Version

Nicolas A. Barriga, Carlos Quiroz, Arturo Núñez

27052011/01

Created: May 27, 2011

Modified: May 27, 2011



**Change Record**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Notes** |
| 1 | May 27, 2011 | Nicolas A. Barriga | * First Version |

Table of Contents

1 Introduction 1

1.1 Document Purpose 1

1.2 Intended Readership 1

1.3 Conventions 1

1.4 Acronyms 1

1.5 Reference Materials 1

2 Overview of the Gemini Data Service 2

2.1 Data Values Composition 2

2.2 Data Label Generation 2

2.3 Sending FITS Headers from Seqexec to GDS 3

3 GUI 5

4 Implementation Details 9

4.1 Protocols 9

4.2 Scala 10

# Introduction

This document provides a description of the Gemini Data Service (GDS) Design.

## Document Purpose

The purpose of this document is to present the GDS design for review by interested parties.

This document does not rehash the critical information in references and . It is assumed that the reader has read these other documents and understands their content.

## Intended Readership

The intended audience for this document is *groups who are writing software, design review documents or providing operational support* *for Aspen instruments*.

## Conventions

The GDS is still under active development and things that are expected to undergo some changes are marked like this paragraph with a yellow exclamation point. There are not many of these situations in this document.

Code examples and individual methods are written in a fixed-width font like this: unsubscribeToStatus.

## Acronyms

ACM Action Command Model

CMS C++ Messaging Service

DHS Data Handling System

GIAPI Gemini Instrument Application Programmer Interface

GMP Gemini Master Process

GSDN Gemini Data Storage Network

ICD Interface Control Document

JMS Java Message Service

PCS Primary Control System

TCS Telescope Control System

TLC Top Level Computer

WCS World Coordinate System

GDS Gemini Data Service

## Reference Materials

x

|  |  |
| --- | --- |
| [1] | Kim Gillies and Arturo Nunez, "Aspen GIAPI Design and Use," 2006. |
| [2] | Kim Gillies, "Guidelines for Designing Gemini Aspen Instrument Software," 2004. |
| [3] | FITS Standard Specification. [Online]. http://archive stsci.edu/fits/fits\_standard/fits\_standard.html |

x

# Overview of the Gemini Data Service

The Gemini Data Service is the system that will (partially) replace the Gemini Data Handling System (DHS) for GIAPI based instruments, such as GPI.

## Data Values Composition

The GDS is a component that listen for Observation Events sent from an instrument and that reacts to them by sampling the state of the Observatory. observation database, instrument, etc. The sampled information is then added to the data files produced by the instrument. This process is called value composition, where all sampled values upon observation events are composed into the data file.

The design uses the concept of actors or agents that are independent objects that can sample the information. This design makes it simple to keep track of the progress and make each of these actors, a single purpose, very simple to implement component.

It also makes it easy to extend the composition as the core components can discover new modules in charge of gathering these values

## Data Label Generation

Currently, the dataset names (or datalabels) generation is performed via a control command by the DHS, at the request of the seqexec. The seqexec later specifies who will contribute data to this dataset, at which point itself and the contributors can start sending data. A first study of the following documents and code was done:

* ICD 3: Bulk Data Transfer
* ICD 1.9/3.2: Science Instrument to Data Handling System
* dhs/dhs/dhsData/list.C
* dhs/dhs/dhsData/ctl.C

This investigation shows that the datalabel generation functionality is fairly independent from the data storing functionality, in a way that no files are created, and no internal state changes (except for a list of the last labels generated), when labels are generated. Furthermore, data can be sent with arbitrary datalabels not generated by the DHS.

So, there are three main alternatives for instruments using the GDS:

Continue with the seqexec request datalabels to the DHS, and then not sending any data to the DHS, but to the GDS. The major disadvantage is a dependence on the DHS for GIAPI based instruments.

Extract the datalabel generation from the DHS to an external service that the DHS can query for most instruments, and the seqexec can query for GIAPI based instruments. The major disadvantage is the risk of modifying a complex piece of software like the DHS. For consistency and simplicity, if this option is taken, the communication with the new external service should use the same protocol as the one used in Sending FITS Headers from Seqexec to GDS.

Generate datalabels independently in GDS and DHS, and ensure no collisions will happen by changing the naming convention.

## Sending FITS Headers from Seqexec to GDS

The FITS headers that the seqexec provides, must be passed to the GDS. There aren't many suitable remote communication alternatives supported by tcl and java/scala. The most suitable seems to be XMLRPC, which has implementations in both languages, and is relatively simple, but much higher level than using plain http or tcp.

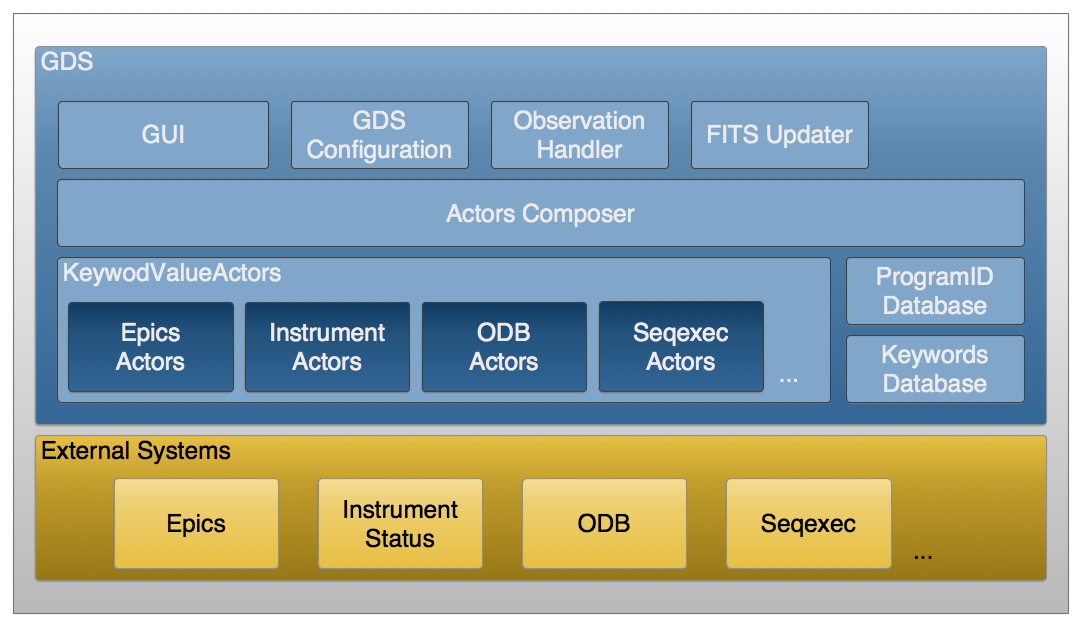


Figure : GDS Module Diagram

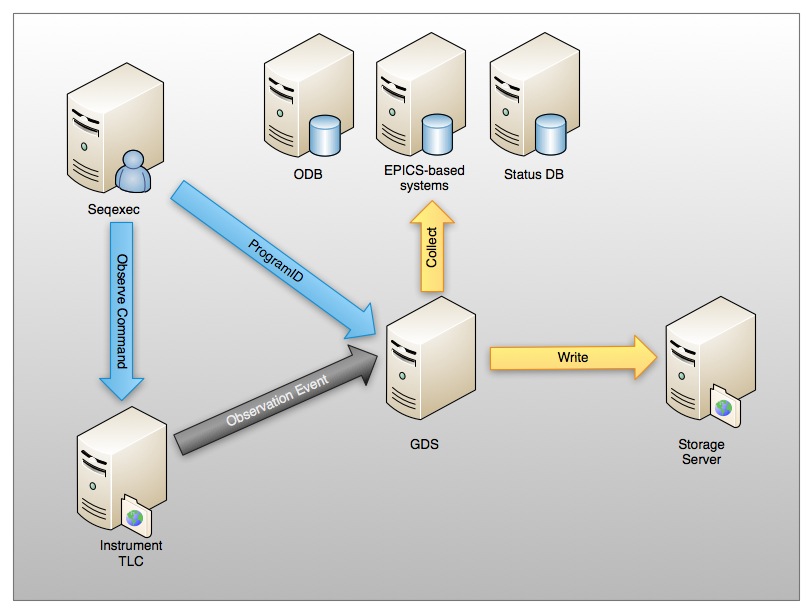
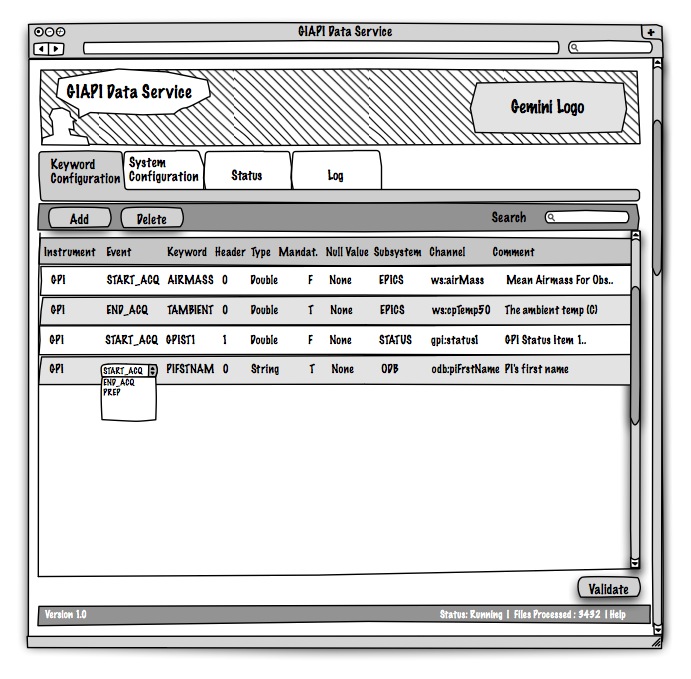
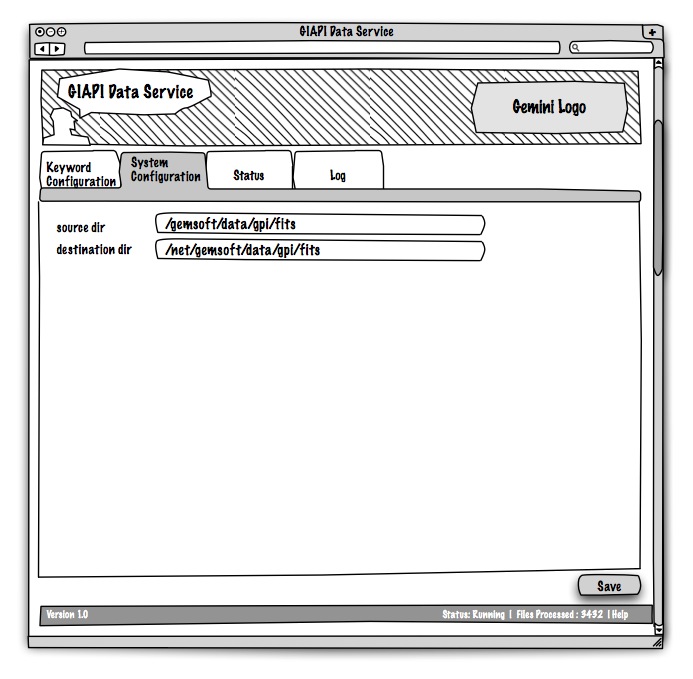


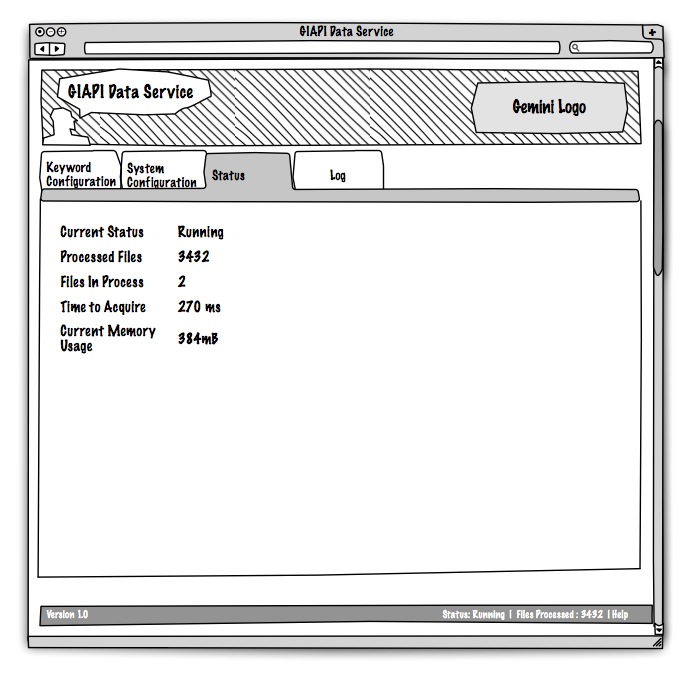
Figure : GDS System Diagram

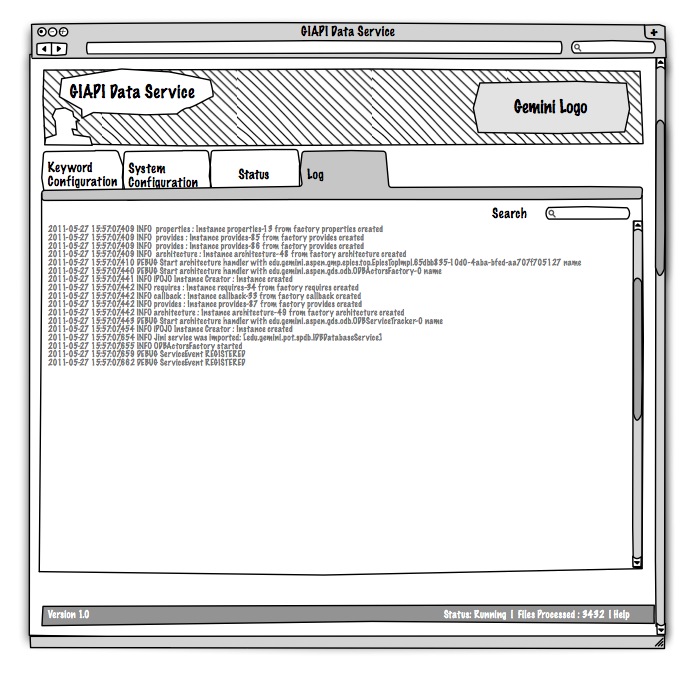
# GUI

These are mockups of the proposed GDS user interface.



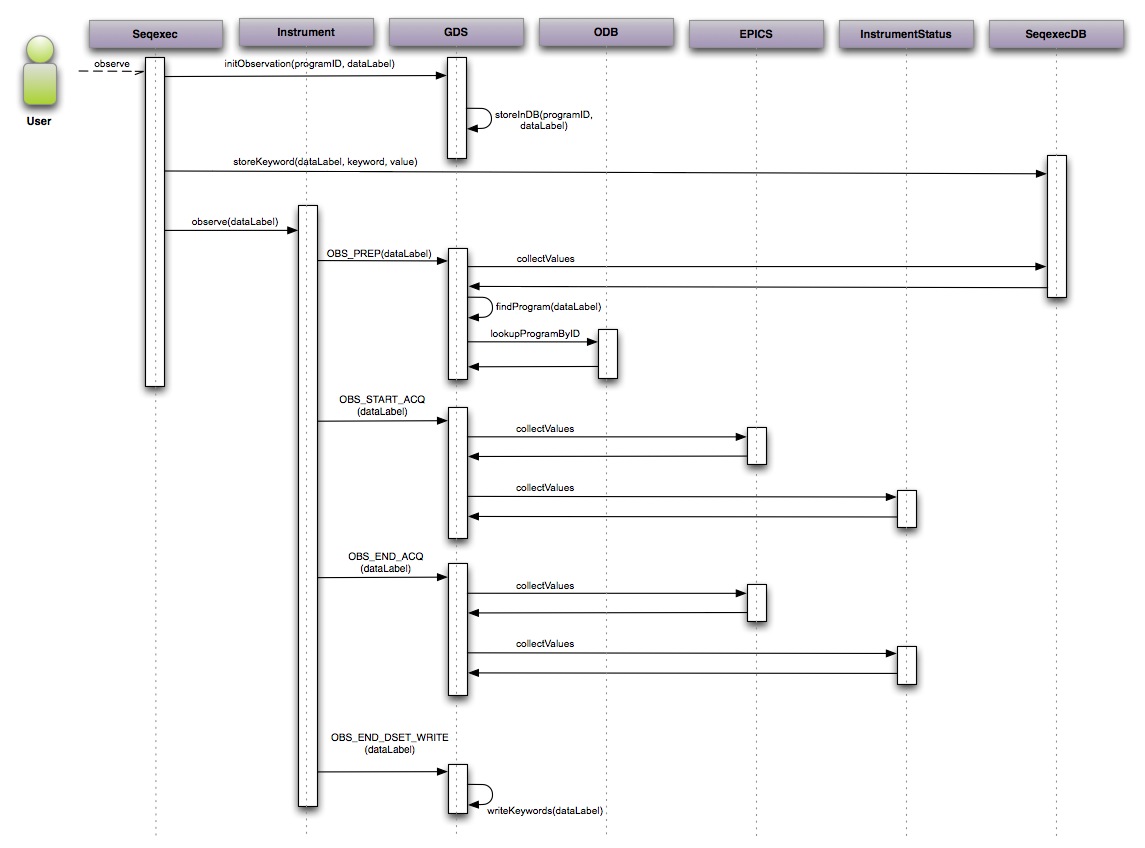






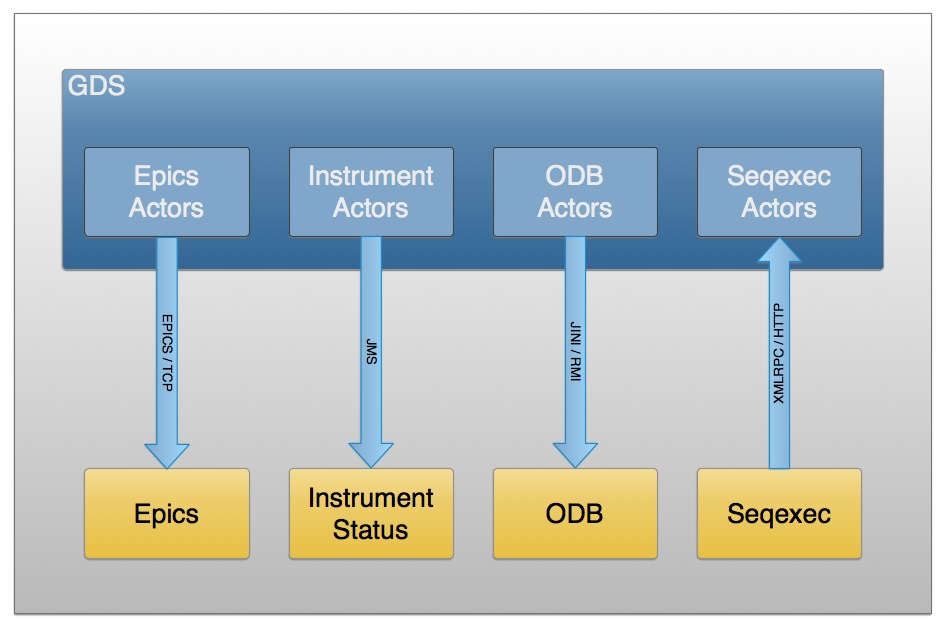
# Implementation Details

The following sections are of interest only to software engineers who need to debug or improve the GDS code base.



## Protocols

Description of the communication protocols used (EPICS, XMLRPC, JMS, RMI)



## Scala

Motivation to use scala: jvm, functional programming, concurrency(actors)…