance procedure to match their policy of continued software development.

2 Preparing for the Acceptance Tests

Check that the Laser-Scan-supplied package initialisation command file LSL$COM:STRUCTUREINI.COM has been invoked. This has probably been done automatically on your behalf at login time. A good check is to use the DCL command:

```
        $ SHOW SYMBOL ILINK
```

to verify that the DCL symbol ILINK exists and points to the program image file of the main STRUCTURE module "$LSL$EXE:ILINK". If symbol ILINK is not defined then invoke the package initialisation command file by giving the DCL command:

```
        $ @LSL$COM:STRUCTUREINI
```

then repeat the check for the existence of DCL symbol ILINK.

Use the DCL command SHOW LOGICAL LSL$IF to ensure that logical name LSL$IF points to a suitable working directory to receive the acceptance test IFF files. If not, then use the SI utility to set LSL$IF appropriately.

NOTE

> In the description and examples below, the working directory is referred to as "LSL$IF:" as the precise device and directory will vary from system to system. However, during the acceptance test command procedure, it will be referred to in some places on the output from running the tests as the actual device and directory being used.

The acceptance test command procedure will check for the existence of the required acceptance test data files in their usual directory on the Laser-Scan software distribution directory tree. It will set up a logical name LSL$STRUCTURE_ACCEPT to point to this directory. It will also copy the initial IFF data file into the working directory pointed at by LSL$IF. The name of the file is "STRUCTURE_ACCEPT.START_IFF".

3 Invoking the Acceptance Tests

Invoke the acceptance test command procedure by giving the DCL command

```
        $ @LSL$COM:STRUCTURE_ACCEPT
```

The test command procedure explains briefly the various phases of the acceptance procedure. Before starting processing, it offers you the option of using the LITES2 map editor to inspect the initial data. If you accept this option then you will get to an interactive LITES2 prompt (eg. 'READY'). It then runs the ILINK utility to carry out the processing. After each phase it offers you the option of using the LITES2 map editor to inspect the results of the processing.

4 The Acceptance Sequence

In these tests, the original data file contains vector "spaghetti" representing some soil type polygons and a roads overlay. The acceptance procedure steps through geometry idealisation and structure formation using ILINK in 6 stages.

1. LLJOIN - to align coincident lines.

2. MERGE - to amalgamate coincident lines.

3. LPJOIN - to extend or truncate line ends to meet at T junctions.

4. PPJOIN - to form unique junction points at line ends.

5. BREAK - to break crossing lines.

6. STRUCT - to generate IFF (link/node) junction structure.

4.1 LLJOIN Phase

In this phase, ILINK is used to align coincident lines.

The command line specified is:

```
$       ILINK/LLJOIN/BPF=3/JNTOL=40/LITES2=STRUCTURE_ACCEPT_LLJ -
                LSL$IF:STRUCTURE_ACCEPT.IFF -
                LSL$IF:STRUCTURE_ACCEPT.LLJ -
                /FCP=LSL$STRUCTURE_ACCEPT:FCP1
```

Note the output from ILINK showing its command interpretation, the log of progress through the operation, and the statistics.

Note the warning messages from ILINK indicating that some segments which are shorter than the specified tolerance have been elided.

Note that $STATUS is returned as "%SYSTEM-S-NORMAL, normal successful completion".

You are then asked if you wish to inspect the results of this phase using LITES2. If you answer yes, then the IFF data file is read in to LITES2 and displayed (assuming LITES2 and suitable graphics hardware are available).

Pass [ ]/Fail [ ]

4.2 **MERGE Phase**

In this phase, ILINK is used to amalgamate coincident lines. Note that one polygon which had been double digitised becomes a single feature after processing, and its feature code has been changed due to use of a Feature Code Combinations (FCC) file.

The command line specified is:

```
$       ILINK/MERGE/BPF=3/LITES2=STRUCTURE_ACCEPT_MER -
                LSL$IF:STRUCTURE_ACCEPT.LLJ -
                LSL$IF:STRUCTURE_ACCEPT.MER -
                /FCC=LSL$STRUCTURE_ACCEPT:FCC -
                /FCP=LSL$STRUCTURE_ACCEPT:FCP1
```

Note the output from ILINK showing its command interpretation, the log of progress through the operation, and the statistics.

Note that $STATUS is returned as "%SYSTEM-S-NORMAL, normal successful completion".

You are then asked if you wish to inspect the results of this phase using LITES2. If you answer yes, then the IFF data file is read in to LITES2 and displayed (assuming LITES2 and suitable graphics hardware are available).

Pass [ ]/Fail [ ]

4.3 **LPJOIN Phase**

In this phase, ILINK is used to extend or truncate line ends to meet at T junctions. Note that this produces a tidy effect at the neat line, and that the road network now connects (with the exception of one junction which is intentionally outside tolerance).

The command line specified is:

```
$       ILINK/LPJOIN/LITES2=STRUCTURE_ACCEPT_LPJ/EXTOL=50/JNTOL=50 -
                LSL$IF:STRUCTURE_ACCEPT.MER -
                LSL$IF:STRUCTURE_ACCEPT.LPJ -
                /FCP=LSL$STRUCTURE_ACCEPT:FCP2
```

Note the output from ILINK showing its command interpretation, the log of progress through the operation, and the statistics.

Note that $STATUS is returned as "%SYSTEM-S-NORMAL, normal successful completion".

You are then asked if you wish to inspect the results of this phase using LITES2. If you answer yes, then the IFF data file is read in to LITES2 and displayed (assuming LITES2 and suitable graphics hardware are available).

Pass [ ]/Fail [ ]

4.4 **PPJOIN Phase**

In this phase, ILINK is used to form unique junction points at line ends. Note that this closes open looped polygons, and makes all lines ending at a logical junction have a unique junction point as end points.

The command line specified is:

```
$       ILINK/PPJOIN/LITES2=STRUCTURE_ACCEPT_PPJ/EXTOL=70/JNTOL=50 -
                LSL$IF:STRUCTURE_ACCEPT.LPJ -
                LSL$IF:STRUCTURE_ACCEPT.PPJ -
                /FCP=LSL$STRUCTURE_ACCEPT:FCP3
```

Note the output from ILINK showing its command interpretation, the log of progress through the operation, and the statistics.

Note that $STATUS is returned as "%SYSTEM-S-NORMAL, normal successful completion".

You are then asked if you wish to inspect the results of this phase using LITES2. If you answer yes, then the IFF data file is read in to LITES2 and displayed (assuming LITES2 and suitable graphics hardware are available).

Pass [ ]/Fail [ ]

4.5 **BREAK Phase**

In this phase, ILINK is used to break crossing lines. Note that the roads are broken only where they cross other roads, and that the polygon links are broken only where they cross other polygon links or the neat line. This is achieved using a Feature Code Pair (FCP) file.

The command line specified is:

```
$       ILINK/BREAK -
                LSL$IF:STRUCTURE_ACCEPT.PPJ -
                LSL$IF:STRUCTURE_ACCEPT.BRK -
                /FCP=LSL$STRUCTURE_ACCEPT:FCP4
```

Note the output from ILINK showing its command interpretation, the log of progress through the operation, and the statistics.

Note that $STATUS is returned as "%SYSTEM-S-NORMAL, normal successful completion".

You are then asked if you wish to inspect the results of this phase using LITES2. If you answer yes, then the IFF data file is read in to LITES2 and displayed (assuming LITES2 and suitable graphics hardware are available).

Pass [ ]/Fail [ ]

4.6 **STRUCTURE Phase**

In this phase, ILINK is used to generate IFF (link/node) junction structure. Note that this phase differs from the previous phases as it is logical pointer structure that is being generated, rather than geometry being idealised.

The command line specified is:

```
$       ILINK/STRUCTURE -
                LSL$IF:STRUCTURE_ACCEPT.BRK -
                LSL$IF:STRUCTURE_ACCEPT.IFJ
```

Note the output from ILINK showing its command interpretation, the log of progress through the operation, and the statistics.

Note that $STATUS is returned as "%SYSTEM-S-NORMAL, normal successful completion".

You are then asked if you wish to inspect the results of this phase using LITES2. If you answer yes, then the IFF data file is read in to LITES2 and displayed (assuming LITES2 and suitable graphics hardware are available). Note that the results of this phase will appear graphically identical to those of the BREAK phase, as the STRUCTURE phase does not modify geometry.

Pass [ ]/Fail [ ]

5 Final Verification

This is the end of the formal acceptance test command procedure. The following files have been created from the original IFF file:

Original IFF filename: LSL$IF:STRUCTURE_ACCEPT.START_IFF

PHASE	FILE CHARACTERISTIC	FILE
SETUP	copy of initial data	STRUCTURE_ACCEPT.IFF
LLJOIN	near coincident lines aligned	STRUCTURE_ACCEPT.LLJ
MERGE	coincident lines amalgamated	STRUCTURE_ACCEPT.MER
LPJOIN	line ends truncated/extended	STRUCTURE_ACCEPT.LPJ
PPJOIN	line ends pulled together	STRUCTURE_ACCEPT.PPJ
BREAK	lines broken at crossings	STRUCTURE_ACCEPT.BRK
STRUCTURE	link/node pointers formed	STRUCTURE_ACCEPT.IFJ

These files are available in the directory pointed to by logical name LSL$IF: for further verification, if required.

The following files can be used to view the output of the acceptance test using LITES2. These can be found in the directory pointed to by the logical name LSL$STRUCTURE_ACCEPT:.

FRT	(feature representation table)	STRUCTURE_ACCEPT.FRT
SRI	(feature representation IFF file)	STRUCTURE_ACCEPT.SRI
TRI	(text representation IFF file)	STRUCTURE_ACCEPT.TRI

6 Conclusions

This completes the acceptance tests for the Laser-Scan STRUCTURE software package.

Overall Pass [ ]/Fail [ ]

Comments:

Customer Representative: **Date:**

Laser-Scan Representative: **Date:**