1 Purpose
This document presents the level 3 requirements of the GLAST LAT I&T Online system.

2 Scope
The scope of this document is to cover ...

3 Definitions

3.1 Acronyms
API - Application Program Interface
cPCI - Compact PCI
COMM - COMmunications Module
EEPROM - Electrically Erasable Programable Read Only Memory
FITS - Flexible Image Transport System - Standard NASA/HEASARC data file format.
GUI - Graphical User Interface  
IDL - Interactive Data Language - Research Systems data analysis and visualization package.  
LAN - Local Area Network  
LAT-COMM - Large Area Telescope - COMunications Module  
MB - Mega Bytes  
NFS - Network File System  
PCI - Peripheral Communications Interface  
PPC - Power PC  
RAM - Random Access Memory  
ROOT - CERN/GLAST supported data analysis and visualization package.  
RTE - Real Time Engine  
SBC - Single Board Computer  
SCL - Spacecraft Control Language  
TEM - Tower Electronics Module  
TCP/IP - Transmission Control Protocol/Internet Protocol  
VME - Versa Module Eurocard  

3.2 Definitions  
DataIO - The program that receives and decommutates the telemetry data.  
Flash Memory - A type of memory that retains its contents when powered off. It is normally read only but can be written by a special software procedure.  
Network router - A device used to isolate traffic on two subnets from one another.  
Test-stand - The combination of a workstation, embedded system, electronics and sensor under test, with corresponding software.

4 Applicable Documents  
LAT-SS-00???-?? - I&T/Online PDR report  
LAT-SS-00XXX-P1 - LAT Electrical Ground Support Equipment Level 3 Specification  

5 Requirements  

5.1 Hardware  
A test stand consists of a workstation, a crate of interface electronics, a network router and electronics and/or sensor under test. The function of the router is to isolate the network traffic generated by the test stand from the rest of the subnet to which it is connected.  

5.1.1 Workstation  
The workstation will be an Intel Pentium 3 or higher class machine running Microsoft NT-4 or Windows 2000 Professional (TBR). The machine will have a minimum of 256 MB of memory, 10 GB of disk (TBR).  

5.1.2 Test stand  
Several versions of the test stands will be made. One version will be based on a VME crate containing an MVME 2306 SBC (PPC 604r, 32 MB RAM) and a VME LAT-COMM card. Other versions (including the flight version) will be based on a cPCI form factor with a TBD SBC apropos to the situation. In some cases, the cPCI based test stand will have a cPCI version of the LAT-COMM card to communicate with the electronics and sensor under test. In other cases, the SBC will interface to the flight hardware through the cPCI bus.  

5.2 Software Environment Setup  

5.2.1 NT
The workstation will be configured with SCL and NFS server software.

5.2.2 VxWorks
The embedded processor will run the VxWorks operating system. The bootstrap loader will be resident in SBC's Flash or EEPROM memory. The OS and additional software will be loadable from the NT workstation's hard disk through an NFS mount point.

5.3 Coding interface requirements

5.3.1 Programming technique
Object Oriented techniques and C++ shall be used to construct the system.

5.3.2 Scope limitation
A method is needed to avoid multiply defined symbols. Symbols, macros, etc., that live in the global namespace will be prepended with an acronym to signify to which facility/package the symbol belongs. The owner/maintainer of the facility/package is responsible for avoiding name clashes.

5.4 SCL
SCL is a toolkit with which a control system can be built. In order to create a control system, several features need to be added, such as a command and monitor interface, an archiver, etc. SCL provides an API with which to build these features. These features communicate with the core SCL system via a Shared Memory Database and/or TCP/IP Stream sockets.

5.4.1 Network communication
A set of C++ classes will be provided with which to set up communication using TCP/IP sockets. The class interface is TBD, but will need to take into account file descriptors other than those in use for SCL on which to hang a read.

TCP/IP socket port number will be allocated according to a method TBD. Probably the SCL defaults are good enough.

A given test stand can, in principle, serve multiple simultaneous users. To avoid conflicts between these users, the TCP/IP socket port numbers should be allocated so that a given port number is not used for two different tasks. The method to assign an offset to the group of ports used is TBD. The SCL mechanism to offset the ports by a function of the user ID (UNIX, NT = ?) may be satisfactory.

5.4.2 Shared Memory Database
Shared memory database entries are ultimately recognised by the SCL system via their "record ID" numbers. These record ID numbers are assigned by the database designer and must be unique in a given database. In order to allow different subsystems to develop their databases independently, the capability to load multiple databases will be used. The databases are assigned an ID that is a number corresponding to the order in which they are loaded, starting from zero. A method to discover which databases are loaded will be provided. All references to database records will be prepended by a database designator. The method to define this designator is TBD.

5.4.3 SCL Commands
There are several types of commands that can be issued to SCL. The core set provided by the SCL command interpreter can be augmented with custom commands. The custom commands are typically directed to a "listener" application that in turn acts upon the command. Scripts are another form of commands.
Namespace conflicts will be avoided with a TBD method. If nothing better can be found, the same rule as in 5.3.2 will apply. It is expected that users of the System will rarely, if ever, have to type these commands as they will be issued in response to interaction with a GUI.

5.4.4 Applications
A number of applications will be built with the SCL tools, as indicated in 5.4.

5.4.4.1 Command/Monitor GUI
A commanding and monitoring GUI tool will be provided. For the test stands for EMI, this will be LabVIEW.

5.4.4.1.1 Color and style rules
Color and style rules are used to give a GUI a common look and feel throughout the different portions of the GUI. TBD, if necessary.

5.4.4.2 Archivers
SCL provides for the capability of supplying several different types of archiver applications. Archiver applications listen to DataIO sockets.

5.4.4.2.1 Data logger
The Data Logging archiver will receive data from the socket designated by the SCL_DATAIO_DATA_SOCKET environment variable and format it for writing to a disk file.

5.4.4.2.2 Event archiver
The Event archiver will receive data from the socket designated by the SCL_DATAIO_EVENT_SOCKET environment variable. This information will be used to create an alarm handler.

5.4.4.2.3 Telemetry archiver
The Telemetry archiver receives raw telemetry frame data from the socket designated by the SCL_DATAIO_FRAME_SOCKET environment variable. This data can be logged to a disk file for debugging or later playback purposes. The NT version of SCL comes with this application prebuilt.

5.4.4.3 Data Visualizer
The Data Visualizer program will receive data from the socket designated by the SCL_DATAIO_DATA_SOCKET environment variable and may possibly listen for commands on the socket designated by the SCL_RTE_IPC environment variable. The program will be capable of iterating over the data to pick out requested portions for visualization. Consideration will be given to whether or not this activity should be able to impact data-taking deadtime. The drawing package to be used is TBD from HippoDraw, ROOT and IDL.

5.5 Embedded processor
The embedded processor will load software supplied by the FSW group from an NFS partition on the workstation. Initially, this software will serve the LAT-COMM card hardware interface to SCL. A description of the device attached to the LAT-COMM card, typically some form of TEM, will be built up with SCL. Applications will be built on top of this description to handle data acquisition from, and control of the device under test.

In the final stages of the test stand design, the embedded processor will perform the data acquisition and control functions under command of an SCL application running on the workstation. Additional hardware in the embedded
system crate will also participate in the data acquisition.

There will be several stages of the system in between these two extremes in which progressively more and more of the functionality initially implemented in SCL migrates into the embedded processor.

Commands and data communicated between the Workstation and the Embedded processor will be in the format that requires the least effort on the part of the Embedded processor. (By happenstance, this turns out to be network byte ordering.) All required byte swapping will take place on the Workstation.

5.6 Data format
Data collected from the test stand system will be stored in FITS format files.

5.7 Databases
There are two relational databases that pertain to the test stands. The local database is used for online activities: test configuration, data logging, etc. The purpose of the central database is to maintain a historical account of test activities. A procedure will be provided with which to upload the local database to the central database.

5.7.1 Local
The local database software will be MS Access. The schema definition is TBD.

5.7.2 Central
The central database software will be Oracle. The schema definition is TBD. The central database will be considered write once, read many times.

5.8 Code Management, Building and Release Control
TBD

5.9 Documentation
Code documentation will be carried out with the doxygen package.

5.10 Remote debugging
A method for debugging test stands remotely will be provided. The mechanism is TBD.