Data Release Production WIP W16/X16 release notes

> ⚠ **Release Frozen**
>
> The upcoming v12.0 release of the stack has been frozen. Please do not add anything here. Add new notes to the F16 release notes.

These *draft* notes cover the major updates made by Data Release Production to the LSST stack since release 11.0 (Summer 2015). Please record significant updates here so that they can ultimately be incorporated into the notes accompanying the Winter 2016 release.

- Major Functionality and Interface Changes
  - Add the option of excluding mask planes before coaddition
  - Dumping task configuration parameters now includes documentation
  - Clean up interpolation tasks and implement useFallbackValueAtEdge
  - HSC backport: avoid I/O race conditions config write out
  - HSC backport: Introduce SafeClipAssembleCoaddTask which extends AssembleCoaddTask to make clipping safer
  - HSC backport: Allow for some fraction of PSF Candidates to be reserved from PSF fitting
  - Updated pipeline configuration based on Hyper Suprime-Cam experience
  - Define polygon bounds for CCDs based on vignetted regions
  - Introduce --rerun option for command line tasks
  - Introduce framework for injecting fake sources into data processing
  - Add convenience routines for working with tracts
  - Warp images based on an XYTransform
  - Add getCoordSystem to Coord and add UNKNOWN CoordSystem enum
  - Adapt joinMatchListWithCatalog to facilitate and simplify denormalizing a match list
  - Add a script for visualizing skymaps and CCDs
  - Add functions to generate "unpacked matches" to and from a catalog
  - Add a measurement algorithm which records the focal plane coordinates of sources
  - Add a measurement algorithm which records the Jacobian at the positions of sources
  - Add a measurement algorithm which records the number of input images contributing to a coadd sources
  - Add a measurement algorithm which records the variance at the positions of sources
  - Add a measurement algorithm which records source flux in an aperture scaled to the PSF
  - Add a measurement algorithm which quantifies the amount of "blendedness" of an object
  - Add a "simple" shape measurement algorithm
  - Add Hirata-Seljak-Mandelbaum shape measurement algorithms
  - Add option to temporarily remove an interpolated background prior to detection
  - Add function to average coordinates
  - Integrate support for Hyper Suprime-Cam
  - Reimplement PSF Shapelet approximations for CModel
  - Propagate flags from individual visits to coadds
  - Prototype Brighter-Fatter correction
  - Aperture correction on coadd measurements
  - Return grown Footprints from detection
  - Enable measurement of "sky objects" in coadd processing
  - Specification of output directory is now mandatory
  - Bright object masks
  - Overhaul ProcessCcdTask and subtasks, including eliminating camera-specific variants
  - CModel fitting improvements
  - Astropy Table views into LSST Catalog objects
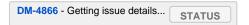 - Add an "afterburner" measurement facility
- Bug Fixes
  - Persist LTVn headers as floating point numbers
  - Fix bug when identifying existing peaks in a merge
  - Fix situation in which the getChildren method of SourceCatalog may return the wrong information
  - Fix warping when the WCS have different coordinate systems
  - Correct bad default minInitialRadius for CModel
  - Correct algebraic error in CModel uncertainty calculation
  - NaiveDipoleCentroid and NaiveDipoleFlux algorithms no longer require a centroid slot
  - Update (some) example code to run with recent stack versions
  - Fix a failure to appropriately log failed task execution
  - Updates to Skymap packages
  - Fix variance in coadded images
  - Fix variance in deblended sources
  - Fix logic for applying aperture corrections
  - More uniform support for assigning to catalog columns
  - Upgraded WCSLIB to version 5.13
  - Fix rotation for instrument signature removal in obs_subaru
  - Fix a silent failure to apply config overrides
  - Correctly set flags for bad SdssShape measurements.
  - Fix CONSTANT background interpolation of bad data
  - Accommodate pixel padding when unpersisting reference catalog matches
  - Correct misleading filter fallback error message
- Build and code improvements
  - Work-around incompatibilities with NumPy 1.10
  - When building boost warn user if user-config.jam or site-config.jam exists
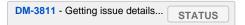
Major Functionality and Interface Changes

Add the option of excluding mask planes before coaddition

The user-configurable paramter `removeMaskPlanes` has been added to `AssembleCoaddConfig`. This is a list of mask planes which will not be propagated to the coadd; by default, the `CROSSTALK` and `NOT_DEBLENDED` mask planes are removed.

DM-4866 - Getting issue details... STATUS

Dumping task configuration parameters now includes documentation

That is, running `${TASK} ... --show config` displays not only the names and values of the configuration but also associated documentation.

DM-3811 - Getting issue details... STATUS

Clean up interpolation tasks and implement useFallbackValueAtEdge

The `interpImageTask` in `pipe_tasks` has been refactored to implement a single `run` function for interpolation over a list of defects in an image. This run function will accept an afw image type of either MaskedImage or Exposure. A defects list can be passed in directly OR the name of mask plane can be passed from which a defects list will be created. If a PSF is attached to the image, it will be used as the (required) argument for the `interpolateoverDefects` function in `meas_algorithms` used for the interpolation. Otherwise a FWHM (in pixels) can be provided or the **defaultFWHM** value in `meas_algorithms'` `GaussianFactory` is used. Note that while the PSF is a required argument for `meas_algorithms'` `interpolateOverDefects`, function it is currently not being used, so it is not necessary to pass in an accurate PSF.

The **useFallbackValueAtEdge** option is now implemented. This tapers the interpolation to a **fallbackValue** towards the image edge. The **fallbackValue** can be set via config parameters to be either computed as a statistical representation of the image data (MEAN, MEDIAN, or MEANCLIP) or set by providing a specific value. Allowance for a negative **fallbackValue** is also controlled though a config parameter.

DM-3677 - Getting issue details... STATUS

HSC backport: avoid I/O race conditions config write out

This adds functionality to avoid potential I/O race conditions when running multiple simultaneous process. This is accomplished by writing to temporary files and then renaming them to the correct destination filename in a single operation. Also, to avoid similar race conditions in the backup file creation (e.g. config.py~1, config.py~2, ...), a --no-backup-config option (to be used with --clobber-config) is added here to prevent the backup copies being made. The outcome for this option is that the config that are still recorded are for the most recent run.

DM-3911 - Getting issue details... STATUS

HSC backport: Introduce SafeClipAssembleCoaddTask which extends AssembleCoaddTask to make clipping safer

SafeClipAssembleCoaddTask does the following,

- build both clipped and unclipped coadds and difference them first
- detect on the difference
- identify difference footprints that overlap appreciably with only one input image
- use identified difference footprints as a mask into a final coadd
- set a "clipped" bit on the final coadd for any pixel that did not include all input frames within the valid polygons.

To support the new coadd task, the pixel flags measurement plugin has been modified to accept two new configuration parameters. These new parameters, named masksFpCenter and masksFpAnywhere, each accept a list of mask planes. When the pixel flags measurement plugin searches for mask planes to set corresponding flags, it will now additionally search the user supplied mask plane lists. The masksFpCenter parameter specifies mask planes that, if found within the center of a footprint, will have a corresponding pixel flag set. The masksFpCenter parameter specifies mask planes that, if found anywhere in a footprint, will have a corresponding pixel flag set. The masksFpAnywhere parameter now has it's defaults set within the stack which specify the clipped mask plane created by SafeClipAssembleCoaddTask.

SafeClipAssebleCoaddTask is now the default method for building a coadd within the LSST stack. The assembleCoadd.py CommandLineTask now supports the --legacy command line flag which will allow the original AssembleCoaddTask to be run. If the legacy task is run, the clipped mask plane must be removed from the masksFpAnywhere configuration parameter.

DM-2915 - Getting issue details... `STATUS`   **DM-4961** - Getting issue details... `STATUS`

HSC backport: Allow for some fraction of PSF Candidates to be reserved from PSF fitting

This adds functionality that allows for the reservation of a fraction of the PSF Candidates from PSF fitting. The reserved fraction can then be used to test for over-fitting, do cross-validation etc. To support this functionality, the run methods of `CalibrateTask` and `MeasurePsfTask` now accept the additional keyword argument `expId`. Dummy versions of this keyword were added to the `SdssCalibrateTask` and `CfhtCalibrateTask` to maintain a consistent API.

DM-3692 - Getting issue details... `STATUS`

Updated pipeline configuration based on Hyper Suprime-Cam experience

Configuration defaults and metadata through the LSST single-processing pipeline have been updated to match the current best practice established on HSC. Major changes include:

- Support for narrow band filters;
- Updated CCD defect lists;
- Optimized (in terms of CPU time) deblender settings;
- Avoiding failure in certain corner cases (e.g. operating on zero-length arrays, taking log10 of zero).

DM-3942 - Getting issue details... `STATUS`

Define polygon bounds for CCDs based on vignetted regions

This adds a function in ip_isr's isrTask.py to set a "valid polygon" for a given CCD exposure as the intersection of a polygon defined in focal plane coordinates and the CCD corners. It is currently being used in obs_subaru's isr.py to set the polygon bounds (added in DM-2981) for a CCD exposure to include the non-vignetted regions. The settings for the vignetted region is in a separate config file so that it can be used in different places in the code.

DM-3259 - Getting issue details... `STATUS`

Introduce --rerun option for command line tasks

This new command line option streamlines the process of specifying input and output repositories for command line tasks. In its simplest form, the `--rerun` option is a shorthand for specifying an output repository: output is written to a location relative to the input. Thus

```
CmdLineTask /path/to/inputroot --rerun useroutput
```

is equivalent to

```
CmdLineTask /path/to/inputroot --output /path/to/inputroot/rerun/useroutput
```

Often, one task will process the output of a previous rerun. For this situation, `--rerun` provides a two-valued form which specifies relative locations of both input and output. In this mode,

```
CmdLineTask /path/to/inputroot --rerun process1:useroutput
```
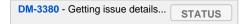
is equivalent to

```
CmdLineTask /path/to/inputroot --input /path/to/inputroot/rerun/process1 --output /path/to/inputroot/rerun
/useroutput
```

RFC-95 - Getting issue details... `STATUS`

DM-3371 - Getting issue details... `STATUS`

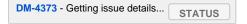## Introduce framework for injecting fake sources into data processing

A new boolean option (doFakes) and a retargetable task (fakes) have been introduced into the ProcessCcdTask. This (along with a new class called BaseFakeSourcesTask) sets up a frame work that others may use to introduce known fake sources into the data processing stream. However, this framework itself does not actually insert any fake data itself, but provides an interface others may use to define their own fake source injection task. To implement a fake injection task one must create a child class of BaseFakeSourcesTask (located in lsst.pipe.tasks.fakes) and overload the run method to do the work of injecting the sources. Each source that is injected should have a corresponding bit set in the FAKE maskplane which can be accessed with the self.bitmask variable. Once a task has been created, the config field fakes in processCcd must be retarged to point to the user created task. Additionally, the task will not be run unless the doFakes configuration option in processCcd is set to True. However, if this variable is set to True, and the task is not retarged processCcd will fail.

DM-3380 - Getting issue details... STATUS

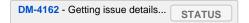## Add convenience routines for working with tracts

Added two new data id containers:

- `PerTractCcdDataIdContainer`: determines the set of tracts each visit touches and adds a data reference with those tracts for each of the visit components. This allows for the user to run a command line task `forcedPhotCcd.py` for a given visit without having to know which tracts overlap the visit. [Note this will also be used by `meas_mosaic` if/when it gets incorporated into the LSST stack.]
- `TractDataIdContainer`: generates a list of data references for patches within a given tract (effectively a "data reference list" that points to the entire tract). [Note that, at the time of writing, this is only being used by a qa analysis script currently under development.]

DM-4373 - Getting issue details... STATUS

Warp images based on an XYTransform

Added the ability to warp images using a transformation defined by an lsst::afw::geom::XYTransform

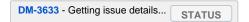**DM-4162** - Getting issue details... STATUS

Add getCoordSystem to Coord and add UNKNOWN CoordSystem enum

Added method getCoordSystem to lsst::afw::coord::Coord. Also added UNKNOWN=-1 as a new CoordSystem enum (the existing enums retain their existing value).

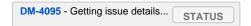**DM-4606** - Getting issue details... STATUS

Adapt joinMatchListWithCatalog to facilitate and simplify denormalizing a match list

The match lists created when performing image calibration (astrometry and photometry) are normalized (i.e. stripped down to a list of the matched reference and source ids and their distance) prior to being persisted. The ability to denormalize a match catalog is very useful (for post qa analysis, for example). This can now be done using the `joinMatchListWithCatalog()` function in `meas_algorithms`'s **LoadReferenceObjectsTask**. It has been moved from `meas_astrom`'s **ANetBasicAstrometryTask** so that it can be easily accessed (requiring only that a reference object loader be initiated) and to allow it to work with any kind of reference catalog (i.e. other than `a_net`).
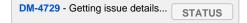
DM-3633 - Getting issue details... STATUS

Add a script for visualizing skymaps and CCDs

The `skymap` package now contains the script `showVisitSkyMap.py` which provides a convenient way of visualizing the tracts, patches and CCDs contained in a set of visits.

DM-4095 - Getting issue details... STATUS

Add functions to generate "unpacked matches" to and from a catalog

Functions have been added to lsst::afw::catalogMatches to provide the ability to convert a match list into a catalog and vice versa (this can be useful for post-processing analyses: qa analysis, for example).

DM-4729 - Getting issue details... STATUS

Add a measurement algorithm which records the focal plane coordinates of sources

The `SingleFrameFPPositionPlugin` measurement plugin, available in `meas_base`, records the positions of source centroids in focal plane coordinates (which may be convenient for plotting). This plugin is not enabled by default, but may be switched on by requesting `base_FPPosition` in measurement configuration.

DM-4234 - Getting issue details... **STATUS**

Add a measurement algorithm which records the Jacobian at the positions of sources

The `SingleFrameJacobianPlugin` calculates the ratio between the nominal Jacobian determinant at the source centroid (as determined by a user-specified pixel scale) and the actual Jacobian determinant as derived from the astrometric solution. This plugin is not enabled by default, but may be switched on by requesting `base_Jacobian` in measurement configuration.

DM-4234 - Getting issue details... **STATUS**

Add a measurement algorithm which records the number of input images contributing to a coadd sources

When run on a source detected on a coadd, `SingleFrameInputCountPlugin` records the number of input images which were stacked to create the coadd at the position corresponding to the source centroid. The plugin is referred to as `base_CountInputs`, and is enabled by default when performing measurement on coadded images. It is not appropriate to enable this plugin when processing single visit (i.e., not coadded) images.

DM-4235 - Getting issue details... **STATUS**

Add a measurement algorithm which records the variance at the positions of sources

The `SingleFrameVariancePlugin` records the median variance in the background around the position of the source being measured. The plugin is referred to as `base_Variance` and is enabled by default when performing single frame measurement.

DM-4235 - Getting issue details... **STATUS**   **DM-5050** - Getting issue details... **STATUS**

DM-5427 - Getting issue details... **STATUS**

Add a measurement algorithm which records source flux in an aperture scaled to the PSF

The `ScaledApertureFluxAlgorithm` measures the flux in a circular aperture with radius scaled to some user-specified multiple of the PSF FWHM. This plugin is not enabled by default, but may be switched on by requesting the `base_ScaledApertureFlux` in measurement configuration.

DM-3257 - Getting issue details... **STATUS**

Add a measurement algorithm which quantifies the amount of "blendedness" of an object

The `BlendednessAlgorithm` measures the amount to which an object is blendeded. Both the flux and shape of each child object are compared to measurements at the same point on the full image. The size of the weight function used on both images is determined from the child object. The blendedness metric implemented is defined as 1-childFlux/parentFlux. The plugin is referred to as `base_Blendedness` and is not enabled by default.

DM-4847 - Getting issue details... **STATUS**

Add a "simple" shape measurement algorithm

The `SimpleShape` algorithm, provided in the `meas_extensions_simpleShape` package, computes the non-adaptive elliptical Gaussian-weighted moments of an image. The plugin is referred to as `ext_simpleShape_SimpleShape` and is not enabled by default.

DM-5284 - Getting issue details... **STATUS**

Add Hirata-Seljak-Mandelbaum shape measurement algorithms

The `meas_extensions_shapeHSM` package has been added to the distribution. This provides a series of measurement algorithms based on the work by Hirata and Seljak (2003) and Mandelbaum (2005): please cite those works if publishing results based on this code. These algorithms are disabled by default; they can be enabled by requesting ext_shapeHSM_HsmShapeBj, ext_shapeHSM_HsmShapeLinear, ext_shapeHSM_HsmShapeKsb, ext_shapeHSM_HsmShapeRegauss, ext_shapeHSM_HsmSourceMoments and/or ext_shapeHSM_HsmPsfMoments in the measurement configuration.
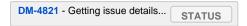
DM-2141 - Getting issue details... `STATUS`   **DM-3384** - Getting issue details... `STATUS`

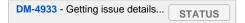**DM-4780** - Getting issue details... `STATUS`

Add option to temporarily remove an interpolated background prior to detection

This has the potential for removing a large number of junk detections around bright objects due to noise fluctuations in the elevated local background. The extra subtracted interpolated background is added back in after detection. Currently, the default setting for the config parameter `doTempLocalBackround` is set to False.

DM-4821 - Getting issue details... `STATUS`

Add function to average coordinates

Added function `lsst.afw.coord.averageCoord`, which will return an average coordinate (accounting for spherical geometry) given a list of input coordinates.

DM-4933 - Getting issue details... `STATUS`

Integrate support for Hyper Suprime-Cam

The `obs_subaru` camera package, which enables the stack to operate on data taken with the Suprime-Cam and Hyper Suprime-Cam instruments on Subaru, has been modernized, resolving build and test issues and integrating it with LSST's continuous integration system. It will now be included as part of the `lsst_distrib` release. (Note, though, that usage of Suprime-Cam with the stack is unsupported and unmaintained at present.)

DM-3518 - Getting issue details... `STATUS`   **DM-4358** - Getting issue details... `STATUS`

DM-5007 - Getting issue details... `STATUS`

Reimplement PSF Shapelet approximations for CModel

A new algorithm for computing multi-shapelet approximations (`DoubleShapeletPsfApprox`) has been added to meas_modelfit. This is simpler and more robust than the old algorithm, which has been renamed to `GeneralShapeletPsfApprox`. The new algorithm is recommended for production use, and is now the default.

DM-5197 - Getting issue details... `STATUS`

Propagate flags from individual visits to coadds

A task has been added which can propagate flags from individual visit catalogs to coadd catalogs. This is useful, for example, to track which stars in the coadd were used for measuring PSFs on the individual visits.

DM-4878 - Getting issue details... `STATUS`   **DM-5084** - Getting issue details... `STATUS`

Prototype Brighter-Fatter correction

Code for correcting for the Brighter-Fatter effect on CCDs is now available in the stack. It is enabled using the `doBrighterFatter` configuration option to `IsrTask`. It requires a pre-generated correction kernel. Calculation of this kernel is not currently performed within the stack: a prototype exists, and will be merged to the Calibration Products Pipeline in a future cycle.

DM-4837 - Getting issue details... `STATUS`   **DM-5082** - Getting issue details... `STATUS`

DM-5130 - Getting issue details... `STATUS`

Aperture correction on coadd measurements

Aperture corrections are now enabled for measurements performed on coadds.

DM-5086 - Getting issue details... `STATUS`

Return grown `Footprints` from detection

By default, `Footprint`s return by `SourceDetectionTask` are now grown by a multiple of the PSF size.

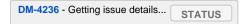**DM-4410** - Getting issue details... `STATUS`

Enable measurement of "sky objects" in coadd processing

Sky objects correspond to source properties measured at positions when no objects have been detected. This enables us to better characterize the depth of the survey. This functionality is enabled by default; it can be disabled by setting `nSkySourcesPerPatch` to zero in the configuration of `MergeDetecti onsTask`.

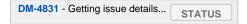**DM-4840** - Getting issue details... `STATUS` **DM-5288** - Getting issue details... `STATUS`

Specification of output directory is now mandatory

When running a command line task which produces output it is now mandatory to specify an output directory (previously, if no output location was specified, data products were written back to the input repository). Output locations may be specified with the `--rerun` or `--output` command line options. More information is available on community.lsst.org.

DM-4236 - Getting issue details... `STATUS`

Bright object masks

Given an input catalog listing the known positions and sizes of bright objects, a bit is set in the mask plane for every pixel lying within the object's footprint.

DM-4831 - Getting issue details... `STATUS`

Overhaul ProcessCcdTask and subtasks, including eliminating camera-specific variants

Overhauled ProcessCcdTask and CalibrateTask to simplify the code and eliminate the need for camera-specific variants. See Changes to ProceessCcdTask and Subtasks for details.

DM-4692 - Getting issue details... `STATUS` **DM-5348** - Getting issue details... `STATUS`

CModel fitting improvements

CModel is a model fitting approach in which a pure exponential and pure de Vacouleur are each fit separately, and then their linear combination is fit while the ellipse parameters are held fixed. Improvements in this release make CModel fitting faster and improves results on objects which are detected with an unphysically large likelihood radius. This has been achieved in three ways:

- The initial approximate exponential fit that is used to determine the starting parameters and pixel region to use for the exp and dev fit now uses per-pixel variances;
- The method used to determine the pixel region used in fitting has been adjusted to make smarter choices, using fewer pixels on average for all objects and *many* fewer pixels for unphysically large objects;
- A new semi-empirical Bayesian prior on radius and ellipticity based on COSMOS distributions has been introduced.

DM-4768 - Getting issue details... `STATUS`

Astropy Table views into LSST Catalog objects

Astropy `Table` views into LSST catalog objects can now be created. These views share underlying data buffers (aside from flag fields), making them read-write, but rows and columns added on either side will not be visible in the other. Two equivalent interfaces are available:

```
astropy_table = lsst_catalog.asAstropy()
```

and (in Astropy >= v1.2):

```
astropy_table = astropy.table.Table(lsst_catalog)
```
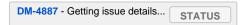
`QTable` objects can also be used, resulting in columns that use Astropy's `units` package to enforce unit correctness. These interfaces have multiple options to control the details of the view, including how to handle columns that require copies; see the Python on-line help for `asAstropy` for more information.

While LSST's catalog objects have features that make them particularly useful in building pipelines, Astropy's are much more appropriate for most analysis tasks, and we strongly recommend using them for any analysis tasks that need to add columns to tables or combine columns from multiple tables.

DM-5641 - Getting issue details... **STATUS**   **DM-5642** - Getting issue details... **STATUS**

DM-5643 - Getting issue details... **STATUS**

Add an "afterburner" measurement facility

This new functionality makes it possible to register plugins to calculate quantities based on the results of pixel measurement algorithms. This might include, for example, star-galaxy separation or applying aperture corrections. Afterburners of this type are run after measurement plugins, and do not have access to pixel data.
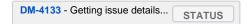
DM-4887 - Getting issue details... **STATUS**

Bug Fixes

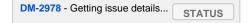The following fixes resolve problems visible to end users.

Persist `LTVn` headers as floating point numbers

When persisting to a FITS file, these header cards were previously, incorrectly, stored as integers.

DM-4133 - Getting issue details... **STATUS**

Fix bug when identifying existing peaks in a merge

If two separate footprints from the same catalog are merged due to an existing merged object which overlaps both of them the flags of which peaks are being detected were not being propagated. This issue caused apparent dropouts of sources and has now been fixed.

DM-2978 - Getting issue details... **STATUS**

Fix situation in which the `getChildren` method of `SourceCatalog` may return the wrong information

The `getChildren` method requires that the result must be sorted by parent. This is naturally the case when the catalog is produced by detection or deblending tasks. However, if multiple catalogs are concatenated together this condition may no longer be true. The `getChildren` method was updated to raise an exception if the precondition of sorting is not met.

DM-2976 - Getting issue details... **STATUS**

Fix warping when the WCS have different coordinate systems

Warping assumed that the the sky representation of both WCS was identical.

DM-4162 - Getting issue details... **STATUS**

Correct bad default `minInitialRadius` for CModel

The minInitialRadius configuration parameter had a default that is too small, causing many galaxies to be fit with point source models, leading to bad star /galaxy classifications.

DM-3821 - Getting issue details... **STATUS**

Correct algebraic error in CModel uncertainty calculation

There was a simple but important algebra error in the uncertainty calculation, making the uncertainty a strong function of magnitude.
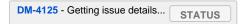
`NaiveDipoleCentroid` and `NaiveDipoleFlux` algorithms no longer require a centroid slot

Previously, initializing these algorithms was only possible if a centroid was already defined. That was not only unneccessary, but also made them more complicated to use, particularly in testing.
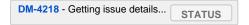
Update (some) example code to run with recent stack versions

Changes in `afw::table` had broken `examples/calibrateTask.py` in `pipe_tasks`. It has now been updated to comply with the latest `afw::table` API.

Fix a failure to appropriately log failed task execution

When task execution fails, we add a message to the log (with level `FATAL`). In some cases, the very act of attempting to log this message could throw an exception, and information about the original error was lost. This has now been resolved.
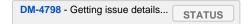
Updates to Skymap packages

Add functions to return patches and tracts which contain given coordinates, i.e. conversions between celestial coordinates and tract,patch indicies. Functions include findClosestTractPatchList, findAllTract, and findTractPatchList which finds the closets tract and patch that overlaps coordinates, finds all tracts which include the specified coordinate, and finds tracts and patches that overlap a region respectively.
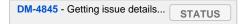
Fix variance in coadded images

Warping images in order to coadd them loses variance into covariance. This is mitigated by scaling the variance plane of the coadd. The scaling was being applied incorrectly in some cases. This has now been fixed.

Fix variance in deblended sources

The deblender incorrectly scaled the variance plane in deblended sources with the fraction of the total flux assigned to the source. This has been corrected.

Fix logic for applying aperture corrections

This fixes a bug whereby the aperture corrections were being applied only after all the measurement plugins had run through, independent of their execution order. This resulted in plugins whose measurements rely on aperture corrected fluxes (i.e. with execution order > `APCORR_ORDER`) being applied prior to the aperture correction, leading to erroneous results. The only plugin that was affected by this at this time was `base_ClassificationExtendedness`.
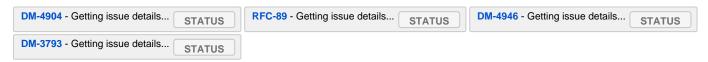
More uniform support for assigning to catalog columns

Assignment of scalars or NumPy arrays to columns of `afw.table.Catalog` objects (e.g. `"catalog["column"] = value"`) is now more uniformly supported across types (support was inconsistent before, and never allowed scalar or augmented assignment). `Flag` columns still do not support column assignment, and `Flag` column access still returns a copy, not a view, because `Flag` values are stored internally as individual bits within a larger integer.

DM-4856 - Getting issue details... `STATUS`

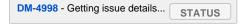## Upgraded WCSLIB to version 5.13

Version 5.13 of WCSLIB resolves memory corruption errors that could crash the stack in some circumstances.

DM-4904 - Getting issue details... `STATUS`   **RFC-89** - Getting issue details... `STATUS`   **DM-4946** - Getting issue details... `STATUS`

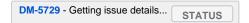**DM-3793** - Getting issue details... `STATUS`

Fix rotation for instrument signature removal in obs_subaru

Approximately half of the HSC CCDs are rotated 180 deg with respect to the others. Two others have 90 deg rotations and another two have 270 deg rotations (see HSC CCD layout). The raw images for the rotated CCDs thus need to be rotated to match the rotation of their associated calibration frames (in the context of how they have currently been ingested) prior to applying the corrections. This is accomplished by rotating the exposure using the **rotated** context manager function in obs_subaru's **isr.py** and the **nQuarter** specification in the policy file for each CCD. Currently, **rotated** uses afw's **rotateImage By90** (which apparently rotates in a counter-clockwise direction) to rotated the exposure by 4 - nQuarter turns. This turns out to be the wrong rotation for the odd nQuarter CCDs. This issue fixes this bug, leading to much improved processing of HSC CCD's 100, 101, 102, and 103. Note that, in the future, the ingestion of the calibration data will be updated such that no rotations are necessary (so they will then be removed from `obs_subaru` accordingly).

DM-4998 - Getting issue details... `STATUS`

Fix a silent failure to apply config overrides

When applying a config override, using a variable which hadn't been defined should throw a `NameError`, which ultimately propagages to the end user to notify them that something has gone awry. This warning was being incorrectly suppressed.

DM-5729 - Getting issue details... `STATUS`

Correctly set flags for bad `SdssShape` measurements.

The `SdssShape` algorithm provides both shape and flux measurements. In some cases, a failed shape measurement could go un-noticed, resulting in an incorrect and unflagged flux measurement being associated with that source. This is now checked for, and bad fluxes are appropriately flagged.

DM-3935 - Getting issue details... `STATUS`

Fix `CONSTANT` background interpolation of bad data

When performing interpolation over bad data (e.g. every pixel masked), all interpolation types other than `CONSTANT` would return `NaNs`; `CONSTANT` would throw. This has now been changed so that `CONSTANT` also returns `NaNs`.

DM-5797 - Getting issue details... `STATUS`

Accommodate pixel padding when unpersisting reference catalog matches

The reference object loader in meas_algorithm's **loadReferenceObjects.py** grows the bbox by the config parameter **pixelMargin:** doc = "Padding to add to 4 all edges of the bounding box (pixels)" when setting the radius of the sky circle to be searched in the reference catalog. This is set to 50 by default but was not reflected by the radius parameter set in the metadata, which left open the possibility that some matches could reside outside the circle searched within the unpersisted radius. Additionally, the match metadata was being set after the exposure's wcs had been updated, also leading to an inconsistency with the sky circle that was actually searched. We now ensure that the actual sky circle that was searched for reference objects is the one set and persisted in the match metadata.

DM-5686 - Getting issue details... `STATUS`

Correct misleading filter fallback error message

When failing to load a calib, if fallbackFilterName was not set, a confusing and apparently unrelated error message would be generated (`Unknown value type for filter: <type 'NoneType'>`). This has been corrected to properly inform the user about the issue.

DM-6151 - Getting issue details... `STATUS`

Build and code improvements

These improvements should not usually be visible to end users. They may, however, be important for developers.

Work-around incompatibilities with NumPy 1.10

NumPy 1.10 introduced API changes which were incompatible with existing usage in the stack. The latter has been updated to match

DM-4063 - Getting issue details... **STATUS**   **DM-4071** - Getting issue details... **STATUS**

DM-4238 - Getting issue details... **STATUS**

When building boost warn user if user-config.jam or site-config.jam exists

Building boost can fail if a user-config.jam or site-config.jam exist and have options which conflict with the lsst build configuration process. Introduce a warning message if either of these files are found to notify the user.

DM-4198 - Getting issue details... **STATUS**

Remove deprecated `Task.display()` method

This method has been deprecated since release 9.2 (S14). It has been removed from the codebase, and all stack code updated to directly interface with `af w.display` or to use helper functions defined in `meas_astrom`.

DM-4428 - Getting issue details... **STATUS**

Efficiency improvement in converting `Masks` to `DefectLists`

The previous version of routine was extremely memory intensive when large numbers of pixels were masked.

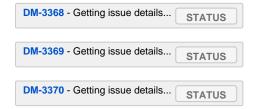**DM-4800** - Getting issue details... **STATUS**

Add a new task parallelization framework

The `ctrl_pool` package has been added to the LSST stack. This is a high-level parallelization framework used for distributing `Task` execution across a cluster, based on an MPI process pool. It is based on work carried out on Hyper Suprime-Cam. It is not intended to be the long-term solution to parallelized processing in the LSST stack, but meets our data processing needs until the fully-fledged parallelization middleware is available.

DM-2983 - Getting issue details... **STATUS**   **DM-4835** - Getting issue details... **STATUS**

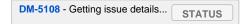**DM-5409** - Getting issue details... **STATUS**

Add parallel-processing top level tasks

The new `pipe_drivers` package builds upon `ctrl_pool`, above, to provide command-line scripts which coordinate distributed execution of the single-frame, coaddition and multiband processing steps using either the Python `multiprocessing` module or with a SLURM batch scheduler on a cluster.

DM-3368 - Getting issue details... **STATUS**

DM-3369 - Getting issue details... **STATUS**

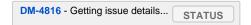**DM-3370** - Getting issue details... **STATUS**

Adjust test tolerances to be compatible with MKL-based NumPy

Anaconda 2.5 ships, by default, with a version of NumPy built against Intel MKL rather than OpenBLAS. This can change some numerical results slightly, necessitating a change to test tolerances.

DM-5108 - Getting issue details... **STATUS**

Now possible to directly get a `Filter`'s canonical name and aliases

Added the convenience methods `getCanonicalName()` and `getAliases()` to `lsst.afw.image.Filter`, accessible from both C++ and Python. These return the canonical name and the aliases, respectively, of the filter. This information was previously only available through an awkward sequence of method calls.

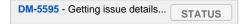**DM-4816** - Getting issue details... STATUS

Fix build issues with recent clang

Recent releases of the clang C/C++ compiler, as shipped with Apple XCode, caused build failures in the stack. Although we believe this may be a problem with clang, we have worked around it within the stack code. We hope to track down the source of the error and, if appropriate, report it to the clang developers in future.
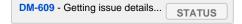
DM-5590 - Getting issue details... STATUS   **DM-5609** - Getting issue details... STATUS

Fix incorrect linking against Anaconda-provided libraries when using CMake

Some external packages -- `mariadb`, `mariadbclient` -- use a CMake based build system. This can incorrectly link against some libraries bundled with the Anaconda Python distribution, rather than the system-provided equivalents; in the end, this results in a build failure. We have adjusted the build process of the affected packages to work around this error.

DM-5595 - Getting issue details... STATUS

Execute afw test suite when afwdata is not available

Some tests in the afw package rely on data from the afwdata package. The test suite would search for afwdata, and skip *all* of the afw tests if afwdata is not available. This check has been made smarter, so that only tests which actually require afwdata are now skipped.

DM-609 - Getting issue details... STATUS

Disable implicit threading

Low-level threading packages (such as OpenBLAS or MKL) can implicitly use many threads. Since the LSST stack also parallelizes at a higher level (e.g. using Python's `multiprocessing` module), this can cause underdesireable contention. We now disable implicit threading when explicitly parallelizing at a higher level to protect the user from this. Implicit threading can be re-enabled by setting the `LSST_ALLOW_IMPLICIT_THREADS` environment variable. For more details, see this CLO post.

DM-4719 - Getting issue details... STATUS

Migrate to standard smart pointers

C++11 introduced new smart pointer types (`std::unique_ptr`, `std::shared_ptr` and `std::weak_ptr`). We have migrated from the previously used Boost smart pointers to their standard equivalents.

DM-5879 - Getting issue details... STATUS   **DM-4008** - Getting issue details... STATUS

RFC-100 - Getting issue details... STATUS   **DM-5966** - Getting issue details... STATUS