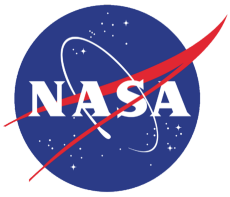
NASA/SP-2015-218490



The *Kepler* Science Data Processing Pipeline Source Code Road Map

KSOC-21226

Bill Wohler and Jon M. Jenkins

2016-10-31

NASA Ames Research Center
Moffett Field, CA 940535

Bill Wohler, Senior Software Engineer (Author)	Date
Jon Jenkins, Co-Investigator for Data Analysis (Author)	Date
Joseph Twicken, Lead Scientific Programmer	Date
Dwight Sanderfer, SOC Manager	Date
Charlie Sobeck, Project Manager	Date
Natalie M. Batalha, Project Scientist	Date

List of Contributors

The following individuals contributed to the *Kepler* Science Data Processing Pipeline codebase:

Name	**Role**
Christopher Allen	Software Engineer
Lee Brownston	Software Engineer
Stephen T. Bryson	Support Scientist
Christopher J. Burke	Support Scientist
Douglas A. Caldwell	Instrument Scientist, Co-Investigator
Jennifer Campbell	Operations Engineer Lead
Joseph Catanzarite	Scientific Programmer
Hema Chandrasekaran	Scientific Programmer
Jessie L. Christiansen	Support Scientist
Bruce D. Clarke	Scientific Programmer
Miles T. Cote	Lead Engineer
Forrest R. Girouard	Software Engineer
Jay P. Gunter	Software Engineer
Jon Jenkins	Co-Investigator for Data Analysis
Todd C. Klaus	Software Engineer
Jeffrey J. Kolodziejczak	Support Scientist
Jie Li	Scientific Programmer
Sean McCauliff	Software Engineer
Christopher Middour	Systems Engineer
Robert L. Morris	Scientific Programmer
Elisa V. Quintana	Scientific Programmer
Jason Rowe	Support Scientist
Anima Sabale	Operations Engineer
Jesse Sanderfer	Software Engineer Intern
Shawn Seader	Scientific Programmer
Jeffrey C. Smith	Scientific Programmer
Martin Stumpe	Scientific Programmer
Peter Tenenbaum	Scientific Programmer
Susan E. Thompson	Support Scientist
Joseph D. Twicken	Scientific Programmer
Kamal Uddin	Test Engineer
Jeffrey E. van Cleve	Support Scientist
Bill Wohler	Software Engineer
Hayley Wu	Scientific Programmer
Khadeejah Zamudio	Operations Engineer

1 Introduction

The *Kepler* telescope launched into orbit in March 2009, initiating NASA's first mission to discover Earth-size planets orbiting Sun-like stars. *Kepler* simultaneously collected data for ∼160,000 target stars over its four-year mission, identifying over 4700 planet candidates, 2300 confirmed or validated planets, and 2100 eclipsing binaries. While *Kepler* was designed to discover exoplanets, the long term, ultra-high photometric precision measurements it achieved also made it a premier observational facility for stellar astrophysics, especially in the field of asteroseismology, and for variable stars, such as RR Lyrae. The *Kepler* Science Operations Center (SOC) was developed at NASA Ames Research Center to process the data acquired by *Kepler* starting with pixel-level calibrations all the way to identifying transiting planet signatures and subjecting them to a suite of diagnostic tests to establish or break confidence in their planetary nature. Detecting small, rocky planets transiting Sun-like stars presents a variety of daunting challenges, including achieving an unprecedented photometric precision of ∼20 ppm on 6.5-hour timescales, supporting the science operations, management, processing, and reprocessing of the accumulating data stream.

The scientific objective of the *Kepler* Mission is to explore the structure and diversity of planetary systems. This is achieved by surveying a large sample of stars to:

- Determine the abundance of terrestrial and larger planets in or near the habitable zone of a wide variety of stars;
- Determine the distribution of sizes and shapes of the orbits of these planets;
- Estimate how many planets are in multiple-star systems;
- Determine the variety of orbit sizes and planet reflectivities, radii, masses and densities of short-period giant planets;
- Identify additional members of each discovered planetary system using other techniques; and
- Determine the properties of those stars that harbor planetary systems.

This document presents an overview of the source code of the Science Data Processing Pipeline. It describes the context of each software component or module in the directory tree. It provides a road map for the reader's study of the source code where he or she can gain insight into the scientific algorithms used by the Pipeline. The reader interested in the theoretical development of the algorithms should refer the *Kepler* Data Processing Handbook (KDPH), available at `https://archive.stsci.edu/kepler/documents.html`. The KDPH, together with the *Kepler* SOC 9.3 source code, represent the documentation of the algorithms used to produce the legacy archive data products including the calibrated Full Frame Images (FFI), the target pixel files, the simple aperture photometry and systematic error-corrected light curves, and the data validation (DV) reports on potential transiting planet candidates. A list of other *Kepler* Project documentation relevant to the study and interpretation of *Kepler* archive data products is provided and discussed in section 5.

However, this document is not a tutorial for building and running the software as it is not expected that the reader would be able to do these things. There are far too many external dependencies and procedures to explain in this short, introductory document.

The block diagram in Figure 1 introduces the primary components that comprise the Science Data Processing Pipeline that produces the *Kepler* archival science data products listed above.

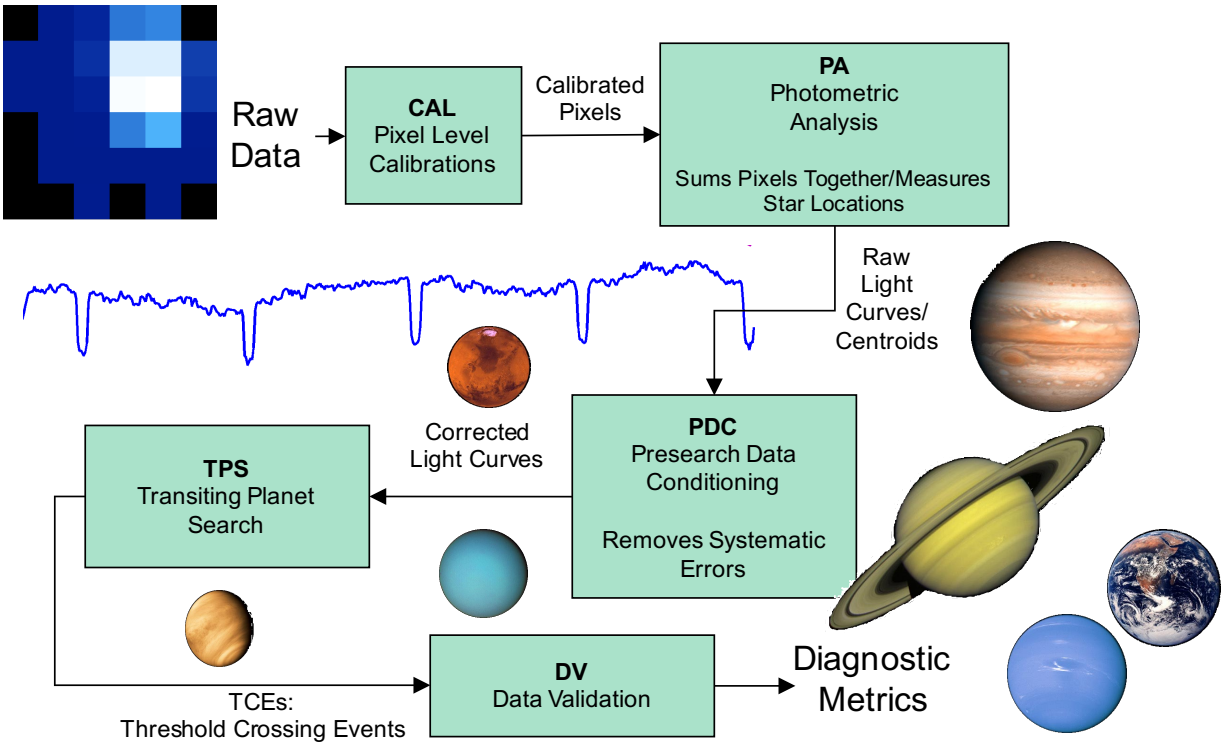


Figure 1: From pixels to planets

2 Science Data Processing Pipeline

In support of the mission's objectives, the *Kepler* Science Data Processing Pipeline selects the pixels that will be collected on the spacecraft, performs pixel-level calibration on downlinked data to remove instrumental effects, calculates flux and centroids per cadence for each target star, corrects the flux for systematic errors, performs a transiting planet search on each flux time series to identify targets with transiting planet signatures, then produces a battery of diagnostic metrics for each planet candidate. The Pipeline products are then used to answer the scientific questions listed above. The architecture of the SOC is given in Middour et al. (2010), while an overview of the science data processing steps is given in Jenkins et al. (2010a).

The Pipeline source code is primarily written in Java and MATLAB. For the most part, the data processing and analysis algorithms are written in MATLAB, while the data management, execution automation, and operations consoles and tools are written in Java.

The source code is organized in the directory tree shown in Figure 2. Only one of the Pipeline components (Data Validation, or DV) is shown.

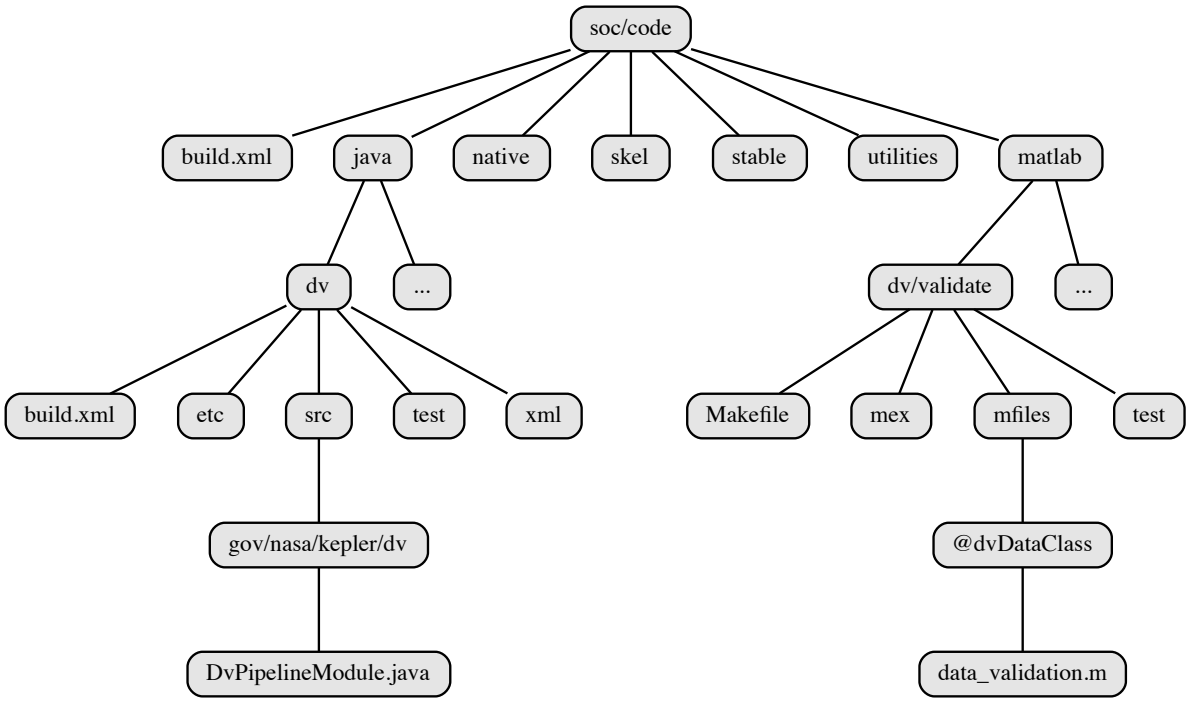


Figure 2: A portion of the source code directory tree

The directories shown in Figure 2 are described as follows.

- soc/code/

  The root of the source code. It contains the following directories at the top level:

  - java

    Contains the Java source code. All *Kepler* Java source code is found in the gov.nasa.kepler package, so the src and test directories contain a directory hierarchy that matches this convention (that is, gov/nasa/kepler).

    The top-level directories under the `java` directory contain one or more of the following sub-directories:

    * bin

      Contains scripts that are used to support the component. Some are developer scripts. Others are fully-fledged programs used during operations, such as `cm/bin/ingest` that ingests the *Kepler* Input Catalog (KIC) into the Oracle database.

    * etc

      The most common two files contained in etc are kepler.properties, which provides run-time properties used by the *Kepler* Java code, and `log4j.xml`, that contains the Log4j configuration that controls the logging output. These are only used in the component's development environment. When these files are used by the Pipeline at runtime, the installed versions from `skel` are used instead.

    * src

      Contains the Java source code.

    - test
      Contains unit and system tests.
    - xml
      Contains XML code and definitions used by various XML tools.
  - matlab

    Contains the MATLAB source code. The top-level directories under `matlab` contain one or more of the following sub-directories:

    - common
      Some components have several sub-components and use a common directory to share code.
    - mex
      Contains C code compiled with MEX.
    - mfiles
      Contains the MATLAB source code.
    - test
      Contains unit and system tests.
  - native

    Common low-level C code.
  - skel

    This hierarchy contains a mixture of skeleton directories and files that are copied to the installation directory (`dist`). It includes scripts and libraries that are needed at run-time that don't fit elsewhere.
  - stable

    Contains a default `kepler.properties` file that points to a stable database for scientific programmers.
  - utilities

    Holds utilities and tools that need to be part of the release branch for use by the SOC Operations team, but are not part of the build.

Figure 2 presents a more detailed overview of the Pipeline components that are implemented within the `java` and `matlab` directories and how they interface to the other segments of the *Kepler* Mission.

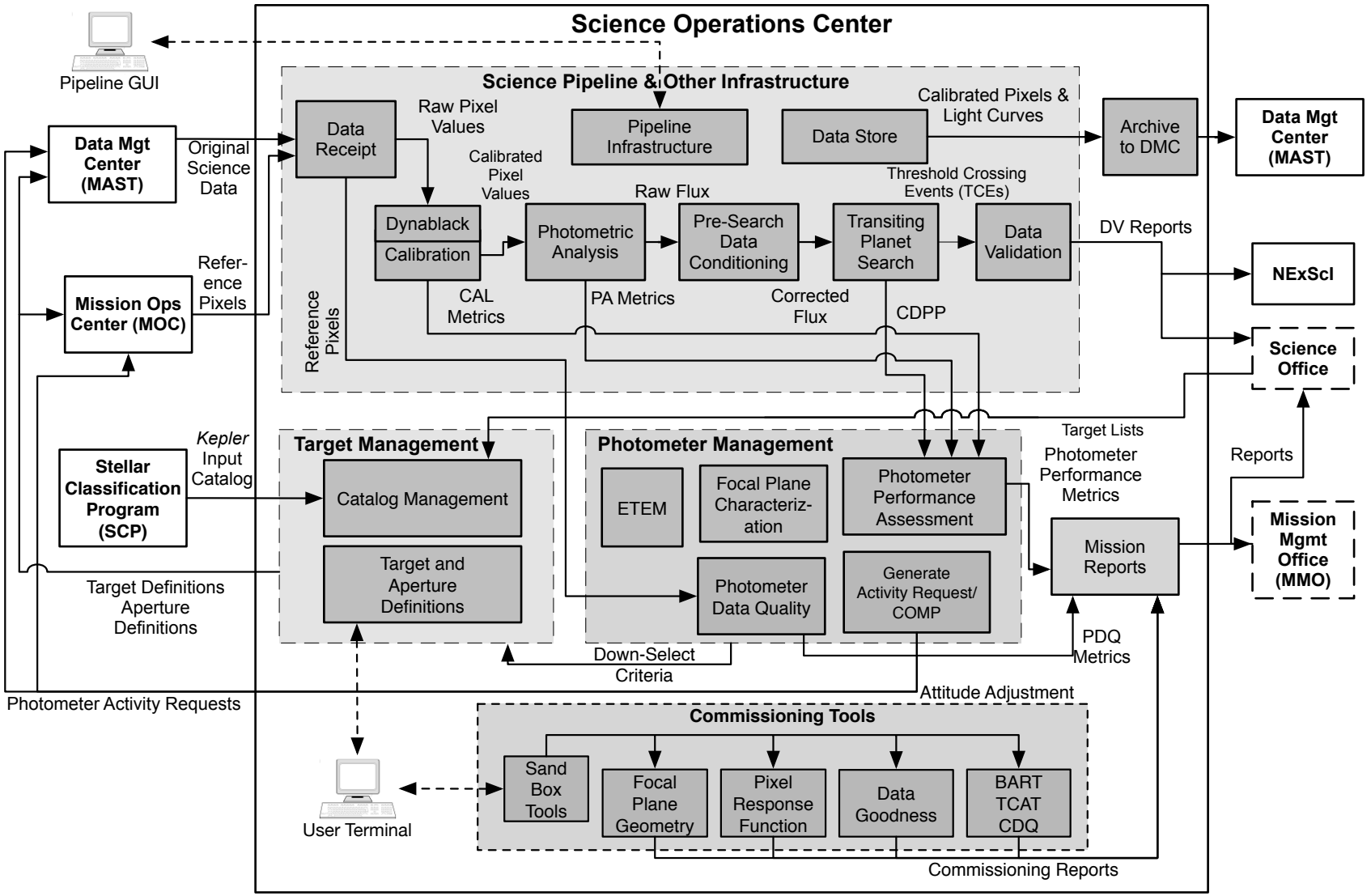


Figure 3: The *Kepler* Science Data Processing Pipeline

3 Pipeline Components

What follows is an alphabetic encyclopedia of the Pipeline and support components, sorted by acronym. The source code for each component is found in the `java` or `matlab` directories or both. Refer to Figure 2 to see the component's place in the Pipeline.

3.1 Archive to DMC (`java/ar`, `matlab/ar`)

The Archive (AR) module generates the files in the archival format delivered to the Mikulski Archive for Space Telescopes (MAST) and made available to the astronomical community. The *Kepler* archival products include calibrated pixels, simple aperture photometry, systematic-error-corrected photometry, astrometry (centroids), and associated uncertainty estimates. Target pixel files contain the pixel data for each target, both original and calibrated, organized as image data. These files also include information about sky background flux and cosmic ray hits detected by the pipeline. The transit-like features identified by the transit search and the results of DV's diagnostic tests are archived as XML files along with PDF reports to the Exoplanet Archive managed by NASA's Exoplanet Science Institute. These PDF reports contain a wealth of information regarding each transit-like signature.

3.2 BIN to MAT Converter (`matlab/bin_to_mat`)

A pipeline module that converts `.bin` files to `.mat` files for modules that execute remotely.

3.3 MATLAB Build System (`matlab/build`)

This directory contains Makefile templates and other files needed to build the MATLAB code.

3.4 Calibration (`java/cal`, `matlab/cal`)

The calibration (CAL) module performs traditional CCD data reduction such as removal of bias and dark current signals and correction for gain, linearity, and flat-field (Clark et al., 2016; Clarke et al., 2016). The module also corrects for Kepler-specific effects such as smear from the shutterless readout and electronic pattern noise. CAL operates on 1-minute Short Cadence (SC) and 30-minute Long Cadence (LC) data as well as Full Frame Images (FFIs – nominally acquired once per month), and produces calibrated pixel flux time series, associated uncertainties, and metrics that are used in subsequent pipeline modules.

3.5 Catalog Management (`java/cm`)

Catalog Management (CM) contains the *Kepler* Input Catalog (KIC) provided by the Stellar Classification Program and subsequent updates to the catalog. CM contains the characteristics of the target stars, and background field stars, such as location (right ascension, declination), effective temperature, surface gravity, radius, mass, and proper motion, all of which are necessary to identify the pixels to be used in measuring the brightness of each target star.

3.6 Commissioning Tools (`matlab/ct`)

Several tools were developed and deployed specifically for the commissioning phase of the *Kepler* Mission. The Pixel Response Function (PRF) is covered separately.

3.6.1 Focal Plane Geometry (`matlab/ct/fpg`)

FPG was used to determine the detailed sky to pixel mapping coefficients, across each of the 84 individual CCD readout channels (Tenenbaum & Jenkins, 2010). The FPG coefficients include terms for detector orientation and position, pixel plate-scale, and pin-cushion distortion.

3.6.2 Data Goodness (`matlab/ct/dg`)

The DG tool allows the user to assess the quality of FFIs.

3.6.3 2-D Black and Artifact Removal Tool (`matlab/ct/bart`)

The 2-D Black and Artifact Removal Tool (BART) detects and models temperature-dependent image artifacts in pixel data. BART provides insight into whether the photometer data are consistent with pre-launch expectations regarding temperature variations.

3.6.4 Check Data Quality (`matlab/ct/cdq`)

Check Data Quality (CDQ) checks and analyzes the RMS of data fitting residuals and thermal coefficients produced by BART for the pixels in the collateral regions of each CCD.

3.6.5 Temperature Coefficient Analysis Tool (`matlab/ct/tcat`)

The Temperature Coefficient Analysis Tool (TCAT) investigates the thermal variations of pixels that are affected by crosstalk from the Fine Guidance Sensors (FGS), which are used to control the spacecraft pointing. FGS Crosstalk is a significant source of *Kepler* instrument noise.

3.7 Common Libraries (`java/common`, `java/common-spiffy`, `matlab/common`)

These directories contain code that is used by more than one pipeline module.

3.8 Debug and Prototyping Module (`java/debug`, `matlab/debug`)

The debug pipeline component was used for prototyping the interaction between the database and the Data Store with MATLAB.

3.9 Data Receipt (`java/dr`)

The DR component provides data ingestion and automated pipeline launch capabilities. It is divided into two main components: 1) a generic layer that watches for new files, dispatches the proper handler, and launches pipelines and 2) a plug-in layer for specific data types.

3.10 Data Validation (`java/dv`, `matlab/dv`)

The DV component performs a suite of diagnostic tests on each transiting planet signature identified by TPS to make or break confidence in its planetary nature (Twicken et al., 2016). These include a comparison of the depth of the even transits to the odd transits, an examination of the correlation of changes in the photocenter (centroid) of the target star to the photometric transit signature, a statistical bootstrap to assess confidence in the detection, difference image centroiding to rule out background sources of confusion, and a ghost diagnostic test to rule out optical ghosts of bright eclipsing binaries as the source of the transit-like features. These tests can determine if the

transit signature is likely to be due to a background eclipsing binary whose diluted eclipses are masquerading as transits of a planetary body. In order to detect multi-planet systems, DV calls TPS to search the residual light curve for evidence of additional transiting bodies after fitting and removing the first planetary transit signature from the light curve (Li et al., 2016). This process is repeated until TPS fails to identify another transit signature.

3.11 Dynamic Black Level Calibration (`java/dynablack`, `matlab/dynablack`)

Dynablack is a time-dependent 2-D black calibration that attempts to mitigate the effects of image artifacts due to FGS clocking crosstalk and high-frequency oscillations in the electronics.

3.12 End-to-End Model (`java/etem`, `matlab/etem2`)

ETEM is a suite of software that generates synthetic flight-like data for *Kepler* with a high degree of fidelity, including matching the formats of the science data at each ground segment interface, from the solid state recorder (SSR) onboard the spacecraft, through the Mission Operations Center (MOC) and the Data Management Center (DMC) (Jenkins et al., 2004; Bryson et al., 2010). ETEM was indispensable in testing the entire *Kepler* ground segment as well as for designing, implementing, and testing the SOC. ETEM simulates the astrophysics of planetary transits, stellar variability, background and foreground eclipsing binaries, cosmic rays, and other phenomena.

3.13 Focal Plane Characterization (`java/fc`, `matlab/fc`)

The FC module consists of a set of database tables, persistence classes, and associated handling code that manages the calibration models used to process data and create and manage target definitions (Allen et al., 2010). The models are also used for managing the target lists and identify which targets are expected to fall on silicon. These include the 2-D bias voltage image, the pixel-level gain model, and the pixel response function (PRF). One of the most critical model sets maintained in FC is the model describing the mapping from celestial coordinates to focal plane coordinates (pixels), called `RaDec2Pix` (see also Thompson et al., 2016, Section 2.3.5).

3.14 Data Store (`java/fs`)

The *Kepler* Data Store (DS) contains a custom array data base (ADB), a transactional database management system for arrays, sparse arrays, and binary data (McCauliff et al., 2010). The vast majority of data for use by the pipeline modules is stored in the ADB (formerly known as the File Store), which is augmented by a relational (Oracle) database for metadata.

3.15 Compression/Generate Activity Request (`java/gar`, `matlab/gar`)

Pixel data compression tables are generated by the Huffman Generator (HGN) and the Huffman Aggregator (HAG) modules. The data compression scheme involves three steps: 1) re-quantizing the data so that the quantization noise is approximately a fixed fraction of the intrinsic measurement uncertainty (which is dominated by shot noise for bright pixels), 2) taking the difference between each re-quantized pixel value and a baseline value that was updated once per day, and 3) entropic encoding via a length-limited Huffman table (Jenkins & Dunnuck, 2011). Typical compression rates of 4.5–5 bits per pixel measurement were achieved throughout the *Kepler* Mission, allowing for >66 days of data to be stored on the SSR and decreasing the time required for DSN contacts.

3.16 Hibernate Object to Relational Model (ORM) classes (`java/hibernate`)

The classes within this directory define the objects that are actually stored in the database by Hibernate, which provides an Object to Relational Model (ORM). In addition to the model data, they contain database metadata which are used to persist the information into database tables and then extract the data.

3.17 Ant Build Files (`java/include`)

The `include` directory contains shared ant build files that are imported by the pipeline module build files.

3.18 Java 3rd-party Libraries (`java/jars`)

The `jars` directory contains all of the 3rd-party jar files used by the pipeline. There are three sub-directories. The first called `runtime` contains jar files such as `log4j` and `hibernate` that are used during compilation and run time.

The libraries in `dev` contain development tools that are needed to actually run the pipeline. These include static code analysis, code coverage, and unit testing.

The `src` directory contains select archives that are used by Eclipse to view the source code of the 3rd-party libraries.

3.19 Module Common (`java/mc`)

The MC directory contains common pipeline module code that is shared by most of the pipeline modules.

3.20 CADU Maker (`matlab/mkcadu`)

The MKCADU pipeline module creates Channel Access Data Units (CADU) formatted files for use by the ETEM pipeline.

3.21 Mission Reports (`java/mr`)

MR provides a web-based interface to a library of reports concerning the pipeline and its processes that can be generated on the fly. These reports are used extensively by the operations personnel to help manage the science data processing.

3.22 Photometric Analysis (`java/pa`, `matlab/pa`)

PA measures the brightness of the image of each target star on each cadence. It also fits and removes background flux due to zodiacal light and the diffuse stellar background, identifies and removes cosmic rays from all target star apertures, and measures the photocenter or centroid of each target star on each cadence. PA also uses PRF-fitting to measure precisely the location of $\sim$200 bright, unsaturated target stars on each CCD readout area in order to establish the pointing and focus of the telescope. This pointing information is used to update the photometric apertures used for data analysis.

3.23 Presearch Data Conditioning (`java/pdc`, `matlab/pdc`)

PDC performs a critical set of corrections to the light curves produced by PA, including the identification and removal of instrumental signatures caused by changes in focus or pointing, and step discontinuities that result occasionally from radiation events in the CCD detectors (Jenkins et al., 2012; Stumpe et al., 2012; Smith et al., 2012, 2016). PDC also identifies and removes isolated outliers and corrects the flux time series for crowding effects and for the fact that not all the light from a star can be captured by a finite aperture. PDC employs a multi-scale Bayesian approach to identify and remove systematic errors, allowing it to retain important astrophysical signals in the face of much larger instrumental effects (Stumpe et al., 2014).

3.24 Photometer Data Quality (`java/pdq`, `matlab/pdq`)

PDQ provides a "quick look" assessment of the health and performance of the instrument through data downlinked by X-band twice-weekly (Chandrasekaran et al., 2010). PDQ also assesses the validity of the spacecraft pointing after each return to science attitude and computes a corrective "tweak" for use if the spacecraft is pointed too far from its desired orientation.

3.25 Pipeline Infrastructure (`java/pi`)

The Pipeline Infrastructure (PI) provides fully automated distributed processing of science data and sequencing of pipeline modules based on theresults of previous modules (Klaus et al., 2010a,b). Features include a customizable unit-of-work that controls how the data are distributed across the cluster, a configuration management and versioning system for algorithm parameters and pipeline configurations, and a graphical user interface for the configuration, execution, and monitoring of pipeline jobs. PI provides scalability for running the pipeline on a developer workstation, a large cluster of computing nodes, as well as on NASA's Pleiades supercomputer.

PI is designed to be a generic, reusable platform suitable for developing science processing and analysis pipelines. PI provides a plug-in architecture for pipeline modules, parameter definitions, unit-of-work definitions, and contains no *Kepler*-specific code. PI also provides a generic mechanism for running any pipeline module on large computing clusters that support the PBS (Portable Batch System) interface. This allows us to routinely process nearly all *Kepler* science data on NASA's Pleiades system.

3.26 PI GUI (`java/pig`)

The PI GUI (PIG) is a Java Swing application that is used by the *Kepler* operator to build, launch, and monitor pipelines to process *Kepler* data.

3.26.1 Pixel Overlay On FFIs (`matlab/poof`)

The POOF tool allows the user to retrieve *Kepler* FFIs and overlay the aperture masks from target tables on the images, along with information about the stellar targets themselves. POOF enabled the validation of target tables starting early and continuing throughout the *Kepler* Mission.

3.27 Photometer Performance Assessment (`java/ppa`, `matlab/ppa`)

PPA assesses the health and performance of the instrument based on the science data sets collected each month, identifying out-of-bounds conditions and generating alerts (Li et al., 2010). The metrics include photometric precision, brightness, black level, background flux, smear level, dark

current, cosmic ray counts, outlier counts, centroids, reconstructed attitude, and the difference between the reconstructed and nominal attitudes. These metrics are tracked and trended by PPA and the numerical results are persisted to the database as well as populating a PDF report. PPA results are used to identify and set data anomaly flags required for archival processing.

3.28 Pixel Response Function (`java/prf`, `matlab/prf`)

During commissioning, PRF was used to determine the shape of the stellar optical point spread function (PSF) convolved with the pixel response, across each of the 84 individual CCD readout channels (Bryson et al., 2010a). The PRF tool constructed five individual PRFs for each CCD readout area in order to capture non-uniformity in the focus and PSF. PRF models at intermediate locations are obtained by interpolation. The pipeline uses these model waveforms to monitor the locations of the brightest, unsaturated 200 stars on each channel to reconstruct pointing and capture distortion due to focus changes.

3.29 Sandbox Tools (`matlab/sbt`)

The sandbox tools allow the staff to make queries against the file store on their own workstations.

3.30 Basic Services (`java/services`)

The services directory contains code for various services of the pipeline infrastructure. These services include inter-process messaging services, user-level access control for pipeline configuration and operations, pipeline configuration, and automated alerts. The services directory also contains general classes used by all pipeline processes.

3.31 Sky Group Generator (`java/sggen`, `matlab/sggen`)

The sggen pipeline component generates sky groups. These are groups of stars in the KIC that are in the same relative position (CCD/Output) throughout the year as the telescope is rotated 90° each quarter.

3.32 System Tests (`java/systest`)

The systest directory includes many tests of the pipeline, including the Automated Feature Test (AFT).

3.33 Target and Aperture Definitions (`java/tad`, `matlab/tad`)

Target and Aperture Definitions (TAD) predicts what pixels need to be stored and downlinked by the *Kepler* spacecraft for each quarterly observation and formulates the 1024 mask definitions used to capture the pixels of interest for each target (Bryson et al., 2010b, 2016). The predicted photometric apertures are used by the Science Data Processing Pipeline to measure the centroids of a fiducial set of bright, unsaturated targets in order to reconstruct the pointing history of the spacecraft and update the photometric apertures accordingly.

The associated sub-module, Compute Optimal Apertures (COA), predicts the pixels of interest for extracting photometric measurements from the CCD images for each target star in the Pipeline. These pixels, along with a buffer halo or halos, are used to create the target definition tables flown on the spacecraft. *Kepler* had very tight margins for the pixel data stored onboard and returned to the ground: only ∼6% of the full-frame pixel data could be stored onboard for later downlink.

3.34 Transit Injection Parameters (`java/tip`, `matlab/tip`)

The TIP pipeline component is used to inject transits into the photometric data stream to provide ground truth for testing the performance of the TPS pipeline component.

3.35 Transiting Planet Search (`java/tps`, `matlab/tps`)

TPS implements a wavelet-based, adaptive matched filter algorithm to detect signatures of transiting planets (Jenkins, 2002; Jenkins et al., 2010b; Seader et al., 2015; Seader et al., 2013; Jenkins et al., 2016). TPS stitches the ∼93-day light curves together for stars observed on consecutive quarters prior to searching for transits. TPS also provides estimates of combined differential photometric precision (CDPP), a key performance diagnostic for transit survey missions (Christiansen et al., 2012). CDPP is necessary for estimating the completeness of the transit survey and for extrapolating the results to infer the intrinsic frequency of planets in the star sample.

3.36 The *Kepler* Science Operations Console (`java/ui`)

The ui directory contains The *Kepler* Science Operations Console (KSOC). It is used to build and manage target lists and export compression tables for upload to the spacecraft.

4 Build System

The codebase is built using a combination of Apache Ant for the Java components, and make for the MATLAB, C, and C++ components. Ant is also used to build the entire codebase. Some of the build.xml and Makefiles are shown in Figure 2.

These files are not expected to be used to build the system, but can be used to see what third-party software was employed and how this large system was built.

5 Guide to Literature and Documentation

A significant number of papers documenting the *Kepler* Science Data Processing Pipeline have been published over the years including ∼20 papers written for the SPIE 2010 Software and Cyberinfrastructure for Astronomy conference that included a special session on the *Kepler* SOC.[1] However, ∼80% of the science algorithms were subject to significant revision or replacement over the last 7.5 years as the *Kepler* Science Data Processing Pipeline evolved to improve the quality of the data products and the sensitivity of the planetary search. The interested reader should refer to the references cited in Section 3 and in the revised KDPH. Here we list the most relevant and up-to-date documents for readers interested in pursuing the details of the algorithms and software design, as well as the characteristics of the data products.

Additional documentation on the *Kepler* Science Data Processing Pipeline, SOC operations, and data products generated by the SOC can be found in the literature and at MAST under `https://archive.stsci.edu/kepler/documents.html` and at NExScI's Exoplanet Archive.

5.1 *Kepler* Data Processing Handbook (KSCI–19081–002)

The algorithms of the components of the Science Data Processing Pipeline, including DYN, CAL, PA, PDC, TPS, DV, and TAD have been updated in the revised *Kepler* Data Processing Handbook

[1] See `https://kepler.nasa.gov/science/ForScientists/papersAndDocumentation/SOCpapers/`

that will be available at MAST along with the other *Kepler* documentation. The KDPH develops the theoretical underpinnings of the science algorithms, gives examples, and briefly describes the principal characteristics of the resulting data products.

5.2 *Kepler* Instrument Handbook (KSCI-19033-002)

The KIH provides information about the design, performance, and operational constraints of the *Kepler* hardware, and an overview of the available pixel data sets (Van Cleve & Caldwell, 2016). That document presents an overview of the *Kepler* instrument, and then tracks photons through the telescope, focal plane, and focal plane electronics. Details regarding targets, the pixels of interest around them, and operational details are specified, which will be helpful in both planning observations for the repurposed *Kepler* Mission, dubbed K2, and for understanding the data reduction procedures described in this document.

5.3 *Kepler* Data Characteristics Handbook (KSCI–19040-005)

The Data Characteristics Handbook provides a description of a variety of phenomena identified within the *Kepler* data, and a discussion of how these phenomena are handled by the data reduction Pipeline (Van Cleve et al., 2016).

5.4 *Kepler* Data Release Notes (KSCI–19041, etc.)

With each quarterly release of data, a set of accompanying notes is created to give *Kepler* users information specific to the time period during which the data was obtained and processed. The notes provide a summary of flight system events that affect the quality of the data and the performance of the Pipeline. The Data Release Notes, along with other *Kepler* documentation, are located at the Multi-mission Archive (MAST) at Space Telescope. Once the user becomes familiar with the content of the Data Characteristics Handbook, they need only read the short Release Notes for details specific to that quarter.

5.5 *Kepler* Archive Manual (KDMC–10008-006)

Data from the Kepler mission are archived at MAST, which serves as NASA's primary archive for ultraviolet and optical space-based data. The *Kepler* Input Catalog (KIC – Brown et al., 2011), processed light curves, and target pixel data are all accessed through MAST. The Kepler Archive Manual describes data products, file formats and the functionality of Kepler data access (Thompson et al., 2016). The Archive Manual can be accessed from the MAST Kepler page: `https://archive.stsci.edu/kepler/documents.html`, and is available in HTML, DOC, and PDF formats.

References

Allen, C., Klaus, T., & Jenkins, J. 2010, in Proc. SPIE, Vol. 7740, Software and Cyberinfrastructure for Astronomy, 77401E–77401E–8, available online at `http://kepler.nasa.gov/science/ForScientists/papersAndDocumentation/SOCpapers`

Brown, T. M., Latham, D. W., Everett, M. E., & Esquerdo, G. A. 2011, AJ, 142, 112

Bryson, S. T., Jenkins, J. M., Peters, D. J., et al. 2010, in Proc. SPIE, Vol. 7738, Modeling, Systems Engineering, and Project Management for Astronomy IV, 773808, available online at `http://kepler.nasa.gov/science/ForScientists/papersAndDocumentation/SOCpapers`

Bryson, S. T., Tenenbaum, P., Jenkins, J. M., et al. 2010a, Astrophysical Journal Letters, 713, 97

Bryson, S. T., Jenkins, J. M., Klaus, T. C., et al. 2010b, Proc. SPIE, 7740, 77401D, available online at `http://kepler.nasa.gov/science/ForScientists/papersAndDocumentation/SOCpapers`

—. 2016, in *Kepler Data Processing Handbook*: KSCI–19081–002, ed. J. M. Jenkins (NASA Ames Research Center), 33–51, available online at `https://archive.stsci.edu/kepler/documents.html`

Chandrasekaran, H., Jenkins, J. M., Li, J., et al. 2010, in Proc. SPIE, Vol. 7740, Software and Cyberinfrastructure for Astronomy, 77401B, available online at `http://kepler.nasa.gov/science/ForScientists/papersAndDocumentation/SOCpapers`

Christiansen, J. L., Jenkins, J. M., Caldwell, D. A., et al. 2012, PASP, 124, 1279, available online at `https://arxiv.org/abs/1208.0595`

Clark, B. D., Kolodziejczak, J. J., Caldwell, D. A., et al. 2016, in *Kepler Data Processing Handbook*: KSCI–19081–002, ed. J. M. Jenkins (NASA Ames Research Center), 53–75, available online at `https://archive.stsci.edu/kepler/documents.html`

Clarke, B. D., Caldwell, D. A., Quintana, E. V., et al. 2016, in *Kepler Data Processing Handbook*: KSCI–19081–002, ed. J. M. Jenkins (NASA Ames Research Center), 77–100, available online at `https://archive.stsci.edu/kepler/documents.html`

Jenkins, J. M. 2002, Astrophysical Journal, 575, 493, available online at `http://kepler.nasa.gov/files/mws/JenkinsSolVarApJ575.pdf`

Jenkins, J. M., & Dunnuck, J. 2011, in Proc. SPIE, Vol. 8146, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 814602

Jenkins, J. M., Peters, D. J., & Murphy, D. W. 2004, in Proc. SPIE, Vol. 5497, Modeling and Systems Engineering for Astronomy, ed. S. C. Craig & M. J. Cullum, 202–212, available online at `http://kepler.nasa.gov/files/mws/SPIE.Glasgow.Jenkins.pdf`

Jenkins, J. M., Smith, J. C., Tenenbaum, P., Twicken, J. D., & Van Cleve, J. 2012, in Advances in Machine Learning and Data Mining for Astronomy, ed. M. J. Way, J. D. Scargle, K. M. Ali, & A. N. Srivastava (Chapman and Hall, CRC Press), 355–381

Jenkins, J. M., Tenenbaum, P., Seader, S., et al. 2016, in *Kepler Data Processing Handbook*: KSCI–19081–002, ed. J. M. Jenkins (NASA Ames Research Center), 233–270, available online at `https://archive.stsci.edu/kepler/documents.html`

Jenkins, J. M., Caldwell, D. A., Chandrasekaran, H., et al. 2010a, Astrophysical Journal, 713, L87

Jenkins, J. M., Chandrasekaran, H., McCauliff, S. D., et al. 2010b, SPIE Conference Series, 7740, available online at `http://kepler.nasa.gov/science/ForScientists/papersAndDocumentation/SOCpapers`

Klaus, T. C., McCauliff, S., Cote, M. T., et al. 2010a, in Proc. SPIE, Vol. 7740, Software and Cyberinfrastructure for Astronomy, 774017, available online at `http://kepler.nasa.gov/science/ForScientists/papersAndDocumentation/SOCpapers`

Klaus, T. C., Cote, M. T., McCauliff, S., et al. 2010b, in Proc. SPIE, Vol. 7740, Software and Cyberinfrastructure for Astronomy, 774018, available online at `http://kepler.nasa.gov/science/ForScientists/papersAndDocumentation/SOCpapers`

Li, J., Burke, C. J., Jenkins, J. M., et al. 2016, in *Kepler Data Processing Handbook*: KSCI–19081–002, ed. J. M. Jenkins (NASA Ames Research Center), 313–352, available online at `https://archive.stsci.edu/kepler/documents.html`

Li, J., Allen, C., Bryson, S. T., et al. 2010, in Proc. SPIE, Vol. 7740, Software and Cyberinfrastructure for Astronomy, 77401T, available online at `http://kepler.nasa.gov/science/ForScientists/papersAndDocumentation/SOCpapers`

McCauliff, S., Cote, M. T., Girouard, F. R., et al. 2010, in Proc. SPIE, Vol. 7740, Software and Cyberinfrastructure for Astronomy, 77400M, available online at `http://kepler.nasa.gov/science/ForScientists/papersAndDocumentation/SOCpapers`

Middour, C., Klaus, T. C., Jenkins, J., et al. 2010, in Proc. SPIE, Vol. 7740, Software and Cyberinfrastructure for Astronomy, 77401A, available online at `http://kepler.nasa.gov/science/ForScientists/papersAndDocumentation/SOCpapers`

Seader, S., Tenenbaum, P., Jenkins, J. M., & Burke, C. J. 2013, ApJS, 206, 25, available online at `http://arxiv.org/abs/1501.03586`

Seader, S., Jenkins, J. M., Tenenbaum, P., et al. 2015, The Astrophysical Journal Letters Supplement, 217, available online at `http://arxiv.org/abs/1501.03586`

Smith, J. C., Morris, R. L., Jenkins, J. M., et al. 2016, in *Kepler Data Processing Handbook*: KSCI–19081–002, ed. J. M. Jenkins (NASA Ames Research Center), 153–232, available online at `https://archive.stsci.edu/kepler/documents.html`

Smith, J. C., Stumpe, C., Cleve, J. E. V., et al. 2012, PASP, 124, 1000, available online at `https://arxiv.org/abs/1203.1383`

Stumpe, M. C., Smith, J. C., Catanzarite, J. H., et al. 2014, PASP, 126, 100, available online at `http://iopscience.iop.org/article/10.1086/674989`

Stumpe, M. C., Smith, J. C., Cleve, J. E. V., et al. 2012, PASP, 124, 985, available online at `https://arxiv.org/abs/1203.1382`

Tenenbaum, P., & Jenkins, J. M. 2010, in Proc. SPIE, Vol. 7740, Software and Cyberinfrastructure for Astronomy, 77401C, available online at `http://kepler.nasa.gov/science/ForScientists/papersAndDocumentation/SOCpapers`

Thompson, S. E., Fraquelli, D., van Cleve, J. E., & Caldwell, D. A. 2016, Kepler Archive Manual, Tech. Rep. KDMC–10008–006, NASA Ames Research Center Kepler Mission

Twicken, J. D., Clarke, B. D., Girouard, F., et al. 2016, in *Kepler Data Processing Handbook*: KSCI–19081–002, ed. J. M. Jenkins (NASA Ames Research Center), 297–312, available online at `https://archive.stsci.edu/kepler/documents.html`

Van Cleve, J. E., & Caldwell, D. A. 2016, Kepler Instrument Handbook, Tech. Rep. KSCI–19033–002, NASA Ames Research Center Kepler Mission

Van Cleve, J. E., Christiansen, J. L., Jenkins, J. M., et al. 2016, Kepler Data Characteristics Handbook (KSCI-19040-005), Tech. rep., Moffett Field, CA