

## Design and Development of an Observation Log Auxiliary System for the Lijiang 2.4-meter Telescope (Postprint)

**Authors:** Wang Chuanjun, Wang Deqing, Xiao Jian, Yin Shucheng, Wang Feng, Fan Yufeng

**Date:** 2019-05-30T00:00:00+00:00

### Abstract

Observation logs are essential for observers during data processing and for observatories during the storage, archiving, and publication of observational data, as they provide extensive information for assessing data quality. In practice, however, observation logs are manually recorded by observing assistants or users, which may lead to incomplete information, clerical errors, and other issues; moreover, paper-based logs are not conducive to the subsequent archiving and publication of observational data. This paper describes the design and development of an auxiliary system for automatic observation log recording for YFOSC data from the Lijiang 2.4-meter telescope. The system automatically records log information for observational data and provides functionality for correcting erroneous information. It manages observational data using the observer's observation time proposal number as a unique identifier, thereby enabling the packaging of observational data according to different observers. This approach protects data access rights while reducing the impact of clerical errors, improving the accuracy of observation logs, and ultimately enhancing observing efficiency.

### Full Text

## Design and Development of an Observation Log Assistant System for the Lijiang 2.4-Meter Telescope

Wang Chuanjun<sup>1,2,3</sup>, Wang Deqing<sup>1,2,3</sup>, Xiao Jian, Yin Shucheng, Wang Feng<sup>1</sup>, Fan Yufeng<sup>1,2</sup>

<sup>1</sup>Yunnan Observatories, Chinese Academy of Sciences, Kunming 650216, China

<sup>2</sup>Key Laboratory for the Structure and Evolution of Celestial Objects, Chinese Academy of Sciences, Kunming 650216, China

<sup>3</sup>University of Chinese Academy of Sciences, Beijing 100049, China  
School of Software, Tianjin University, Tianjin 300354, China  
Center for Astrophysics/School of Physics and Electronic Engineering,  
Guangzhou University, Guangzhou 510006, China

**Abstract:** Telescope observation logs are crucial for observers during data processing and for observatories during data storage, archiving, and publication. These logs provide essential information for assessing observation data quality. However, current practice relies on manual recording by observers or assistants, which risks incomplete information, transcription errors, and inefficient archiving of paper documents. This paper presents the design and development of an automated observation log assistant system for the YFOSC instrument on the Lijiang 2.4-meter telescope. The system automatically records log information from observation data while providing functionality to correct errors. It manages observation data using the unique proposal ID from observers' time applications, enabling automated data packaging by user. This approach protects data access rights, reduces transcription errors, improves log accuracy, and ultimately enhances overall observation efficiency.

**Keywords:** observation log; observation data; observation time

---

## 0 Introduction

Astronomy has evolved from naked-eye observations to telescopic studies, and from “all-waveband astronomy” to “multi-messenger astronomy,” continuously expanding humanity's means of understanding the universe. On October 16, 2017, scientists worldwide announced the first direct detection of gravitational waves from a binary neutron star merger, simultaneously observing electromagnetic signals from this cosmic event. Gravitational waves provide an entirely new observational modality, ushering astronomy into the era of big data and multi-messenger astronomy.

As observational capabilities advance, data volumes from large telescopes continue to grow exponentially, imposing new demands on storage, computing, networking, software, algorithms, and operational workflows. Astronomers require standardized formats for storing and transmitting massive datasets, with FITS (Flexible Image Transport System) having long served as the universal standard for astronomical data. FITS files consist of headers and data matrices, with extensible structures constrained to integer multiples of 2880 bytes. During data ingestion, FITS header keywords are archived into databases to facilitate subsequent retrieval. However, header content varies significantly across different instruments, often lacking completeness, particularly for commercial devices whose headers cannot be modified or supplemented. Consequently, detailed observation logs are necessary to record observational circumstances, enabling better data classification and management.

Observation logs for optical telescopes document critical information during observations, including telescope parameters, instrument configuration, targets, exposure times, filters or gratings, observation types, slit settings, and acquisition timestamps. For observers, these logs are indispensable for data processing. For observatory operations, they provide essential references for data storage, archiving, preprocessing, and publication, allowing operators to discard useless data during preprocessing and perform better calibration based on actual conditions. For statistical studies, logs are even more vital, enabling rapid identification of multiple observations of the same target across years.

At the Lijiang 2.4-meter telescope, observers provide target coordinates and requirements to telescope operators, who then input these into the control system, configure the instrument, and manually record logs during exposure. This manual process is time-consuming, prone to errors, and inefficient. To improve accuracy, reduce logging time, and enhance observation efficiency, we designed and developed an automated observation log assistant system specifically for the YFOSC (Yunnan Faint Object Spectrograph and Camera) instrument.

## 1 System Functions and Design

The observation log assistant system automates log recording, freeing telescope operators from tedious handwritten documentation while streamlining and standardizing log management workflows. It reduces operational complexity and minimizes human error, providing high reliability and adaptability for log management. The system achieves this without modifying original FITS files by integrating monitoring and query functions, correcting FITS header information as needed, and combining it with supplementary data into comprehensive log entries stored in a database for later use.

The system's primary functions include:

1. Automatically reading FITS header information when observation files are saved to disk, writing this information to the database, and displaying it in the user interface.
2. Allowing users to inspect recorded logs, add missing information, and correct errors through the interface, with updates submitted to the database.
3. Associating observation data with specific users by selecting proposal IDs from time applications, facilitating data management by user.
4. Enabling query, packaging, and download of observation data by username or proposal ID, while generating standardized log files.
5. Displaying frequently used observers in the interface for convenient access during observations.
6. Showing system status and user operation logs in real-time to ensure proper system functioning.
7. Supporting batch submission of log entries, defaulting to FITS header values if no modifications are made.

The system employs a layered architecture comprising four tiers: view, control,

service, and data layers [Figure 1: see original paper]. The view layer displays FITS header information, accepts user modifications, and shows system status. The control layer routes user requests between the view and service layers. The service layer handles all interactions with the data layer, providing interfaces for six core services: file monitoring, data export, FITS processing, log management, file operations, and configuration management. The data layer encompasses all data files, including raw FITS observations, configuration files, and structured database contents, responding to service layer requests.

This layered design ensures that all functions require coordinated interaction between tiers while maintaining security through strict independence and preventing cross-layer calls.

## 2 System Implementation

The system was implemented using modular design principles, divided into six functional modules: interface display, data monitoring, data query, data export, system logging, and system configuration.

### 2.1 Development and Runtime Environment

The system was developed and deployed on Ubuntu 13.10 using JDK 1.7 and Java [7], with PostgreSQL as the backend database [8]. Several open-source Java libraries were integrated:

- **Commons-compress-1.8.jar**: Apache library for packaging queried observation data
- **Dom4j-2.0.00-ALPHA-2.jar**: High-performance XML parsing library from dom4j.org
- **jxl.jar**: Java Excel library for reading, creating, and updating Excel files
- **postgresql-9.3-1101.jdbc4.jar**: JDBC driver for PostgreSQL database connectivity

### 2.2 Interface Display Module

Developed using Java's Swing GUI library [9], the interface [Figure 2: see original paper] comprises four functional zones:

1. **Observation Log Display Area**: Shows FITS header information from newly acquired data, allowing user modifications.
2. **Data Monitoring Control Area**: Provides proposal ID selection, start/stop monitoring, modified log submission, and long-term project user management.
3. **Data Query and Export Area**: Enables querying by time range and observer, with results cached for log file generation. Users can select files for export (Excel for logs, TAR for data) and specify output paths via configuration files.

4. **System Status Display Area:** Shows real-time operational status and user actions for immediate error detection and response.

### 2.3 Data Monitoring Module

Monitoring initiates when users select a project and click “Start.” A background thread scans the data directory, storing results in a hash table, then rescans every second to detect new files. When new data appears, the system processes the FITS file, ingests the log entry into the database, and displays it in the interface. Monitoring can be stopped via the “Stop” button, allowing project switching between observation sessions.

The processing workflow is illustrated in Figure 3 [Figure 3: see original paper]. The observation log database schema includes fields for observation time, exposure duration, user information, telescope pointing coordinates (RA, DEC), focus parameters, and instrument-specific settings (filters, gratings, slits).

### 2.4 Data Query Module

This module queries the observation log database via JDBC. Users can retrieve historical records by time period and principal investigator name (Observer field). Results display in the interface and are cached for log file generation, with error correction capabilities available during review.

### 2.5 Data Export Module

The export module packages both observation data and corresponding logs. Data files are compressed and archived via background processes, while logs are generated as Excel files. All outputs are saved to directories specified in configuration files, providing users with complete, ready-to-process datasets.

### 2.6 System Logging Module

System logs are categorized into three types:

- **Status Logs:** Redirected to Swing text components for real-time display without file storage
- **Operation Logs:** Record software states, data generation times, and user actions to enable scenario reconstruction when errors are discovered
- **Error Logs:** Capture specific error details to assist developers in debugging

Operation and error logs generate daily files managed by a Log4jUtil class [6] through dedicated writing threads.

### 2.7 System Configuration Module

System configurations are stored in two XML files:

- **configure.xml**: Contains database connection parameters, monitoring directories, and output paths
- **longterm.xml**: Stores long-term project user information (names and proposal IDs), currently supporting up to six users. This feature also supports Target of Opportunity (ToO) observations by allowing temporary user creation through configuration updates that appear immediately in the interface.

### 3 System Operation Status

After development, the system was deployed on a computer in the Lijiang 2.4-meter telescope dome. All functions were verified during actual operations. Figure 2 shows the system monitoring data and logging operational messages. Figure 4 [Figure 4: see original paper] demonstrates data query results, while Figure 5 [Figure 5: see original paper] shows exported data packages (TAR archives) and corresponding Excel log files.

The system has been operational at the Lijiang 2.4-meter telescope, successfully automating log acquisition and significantly reducing manual workload and error rates. It enables post-observation data query and export by user, generating standardized logs and simplifying data transfer while ensuring security during proprietary periods. Integration with astronomical domain cloud systems for time allocation and data management is planned to create a complete observation lifecycle workflow.

Current limitations include: 1. Manual import of time application tables is required each observing season to update user information 2. The dropdown menu for selecting observers becomes cumbersome with many annual proposals 3. The database uses filenames as primary keys, requiring redevelopment for other instruments with different naming conventions 4. No reserved fields for future expansion, limiting system extensibility

Future work will address these issues, optimizing the system for broader applicability to other instruments and telescopes.

### References

- [1] Zhong Shoubo, Han Bo, Zhang Yanxia, et al. Design and implementation of astronomical big data management tools[J]. *Astronomical Research & Technology*, 2015, 12(4): 510-517.
- [2] Zhao Yongheng. Automatic observation technology for astronomical telescopes[J]. *e-Science Technology & Application*, 2012, 3(4): 11-16.
- [3] Cui Chenzhou, Yu Ce, Xiao Jian, et al. Astronomy research in the era of big data[J]. *Chinese Science Bulletin*, 2015, 60(5-6): 445-449.
- [4] Ji Kaifan, Cao Wenda, Song Qian. FITS, BMP and SCR image formats and their mutual conversion[J]. *Publications of Yunnan Observatory*, 1996, 2: 60-64.

[5] Zhang Hailong, Ye Xinchun, Li Huijuan, et al. A review of astronomical data retrieval and publication[J]. *Astronomical Research & Technology*, 2017, 14(2): 212-228.

[6] Wang Chuanjun, Fan Yufeng, Yi Weimin. Design and development of the TCS Sequencer for the Lijiang 2.4-meter telescope[J]. *Astronomical Research & Technology*, 2013, 10(4): 378-385.

[7] Bruce Eckel. *Thinking in Java (4th Edition)*[M]. Machinery Industry Press, 2013.

[8] Peng Zhiyong. *PostgreSQL Database Kernel Analysis*[M]. Machinery Industry Press, 2012.

[9] Wang Peng, He Yunfeng. *Java Swing Graphical Interface Development and Case Studies*[M]. Tsinghua University Press, 2008.

*Note: Figure translations are in progress. See original paper for figures.*

*Source: ChinaXiv – Machine translation. Verify with original.*