

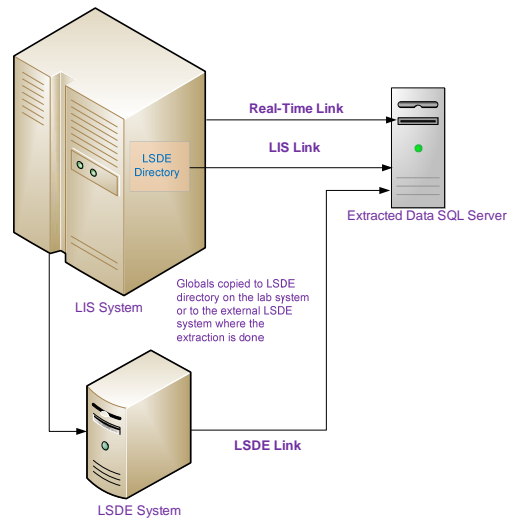


# Laboratory Data Acquisition

## Technical User Manual

### LIS-LSDE Data Extraction

---





## LIS Data Extract Summary

LSDE (Laboratory System Data Extraction) is a full function application that extracts data from the laboratory system and transfers it to an external server. It can be used as a turnkey application or modified to meet your specification. LSDE software package comes with a menu driven utility that allows you to configure and control the extraction process. Extraction control utility is install as LAB function LSDE.

Extracted data from LIS Cache/M global arrays are formatted and transferred to an external server where the data can be accessed via SQL statements or other programming languages such as java, C#, C++, VB, etc.

## Table of Contents

FULL EXTRACT .....	6
TABLE 2 FULL EXTRACTION ROUTINES .....	7
<b>INCREMENTAL EXTRACT .....</b>	<b>9</b>
<b>MANUAL START/STOP INCREMENTAL EXTRACTIONS: .....</b>	<b>9</b>
<b>AIX SHELL SCRIPTS .....</b>	<b>11</b>
<b>ROUTINES: .....</b>	<b>12</b>
<b>DATA EXTRACTION FILES .....</b>	<b>19</b>
<b>TEXT FILES.....</b>	<b>19</b>
<b>DATA DICTIONARIES.....</b>	<b>20</b>
<b>FILE SPECIFICATIONS .....</b>	<b>21</b>
TEST/BATTERY/PACKAGE DICTIONARY .....	21
BATTERY .....	21
TEST/BATTERY/PACKAGE DICTIONARY .....	21
TEST .....	21
TEST/BATTERY/PACKAGE DICTIONARY .....	22
PACKAGE.....	22
ENGLISH TEXT TRANSLATIONS .....	22
LABORATORY DEPARTMENT DICTIONARY .....	23
LABORATORY LOCATION DICTIONARY .....	23
PERFORMING LABORATORY DICTIONARY .....	23
DIAGNOSIS DICTIONARY .....	24
TUBE TYPE DICTIONARY .....	24
REFERENCE LAB DICTIONARY .....	24
TECH CODE DICTIONARY .....	25
LOCATION CODE/ROOM NUMBER DICTIONARY .....	25
EVENT TYPE MAINTENANCE .....	26
HOSPITAL ID CODES FOR MULTIPLE HOSPITAL SITES.....	26
PHYSICIAN DICTIONARY .....	26
NORMAL RANGE FILE .....	27
PATIENT MASTER FILE.....	28
PATIENT RESULTS FILE .....	29
TEST METHOD CODE FILE.....	32
ACCESSION HISTORY FILE.....	32
SITE LAST LOAD FILE.....	32
SITE LAST DATE PROCESSED FILE .....	33
FTP FILE TRANSFER CONFIRMATION FILE .....	33
<b>TRANSACTION GLOBAL FILE SPECIFICATION .....</b>	<b>34</b>
<b>LSDE UTILITY DOCUMENTATION .....</b>	<b>36</b>

# FULL EXTRACTION

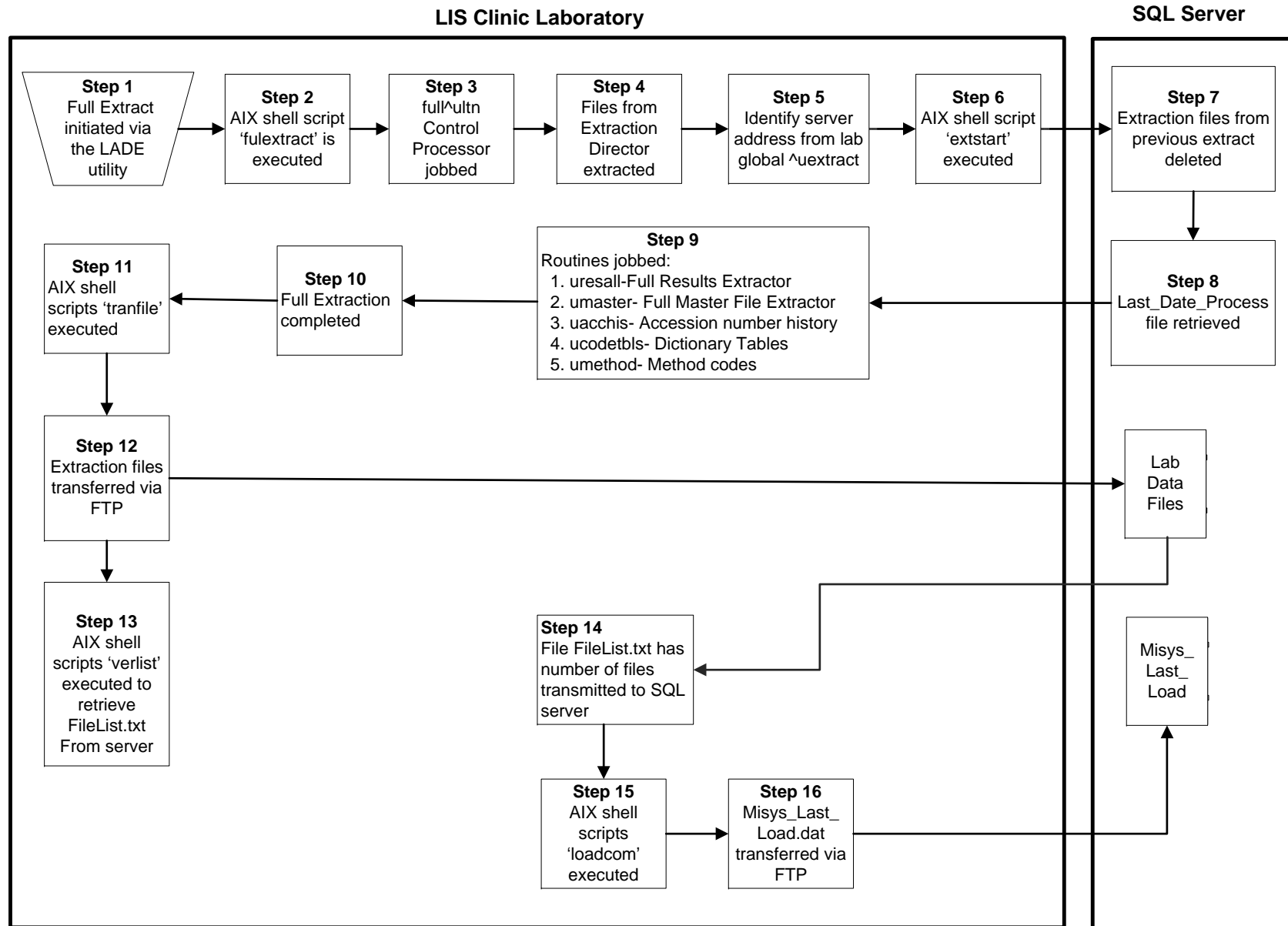


Figure 1 Full Extraction

## Full Extract

To start a full extraction use lab system function LSDE options 2 ,1 “Start /Stop Full Extract” (see LSDE Utility Documentation for details).

Full extractions are started manually using LSDE Utility option 2 which calls AIX script **fulextract**.

### **^uln: Full Control Processor** (See Figure 1 above)

- **Step 1** Start by using LSDE function 2,1
- **Steps 2 - 3** AIX script “fulextract” is executed which in turn jobs up the control processor uln.
- **Step 4** All previous extracted files are deleted from the LSDE directory in the aix003 server.
- **Step 5** Gets the Server address from lab global ^uextract.
- **Steps 6 - 8** Executes AIX script **extstart** to delete all \*.dat files from the Server and retrieve Site\_LastDateProcessed.txt date stamp file BUT DOES NOT READ IT.
- **Step 9** Five extract routines are jobbed and monitored (for completion flags) all tables and history files are created in LSDE lab directory /extract in the aix003 server.
- **Steps 10 - 11** When jobs complete control processor executes AIX script **tranfile** which transfers files in the LSDE /extract directory in the aix003 server via FTP to the Server.
- **Step 12** After file transfer has completed AIX script **verlist** is run to verify all files have been received by the Server.
- **Step 13** File FileList.txt is retrieved from the Server to verify all files were received.
- **14 – 15** AIX script **loadcom** is run to send the Site\_Last\_Load.dat file to the Server which signals the Server to start processing.

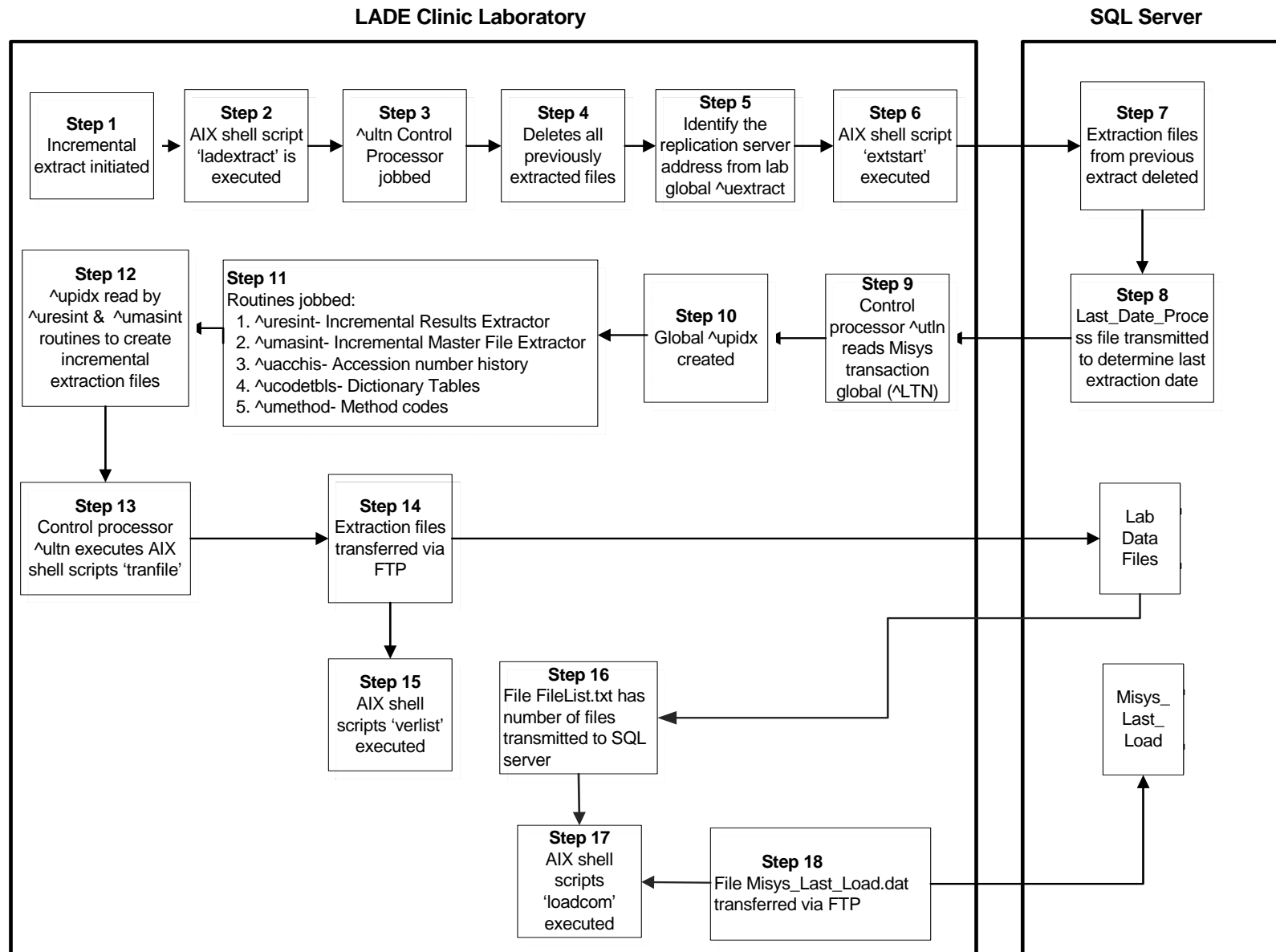
Full extract Function LSDE Options 2,1	Routine	Approximate run time
	full^ultn	28+ hrs -Monitors until all routine are done.
	Umaster	5 hrs 30 min.
	Uresall	20 hrs. 45 min.
	Ucodetbls	<30sec.
	Umethod	<10 sec.
	Uacchis	2hrs 30 min.

**Table 1**  
**Full Extraction Routine Run Times**

Routine	Description
umaster	Creates PatientMaster.dat file for all patients.
uresall	Creates multiple PatientResults.dat files for all patients
ucodetbls	Creates 16 data dictionary files (see table 2)
umethod	Creates TestMethod.dat for the method code dictionary (see table 3)
uacchis	Creates file AccessionHistory.dat (see table 3)

**Table 2**  
**Full Extraction Routines**

# INCREMENTAL EXTRACTION





## Incremental extract

### **Manual Start/Stop Incremental Extractions:**

Incremental extractions should not need to be initiated manually since the process is run daily. However, if for some reason SQL fails to process extracted files for a day or more, use LSDE function options 3,1 to enter the appropriate date incremental extraction start date. (see LSDE Utility Documentation for details).

### **Incremental Extract:**

#### **^ultn: Incremental Control Processor (See Figure 2 above)**

If the Tivoli monitor does not kickoff or fails, an email is sent to the individual setup in the LSDE Utility

- **Step 1** Tivoli monitor triggers AIX script **lsdextract** or it is started via SQ startup table
- **Step 2** AIX script **lsdextract** jobs control processor routine **^ultn**
- **Step 3** **^ultn** starts
- **Step 4** **^ultn** delete all \*.dat files from LSDE directory in the aix003 server.
- **Step 5** Get the server address from lab global **^uextract**.
- **Step 6 - 8** Executes AIX script **extstart** to delete all \*.dat files from the server and retrieve **Site\_LastDateProcessed.txt** file from the server which contains a date stamp. The date stamp is used to identify the number of days worth of data to extract. The default is one day.
- **Steps 9 and 10** Scan Site transaction global (**^LTN**) and creates scratch global **\*^upidx**.
- **Step 11** Start and monitor (for completion flags) five incremental extraction routines (see table 3). All incremental tables and history files are created in LSDE /extract directory .
- **Step 12** Routine **uresint** and **umasint** read global **^upidx** which they use to create incremental extraction files, **umasint** creates one **PatientMaster.dat**. **uresint** creates one file for every 3 million patient records as a result it creates multiple **PatientResults\_x.dat** files (x = 1, 2, 3... etc).
- **Step 13 and 14** Control processor executes AIX script **tranfile** which transfers files in the LSDE /extract directory in the aix003 server via FTP to the server.
- **Step 15 and 16** After file transfer has completed AIX script **verlist** is run to verify all files have been received by the Server.
- **Step 17 and 18** It then executes AIX script **loadcom** that sends the **Site\_Last\_Load.dat** file to the Server which signals the Server to start processing.

\*^upidx contains all the changed patient data.

<b>Incremental</b>	<b>Routine</b>	<b>Approximate run time</b>
Uln	uresint	2 hours 30 minutes
	Umasint	30 minutes
	ucodetbls	30 seconds
	umethod	10 seconds.
	Uacchis	2hrs 30 min.

**Table 3**  
**Routine Run Times**

<b>Routine</b>	<b>Description</b>
Uresint	Patients Results
Umasint	Patients Master File
Uacchis	Accession Number History
Ucodetbls	Dictionary Tables
Umethod	Method Codes

**Table 4**  
**Incremental Routines**

## **AIX Shell scripts**

The information listed below provides specific details about the AIX Shell Scripts utilized throughout the extraction logic. The same scripts are used for both the incremental and full extract processes. Note: Full extraction process retrieves but does not read Site\_LastDateProcessed.txt file to determine how many days worth of data to extract.

Note:

Tivoli monitor calls AIX script **LSDEExtract** for daily automatic incremental extractions.

LSDE Utility option 2 calls AIX script **fullextract** to manually start full extractions.

All scripts are located on AIX Lab system in the LSDE /extract directory.

<b>AIX Shell Script</b>	<b>Function</b>	<b>Calling Routine</b>
<b>LSDEExtract</b>	Starts control processor ultn for incremental extractions.	Tivoli monitor
<b>Extstart</b>	Retrieves Site_LastDateProcessed.txt date file from the Server and deletes all .dat files from the Server. This routine is the only routine that deletes the .txt or .dat files from the Server environment. There are no other temporary files being created that requires deletion.	Ultm
<b>Tranfile</b>	Transfers all extraction files from the LSDE /extract directory in the aix003 server to Server via FTP.	Ultm
<b>Verlist</b>	Retrieves a record of all files from the Server after FTP for verification.	Ultm
<b>Loadcom</b>	Sends Site_Last_Load.dat start process file to Server.	Ultm
<b>Fulextract</b>	Calls full^ultm to initiate a full extraction. This routine is run from the LSDE utility, option 1. (see LSDE Utility Documentation for details).	LSDE Utility

**Table 5**  
**AIX Scripts**

## **Routines:**

The information listed below provides specific details about the routines utilized throughout the extraction logic.

### **Control Processor**

ultn

Controls entire extraction process by calling all data and table extract routines and monitors them until they run to completion. It also calls all the AIX scripts (see table 5 **AIX Scripts**)

ful^ultn

Controls the full extraction process by calling all data and table extract routines and monitors them until they run to completion. This routine is executed from the LSDE Utility provided to Site Health.

### **Routines Executed for Full Extract Only:**

umaster

Is run when a full restore is required. Program extracts all Site patient master records and creates file PatientMaster.dat. A full extraction can only be manual started using lab system function LSDE option 2 "Start Full Extract" (see LSDE Utility Documentation for details).

uresall

Called by ultn control Processor when a full extraction has been initiated, uresall extracts all patient test results creating multiple PatientResults\_x.dat files.

### **Routines Executed for Incremental Extract Only:**

uresint

Called by ultn. Creates incremental file PatientResults\_x.dat (see table 3).

umasint

Called by ultn. Creates incremental file PatientMaster.dat (see table 3).

### **Routines Executed for both Full & Incremental Extracts:**

ucodetbls

Called by ultn. Creates 16 data dictionary files (see table 2).

umethod

Called by ultn. Creates TestMethod.dat for the method code dictionary (see table 3).

uacchis

Called by ultn. Creates file AccessionHistory.dat (see table 3).

## **Error Messages:**

To ensure process and data dependability, all programs have standard error trapping implemented as part of their functionality. Error trapping forces programs that fail to execute a error routine which records errors and or critical informational messages. These errors/messages are then \*e-mailed to support staff before the program can halt.

There are almost an unlimited number of error messages the can be generated as a result of a program error. Error trapping provides Site staff with a clue as to what caused the problem which prevented the program from running to completion.

Below are examples of email message that can be generated by program error trapping functionality.

<b>Subject</b>	<b>Message</b>	<b>Action to Take</b>
LSDE Results Incremental Extractor Error	Routine uresall failed _\$ze	Critical Error Call LSDE Support
LSDE Data Dictionary extract Error	Routine ucodetbbs failed _\$ze	Critical Error Call LSDE Support
LSDE Patient Master Extractor Error	Routine umasint failed _\$ze	Critical Error Call LSDE Support
LSDE Full Results Extractor Error	Routine uresall failed _\$ze	Critical Error Call LSDE Support

**Table 6**  
**Sample Error messages**

NOTE: \$ze = actual program error that occurred, used by LSDE program support to analysis program errors. There are almost an unlimited number of program errors that can occurs. \$ze contain the error message that caused the error.

However informational messages are limited to critical operations that failed or those operations the control processor was unable to verify.

Below is a complete list of email informational messages.

Subject	Message	Description	Person(s) Notified	System Action to be Taken
LSDE Control Processor Extractor Error	Control Processor Has Been Disabled	Daily incremental extract program cannot run until it is "Enabled"	Site lab staff	Call Site Lab Staff. They should use Site lab function LSDE option1 to re-enable the control processor (see LSDE Utility documentation). The enabling of this option allows the daily incremental program to run automatically the next day or the staff can run the LSDE Utility routine manually right away.
	Failed to remove data files from LSDE /extract directory on the aix003 server	Fails to remove *.dat files from previous incremental extraction. Most likely cause by incorrect file permissions	AIX system support.	Call LSDE System Support Team. The Support Team should check to see if files are in LSDE /extract directory on the aix003 server. If files exist, then these files need to be manually deleted. If any files are locked, the AIX (Unix) System Administrator needs to unlock the file and then the file needs to be manually deleted. File permissions at the directory level should be checked to ensure that the utility program has delete capability to perform the delete function (this is a rare occurrence). No additional activity needs to happen, the next day the incremental program will start and continue the process. This process can also be started manually using the LSDE Utility.

Subject	Message	Description	Person(s) Notified	System Action to be Taken
	Failed to remove file FileList.txt from LSDE /extract directory on the server	<p>FileList.txt file is used by the control processor to confirm all files sent to Server were received.</p> <p>Causes are –</p> <ul style="list-style-type: none"> <li>- File is missing and cannot be deleted or file permissions are incorrect. This is a non-critical error, extraction programs continue to run.</li> </ul>	AIX system support/ SQL System Support	<p>This typically happens, if the Server goes down, while the incremental process is updating the files. To rectify the problem, the LSDE System Support Team should check to see if the server is up. And then delete all the .dat files from the Server. Check the file permissions (as explained in the previous step) to ensure that the utility has file deletion capability. No new action is required. If the server is up, the next time the incremental process runs, it will copy the files and continue its processing.</p> <p>Staff can also confirm all files were received, FTP those that were not sent, and then manually run the SQL process.</p> <p>SQL Support is listed just to verify that there was nothing specific to SQL that needs to be addressed based on the review of the log file.</p>

Subject	Message	Description	Person(s) Notified	System Action to be Taken
LSDE Control Processor Extractor Error	Failed to execute Unix shell extstart on server xxxx.  xxxx = is the server where SQL is installed	Causes – <ul style="list-style-type: none"> <li>• Script extstart does not exist</li> <li>• Server is down</li> <li>• Unable to login to server</li> <li>• File transfer failed</li> <li>• Site_LastDateProcessed.txt missing or file permissions are incorrect</li> <li>• General FTP file transfer errors</li> </ul>	AIX System Support Team/ SQL Support Team	<p>Call AIX Support Team. The team needs to check for the following:</p> <ul style="list-style-type: none"> <li>• Script exists – if that script does not exist, then that script needs to be reinstalled from the backup.</li> <li>• Server is down – then system or network routines needs to be run to bring the server back to an operational state.</li> <li>• Unable to login to server – same as above.</li> <li>• File transfer failed – same as above, this happens because of either system or network.</li> <li>• General FTP file transfer error – same as above, this happens because of either system or network problem.</li> <li>• Missing Site_LastDateProcessed.txt – Call SQL Support Team to verify why the process failed. This team will needs to verify the log file to understand what caused this failure, rectify problem and rerun the process.</li> </ul> <p>When the problem is rectified and fixed, no additional action is required, since the incremental process will start and continue processing. Staff can run the incremental extract manually using the LSDE Utility.</p>
	Failed to open file Site_LastDateProcessed.txt	This file is created by the SQL process. If this file is missing, then that file cannot be read by the control processor. The Site_LastDateProcessed.txt is read by control processor to determine extract start date.	SQL Support Team	<p>Call the SQL Support Team. This team will needs to verify the log file to understand what caused this failure, rectify problem. SQL Support Team will have to recreate this file manually with the date when the files were last processed (specification for this file is provided) so that next day the control process can use this file.</p> <p>Staff can run the incremental extract manually using the LSDE Utility.</p>
Lab System Data	Extraction Support			



Subject	Message	Description	Person(s) Notified	System Action to be Taken
LSDE Control Processor Extractor Error	Failed to execute Unix shell verlist that verifies file transfer	Transfers FileList.txt file from the Server to AIX and is used to verify if the file was transferred to the server.	AIX System Support/ SQL Support Team/ LSDE (MUMPS) program support	<ul style="list-style-type: none"> <li>Script verlist does not exist – Call MUMPS Program Support to reinstall script</li> <li>Server is down - Call AIX System Support to check the cause for the server being down.</li> <li>FileList.txt missing or file permissions are incorrect - check server file permissions</li> </ul> <p>SQL Support is listed just to verify that there was nothing specific to SQL that needs to be addressed based on the review of the log file. The Site_Last_Load.txt can be manually created, FTP to the Server and then restart the SQL process.</p>
	Patient Results file transfer Failed, Verified 8 out of 9 sent	<p>Nine (9) files were sent to the Server but only 8 were verified.</p> <ul style="list-style-type: none"> <li>Server went down during file transfer</li> <li>File FileList.txt is corrupted</li> </ul>	LSDE program support	<p>Call LSDE Support Team. The team should manually FTP all the PatientResults_x.dat files from AIX to the Server using the same username and password that the script uses. The Site_Last_Load.txt can be then manually created, FTP to the Server and then restart the SQL process.</p>
	Failed to open file Site_Last_Load.txt	File permissions are incorrect. This is the file that signals SQL to begin processing, this error signals that the SQL process could not read the file.	AIX system support	<p>Call AIX System Support Team. This team needs to check file permissions in the AIX system to ensure the SQL process can access the file. The Site_Last_Load.txt can be then manually created, FTP to the Server and then restart SQL process.</p>

Subject	Message	Description	Person(s) Notified	System Action to be Taken
LSDE Control Processor Extractor Error	Site_Last_Load.txt that signals server "_server_" to start processing Failed server = nameserver for Server	This typically happens when you switch servers in the middle of the FTP process and the servers are being switched while the process is running, you will run into this type of an error.  Transfers Site_Last_load.txt file from AIX to Server that is used to signal the server to start processing.	AIX System Support/ SQL System Support/ LSDE (MUMPS) program support.	<ul style="list-style-type: none"> <li>• AIX script loadcom does not exist - Call MUMPS Program Support to reinstall script</li> <li>• Server is down - Call AIX System Support to check server availability.</li> <li>• Unable to login to server - Call SQL System Support to check server availability</li> <li>• File transfer failed or General FTP file transfer errors - Call AIX Support Team to verify that it is not a system or network issue.</li> <li>• The Site_Last_Load.txt can be then manually created, FTP to the Server and then restart the SQL process.</li> </ul>
Terminal mumps program errors	Control Processor (ultn) "_\$ze	<p>Critical Error Below are examples of program errors</p> <p>Routine uresall failed &lt;UNDEFINED&gt;bat+1^uresint</p> <p>Control Processor (ultn) &lt;UNDEFINED&gt;verftp1+3^ultn</p>	LSDE (MUMPS) Program Support	Since this is a critical error because of program errors. These are error traps. Call MUMPS Program Support to review and fix program errors

### **Data Extraction Files**

#	Global	Description	File
1	^LAB	Patient Master File Patient Demographics	PatientMaster.dat
2	^Multiple Globals	Patient Tests Results	PatientResults.dat
3	^LBI(1,TS,"Q")	Method code Dictionary	TestMethod.dat
4	^ANH	Accession History File	AccessionHistory.dat

**Table 9**  
**Extraction Files**

### **Text Files**

#	Source	Description	File
1	LSDE lab system	Signal for Server to start processing Incremental data files	Site_Last_Load.txt
2	Server	Date stamp file that tell incremental processor what date to began extraction	Site_LastDateProcessed.txt
3	LSDE lab system	List of all files sent to Server.	Filelist.txt

**Table 10**  
**Control Text Files**

### **Data Dictionaries**

<b>#</b>	<b>Global</b>	<b>Description</b>	<b>File</b>
1	^LBI(1,CD)	Battery Dictionary	BatterycodeList.dat
2	^LBI(1,TS)	Test Dictionary	TestcodeList.dat
3	^LBI(1,CD)	Package Dictionary	PackagecodeList.dat
4	^LBI(4)	English Text Translations	TextTranslation.dat
5	^LBI(5)	Laboratory Department Dictionary	LabDept.dat
6	^LBI(6)	Laboratory Location Dictionary	LabLoc.dat
7	^LBI("6A")	Performing Laboratory Dictionary	PerformLab.dat
8	^LBI(8)	Diagnosis Dictionary	DiagnosisCode.dat
9	^LBI(9)	Tube Type Dictionary	TubeType.dat
10	^LBI(10)	Reference Lab Dictionary	RefLab.dat
11	^LBI(11)	Tech Code Dictionary	TecCode.dat
12	^LBI(12)	Location Code/Room Number Dictionary	LocationRoomList.dat
13	^LBI(12E)	Event Type Maintenance	EventType.dat
14	^LBI(12H)	Hospital Id Codes for Multiple Hospital	HospitalList.dat
15	^LBI(13)	Physician Dictionary	DoctorList.dat
16	^LBI(1,TS,"Q")	Normal Range Dictionary	NormalRange.dat

**Table 7**  
**Data Dictionary Lab Files**

## File Specifications

### Test/Battery/Package Dictionary Battery

Purpose: Description of all associated parameters for all batteries, which are defined for the laboratory.

File Name: **BatterycodeList.dat**

Battery Code	*Test Type "B"	Test Name	Laboratory department code	Orderable (Y/N)	Reference Lab Code	Alternate Name	Tests Comprising Battery	***Result At Request Entry
--------------	-------------------	-----------	----------------------------	-----------------	--------------------	----------------	--------------------------	----------------------------

\*\*\*Result at Request Entry

Y = yes

M = yes not mandatory

Null = no mandatory

**Sample Record:** Battery CHD has four tests SDES, SREQ, PROBE and RPT

CHD|B|Chlamydia, DNA Probe|MC|Y||SDES|M

CHD|B|Chlamydia, DNA Probe|MC|Y||SREQ|M

CHD|B|Chlamydia, DNA Probe|MC|Y||PROBE|N

CHD|B|Chlamydia, DNA Probe|MC|Y||RPT|N

### Test/Battery/Package Dictionary Test

Purpose: Description of all associated parameters for all tests, which are defined for the laboratory.

File Name: **TestcodeList.dat** contains 14545 records, 10288 tests, 4085 batteries and 171 packages.

Test Code	*Test Type "T"	Test Name	Laboratory Department Code	**Orderable	Reference Lab Code	Result units	Number Of Decimal Places
-----------	-------------------	-----------	----------------------------	-------------	--------------------	--------------	--------------------------

Result At Request Entry	Automatic Answer	***Worksheet Pointers (Null)	Result Suppression Flag ("Y" Or Null)	Sensitive Test Flag ("Y" Or "N")	Alternate Test Name
-------------------------	------------------	------------------------------	---------------------------------------	----------------------------------	---------------------

\*Test type

T = individual test

B = Battery

P = Package

\*\*Orderable (Y/N/B/T)

B = bill only, no report

T = text lookup only

\*\*\*Worksheet Pointers

There can be multiple lines this field will be null for now.

**Sample Record:**

ABUDSN|T|Drug Scrn UR, Neonate|C|N|C||N||N|Alta Bates Test

## Test/Battery/Package Dictionary Package

Purpose: Description of all associated parameters for all packages, which are defined for the laboratory.

File Name: **PackagecodeList.dat**

Package Code	*Test Type <b>"P"</b>	Test Name	Orderable (Y/N)	Alternate Name	Tests Comprising Package
--------------	--------------------------	-----------	-----------------	----------------	--------------------------

### Sample Record:

MEDIC|P|DIC Panel - SCH Only|Y|Coag Survey II - SCH Only|PT  
MEDIC|P|DIC Panel - SCH Only|Y|Coag Survey II - SCH Only|PTT

## English Text Translations

Purpose: Translations of all predefined English language comments and free text results currently active in the system.

File Name: **TextTranslation.dat** containing 15150 records

English Text Code	English translation	* Inactive code	**Type Code
-------------------	---------------------	-----------------	-------------

\* Inactive code

0 = active

Date inactivated

### \*\* Type Code

AS = anatomic pathology specimen code  
BC = blood bank comment  
BS = blood bank status  
BT = blood type code  
BW = Blood Bank Workload Code  
C = quality control code  
D = department specific code  
G = group text code  
M = quality control method  
MG = Microbiology Organism Group Code  
MO = Microbiology Organism Code (individual)  
MM = Microbiology Media Code  
MS = Microbiology Sensitivity Type Code  
MW = Microbiology Workup Workload Code  
P = priority code  
Q = quality control modifier  
R = QC (Westgard) rule code  
S = specimen type code  
T = test result code  
(used to allow specific listing of these codes)  
W = workload recording code  
X = exclude from CAP workload

### Sample Record:

A12800|12800IU/mL.Suggestive of recent streptococcal infection|0|D|

## Laboratory Department Dictionary

Purpose: Used for association of tests and batteries with the appropriate department and for the organization of certain data for reporting purposes.

File Name: **LabDept.dat** Containing 22 records

Department Code	Department Name
-----------------	-----------------

**Sample Record:**

AP|ANATOMIC PATHOLOGY

## Laboratory Location Dictionary

Purpose: It is used to identify the location of terminals from which orders are placed. This may be used for assignment of tests to worksheets, printing of receipt labels, etc

File Name: **LabLoc.dat** Containing 2069 of which 56 contain duplicate review parameters,

Location Code	Location Name	Duplicate Review Parameters	Repeat Password Parameter	Hospital ID	Track Location	Telephone Number	Callback Comments
---------------	---------------	-----------------------------	---------------------------	-------------	----------------	------------------	-------------------

**Sample Record:**

MESP|SSME - 5301 F ST <14,985,98177,,SS,22,482>||900|ME|MEPSC|7331490|FAX: 733-1491

## Performing Laboratory Dictionary

Purpose: Performing laboratories identify physical facilities

File name: **PerformLab.dat** Containing 422 records

Performing Laboratory	Name of performing laboratory	License number	Director's name	Performing Laboratory reporting code	*Active flag	Address
-----------------------	-------------------------------	----------------	-----------------	--------------------------------------	--------------	---------

**\*Active flag**

0 = I: Inactive

1 = C: Active, Code (flag to indicate display of only the Code on reports)

2 = T: Active, Text (flag to indicate display of only the text on reports)

3 = B: Active, Both (flag to indicate display of both the Active status code and the translation of that code on reports).

**Sample Record:**

CD|CPMC Davies Lab||Richard Garcia-Kennedy|CD|3| CPMC Davies Lab, Castro & Dubose, SF, CA 94115

## Diagnosis Dictionary

Purpose: Used to define diagnostic codes for translation to the appropriate English text. Each code is assigned a 5 character alphanumeric code.

File name: **DiagnosisCode.dat** containing record 20385.

Diagnosis Code	Diagnosis Data
----------------	----------------

**Sample Record:**

66.39|XXBILAT FALL DESTR NEC

## Tube Type Dictionary

Purpose: Tube types which print on labels. It also specifies the maximum amount that the tube will hold so that the number of tubes required can be calculated.

File name: **TubeType.dat** Containing 116 records

Tube Type code	Tube type name	Volume in mls	Tube type group
----------------	----------------	---------------	-----------------

**Sample Record:**

BC|BldCult|10|MC

## Reference Lab Dictionary

Purpose: Identifies all reference labs which tests are sent to.

File name: **RefLab.dat** containing 11 records

Ref. lab mnemonic code	Reference lab name	Address
------------------------	--------------------	---------

**Sample Record:**

Q|Quest Laboratory|33608 Ortega Highway



## Tech Code Dictionary

Purpose: Translation between the numerical tech code and the name.

File name: **TecCode.dat** containing 4822 records

Tech code	Tech name	*Tech Code Active
-----------	-----------	-------------------

\*Tech Code Active

Y = active tech

N = deactivated tech

**Sample Record:**

37|TEST,FUNCTIONS|Y

## *Location Code/Room Number Dictionary*

Purpose: Valid patient locations and room numbers

File name: **LocationRoomList.dat** that contains 9496 records

location code	Hospital ID	*Location Type	**Location Active Flag	Location name text
---------------	-------------	----------------	------------------------	--------------------

\* Location Type definitions

1 = inpatient

2 = outpatient

3 = emergency room  
(treated as outpatient in file structure)

4 = outside  
(treated as outpatient in file structure)

G = group of locations

R = room number definition

\*\* Location Active Flag 0 = Inactive

**Sample Record:**

MARH|8|4|1|MERCY AMERICAN RIVER HOSPITAL

## Event Type Maintenance

Purpose: Contains the definitions of events types within hospital id.

File name: **EventType.dat** containing record 21.

Hospital ID	Event Type	Description	Group ID	Status A = Active I = Inactive	*Event Time Frame Duration	Number of days after close to deactivate for orders
-------------	------------	-------------	----------	--------------------------------------	----------------------------	---

\*Valid values are:

1D, 1W, 2W, 1M, 2M, 3M, 4M, 6M, 12M value can also be null.

D = day W = week M = month

### Sample Record:

10|ER|Emergency Room|3|A|1D|0

10|P|Inpatient|1|A|1D|0

10|OP|Outpatient|2|A|1D|90

10|OS|Outside|4|A|1D|0

## Hospital Id Codes for Multiple Hospital Sites

Purpose: Contains the definitions of all the hospital id's that will be accessing the system.

File name: **HospitalList.dat** contains 23 records

Hospital ID	User Hospital ID	Hospital name	Active/Inactive Flag A = active I = inactive	Commercial Lab Availability Flag A = active I = inactive
-------------	------------------	---------------	--	--

### Sample Record:

11|SR|SITE SANTA ROSA MEDICAL CENTER AND SITE WARRACK HOSPITAL|A|

## Physician Dictionary

Purpose: Alphabetical and numerical physician data. Inside physician codes are alpha numeric (1UN.UN). Outside physician codes are numeric preceded with the letter O (1"O"1N.N)

File name: **DoctorList.dat** containing 102089 records with 9851 as outside Physician.

Physician Code	Hospital ID	Physician Name (Last, First MI)	Address	City, State Zip	Primary Phone	Active Status A=active I=inactive
UPIN	Primary Location	Primary Location Hospital ID	HIS Physician	Hospital ID		

### Sample Record:

1051|4|MILLER, CATHERINE|1750 EL CAMINO REAL|(650) 692-0182|A||A43796|||

## ***Normal Range File***

**Purpose:** File is used as a lookup to calculate normal range for results for individual tests base on sex and age of the patient.

**File Name:** **NormalRange.dat**

Metcode (metcode) 1	Test (tst)	Start Date (sdx)	End Date (ed)	Age (age)	Sex (sex)	Normal Range (nran)	Units (un)	Technical Range (tr) 9
------------------------	---------------	---------------------	------------------	--------------	--------------	------------------------	---------------	---------------------------

**Sample Record:**

MPNEP|A1AT|01/13/2007|10/14/2114|0|F|90-200|mg/dL|4-520

Patient Master File

ation File contains patient demographics, room location, admit date etc.

dat

- Search transaction global (new patients and updates)  
s.dat - Search transaction global (updated patient record, updated or new results)  
are recreated  
dat

tal	Last Name (lname)	First Name (fname)	Middle Name (mname)	Date of Birth (dob)	Sex (sex)	Social Security Number (ssn)	Patient Internal Identifier (pidx) 9
Patient State (patstate)		Patient Zip (patzip)	Phone Home (patphone)	Patient Location Hospital ID (patlid)		Patient Location (patloc)	Patient Room (patroom) 17
Discharge Date (disdate)	Diagnosis (diag) ^LBI(8) DiagnosisCode.dat		Admit Phys 1 (adphy1) ^LBI(13) DoctorList.dat		Admit Phys 2 (adphy2) ^LBI(13) DoctorList.dat	Admit Phys 3 (adphy3) ^LBI(13) DoctorList.dat	Admit Tech Code (amtec) ^LBI(11) TecCode.dat 25

ds:

BA|YUMIKO||19580208|F|564-76-7924|137P10006|5115 SHALIMAR CIR -|FREMONT|CA|9  
420|20|HMO||OP|05-17-2004|05-23-2004|||||0

## Patient Results File

Purpose: Contains patient test results and various lab related data.

File Name: **PatientResults.dat**

Order Level record file **PatientResults.dat** [Full extraction](#) [12hrs. 45mins incremental extraction](#)

Medical Record Number (mrn) 1	Hospital ID (hid) <a href="#">^LBI(12H)</a> HospitalList.dat	Site Order # (ordx)	Order Physician Code (orderphy) <a href="#">^LBI(13)</a> DoctorList.dat	Hospital ID (phyhid) <a href="#">^LBI(12H)</a> HospitalList.dat	Ordering Account Number (orderacc)	Ordering Lab Code (orderlab) 7 <a href="#">^LBI(6)</a> LabLoc.dat
Patient Ordering Location (patorderloc) 8 <a href="#">^LBI("6A")</a> PerformLab.dat	Order Tech (ordertec) <a href="#">^LBI(11)</a> TecCode.dat	Billing Code "Null" (billcode)	Order Date & Time (ordatetime)	Order Comment (ocomment) <a href="#">^LBI(4) 12</a> TextTranslation.d at		

### Example record 12 fields:

[85272964](#)|[16](#)|[137L343](#)|[00004](#)|[16](#)|[654987](#)|[MNML](#)|[MP](#);[16](#)|[16000](#)||[05-14-2004 10:50:00](#)|[137C712](#)|F1  
054|||[05-14-2004 10:00:00](#)|[16000](#)|[05-14-2004 10:01:00](#)|CA|C|||0|1|C||S25747-0|||  
CA|0|0|1|CA|16000|MNML|05-14-2004 10:52:27|MNRI8|MN||9.9|||8.8-10.2|

Accession Level Record 6

Accessio n (acc) 13	Specimen Type (spectype) <a href="#">^LBI(4)</a> TextTranslati on.dat	Drawing Tech (collecttec) <a href="#">^LBI(11)</a> TecCode.da t	Date & Time Collected (cdatetime)	Received Tech Code (rectec) <a href="#">^LBI(11)</a> TecCode.dat	Received Date & Time (reccdatetime) 18
------------------------------	--	--	---	--	---

### Example record 6 fields:

[85272964](#)|[16](#)|[137L343](#)|[00004](#)|[16](#)|[654987](#)|[MNML](#)|[LAB-M](#);[16](#)|[16000](#)||[05-14-2004 10:50:00](#)||F1  
[054](#)|SPUT||[05-14-2004 10:00:00](#)|[16000](#)|[05-14-2004 10:01:00](#)|CA|C|||0|1|C||S25747-0|||  
CA|0|0|1|CA|16000|MNML|05-14-2004 10:52:27|MNRI8|MN||9.9|||8.8-10.2|

## Patient Results File Contd.

### Battery Level data

Battery Code (btocode) 19 ^LBI(1, BatterycodeList.dat	Battery Dept (batdept) ^LBI(5) LabDept.dat	Date & Time Cancelled (btdattm)	Order Diagnosis (orderdiag) ^LBI(8) DiagnosisCode.dat	Order Comments (btordercom) ^LBI(4) TextTranslation.dat	Cancel Event (btordercan) Tech Code If canceled	Credit Event (creditev) 25 Y or N
Lab Department (btlabdept) 26 ^LBI(5) LabDept.dat	Priority Codes (btpriority)	Order Cancelled Reason (btcanreason)	HIS Order Number (hisorder)	Ordering Rank (ordrank) 30		

### Example record 12 fields:

85272964|16|137L343|00004|16|654987|MNML|LAB-M;16|16000||05-14-2004 10:50:00||F1  
054|||05-14-2004 10:00:00|16000|05-14-2004 10:01:00|CA|C|||0|1|C||S25747-0|||  
CA|0|0|1|CA|16000|MNML|05-14-2004 10:52:27|MNRI8|MN||9.9|||8.8-10.2|  
|||CA|0|0|1|CA|

### Test Level data

Package Code (paocode) 31 ^LBI(1 PackagecodeList.dat	Battery Code (btocode) ^LBI(1 BatterycodeList.dat	Test Code (tst) ^LBI(1 TestcodeList.dat	Package_Qty (packqty)	Battery_Qty (batqty)	Test_Code_Qty (testqty) 36
---	--	--	--------------------------	-------------------------	-------------------------------

### Example record 6 fields:

85272964|16|137L343|00004|16|654987|MNML|LAB-M;16|16000||05-14-2004 10:50:00||F1  
054|||05-14-2004 10:00:00|16000|05-14-2004 10:01:00|CA|C|||0|1|C||S25747-0|||  
CA|0|0|1|CA|16000|MNML|05-14-2004 10:52:27|MNRI8|MN||9.9|||8.8-10.2|

## Patient Results File Contd.

Results data level

Result Code Mnemonic (tst) 37 ^LBI(1 TestcodeList.dat	Result Tech (resulttech) ^LBI(11) TecCode.dat	Result Lab Code (resultlab) ^LBI(6) LabLoc.dat	Resulted Date & Time (resdattm)	Method Code (metcode )	Performing Lab Code (perflab) ^LBI("6A") PerformLab.dat	Result QA flags (resqa) 43	Free text Limited to 2000 chars (restxt) 44		
Current History Flag Current Results = "C" Histroy Results = "H" (curhis) 45		Copy to Physician 1 (cphy(1)) ^LBI(13) DoctorList.dat	Copy to Physician 2 (cphy(2)) ^LBI(13) DoctorList.dat	Copy to Physician 3 (cphy(3)) ^LBI(13) DoctorList.dat	Normal Range (norrage)	Supervising Physician (supply) 50	Test Results (tstres) 51		
Comments Append comments as needed (rescom) 52		Site_Order_Event_Mask _Flags_Lookup (evemask)	Container ID Battery (bcidtt)	Container ID Stat Battery (bcidstt)	Container Id Results (ctidtt)	Container Id Stat Results (ctidst)	Sign in date and time Null 58	Patient Location (patloc)	Patient Room (patroom) 60

### Example record 14 fields:

85272964|16|137L343|00004|16|654987|MNML|LAB-M;16|16000||05-14-2004 10:50:00||F1  
054|||05-14-2004 10:00:00|16000|05-14-2004 10:01:00|CA|C|||0|1|C||S25747-0|||  
CA|0|0|1|CA|16000|MNML|05-14-2004 10:52:27|MNRI8|MN||9.9|C|||8.8-10.2|

### ***Test Method Code File***

Purpose: Contains individual test method code descriptions.

File Name: **TestMethod.dat**

Method Code file **TestMethod.dat**

Test Code (tst)	Method Code (met)	Method Code Description (nam)	Validate From Date (valfrom)	Validate To Date (valto)
--------------------	----------------------	----------------------------------	---------------------------------	-----------------------------

**Sample Record:**

ABA1C|CORMHA|Performed at: MEMORIAL MEDICAL CENTER, 1700 Coffee Rd, Modesto, CA  
95355|06-09-2006|01-11-2001

### ***Accession History File***

Purpose: Used to keep the history of all orders, results and credits on an accession number .

File Name: **AccessionHistory.dat**

Accession number history file **AccessionHistory.dat**. Run time 15 minutes Numbers of records 3,837,259 processed  
File size 1,610,034,824

Patient Internal ID (pidx) 1	Order # (ordx)	Accession number (an) 1	Lab Location (labloc)	Order Code (ocode)	Type Code (typec)	Tech Code (techcode)
------------------------------------	-------------------	-------------------------------	--------------------------	--------------------------	----------------------	-------------------------

**Example record 8 fields:**

129P94754|169L27098|A1098|RVML|URC|2|9640|

### ***Site Last Load File***

Purpose: After the file transfer has been completed and verified, Site\_Last\_Load.txt is sent to the Server by control processor ultn which signals the server to start processing. If for any reason the extract process fails this file is not send to the server.

File Name: **Site\_Last\_Load.txt**

Switch signal "Y" = Process	Number of extracted Patients
--------------------------------	------------------------------

**Example record 2 fields**

Y|8640315



### ***Site Last Date Processed File***

Purpose: This file is retrieved from the Server and contains a date and time stamp. The date stamp tells the processor what date to began incremental processing. The time stamp is not used.

File Name: **Site\_LastDateProcessed.txt**

File Header	
Date Stamp	Time

#### **Example record header and 2 fields**

#PROCESSED\_DATE  
03/19/2007 06:01:00

### **FTP File Transfer Confirmation File**

Purpose: File is used by the control processor to confirm the number of result files sent to SQL.

File Name: **Filelist.txt**

Date	Time	Size	Name
------	------	------	------

#### **Example record**

01/02/2008|11:03 AM|3,200,768| PatientResults\_x.dat

## ***Transaction Global File Specification***

Transactions are created for each change that occurs in the lab system. These transactions are written to global ^LTN. The lab system transaction processor continually reads this file and processes any pending transactions.

LSDE transaction processor (ultn) processes global ^LTN looking for the change transaction types listed in bold below. When a change transaction type is identified, transaction processor (ultn) extracts the patient's Patient Internal Identifier (PIDX). The PIDX is stored temporally in global ^Upidx.

Incremental processors uresint (patients results) and umasint (master patient records) process ^Upidx global.

The following transaction types are processed by the LSDE transaction processor (ultn):

### ***Type 4 : Lab ID Entry***

Definition: Admit to the lab or hospital, Delete lab packages, batteries or tests and Transfer hospital, room locations, etc. If this type of transaction is found, the entire Patient record is extracted.

^LTN(tdj,tpr,tsq) d3 delimiter  
dat="4!!0#18##SC1I7EJ1##272073;14|889818#GAP6##11309\168P47686\272073\14\CH\0"  
Piece 1 : Header Record  
2 : patient PIDX (permanent internal patient number)

### ***Type 5 : Battery Requisition/Accession Number Modification***

Definition: Test, battery and package order transactions

^LTN(tdj,tpr,tsq) - ORDX Level (Primary level for LTN) d3 delimiter  
dat="5!!0#18C#/dev/null:3678210#SC1I8BJ1##0016657;8|889819#(GOP5) OE INTERFACE#0  
#11309#\N\168P46505#168L245070\39537;8\1,0016657,8,EP,1,2\48979173\MER\4EAS;8\1  
2172006,0200\"  
Piece 4 : internal patient id  
Piece 1 : patient number (PIDX)  
2 : order number (ORDX)

### ***Type 7 : Results***

Definition: Transactions are created whenever lab results are available.

^LTN(tdj,tpr,tsq) d3 delimiter  
dat="7!B!0#458C#RVGA#1157104#9121##BOP##11309\RVML\168L245047\S779168\TS\3\111111"  
Piece 1 : Header Record  
3 : order index (ORDX)  
4 : accession number (AN)  
5 : order code (BT)

### **Type 8 : Deletions of tests**

Definition: Transactions are created whenever a Test, battery or package is canceled.

^LTN(tdj, tpr, tsq) - an Level (Primary level for LTN) d3 delimiter  
dat="8!B!0#715#SR023#4921170#17021##CR##11309\S776899\SRR"  
Piece 1 : Header Record  
2 : accession number

### **Type 9 : Patient Data Merge**

Definition: Merge-from patient number (old MRN) merge-to patient number (new MRN) transactions

dat="9\71500\EDBAT\5298146#ED01#####98988207;21!1002046|674374;21!1002063\98988207;21\674374;21\1\\\16  
8P47769\168P47768\\\#"  
Piece 5 : merge-from patient number (old MRN) d1 delimiter  
6 : merge-to patient number (new MRN)  
7 : delete flag  
13 : old PIDX  
14 : new PIDX

### **Type 16 : Special Patient Data**

Definition: Hospital specific patient transaction data set in global ^REG("site")

dat="16\18\dev/null:3678210(GOP5)\SC1I8BJ1\0016657;8" d1 delimiter  
Piece 5 : patient's hospital number

^REG is used to maintain special data related to the patient. Currently the primary node is the site, emphasizing the site specific nature of this file.

User Defined Field Definitions

^REG(SITE,PN  
- 0000000;3  
-- 1 = 510-471-0186 PT. PHONE  
-- 2 = 34276 TORREY PINE LN. - PT. ADDRESS  
-- 3 = UNION CITY PT. CITY  
-- 4 = CA PT. STATE  
-- 5 = 94587 PT. ZIP  
Etc

See "Special reg fields.doc" for global record layout.

## ***LSDE Utility Documentation***

The LSDE Utility documentation provides detailed information including screenshots for the following lab extraction features:

1. Enable/Disable Incremental Processor  
Turns daily incremental processing on and off.
2. Start /Stop Full Extract  
Starts and stops extraction of all results, master files, tables, and dictionaries from the Site lab system.
3. Start/Stop Incremental Extract  
Manual start or stop daily incremental extraction.
4. Set Server  
Sets server that receives extracted files (TEST, QA, Production).
5. Set Error Email Address  
Assign email address where error notification will be sent.
6. Show Summary  
Shows current parameter settings.
7. Monitor Incremental Extract  
Monitors progress and status of incremental extract process.