

Table of Contents

What's New in HIPE 7.....	1
Core system.....	1
HIFI.....	4
PACS.....	5
SPIRE.....	8

What's New in HIPE 7

PDF Version [Click here](#)

■ Interested in what's new for other versions of HIPE? See this page for links to all the What's New documents from HIPE 4.0 onwards.

This document lists the changes in HIPE version 7.0 with respect to the 6.x series. Additional pages list changes between minor versions of the 7.x series.

- Changes between HIPE 7.0 and 7.1
- Changes between HIPE 7.1 and 7.2
- Changes between HIPE 7.2 and 7.3

See also the **HIPE known issues** and the **Data products known issues** pages for last-minute information on known problems.

Core system

HIPE

Console

- Incremental search (Ctrl+I) available.
- Console help text enhanced.
- New internal commands `!clear`, `!help` and `!history`.

History

- Detailed error information can be opened with a double click in the *Trace* column of the *History* view.

Status bar

- Progress bar shows the current line number of the executing script.
- New bin-shaped button to the right of the memory bar, at the bottom of the HIPE window, to run the *garbage collector* and free memory manually.

Generic viewers

Dataset viewer

- Multiple cells can be selected and copied in the TableDataset viewer.

Jython editor

- Find/Replace dialog enhanced. It can now show now whitespace characters.
- Autoindentation of Jython code improved.
- Indent and unindent buttons added to the toolbar.
- If the *Monospaced* font is not really monospaced, HIPE proposes an alternate font.

Astronomical utilities

RadialVelocity task and utility class renamed to RadialV and now using the proper Doppler sign convention.

Numeric routines

Wavelet transform

- new friendly interfaces are now available:
 - ◆ `algo = ContinuousWavelet("MexicanHat")`
 - ◆ `algo = DiscreteWavelet("db5")`
 - ◆ `algo = StationaryWavelet("coif3")`

Images

Display

- Adapted the popup menu to make it possible to work with axes, add a compass and add a scale.
- Display now returns the commands done using the popup menu to the HIPE console. This makes it very easy to reconstruct an image plot from within a script.
- When dragging source positions to an image, the circles are locked. When dragging contours to an image, the contours are also locked.
- Added method **addArcSecs** to add a line of a given length in arcseconds to an image.
- When using the opacity slider with a cube, the next image is used as the underlying image, not the next layer of the cube anymore.

Analysis

- You can now load all monochrome or indexed gif files using **importRgbImageTask**.

Spectra

Display

Analysis

- Addition to SpectrumFitter: create ASCII file with fit parameters. Use in the command line:
 - ◆ `sf = SpectrumFitter(...)`
 - ◆ ...
 - ◆ `m = sf.addModel(...)`
 - ◆ ...
 - ◆ `sf.doFit()` or `sf.doGlobalFit()`
 - ◆ `sf.saveFitParametersAsASCII(fileName)` # writes parameters of all models to fileName
 - ◆ `m.toASCII(dirName)` # writes parameters to dirName/[modelname].txt
- The SpectrumFitterGUI has, in the export TAB, also the possibility to export fit parameters to file.
- Addition to SpectrumFitterGUI: user preference for initial models. If switched on, you can select from:
 - ◆ Instrument specific initial model.
 - ◆ User specific initial models - you can specify the models.
- Addition to SpectrumFitterGUI, MultiTAB:
 - ◆ Some information on what will happen.
 - ◆ Fit parameters are written to file.
 - ◆ A count-down progress bar.

Calibration sources

Integrated the Calibration Source Database into HIPE. There is a *Calibrators* view in the *Work Bench*

perspective, next to the *Navigator* view. You may have to reset the perspective for it to appear.

The *Calibrators* view allows access to the database of calibration sources at ESAC, used by HSC calibration scientists. It contains spectra of planets, asteroids, and fixed targets. The data can be viewed in the usual HIPE viewers for spectra and images, and manipulated using Jython commands.

Product Access Layer

- The HTTP pool now supports digest as well as basic authentication.

Tasks

The default Task GUI has been improved with the following upgrades:

- Parameter error detection (red X) also checks the `validator` of the parameter.
- A new `Help` button gives easy direct access to the URM description of the task.
- White space in layout moved to the lower part of the panel.
- Editors based on text fields now require half the width.
- Labels of parameters have now a maximum size to reserve space for the editors.
- Support for grouping inputs in multiple tabs. For tasks that have inputs related to different aspects or simply a lot of them, they can be presented in several tabs.

There is now a detailed outline shown in the *Outline* view when a task is selected in the *Tasks* view. This allows inspection of the structure of a task without the need to either use the *Console* view or open its GUI in the *Editor* view.

Data input-output

- `simpleFitsReader` and `simpleFitsWriter` now also support reading/writing datasets (not just products).
- The *Export* view now shows the tags associated with each observation version.
- The `localStoreWriter` task now allows to add a tag to the saved product.
- `localStoreWriter` and `localStoreCopier` now use a better integrated GUI to select and create a pool.
- File filters for file dialogs are now self-explanatory (accepted extensions ...).
- The `clear` task now has a simplified syntax to delete all variables (`clear()`, without any parameter, will delete all variables).

Virtual Observatory

The VO external tools have been made accessible from the Tools menu.

We have also added a few pre-set applications (Topcat, Aladin and VOSpec) to the Send To menu. This menu is accessible by right-clicking on a variable. The pre-set applications are visible, even if the application in question is not running and/or HIPE is not connected to the VO. Using these pre-set is transparent to the user: When sending a variable, the application will be automatically started first and the data will be sent.

Documentation

- Replaced the *Read Me First* document with the Welcome documentation page, and by a linked page listing the available manuals.
- In the welcome page, added links to a YouTube channel[↗] of tutorials and a Twitter channel[↗] of tips.
- Completely revised text and screenshots of the Quick Start Guide.

- **Data Analysis Guide**
 - ◆ Added section on the benefits of pools.
 - ◆ Added information on inspecting the quality of an observation.
 - ◆ Updated section on working with the VO.
 - ◆ Added section on importing and exporting images.
 - ◆ Improved tip on changing the colour and size of source symbols on images.
 - ◆ Updated the Spectrum Fitter section.
 - ◆ Added draft section on combining PACS and SPIRE data.
- **Scripting and Data Mining guide**
 - ◆ Many updates to the chapter on storing and accessing data products.
 - ◆ In particular, updated section on pools for remote data.
- **HIPE Owner's Guide**
 - ◆ Revised the whole document.
 - ◆ Improved instruction on updating Java on a Mac.
 - ◆ Added warning on HSA password and unencrypted property files.
 - ◆ Mentioned the new `hipe_help` command.
 - ◆ Documented the new Console view commands.
 - ◆ Added a partial list of keyboard shortcuts.
- **Frequently Asked Questions**
 - ◆ Revised the whole document, removed obsolete entries.
 - ◆ Added entry 9.4 on problems with monospaced fonts.
 - ◆ Added entry 10.1 on temporary pools not deleted.
- **Products Definitions Document**
 - ◆ Replaced some hardcoded references with hyperlinks.
 - ◆ Updates to Chapter 5 and Chapter 11 for Level 2.5 products.

Plug-ins

It is now possible for plug-in authors to publish new versions of plug-ins, which will be automatically picked up by existing users of the plug-in. If a new version of a plug-in is found by HIPE, and the author has indicated that the update is compatible with the user's version of HIPE, then HIPE will ask the user if the plug-in should be updated. The update mechanism is automatic, though HIPE must be restarted for the changes to be completed.

HIFI

HIFI Pipeline

- `doPipelineConfiguration` now passes changes to the level 2 pipeline. To use this task, pass an HTP to the GUI (`doPipelineConfiguration`), modify the pipeline steps as required. Press accept (default name of output is `out`) then pass this new configuration to the pipeline task:

```
obs_new = hifiPipeline(obs=obs, params=out, save=False)
```

HIFI Data Processing Tools

Standing Wave Removal

- new default `doplot=2` option opens no more than 2 `fitHifiFringe` plot windows at the time and allows for inspection before processing next spectrum
- `averscan` option `fitHifiFringe` has been removed as standing wave characteristics vary too much between scans
- `fitHifiFringe` now shows up as applicable task to both HTPs and `ObsContexts`

Baseline Removal

- New non-stop option in interactive mode of fitBaseline allows automatic processing of remaining spectra
- The htcopy option of fitBaseline has been removed as modified HTPs are now properly displayed in SpectrumExplorer

Deconvolution

- doDeconvolution now produces a result even if deconvolution fails. In this case an error message is given
- doDeconvolution now read flags from USB, rather than LSB. Channel flags have always been read from the USB so this change makes it easier for you to pass flagged data to the task

HiClass export tool

- Obsid now exported again to CLASS header

PACS

Pipeline

Photometry

Improvements for SSO scan maps

Assuming the object move linear a linear interpolation between start and end position is done. This method provides 'smoother' coordinates especially for point source.

Old behavior :

```
frames = correctRaDec4Sso(frames, horizonsProduct=horizonsProduct, timeOffset=timeOffset, orbitEp  
frames = correctRaDec4Sso(frames, horizonsProduct=horizonsProduct, timeOffset=timeOffset, linear=
```

New implementation :

```
frames = correctRaDec4Sso(frames, horizonsProduct=horizonsProduct, timeOffset=timeOffset, linear=
```

Aberration correction

Change the order applying SIAM and Aberration correction in **photAddInstantPointing** and **specAddInstantPointing** . This shall have very minor impact on the coordinates, but correct reverse order of uplink.

Spectroscopy

- All level-2 tasks now do an internal loop over line ID. They behave now the same as the level-0/1 tasks: If you do not select a particular line/range before then the level-2 tasks just process all lines/ranges. As a consequence the level-2 products have changed a bit:
- We no longer have a list context for each line ID within another list context: All slices of a level-2 product are now stored in a single list context. This is much more practical when selecting slices based on arbitrary criteria.
- The pacsSliceContext task now contains all the rules in the ipipe scripts as default rules: Also, it slices by line ID, raster column, raster line, nodding position, nod cycle number, is out of field and band.

The rules in the scripts are also now optional and merely for information.

- The interactive pipeline scripts (ipipe) are now tested in the nightly tester
- The interactive pipeline scripts (ipipe) are moved in the build from the directory scripts/pacs/spg/pipeline/ipipe/spec to scripts/pacs/scripts/ipipe/spec since they had to be moved to the package pacs_scripts

Calibration

Photometry

Flux calibration

The photometer flux calibration has changed (new response calibration file "FM, 6"). Absolute fluxes of point-sources will change by less than 5%, extended sources will change by 10-20%. Aperture photometry of point-sources requires from now on the application of new aperture correction factors, taking into account that there is still significant flux beyond 60 arcsec.

These new correction factors (together with the old ones) are stored in:

PCalPhotometer_ApertureCorrection_FM_v2.fits and can be retrieved via the new task **photApertureCorrectionPointSource** to be applied after **annularSkyAperturePhotometry** .

See also the update of PACS ipipe script Menu -> Pipeline -> PACS -> Photometer -> Copped point source -> standard pipeline -> L3_pointSourceAperturePhotometry.py

You need to update your local calibration tree Menu to activate it (Menu -> Tools -> PACS_CAL -> run Updater).

You may check by printing the calibration tree :

```
HIPE> c=getCalTree()  
HIPE> print c.photometer
```

MadMap

New calibration files containing the actual 'best' calibration data for MadMap :

PCalPhotometer_InvnttBL_FM_v4.fits

PCalPhotometer_InvnttBS_FM_v4.fits

PCalPhotometer_InvnttRed_FM_v4.fits

You need to update your local calibration tree Menu to activate it (Menu -> Tools -> PACS_CAL -> run Updater).

You may check by printing the calibration tree :

```
HIPE> c=getCalTree()  
HIPE> print c.photometer
```

Spectroscopy

- Wrong coordinates in Spectrometer results: update of spec spatial cal file
- Improvements in wavelength calibration scatter for spectral pixels in band B2B -> wavelength calibration file updated

Analysis tools

Slicing

- Saving intermediate data on machines with low memory is now done more clever: The slices are only saved to the temporary pool when the memory usage is more than 75% and the sliced product contains more than one slice.

- The temporary pools of HIPE are now written under a directory 'null value' which is created where you start HIPE ([HcssSpr:12911](#)).

Workaround: Set the following property in your user.props file:

```
hcss.ia.pg.temporalpoolparent = ${user.home}/.hcss/lstore
```

These temporary pools are called **tmp_product_sink_NNNNN**

Photometry

- A new Mask Array Viewer displays Masks with their correct size (also the timeline).
- Add start time of last cooler recycling to meta data of products A new calibration product was introduced : CoolerRecyclingTimes
 - ◆ A convenience tool `times = extractCoolerRecTimes(obs, calTree)` returns the last cooler recycling of an observation
 - ◆ This file will be updated on a regular base

Spectroscopy

- Map2signalCubeTask gets an additional shortcut: `m2sc` (besides `map2signalCube`)
- Spectrum Explorer : New extension panel to edit masks is now fully working
- Spectrum Explorer : Performance improvements

Product changes

- `calTree` version is now in meta data of products

Photometry

Spectroscopy

- Frames and PacsCube meta data is made consistent for every product change (like select or join)
- Additional meta keywords added: `onOffSource`, `minWave`, `maxWave`, `nodCycle`, `lineId`
- All masks which are not excluded in rebinning are propagated from PacsCube to PacsRebinnedCube. They can be obtained with the `getFlag()` method which returns a Flag dataset, which is similar to a PACS mask.
- We have now two more sliced products: `SlicedPacsCube?` and `SlicedPacsRebinnedCube?`. `SlicedPacsCube?` offers the same functionality as `SlicedFrames?`, but `SlicedPacsRebinnedCube?` has limited functionality, since it does not have a blocktable. Add, remove and replace work, but selection should be done with the `pacSelectSlices` task.

Each slice is described by the following slice info meta keywords (recognizable by description prefix "SLICE_INFO:"):

Keyword	Values	Description
<code>lineId</code>	0..N	Numerical ID of a distinct wavelength range (usually a line scan)
<code>lineDescription</code>	string	Which spectral line has been observed (or tells that it is a range scan)
<code>nodPosition</code>	A/B	Nodding position
<code>nodCycleNum</code>	1..N	Nodding cycle number
<code>isOffPosition</code>	false/true	Is it an off position according to the <code>IsOutOfField</code> status?
<code>onOffSource</code>	0,1,2	Based on new <code>ONSOURCE/OFFSOURCE</code> Status keywords: 0 = not science, 1 = on-source, 2 = off-source
<code>rasterId</code>	N M	first value: raster column, second value: raster line
<code>calBlock</code>	false/true	Belongs to slice to a calibration block?

minWave	float	Minimum wavelength of range
maxWave	float	Maximum wavelength of range
aotMode	string	AOT mode

Also, since these keywords are now always consistent with the BlockTable and Status contents they can and should be used for slice selection. The new task **pacSelectSlices** does this meta-keyword based selection. It has the same interface as the selectSlices Python function.

- All level-2 tasks now do an internal loop over line ID. They behave now the same as the level-0/1 tasks: If you do not select a particular line/range before then the level-2 tasks just process all lines/ranges. As a consequence the level-2 products have changed a bit:
- The wavelength grid is now also stored in a list context, since one needs to pass several ranges from task to task.
- We no longer have a list context for each line ID within another list context: All slices of a level-2 product are now stored in a single list context. This is much more practical when selecting slices based on arbitrary criteria.
- Status item "GratingCycle" and "IndexInCycle" is now created for unchopped modes

Documentation

- New documentation "How to work with masks" :
<http://herschel.esac.esa.int/twiki/bin/view/Pacs/HowToWorkWithMasks>

SPIRE

Calibration Products

- Spectrometer calibration products for nominal detector settings have been updated to make use of in-flight bolometer model parameters.
- An updated TempDriftCorr product allows a new algorithm in the Signal Jump Detector module to be used (nominal mode only).

Calibration Framework

- The `spireCal` task has new features and modified behaviour:
 - ◆ It will look for a calibration tree in a local pool by default. The default is the one that was last saved if applicable; otherwise the default for the HIPE version.
 - ◆ It will only download from the HSA if explicitly requested via the `calTree` parameter.
 - ◆ Optionally the tree can be saved to a local pool. In this case that one will become the default.
 - ◆ The trees available in the archive can be listed.
- The API has new methods for showing differences between two calibration trees and for verifying that expected metadata is present and correct.

Common Pipeline

- The **AssociateSkyPositionTask** now creates a new table "avgVelocity" in photometer level1 scanmap products. The table contains two columns: "sampleTime" (TAI) and "avgVelocity" (deg/sec)
- The **AssociateSkyPositionTask** now now accepts two new optional parameters:
 - ◆ **scanVelocityTolerance**: double value, in arcsec/sec, defining the scan velocity tolerance relative the the mode's expected scan velocity. Signal samples outside this range will have their NON_NOMINAL_VELOCITY mask set. (default: 1.0"/sec)
 - ◆ **scanVelocityLimits**: Double1d([lowerVelocityLimit, upperVelocityLimit]), in arcsec/sec, defining a range of scan Velocities. Signal samples with angular velocity outside this range will have their NON_NOMINAL_VELOCITY mask set. Setting this parameter, parameter

"scanVelocityTolerance" will be ignored. Setting any velocity limit to a negative value, the task will ignore the scanVelocityLimits parameter and revert to using "scanVelocityTolerance".

Photometer Pipeline

- The **LpfResponseCorrectionTask** now accepts a new optional, boolean input parameter "doMirrorSignal". Setting this parameter to true, the task will perform the correction, by mirroring the signal time-lines in the y-axis. The effect of this is to avoid any edge-ringing, resulting from the default correction method when applied to timelines with a large slope. The draw-back of the new method is that it can double the processing time of the Task, however, not to a detrimental effect to single observation processing (It will definitely be noticeable when processing a large number of observations).
- The mosaicing of maps from overlapping observations in a proposal is now fully functional (this is actually available from 6.1).

Spectrometer Pipeline

Level 0 to Level 0.5

No changes.

Level 0.5 to Level 1

Changes to existing Tasks

- **WaveletDeglitcherTask** now offers the option to make the number of corrected samples dependent on the glitch size by setting the `optionReconstruction` parameter to "adaptative" or "adaptativeLinear".
- **SpecFluxConversionTask** no longer automatically removes non-seeing channels. One must now use `FilterChannelsTask` to remove these channels from `SpectrometerDetectorSpectrum` products.
- **SpecFluxConversionTask** now precedes **SpecTelescopeCorrectionTask** in the Standard pipelines.
- **SpecTelescopeCorrectionTask** now, by default, subtracts only a model of the Herschel telescope and not the model multiplied by the RSRF for Telescope (Sky) port.

Level 1 to Level 2

New Tasks

- **SpireProduct2SimpleSpectrumTask** to convert SPIRE Level-1 and Level-2 data to `SimpleSpectrum` products
- **GriddingProjectionTask** for spectral cube creation. Given `SpirePreprocessedCube` input, `GriddingProjectionTask` produces a spectral cube output. This task uses HIFI's `GriddingTask`.

Changes to existing Tasks

- **NaiveProjectionTask** now creates a spectral cube containing error instead of weight.
- **AverageSpectraTask** no longer provides functionality to remove out-of-band spectral samples. One must use `RemoveOutOfBandTask` instead.
- **AverageSpectraTask** no longer provides functionality to reject outlier from the average computation.

Interactive Analysis and Tools

- New task **DestriperTask** to iteratively remove the relative offsets between Level-1 detector timelines of the same detector array, and thus removing the stripes in the Level-2 map. Signal timelines are re-sampled from a preliminary naive map and subtracted from the original timelines. The resulting offset functions are modeled as polynomials with default degree ($n=0$) and subtracted. A new naive map is generated from the new difference of original signal and polynomial for the next iteration. The bright source threshold is used to suppress the influence of strong spikes in the data. The primary output is a `SpireListContext` containing updated `PointedPhotTimelines`. The additional outputs are a `SimpleImage` map, the convergence ratio, `Tod` etc.
- New tasks **BaselineRemovalMedianTask** and **BaselineRemovalPolynomialTask** operate on SPIRE Level-1 data to remove a baseline from the data. This is necessary due to the residual baseline that remains in the Level-1 data, which can produce stripes in the map if not removed. For each bolometer, the signal timeline is taken, either for each scan line separately, or for the whole observation as one long timeline. Then either the median value is subtracted (**BaselineRemovalMedianTask**), or a polynomial is fit to the data and subtracted. These tasks can be thought of as a "lite" version of the **DestriperTask**: they are simpler and faster, but for some observations containing a lot of structure (such as Galactic cirrus), there will be significant improvement if the **DestriperTask** is used.

Documentation

- Pipeline Specification Manual: Entries on spectroscopy have been reviewed, examples have been validated.

Please enable JavaScript to view the comments powered by Disqus. blog comments powered by Disqus

This topic: Public > HipeWhatsNew7x

Topic revision: r43 - 2012-10-30 - DavideRizzo

