Introduction to HIPE 12.1
(Part 2)

Bernhard Schulz (NHSC/IPAC)
on behalf of
the HSC, the ICCs and the NHSC

Going Further: Contents of Part 2

- Pipeline execution
- Saving processing results in local databases (pools)
- Herschel Archive access through HIPE
- Data pool management
- Data exchange with other applications (SAMP)
- Tasks
- Libraries
- Some practical tips

Data Reprocessing

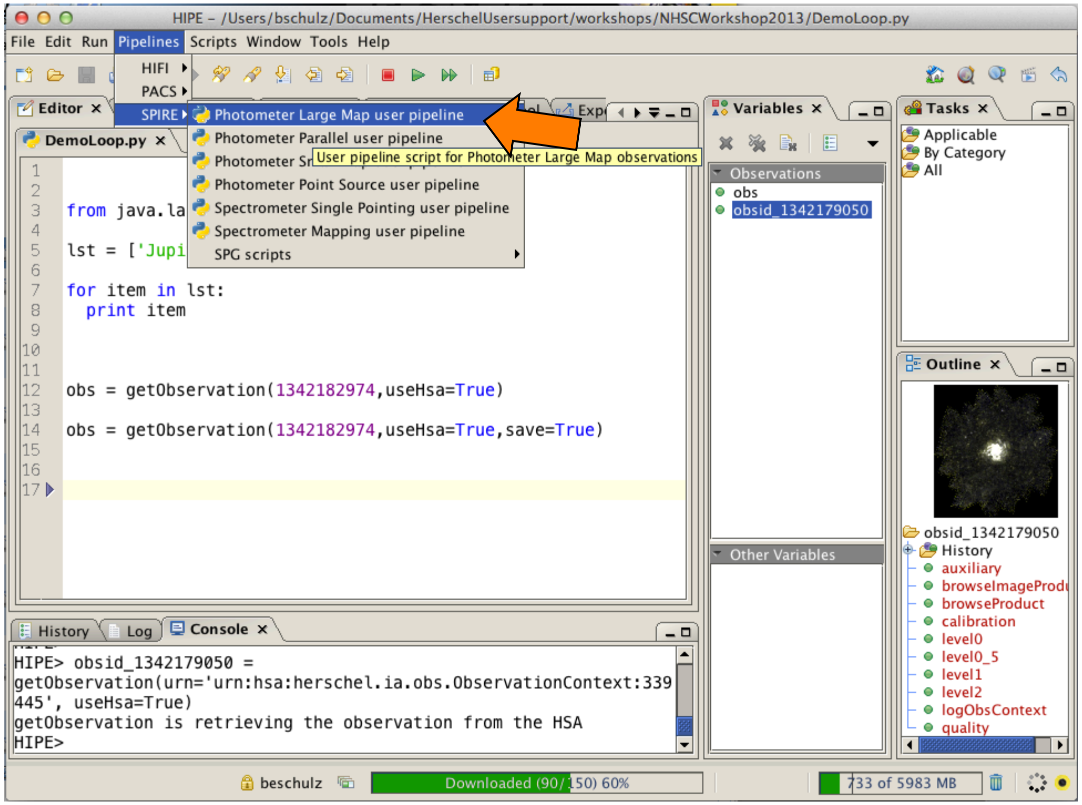
- Sometimes the data inspection reveals shortcomings in the data processing and the astronomer wishes to perform a data re-processing in HIPE.
- Data reprocessing is usually done in Jython pipeline scripts or by specific "Tasks".

Running a Pipeline Script

- Some custom scripts are available in the HIPE menu already.
- Locations:
  - Under "Pipelines" the main pipeline scripts can be found that are used to produce the archive results.
  - Under "Scripts" more Jython scripts are available for more specialized non-standard purposes.
- Script Execution in the Editor View:
  - The single green arrow executes one statement or block at the lowest level.
  - The green double arrow executes the entire file.
  - Indented blocks like for-loops or if-then statements can not be stepped through line by line.
  - The pause() statement sets break-points and helps with debugging inside of indented blocks.
  - Single lines or groups of lines inside blocks can be executed with limitations by marking them blue and hitting the single green arrow.

Example: Running the SPIRE Pipeline

- Select in the Pipelines Menu: "Photometer Large Map user pipeline"
- Edit the script without saving and insert
  - myObsid=1342182974
  - myDataPool='myhsa'
  - outDir='<output directory>'
- Run the script by hitting the green double arrow in the top toobar.
- After a while the updated observation will appear as obs under Variables and in the Outline view.

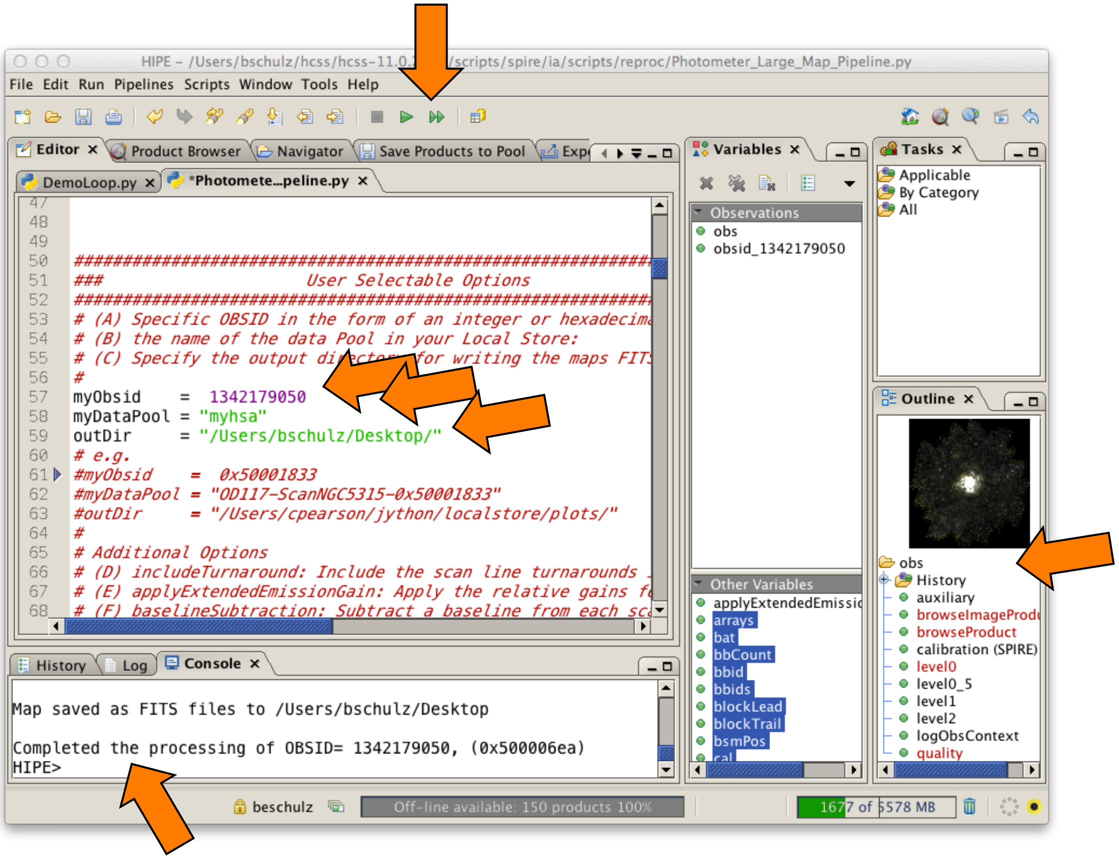


```
from java.la
lst = ['Jupi
for item in lst:
  print item


obs = getObservation(1342182974,useHsa=True)

obs = getObservation(1342182974,useHsa=True,save=True)
```

```
HIPE> obsid_1342179050 =
getObservation(urn='urn:hsa:herschel.ia.obs.ObservationContext:339
445', useHsa=True)
getObservation is retrieving the observation from the HSA
HIPE>
```

Example: Running the SPIRE Pipeline

- Select in the Pipelines Menu: “Photometer Large Map user pipeline”
- Edit the script without saving and insert
  - myObsid=1342182974
  - myDataPool=‘myhsa’
  - outDir=‘<output directory>’
- Run the script by hitting the green double arrow in the top toobar.
- After a while the updated observation will appear as obs under Variables and in the Outline view.



```
######################################################################
###                            User Selectable Options
######################################################################
# (A) Specific OBSID in the form of an integer or hexadecima
# (B) the name of the data Pool in your Local Store:
# (C) Specify the output directory for writing the maps FITS
#
myObsid    =  1342179050
myDataPool = "myhsa"
outDir     = "/Users/bschulz/Desktop/"
# e.g.
#myObsid    =  0x50001833
#myDataPool = "OD117-ScanNGC5315-0x50001833"
#outDir     = "/Users/cpearson/jython/localstore/plots/"
#
# Additional Options
# (D) includeTurnaround: Include the scan line turnarounds 
# (E) applyExtendedEmissionGain: Apply the relative gains fo
# (F) baselineSubtraction: Subtract a baseline from each sca
```

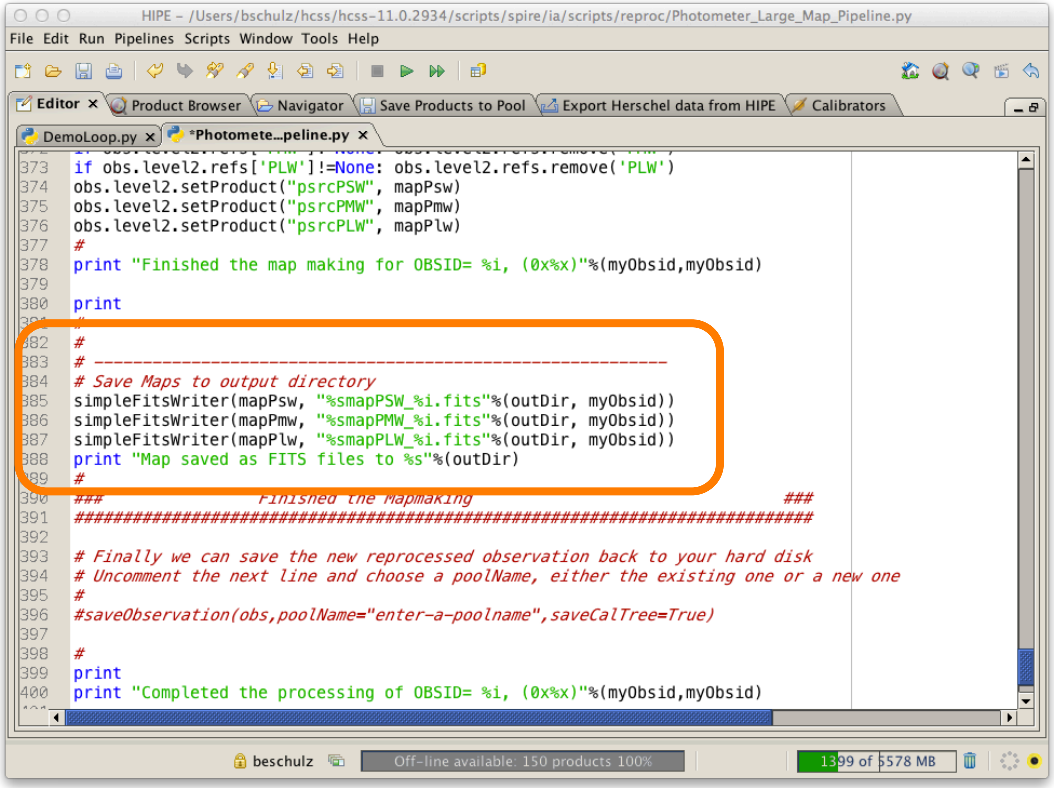
```
Map saved as FITS files to /Users/bschulz/Desktop

Completed the processing of OBSID= 1342179050, (0x500006ea)
HIPE>
```

Saving Products in FITS Files

- If only a few products are of interest after processing is complete, each product can be saved to a FITS file.
  - Right-click on the product (e.g. a map) and select Send-to → FITS-file.
  - In the upcoming GUI of the task simpleFitsWriter() enter a filename with path and hit the “Accept” button.
  - The equivalent command appears in the Command View and can be re-used in a Jython script.

Saving Products to FITS files

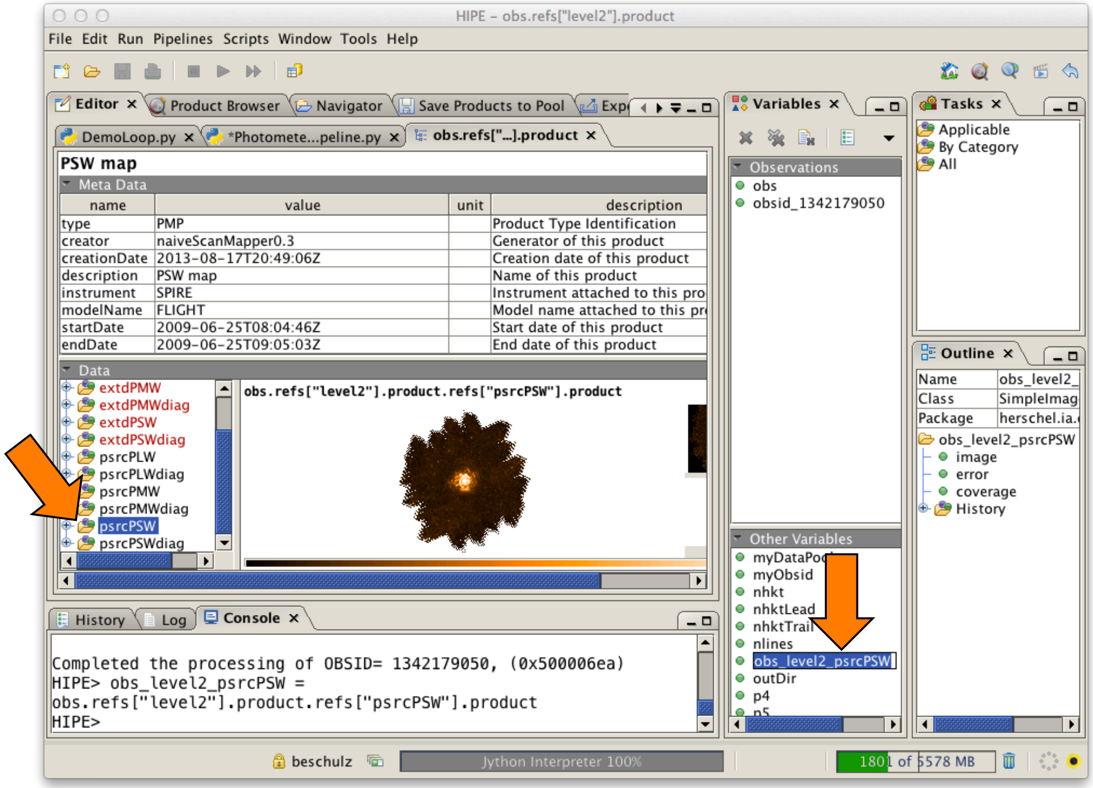
- The script already saved the three map products as FITS files.
- To try using the GUI:
- Double-click Level 2 in the Outline view.
- Drag a map from Level 2 into the variable space.
- Right click on the new variable and select “Send-to” and “FITS-file”.
- In the upcoming GUI enter a filename with path and hit the “Accept” button.
- Note that you just used a Task from its default GUI (more about this later).
- Note also that the scripted version of what you just did appears in the Console View.


HIPE – /Users/bschulz/hcss/hcss-11.0.2934/scripts/spire/ia/scripts/reproc/Photometer_Large_Map_Pipeline.py
File
Edit
Run
Pipelines
Scripts
Window
Tools
Help
Editor
Product Browser
Navigator
Save Products to Pool
Export Herschel data from HIPE
Calibrators
DemoLoop.py
*Photomete...peline.py
beschulz
Off-line available: 150 products 100%
1399 of 5578 MB

```
if obs.level2.refs['PLW']!=None: obs.level2.refs.remove('PLW')
obs.level2.setProduct("psrcPSW", mapPsw)
obs.level2.setProduct("psrcPMW", mapPmw)
obs.level2.setProduct("psrcPLW", mapPlw)
#
print "Finished the map making for OBSID= %i, (0x%x)"%(myObsid,myObsid)

print
#
#
# ------------------------------------------------------------
# Save Maps to output directory
simpleFitsWriter(mapPsw, "%smapPSW_%i.fits"%(outDir, myObsid))
simpleFitsWriter(mapPmw, "%smapPMW_%i.fits"%(outDir, myObsid))
simpleFitsWriter(mapPlw, "%smapPLW_%i.fits"%(outDir, myObsid))
print "Map saved as FITS files to %s"%(outDir)
#
###                Finished the mapmaking                 ###
#############################################################################

# Finally we can save the new reprocessed observation back to your hard disk
# Uncomment the next line and choose a poolName, either the existing one or a new one
#
#saveObservation(obs,poolName="enter-a-poolname",saveCalTree=True)

#
print
print "Completed the processing of OBSID= %i, (0x%x)"%(myObsid,myObsid)
```

Saving Products to FITS files

- The script already saved the three map products as FITS files.
- To try using the GUI:
- Double-click Level 2 in the Outline view.
- Drag a map from Level 2 into the variable space.
- Right click on the new variable and select “Send-to” and “FITS-file”.
- In the upcoming GUI enter a filename with path and hit the “Accept” button.
- Note that you just used a Task from its default GUI (more about this later).
- Note also that the scripted version of what you just did appears in the Console View.



PSW map

Meta Data

name	value	unit	description
type	PMP		Product Type Identification
creator	naiveScanMapper0.3		Generator of this product
creationDate	2013-08-17T20:49:06Z		Creation date of this product
description	PSW map		Name of this product
instrument	SPIRE		Instrument attached to this pro
modelName	FLIGHT		Model name attached to this pr
startDate	2009-06-25T08:04:46Z		Start date of this product
endDate	2009-06-25T09:05:03Z		End date of this product

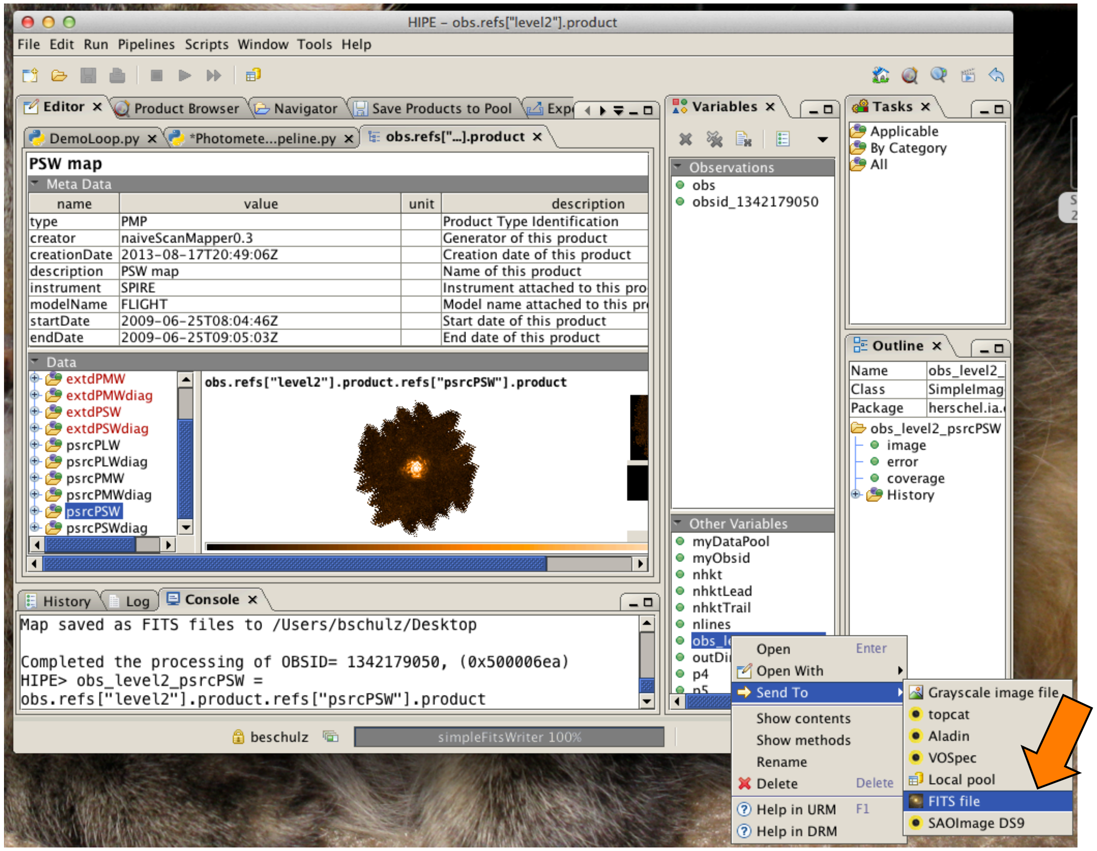
```
obs.refs["level2"].product.refs["psrcPSW"].product
```

Name	obs_level2_
Class	SimpleImag
Package	herschel.ia.

```
Completed the processing of OBSID= 1342179050, (0x500006ea)
HIPE> obs_level2_psrcPSW =
obs.refs["level2"].product.refs["psrcPSW"].product
HIPE>
```

Saving Products to FITS files

- The script already saved the three map products as FITS files.
- To try using the GUI:
- Double-click Level 2 in the Outline view.
- Drag a map from Level 2 into the variable space.
- Right click on the new variable and select "Send-to" and "FITS-file".
- In the upcoming GUI enter a filename with path and hit the "Accept" button.
- Note that you just used a Task from its default GUI (more about this later).
- Note also that the scripted version of what you just did appears in the Console View.



PSW map

Meta Data

name	value	unit	description
type	PMP		Product Type Identification
creator	naiveScanMapper0.3		Generator of this product
creationDate	2013-08-17T20:49:06Z		Creation date of this product
description	PSW map		Name of this product
instrument	SPIRE		Instrument attached to this pro
modelName	FLIGHT		Model name attached to this pr
startDate	2009-06-25T08:04:46Z		Start date of this product
endDate	2009-06-25T09:05:03Z		End date of this product

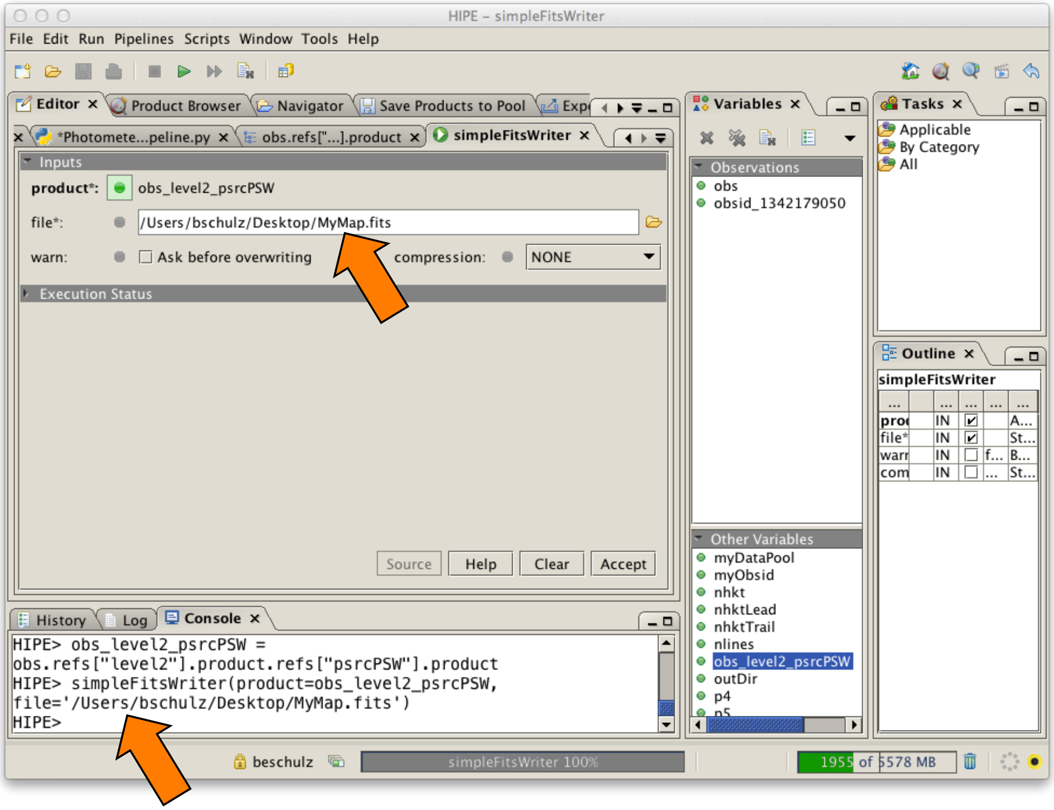
```
obs.refs["level2"].product.refs["psrcPSW"].product
```

Name	obs_level2_
Class	SimpleImag
Package	herschel.ia.

```
Map saved as FITS files to /Users/bschulz/Desktop

Completed the processing of OBSID= 1342179050, (0x500006ea)
HIPE> obs_level2_psrcPSW =
obs.refs["level2"].product.refs["psrcPSW"].product
```

Saving Products to FITS files

- The script already saved the three map products as FITS files.
- To try using the GUI:
- Double-click Level 2 in the Outline view.
- Drag a map from Level 2 into the variable space.
- Right click on the new variable and select "Send-to" and "FITS-file".
- In the upcoming GUI enter a filename with path and hit the "Accept" button.
- Note that you just used a Task from its default GUI (more about this later).
- Note also that the scripted version of what you just did appears in the Console View.



product*: obs_level2_psrcPSW

file*: /Users/bschulz/Desktop/MyMap.fits

warn: ☐ Ask before overwriting

compression: NONE

simpleFitsWriter

...		...	...	...	...
prod		IN	☑		A...
file*		IN	☑		St...
warr		IN	☐	f...	B...
com		IN	☐	...	St...

```
HIPE> obs_level2_psrcPSW =
obs.refs["level2"].product.refs["psrcPSW"].product
HIPE> simpleFitsWriter(product=obs_level2_psrcPSW,
file='/Users/bschulz/Desktop/MyMap.fits')
HIPE>
```

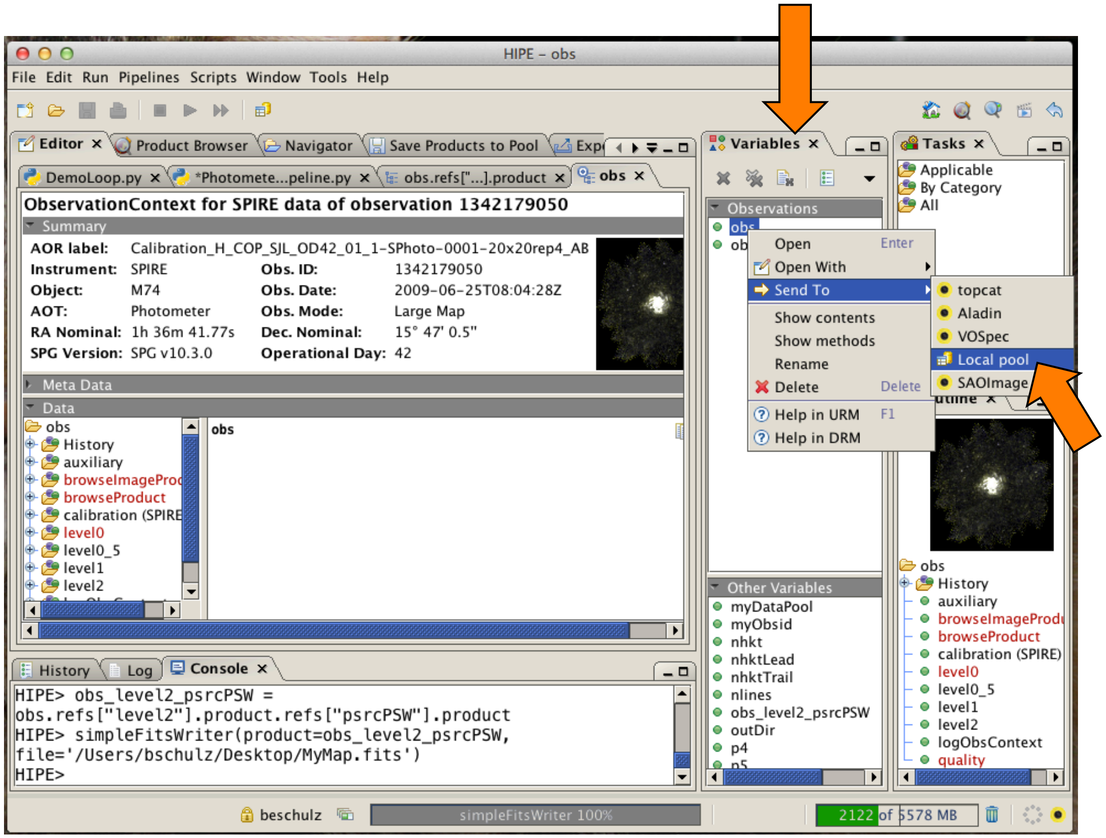
Saving Observations in Local Pools

- If you work on an observation and update many of its products, and you want to save a copy of your work for the next session, saving single FITS files is impractical.
- In this case the better way is to use a Local Pool.
- Local Pools are databases for products and contexts that keep the product tree structure intact.
- Saving and retrieving entire observation trees can be performed very easily using the observation context as a handle.

Saving Observations in Pools

- Select the observation context in the Variable View.
- Select "Send To" and "Local pool".
- Find observation in list.
- Add some text (Tag) for identification.
- Enter a destination pool name (not existing ones will be created new).
- Hit 'Save' button.
- Note the command line representation:
  `saveProduct(product=obs, pool='myTestPool', tag='obs 1342179050 my last reduction')`

Variables View



ObservationContext for SPIRE data of observation 1342179050

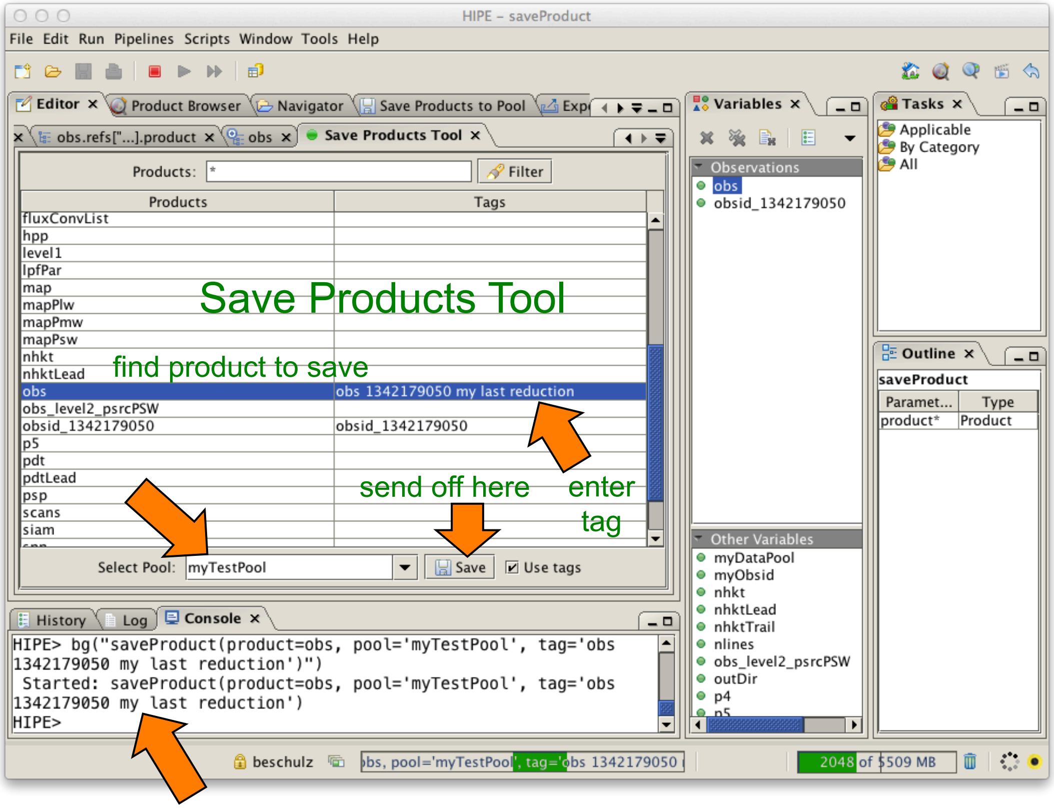
AOR label:	Calibration_H_COP_SJL_OD42_01_1-SPhoto-0001-20x20rep4_AB		
Instrument:	SPIRE	Obs. ID:	1342179050
Object:	M74	Obs. Date:	2009-06-25T08:04:28Z
AOT:	Photometer	Obs. Mode:	Large Map
RA Nominal:	1h 36m 41.77s	Dec. Nominal:	15° 47' 0.5"
SPG Version:	SPG v10.3.0	Operational Day:	42

```
HIPE> obs_level2_psrcPSW =
obs.refs["level2"].product.refs["psrcPSW"].product
HIPE> simpleFitsWriter(product=obs_level2_psrcPSW,
file='/Users/bschulz/Desktop/MyMap.fits')
HIPE>
```

Send to local pool

Saving Observations in Pools

- Select the observation context in the Variable View.
- Select "Send To" and "Local pool".
- Find observation in list.
- Add some text (Tag) for identification.
- Enter a destination pool name (not existing ones will be created new).
- Hit 'Save' button.
- Note the command line representation:
`saveProduct(product=obs, pool='myTestPool', tag='obs 1342179050 my last reduction')`



Products: * Filter

Products	Tags
fluxConvList	
hpp	
level1	
lpfPar	
map	
mapPlw	
mapPmw	
mapPsw	
nhkt	
nhktLead	
obs	obs 1342179050 my last reduction
obs_level2_psrcPSW	
obsid_1342179050	obsid_1342179050
p5	
pdt	
pdtLead	
psp	
scans	
siam	

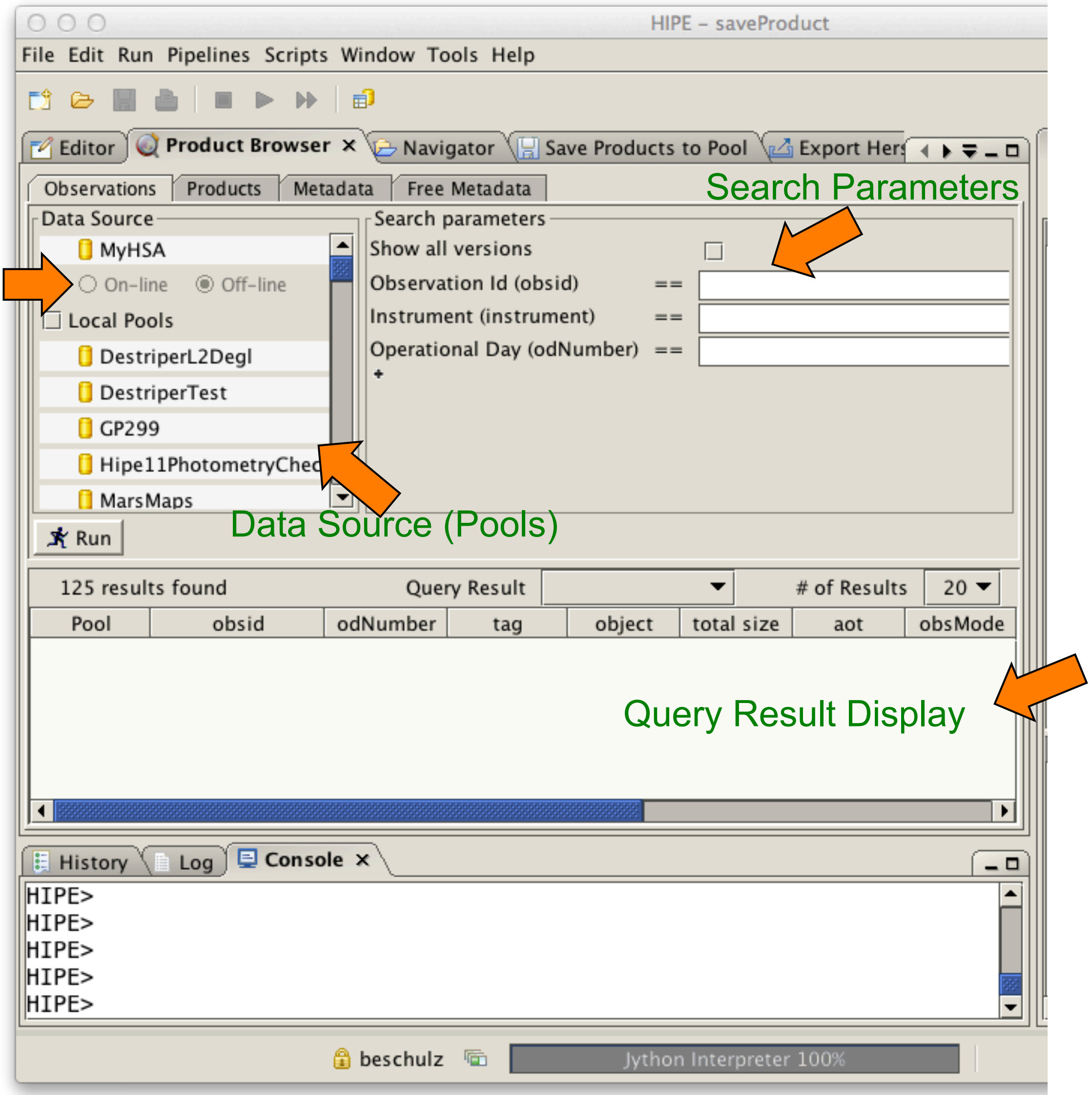
Select Pool: myTestPool ▾ Save ☑ Use tags

```
HIPE> bg("saveProduct(product=obs, pool='myTestPool', tag='obs
1342179050 my last reduction')")
 Started: saveProduct(product=obs, pool='myTestPool', tag='obs
1342179050 my last reduction')
HIPE>
```

command line form uses
task saveProduct()

The Product Browser

- The Product Browser can be found in "Window" as a view under "Data Access".
- The main panels are "Data Source", "Search parameters", and "Query Result Display"
- Data Source shows all the pools in your storage directory and the myHsa pool.
- On the right some search criteria can be chosen.
- The "On-line" button enables querying the HSA through the internet if an observation isn't present locally in the MyHSA pool.



Search parameters		
Show all versions		□
Observation Id (obsid)	==	
Instrument (instrument)	==	
Operational Day (odNumber)	==	

125 results found Query Result # of Results 20

Pool	obsid	odNumber	tag	object	total size	aot	obsMode

```
HIPE>
HIPE>
HIPE>
HIPE>
HIPE>
```

On-line Off-line selector

Search Parameters

Data Source (Pools)

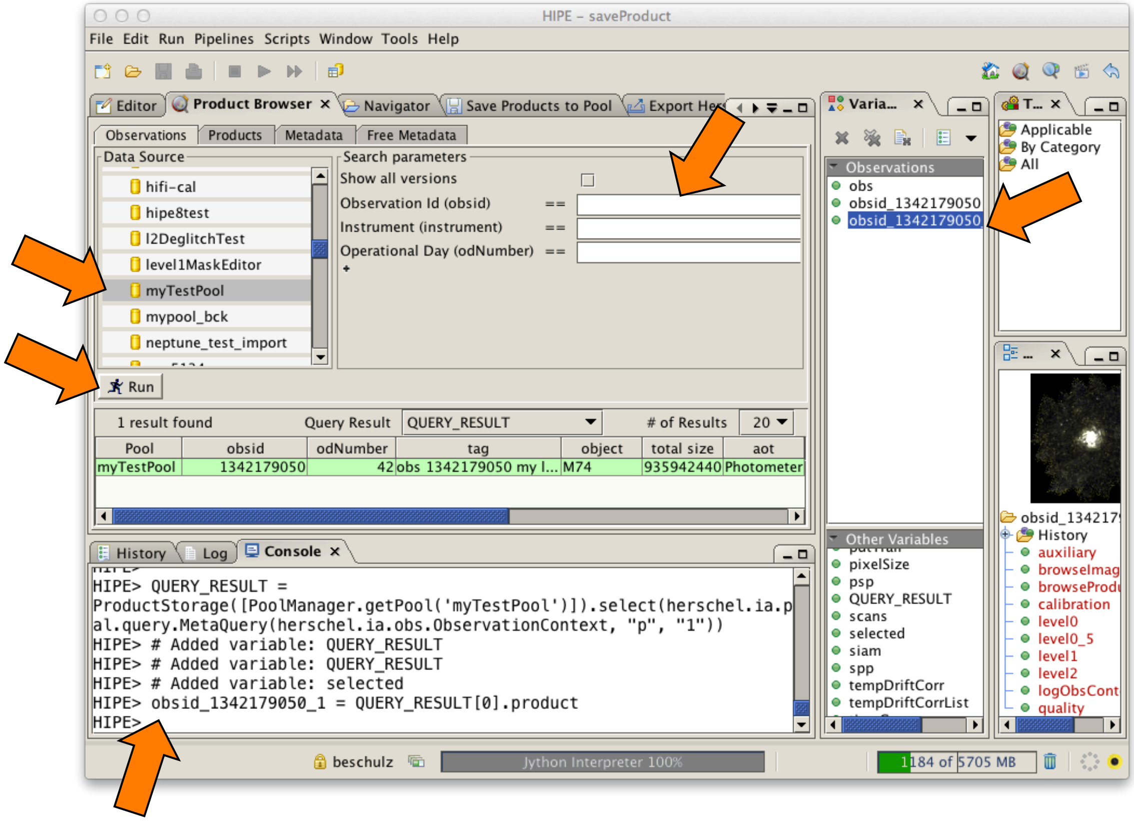
Query Result Display

Managing Observations and Pools

- Pools can be created, inspected, renamed, moved, exported, and deleted.
- The Product Browser helps to easily visualize and manage the content of pools.
- The myHSA pool is special:
  - MyHsa acts as a local cache between the HSA at the Herschel Science Center and your HIPE session on your computer.
  - All pools can be written to, except for the "myHsa" pool.
  - The other pools are intended for saving results that are different from those in the HSA, i.e. products from your own processing and analysis.
  - The data transfer between the HSA and the myHSA pool is optimized.
  - Saving observations that were retrieved straight from the HSA into a pool, is less efficient.
- The Product Browser can be used to retrieve observations from the archive by querying myHsa with the On-line option on.

Pool Content and Retrieval

- Find and select the newly created pool “myTestPool”.
- Hit the “Run” button.
- The query result list will show one line representing the observation that was just saved.
- Double-clicking the observation will retrieve the observation context.
- Note the more complex script equivalent in the Console View.

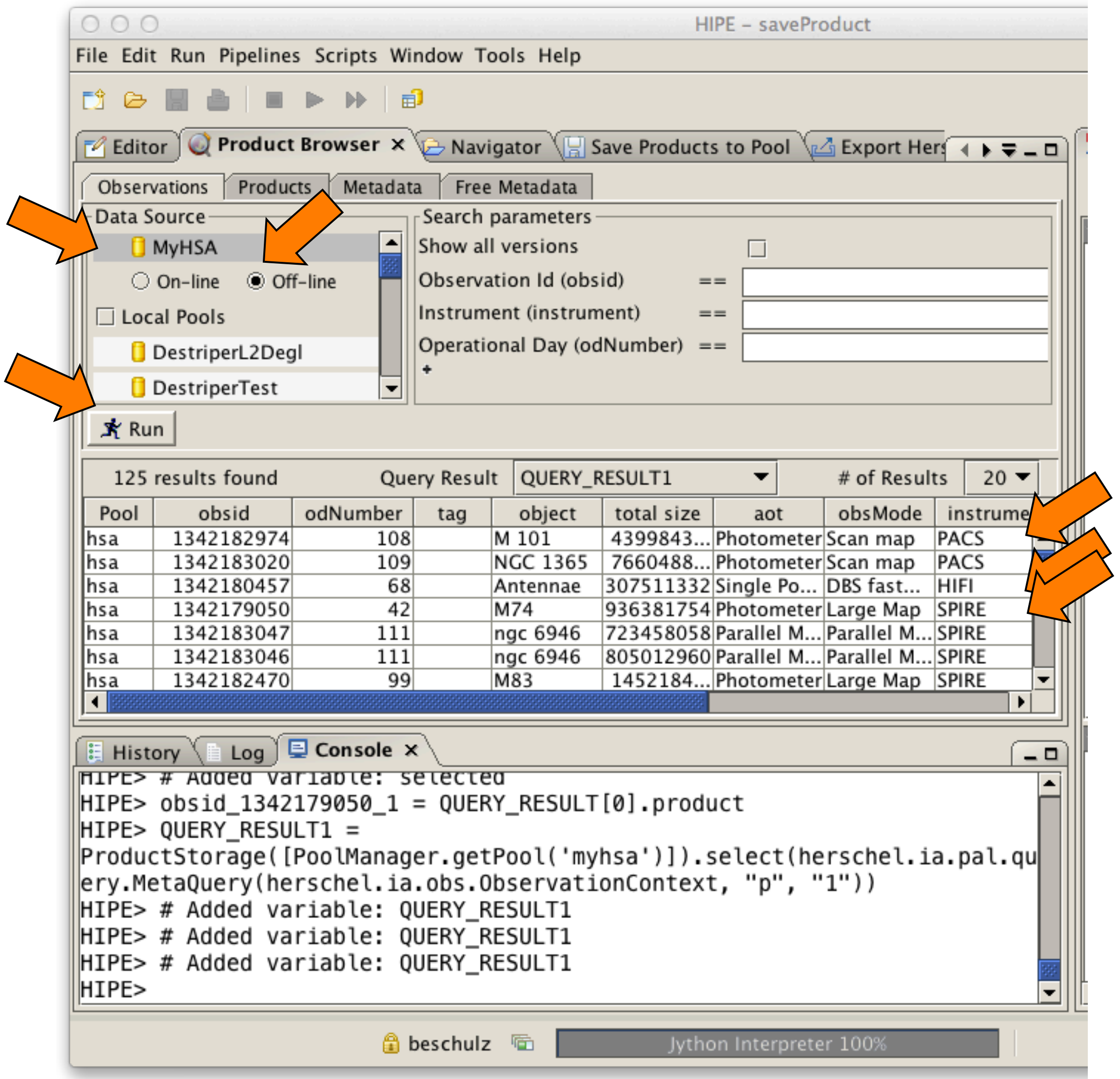


Pool	obsid	odNumber	tag	object	total size	aot
myTestPool	1342179050	42	obs 1342179050 my l...	M74	935942440	Photometer

```
HIPE> QUERY_RESULT =
ProductStorage([PoolManager.getPool('myTestPool')]).select(herschel.ia.p
al.query.MetaQuery(herschel.ia.obs.ObservationContext, "p", "1"))
HIPE> # Added variable: QUERY_RESULT
HIPE> # Added variable: QUERY_RESULT
HIPE> # Added variable: selected
HIPE> obsid_1342179050_1 = QUERY_RESULT[0].product
HIPE>
```

MyHSA Access

- To access observations in the local myHSA pool, select MyHSA
- Ensure "Off-line" is selected.
- Hit the "Run" button.
- Note the presence of the earlier imported observations from SPIRE, HIFI and PACS.



○ On-line ● Off-line

□ Local Pools

Show all versions □
Observation Id (obsid) ==
Instrument (instrument) ==
Operational Day (odNumber) ==

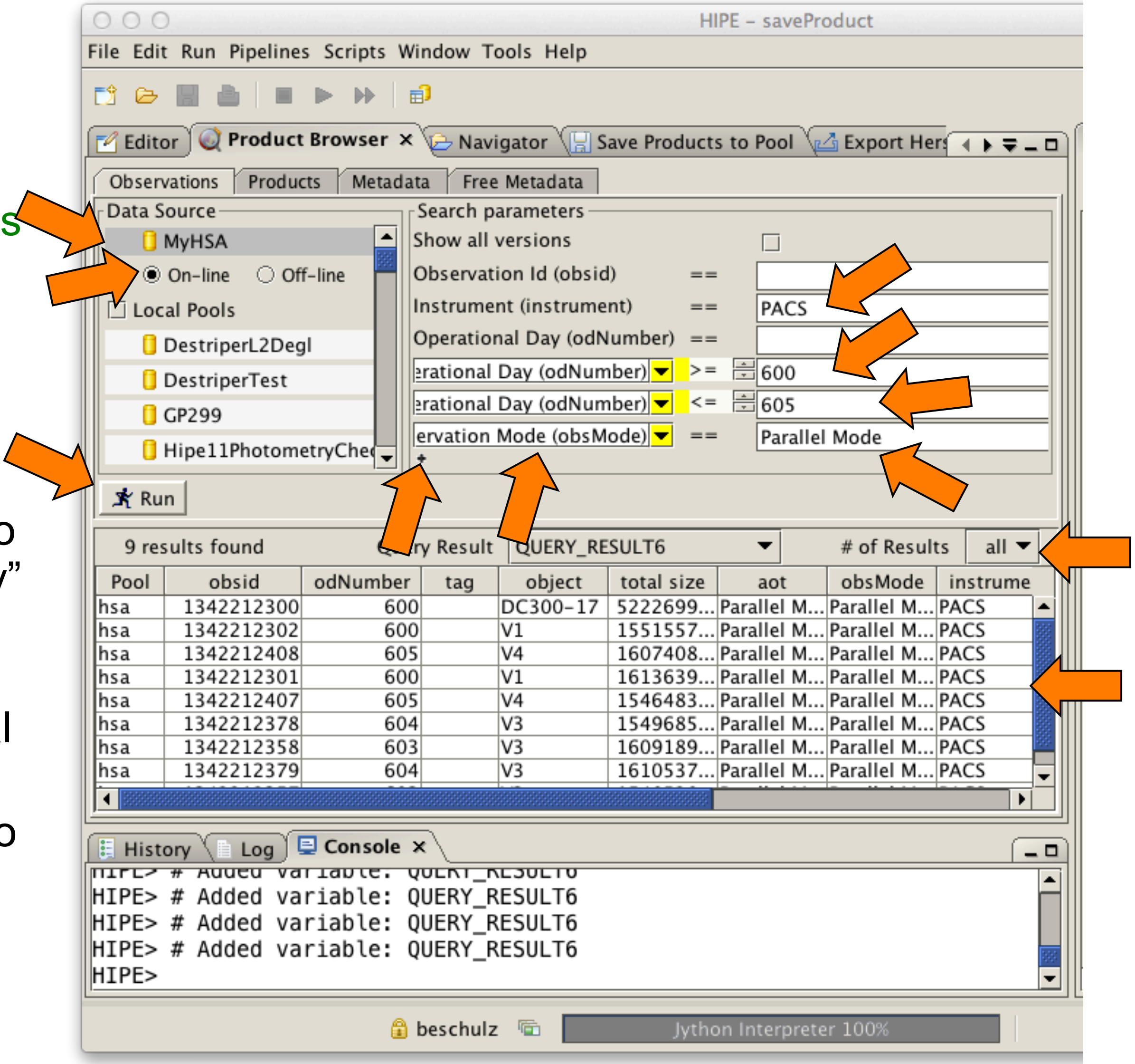
125 results found Query Result QUERY_RESULT1 # of Results 20

Pool	obsid	odNumber	tag	object	total size	aot	obsMode	instrume
hsa	1342182974	108		M 101	4399843...	Photometer	Scan map	PACS
hsa	1342183020	109		NGC 1365	7660488...	Photometer	Scan map	PACS
hsa	1342180457	68		Antennae	307511332	Single Po...	DBS fast...	HIFI
hsa	1342179050	42		M74	936381754	Photometer	Large Map	SPIRE
hsa	1342183047	111		ngc 6946	723458058	Parallel M...	Parallel M...	SPIRE
hsa	1342183046	111		ngc 6946	805012960	Parallel M...	Parallel M...	SPIRE
hsa	1342182470	99		M83	1452184...	Photometer	Large Map	SPIRE

```
HIPE> # Added variable: selected
HIPE> obsid_1342179050_1 = QUERY_RESULT[0].product
HIPE> QUERY_RESULT1 =
ProductStorage([PoolManager.getPool('myhsa')]).select(herschel.ia.pal.qu
ery.MetaQuery(herschel.ia.obs.ObservationContext, "p", "1"))
HIPE> # Added variable: QUERY_RESULT1
HIPE> # Added variable: QUERY_RESULT1
HIPE> # Added variable: QUERY_RESULT1
HIPE>
```

On-line Archive Access Example

- Let's find all parallel mode PACS observations on observational days 600 and 605.
- Switch to "On-line"
- Enter "PACS" into Instrument.
- Use the + sign to add two more "Observational Day" search criteria and one "Observation Mode".
- Select appropriate logical operators and numbers.
- Enter "Parallel Mode" into Observation Mode
- Hit "Run".
- Select "all" to show the full list of results.



● On-line ○ Off-line

Show all versions □

Observation Id (obsid)	==	
Instrument (instrument)	==	PACS
Operational Day (odNumber)	==	
erational Day (odNumber)	>=	600
erational Day (odNumber)	<=	605
ervation Mode (obsMode)	==	Parallel Mode

+

9 results found

Query Result: QUERY_RESULT6

# of Results: all

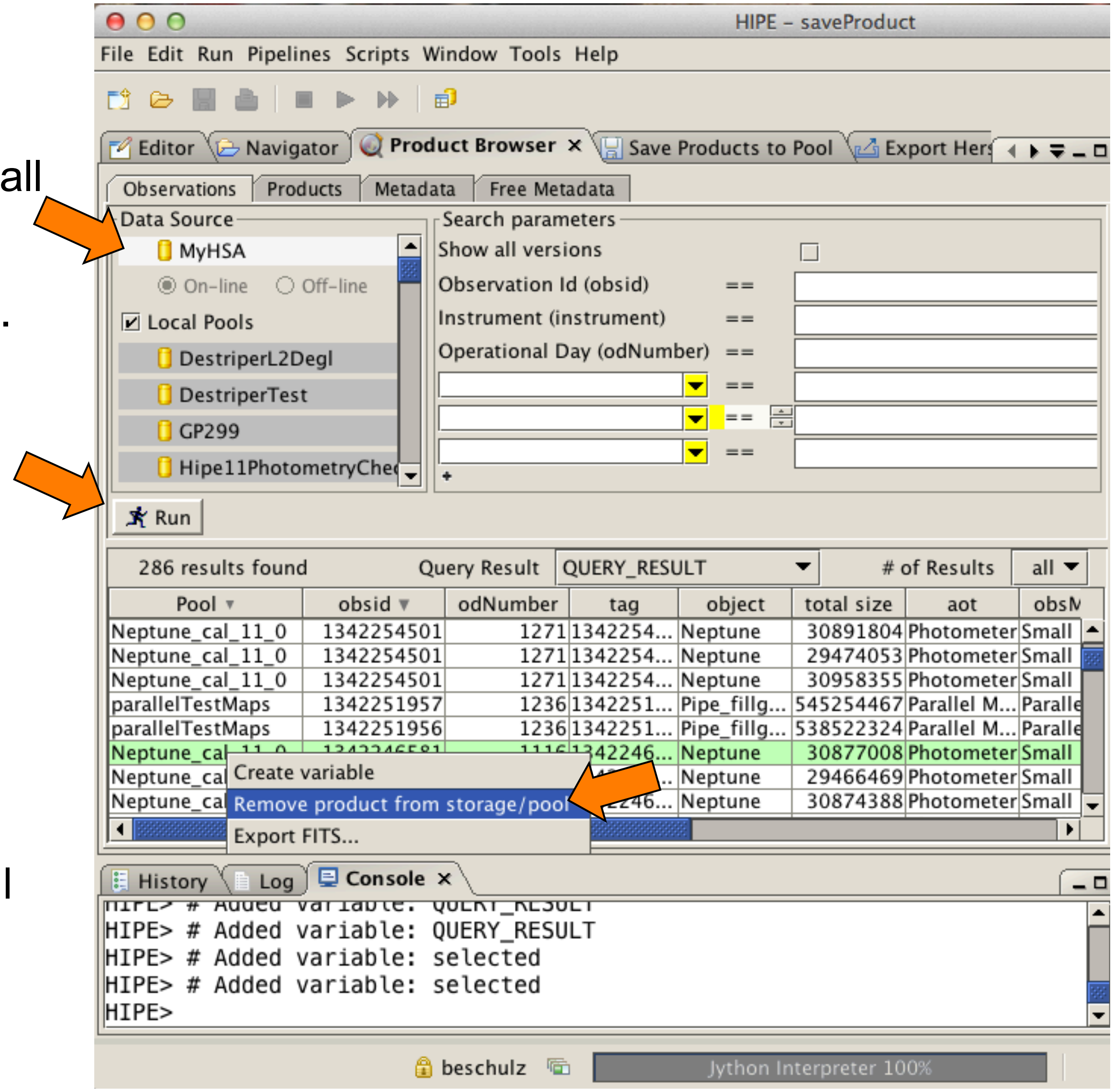
Pool	obsid	odNumber	tag	object	total size	aot	obsMode	instrume
hsa	1342212300	600		DC300-17	5222699...	Parallel M...	Parallel M...	PACS
hsa	1342212302	600		V1	1551557...	Parallel M...	Parallel M...	PACS
hsa	1342212408	605		V4	1607408...	Parallel M...	Parallel M...	PACS
hsa	1342212301	600		V1	1613639...	Parallel M...	Parallel M...	PACS
hsa	1342212407	605		V4	1546483...	Parallel M...	Parallel M...	PACS
hsa	1342212378	604		V3	1549685...	Parallel M...	Parallel M...	PACS
hsa	1342212358	603		V3	1609189...	Parallel M...	Parallel M...	PACS
hsa	1342212379	604		V3	1610537...	Parallel M...	Parallel M...	PACS

```
HIPE> # Added variable: QUERY_RESULT6
HIPE> # Added variable: QUERY_RESULT6
HIPE> # Added variable: QUERY_RESULT6
HIPE> # Added variable: QUERY_RESULT6
HIPE>
```

Managing Observations

- Find all observations in all local pools.
- De-select "MyHSA" by clicking on another pool.
- Select "Local Pools".
- Hit "Run".
- Right click on an observation.
- The menu allows to:

1. load an observation context into RAM (create variable),
2. Delete an observation from a pool (Remove product….),
3. Export FITS (not useful because it is just a context.).



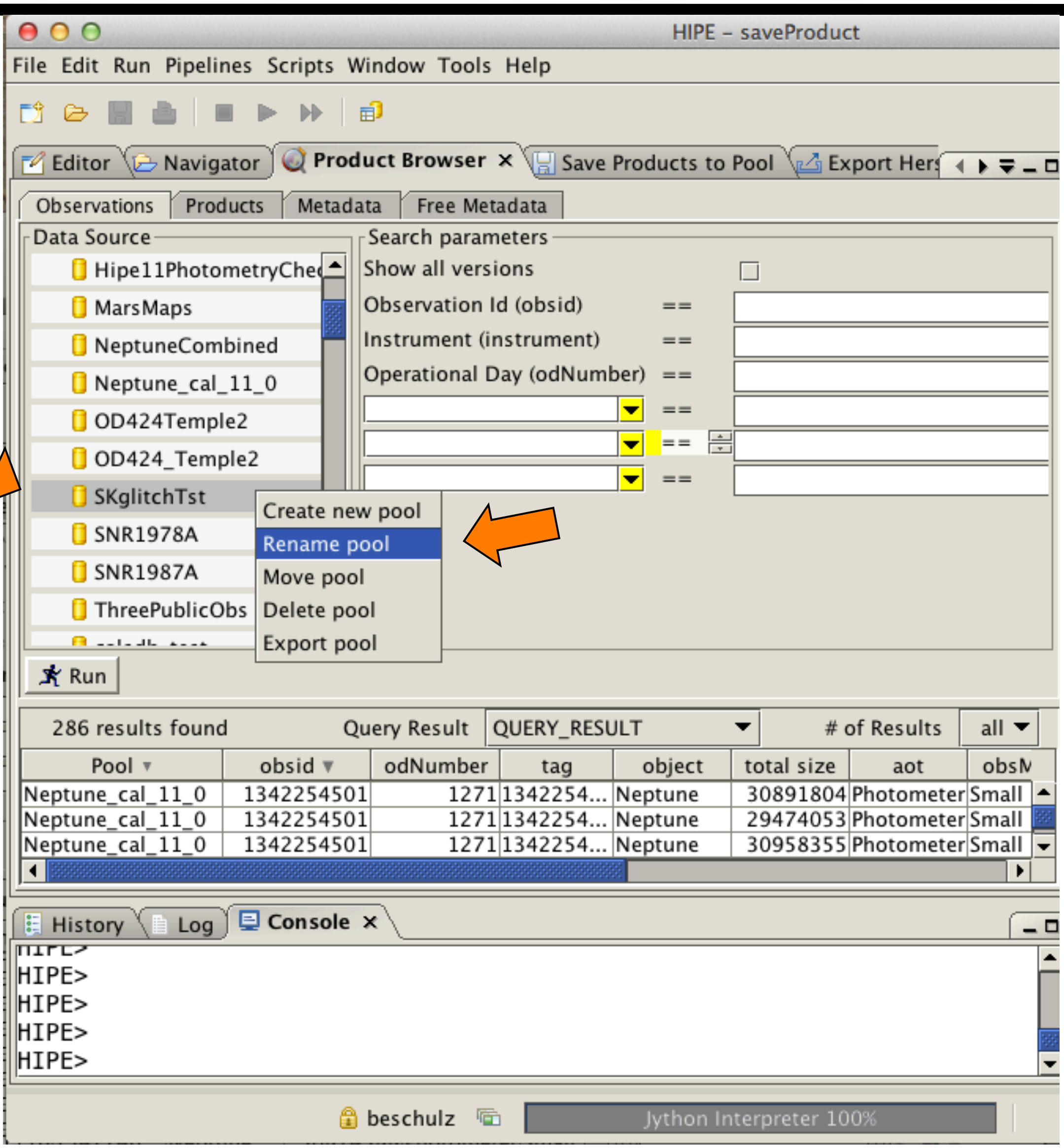
286 results found Query Result QUERY_RESULT # of Results all

Pool	obsid	odNumber	tag	object	total size	aot	obsM
Neptune_cal_11_0	1342254501	1271	1342254...	Neptune	30891804	Photometer	Small
Neptune_cal_11_0	1342254501	1271	1342254...	Neptune	29474053	Photometer	Small
Neptune_cal_11_0	1342254501	1271	1342254...	Neptune	30958355	Photometer	Small
parallelTestMaps	1342251957	1236	1342251...	Pipe_fillg...	545254467	Parallel M...	Paralle
parallelTestMaps	1342251956	1236	1342251...	Pipe_fillg...	538522324	Parallel M...	Paralle
Neptune_cal_11_0	1342246581	1116	1342246...	Neptune	30877008	Photometer	Small
Neptune_cal			...	Neptune	29466469	Photometer	Small
Neptune_cal			1342246...	Neptune	30874388	Photometer	Small

```
HIPE> # Added variable: QUERY_RESULT
HIPE> # Added variable: QUERY_RESULT
HIPE> # Added variable: selected
HIPE> # Added variable: selected
HIPE>
```

Managing Pools

- Select a pool and right-click.
- Pools can be created, renamed, moved to another storage directory, deleted, and exported.
- Moving will move the pool to another directory in your file system.
- Export will copy the contents of a pool into a zip file, so it can be sent to collaborators and other pundits.



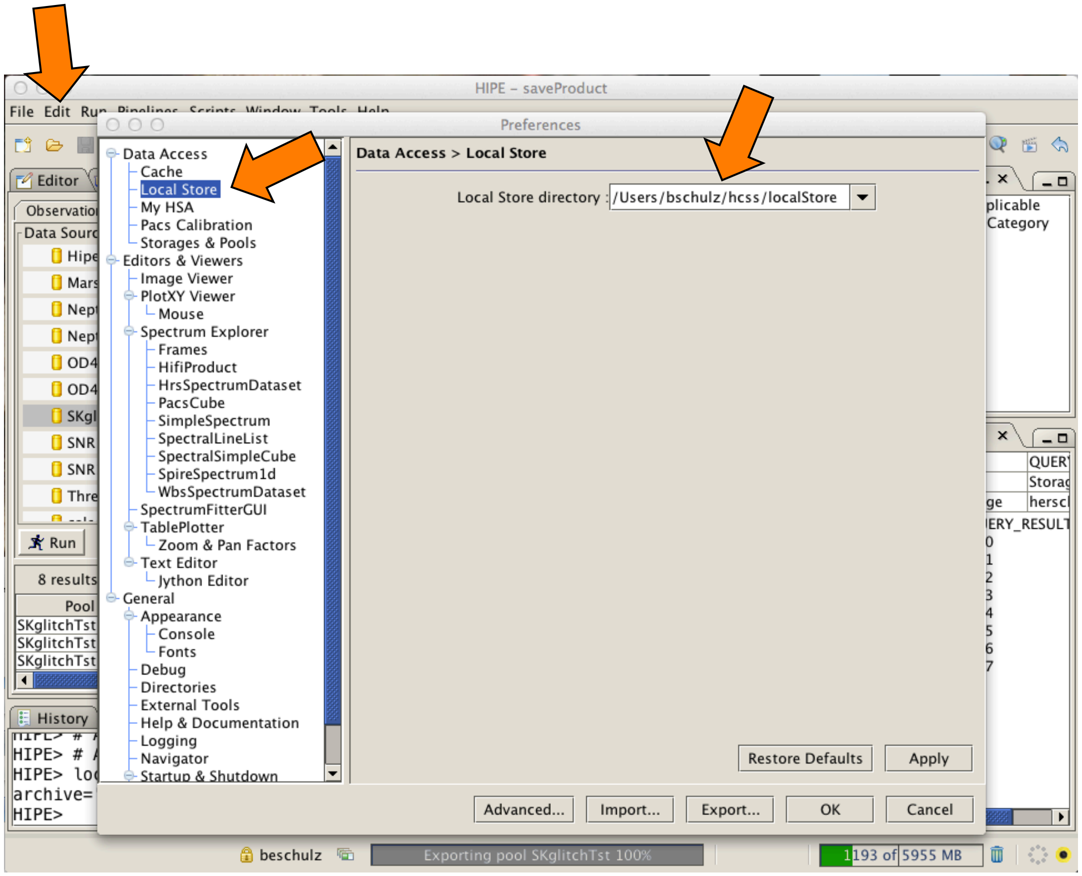
286 results found Query Result QUERY_RESULT # of Results all

Pool	obsid	odNumber	tag	object	total size	aot	obsM
Neptune_cal_11_0	1342254501	1271	1342254...	Neptune	30891804	Photometer	Small
Neptune_cal_11_0	1342254501	1271	1342254...	Neptune	29474053	Photometer	Small
Neptune_cal_11_0	1342254501	1271	1342254...	Neptune	30958355	Photometer	Small

```
HIPE>
HIPE>
HIPE>
HIPE>
HIPE>
```

A Place for Ordinary Pools

- The location of the directory where the pools are stored can be determined in the "Preferences" menu.
- Select "Edit"→"Preferences".
- Select there "Data Access"→"Local Store".
- It makes sense to configure this to be on a large disk with sufficient space.



Data Access > Local Store

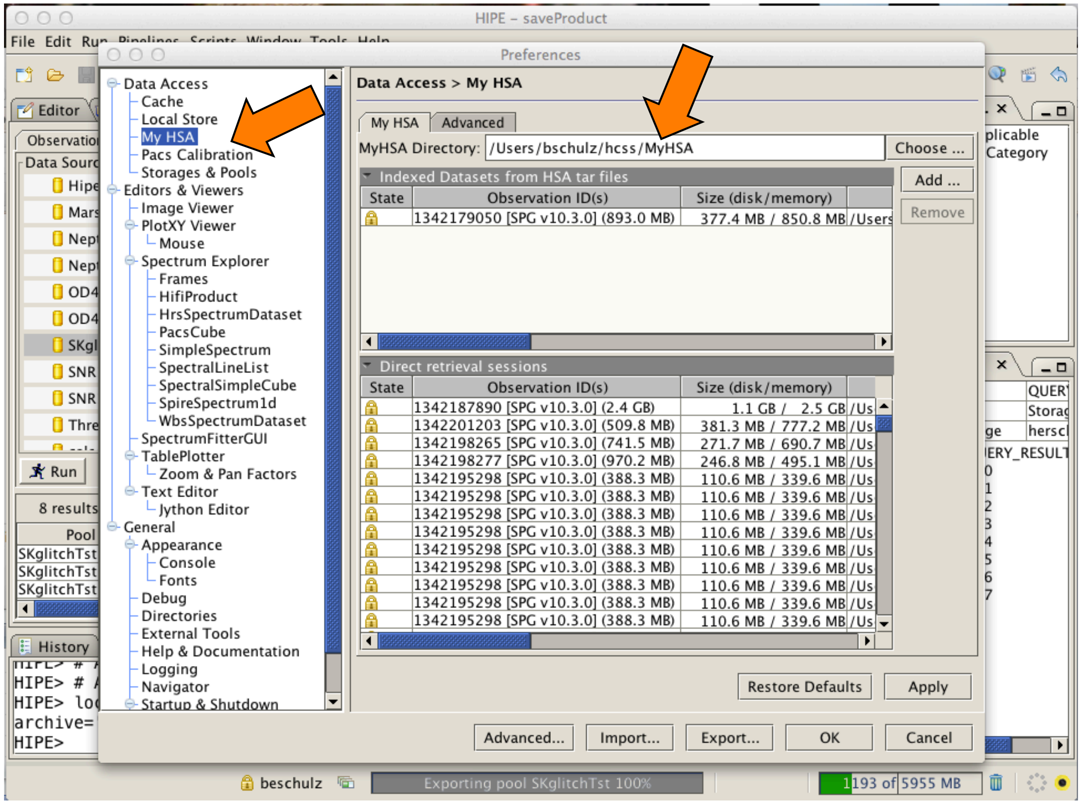
Local Store directory : /Users/bschulz/hcss/localStore

Exporting pool SKglitchTst 100%

1193 of 5955 MB

A Place for the MyHsa Pool

- The myHsa pool has a separate directory that can be configured also in "Preferences"
- Select "Edit"→"Preferences".
- Select there "Data Access"→"My HSA".
- It is good to choose a disk with sufficient space, when the "save" option in getObservation() is used or the save data on-demand option is activated in the "Advanced" tab.



Data Access > My HSA

MyHSA Directory: /Users/bschulz/hcss/MyHSA

Indexed Datasets from HSA tar files

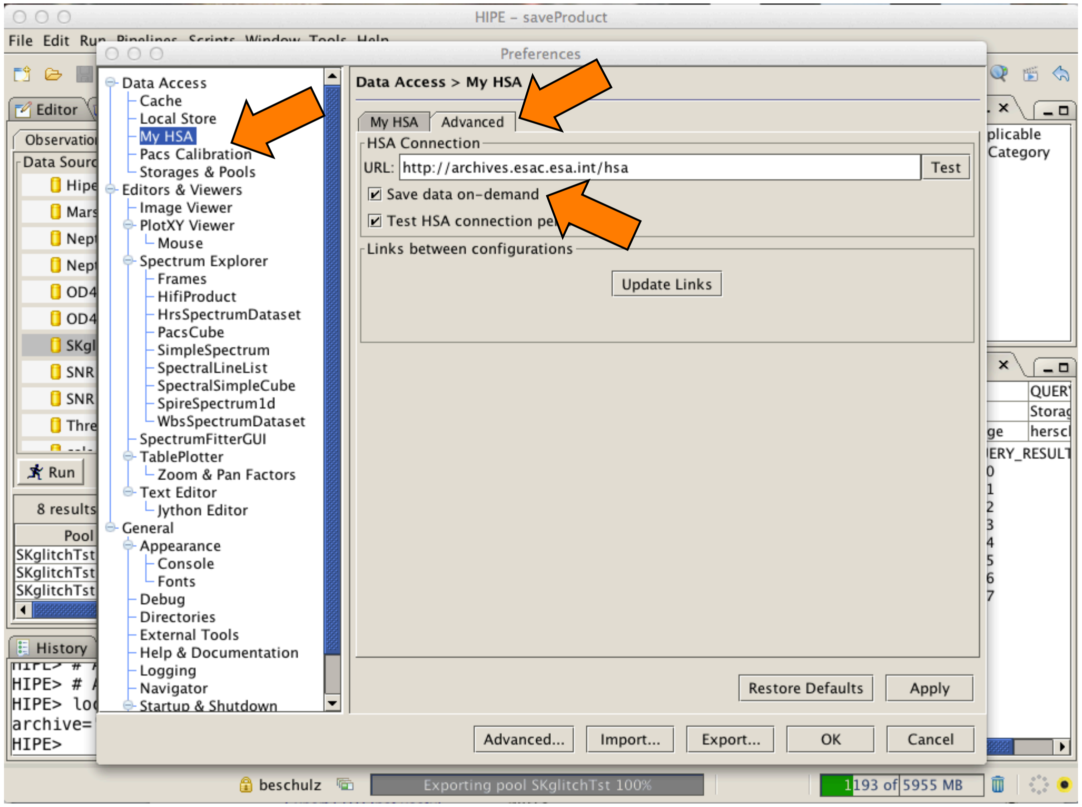
State	Observation ID(s)	Size (disk/memory)	
🔒	1342179050 [SPG v10.3.0] (893.0 MB)	377.4 MB / 850.8 MB	/Users

Direct retrieval sessions

State	Observation ID(s)	Size (disk/memory)	
🔒	1342187890 [SPG v10.3.0] (2.4 GB)	1.1 GB / 2.5 GB	/Us
🔒	1342201203 [SPG v10.3.0] (509.8 MB)	381.3 MB / 777.2 MB	/Us
🔒	1342198265 [SPG v10.3.0] (741.5 MB)	271.7 MB / 690.7 MB	/Us
🔒	1342198277 [SPG v10.3.0] (970.2 MB)	246.8 MB / 495.1 MB	/Us
🔒	1342195298 [SPG v10.3.0] (388.3 MB)	110.6 MB / 339.6 MB	/Us
🔒	1342195298 [SPG v10.3.0] (388.3 MB)	110.6 MB / 339.6 MB	/Us
🔒	1342195298 [SPG v10.3.0] (388.3 MB)	110.6 MB / 339.6 MB	/Us
🔒	1342195298 [SPG v10.3.0] (388.3 MB)	110.6 MB / 339.6 MB	/Us
🔒	1342195298 [SPG v10.3.0] (388.3 MB)	110.6 MB / 339.6 MB	/Us
🔒	1342195298 [SPG v10.3.0] (388.3 MB)	110.6 MB / 339.6 MB	/Us
🔒	1342195298 [SPG v10.3.0] (388.3 MB)	110.6 MB / 339.6 MB	/Us
🔒	1342195298 [SPG v10.3.0] (388.3 MB)	110.6 MB / 339.6 MB	/Us
🔒	1342195298 [SPG v10.3.0] (388.3 MB)	110.6 MB / 339.6 MB	/Us

On-Demand Save Option

- The save data on-demand option keeps copies of all products that are loaded through "lazy-loading" in the myHsa pool, making access to the same products faster next time.



Data Access > My HSA

HSA Connection

URL: http://archives.esac.esa.int/hsa

☑ Save data on-demand

☑ Test HSA connection pe

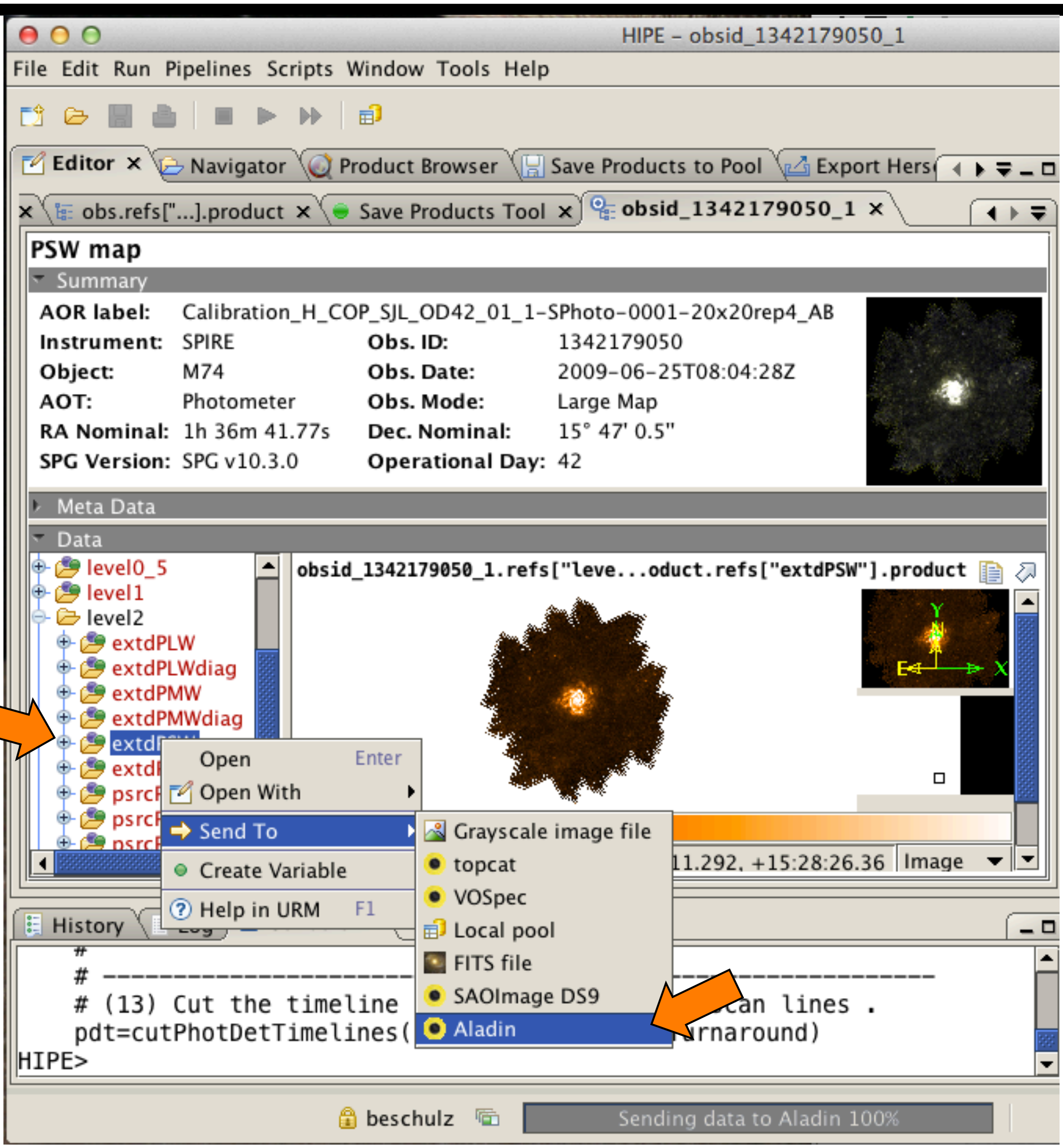
Links between configurations

Save data on-demand is off by default

Data Exchange via SAMP (Images)

- HIPE provides support for a powerful interface to share data between applications, the Simple Application Messaging Protocol (SAMP)
- The send-to menu provides options to send image data or tabular data to Topcat, VOSpec, DS9, or Aladin.
- Right click on a map and send it to Aladin.
- The Java application is downloaded and opens the image.
- Many more visualization and analysis methods become available.

See: http://www.ivoa.net/documents/SAMP/



HIPE – obsid_1342179050_1

PSW map

Summary

AOR label: Calibration_H_COP_SJL_OD42_01_1-SPhoto-0001-20x20rep4_AB
Instrument: SPIRE
Obs. ID: 1342179050
Object: M74
Obs. Date: 2009-06-25T08:04:28Z
AOT: Photometer
Obs. Mode: Large Map
RA Nominal: 1h 36m 41.77s
Dec. Nominal: 15° 47' 0.5"
SPG Version: SPG v10.3.0
Operational Day: 42

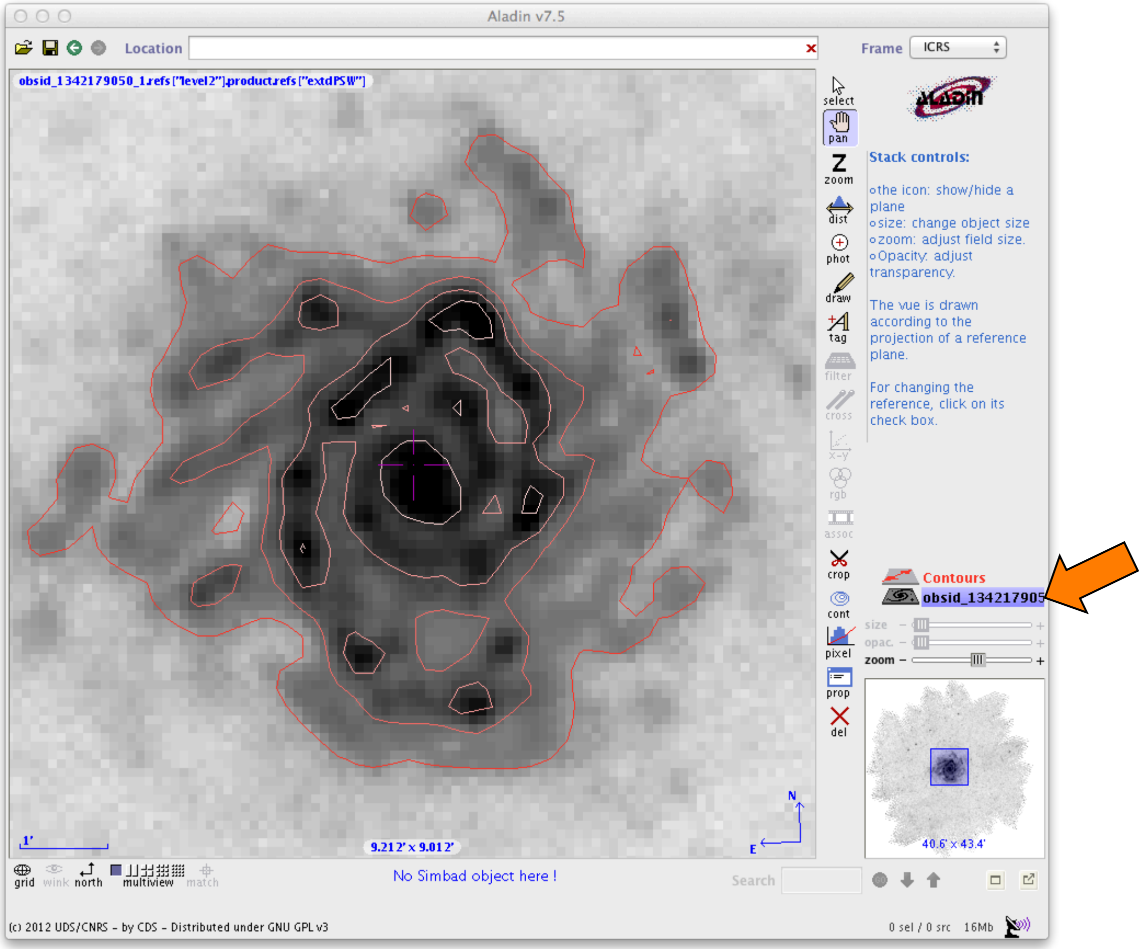
obsid_1342179050_1.refs["leve...oduct.refs["extdPSW"].product

```
#
# ----------------------------------------------------------------------
# (13) Cut the timeline                                    can lines .
pdt=cutPhotDetTimelines(                                   arnaround)
HIPE>
```

Sending data to Aladin 100%

Data Exchange via SAMP (Images)

- HIPE provides support for a powerful interface to share data between applications, the Simple Application Messaging Protocol (SAMP)
- The send-to menu provides options to send image data or tabular data to Topcat, VOSpec, DS9, or Aladin.
- Right click on a map and send it to Aladin.
- The Java application is downloaded and opens the image.
- Many more visualization and analysis methods become available.



obsid_1342179050_1.refs["level2"].product.refs["extdPSW"]

Stack controls:

○the icon: show/hide a plane
○size: change object size
○zoom: adjust field size.
○Opacity: adjust transparency.

The vue is drawn according to the projection of a reference plane.

For changing the reference, click on its check box.

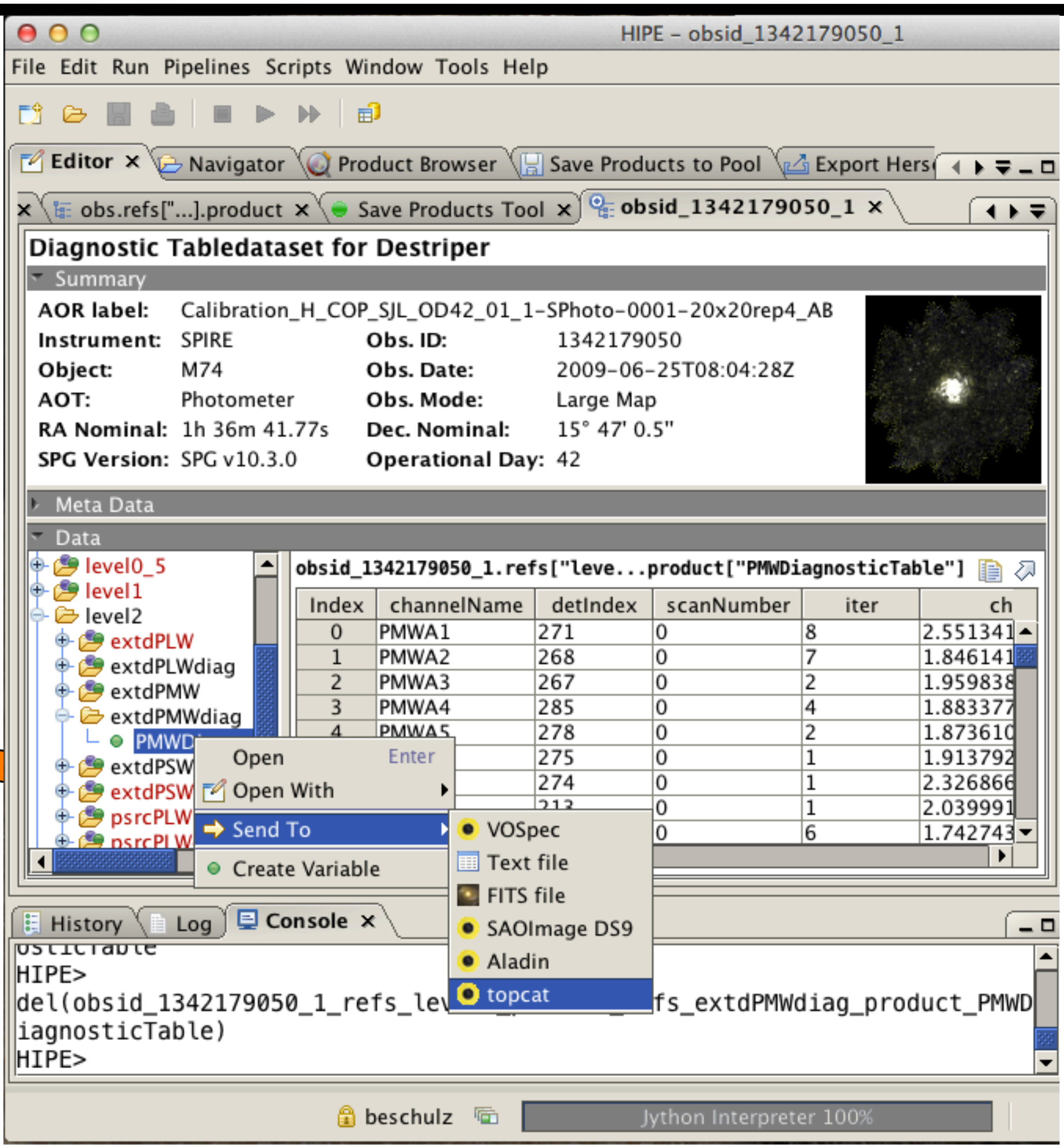
No Simbad object here !

(c) 2012 UDS/CNRS - by CDS - Distributed under GNU GPL v3

0 sel / 0 src 16Mb

Data Exchange via SAMP (Tables)

- Another example sending a table dataset.
- Start Topcat on your computer.
- Select a SPIRE diagnostic table and send it to Topcat.
- It appears as a table and can be analyzed using the Topcat features.
- It can even be modified on Topcat and sent back to HIPE using the “Transmit table to all applications using SAMP” button.



Diagnostic Tabledataset for Destriper

AOR label:	Calibration_H_COP_SJL_OD42_01_1-SPhoto-0001-20x20rep4_AB		
Instrument:	SPIRE	Obs. ID:	1342179050
Object:	M74	Obs. Date:	2009-06-25T08:04:28Z
AOT:	Photometer	Obs. Mode:	Large Map
RA Nominal:	1h 36m 41.77s	Dec. Nominal:	15° 47' 0.5"
SPG Version:	SPG v10.3.0	Operational Day:	42

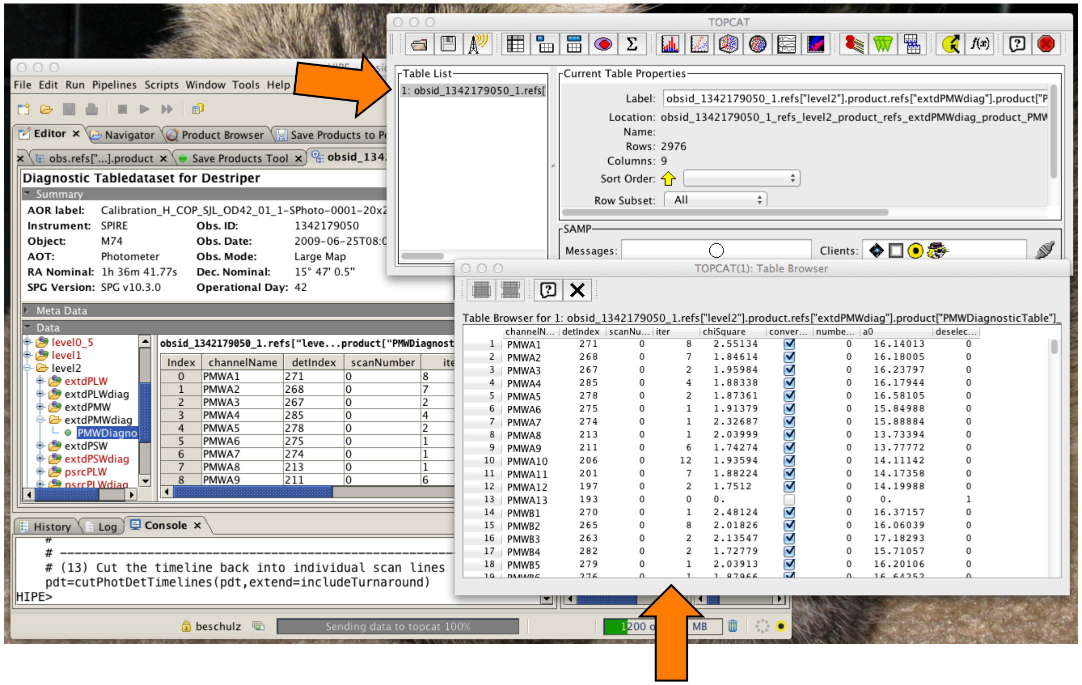
obsid_1342179050_1.refs["leve...product["PMWDiagnosticTable"]

Index	channelName	detIndex	scanNumber	iter	ch
0	PMWA1	271	0	8	2.551341
1	PMWA2	268	0	7	1.846141
2	PMWA3	267	0	2	1.959838
3	PMWA4	285	0	4	1.883377
4	PMWA5	278	0	2	1.873610
		275	0	1	1.913792
		274	0	1	2.326866
		213	0	1	2.039991
			0	6	1.742743

```
osticTable
HIPE>
del(obsid_1342179050_1_refs_lev                      fs_extdPMWdiag_product_PMWD
iagnosticTable)
HIPE>
```

Data Exchange via SAMP (Tables)

- Another example sending a table dataset.
- Start Topcat on your computer.
- Select a SPIRE diagnostic table and send it to Topcat.
- It appears as a table and can be analyzed using the Topcat features.
- It can even be modified on Topcat and sent back to HIPE using the “Transmit table to all applications using SAMP” button.



Diagnostic Tabledataset for Destriper

Summary

AOR label:	Calibration_H_COP_SJL_OD42_01_1-SPhoto-0001-20x2		
Instrument:	SPIRE	Obs. ID:	1342179050
Object:	M74	Obs. Date:	2009-06-25T08:0
AOT:	Photometer	Obs. Mode:	Large Map
RA Nominal:	1h 36m 41.77s	Dec. Nominal:	15° 47' 0.5"
SPG Version:	SPG v10.3.0	Operational Day:	42

Meta Data

Data

- level0_5
- level1
- level2
  - extdPLW
  - extdPLWdiag
  - extdPMW
  - extdPMWdiag
    - PMWDiagno
  - extdPSW
  - extdPSWdiag
  - psrcPLW
  - psrcPLWdiag

obsid_1342179050_1.refs["leve...product["PMWDiagnost

Index	channelName	detIndex	scanNumber	ite
0	PMWA1	271	0	8
1	PMWA2	268	0	7
2	PMWA3	267	0	2
3	PMWA4	285	0	4
4	PMWA5	278	0	2
5	PMWA6	275	0	1
6	PMWA7	274	0	1
7	PMWA8	213	0	1
8	PMWA9	211	0	6

```
    #
    # ----------------------------------------------------------------
    # (13) Cut the timeline back into individual scan lines
    pdt=cutPhotDetTimelines(pdt,extend=includeTurnaround)
HIPE>
```

Sending data to topcat 100%

Label: obsid_1342179050_1.refs["level2"].product.refs["extdPMWdiag"].product["P
Location: obsid_1342179050_1_refs_level2_product_refs_extdPMWdiag_product_PMW
Name:
Rows: 2976
Columns: 9
Sort Order:
Row Subset: All

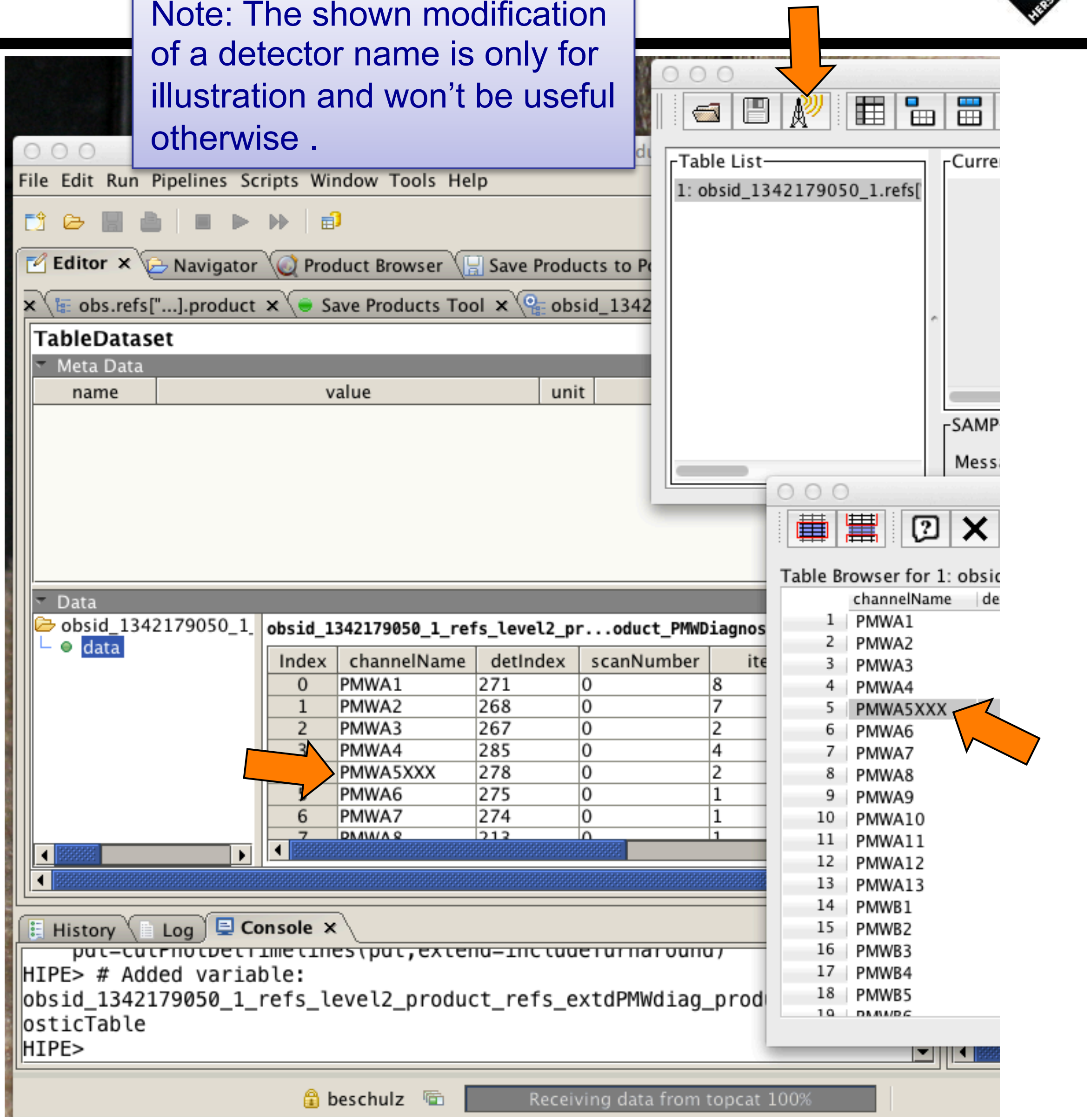
Messages:
Clients:

Table Browser for 1: obsid_1342179050_1.refs["level2"].product.refs["extdPMWdiag"].product["PMWDiagnosticTable"]_

	channelN...	detIndex	scanNu...	iter	chiSquare	conver...	numbe...	a0	deselec...
1	PMWA1	271	0	8	2.55134	☑	0	16.14013	0
2	PMWA2	268	0	7	1.84614	☑	0	16.18005	0
3	PMWA3	267	0	2	1.95984	☑	0	16.23797	0
4	PMWA4	285	0	4	1.88338	☑	0	16.17944	0
5	PMWA5	278	0	2	1.87361	☑	0	16.58105	0
6	PMWA6	275	0	1	1.91379	☑	0	15.84988	0
7	PMWA7	274	0	1	2.32687	☑	0	15.88884	0
8	PMWA8	213	0	1	2.03999	☑	0	13.73394	0
9	PMWA9	211	0	6	1.74274	☑	0	13.77772	0
10	PMWA10	206	0	12	1.93594	☑	0	14.11142	0
11	PMWA11	201	0	7	1.88224	☑	0	14.17358	0
12	PMWA12	197	0	2	1.7512	☑	0	14.19988	0
13	PMWA13	193	0	0	0.	□	0	0.	1
14	PMWB1	270	0	1	2.48124	☑	0	16.37157	0
15	PMWB2	265	0	8	2.01826	☑	0	16.06039	0
16	PMWB3	263	0	2	2.13547	☑	0	17.18293	0
17	PMWB4	282	0	2	1.72779	☑	0	15.71057	0
18	PMWB5	279	0	1	2.03913	☑	0	16.20106	0
19	PMWB6	276	0	1	1.87966	☑	0	16.64252	0

Data Exchange via SAMP (Tables)

- Another example sending a table dataset.
- Start Topcat on your computer.
- Select a SPIRE diagnostic table and send it to Topcat.
- It appears as a table and can be analyzed using the Topcat features.
- It can even be modified on Topcat and sent back to HIPE using the "Transmit table to all applications using SAMP" button.



Note: The shown modification of a detector name is only for illustration and won't be useful otherwise .

obsid_1342179050_1_refs_level2_pr...oduct_PMWDiagnos

Index	channelName	detIndex	scanNumber	ite
0	PMWA1	271	0	8
1	PMWA2	268	0	7
2	PMWA3	267	0	2
3	PMWA4	285	0	4
4	PMWA5XXX	278	0	2
5	PMWA6	275	0	1
6	PMWA7	274	0	1
7	PMWA8	213	0	1

```
    put=cutPhotDetTimelines(put,extend=IncludeTurnaround)
HIPE> # Added variable:
obsid_1342179050_1_refs_level2_product_refs_extdPMWdiag_prod
osticTable
HIPE>
```

	channelName	de
1	PMWA1	
2	PMWA2	
3	PMWA3	
4	PMWA4	
5	PMWA5XXX	
6	PMWA6	
7	PMWA7	
8	PMWA8	
9	PMWA9	
10	PMWA10	
11	PMWA11	
12	PMWA12	
13	PMWA13	
14	PMWB1	
15	PMWB2	
16	PMWB3	
17	PMWB4	
18	PMWB5	
19	PMWB6	

Tasks

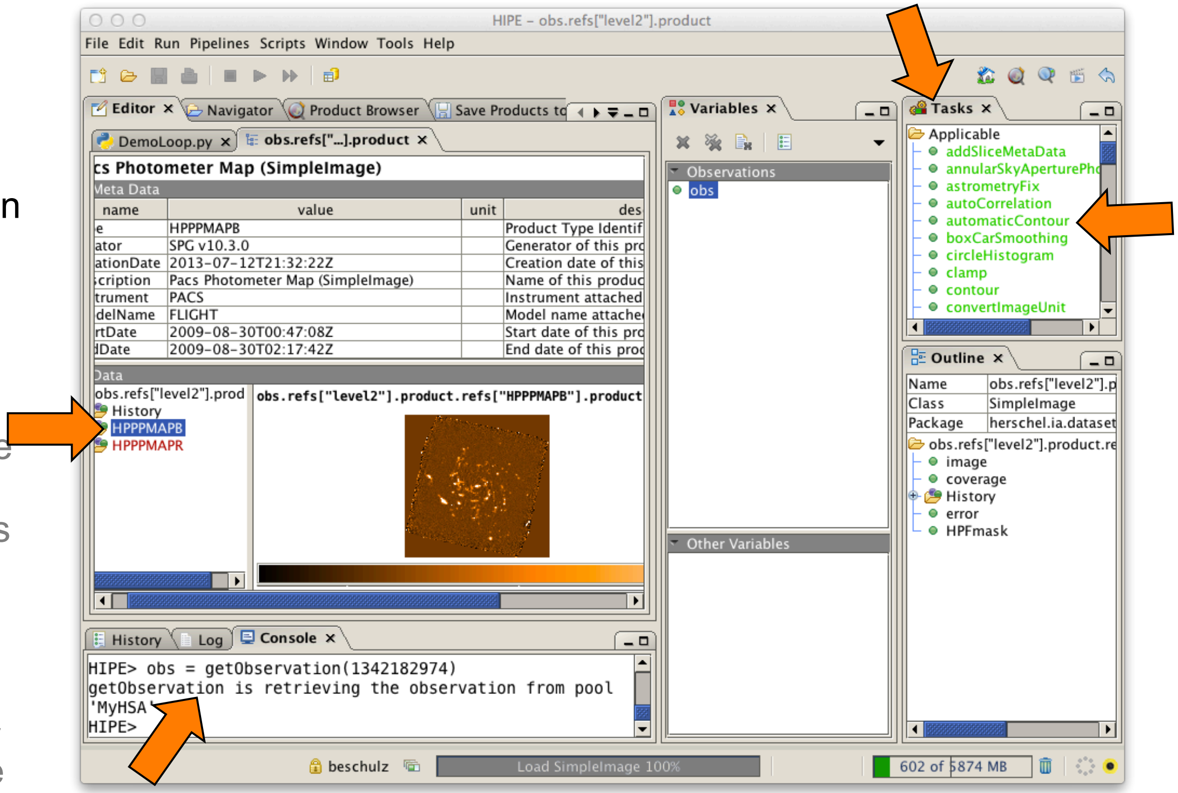
- Tasks are special classes in HIPE conforming to a rigorous specification of input and output parameters.

- They are registered with the system and appear in the Task View.
- Calling them from the Task View will produce a default GUI regardless of whether GUI features were specified in the code.
- They can be made applicable to specific product types and will appear in the "Applicable" folder only for these.
- There is a large collection of pre-defined tasks available in HIPE.

- Users can write their own tasks in Jython if needed and distribute them as a HIPE Plug-In.
- SPIA and CASSIS are examples for HIPE Plug-Ins.

Task Example

- Tasks appear in the Tasks View on the upper right in this perspective.
- Load a PACS observation, click “Level 2’ in the Outline View and select HPPBMAPB in the appearing viewer.
- Double-click the “Applicable” folder in the Task View.
- Find and double-click the task “automaticContour”.
- The task GUI will appear in the Editor View.
- Enter some appropriate values and hit the “Accept” button.
- This will run the task.
- Note the script output in the Console View.
- Go back to the Level 2 Viewer and display the image with the Standard Image Viewer (right click).
- Drag the variable “contours” over the image.



cs Photometer Map (SimpleImage)

Meta Data

name	value	unit	des
e	HPPPMAPB		Product Type Identif
ator	SPG v10.3.0		Generator of this pro
ationDate	2013-07-12T21:32:22Z		Creation date of this
cription	Pacs Photometer Map (SimpleImage)		Name of this produc
trument	PACS		Instrument attached
delName	FLIGHT		Model name attache
rtDate	2009-08-30T00:47:08Z		Start date of this pro
dDate	2009-08-30T02:17:42Z		End date of this proc

Data

obs.refs["level2"].prod
- History
- HPPPMAPB
- HPPPMAPR

obs.refs["level2"].product.refs["HPPPMAPB"].product

Applicable
- addSliceMetaData
- annularSkyAperturePho
- astrometryFix
- autoCorrelation
- automaticContour
- boxCarSmoothing
- circleHistogram
- clamp
- contour
- convertImageUnit

Name	obs.refs["level2"].p
Class	SimpleImage
Package	herschel.ia.dataset

obs.refs["level2"].product.re
- image
- coverage
- History
- error
- HPFmask

```
HIPE> obs = getObservation(1342182974)
getObservation is retrieving the observation from pool
'MyHSA'
HIPE>
```

Task Example

- Tasks appear in the Tasks View on the upper right in this perspective.
- Load a PACS observation, click “Level 2’ in the Outline View and select HPPBMAPB in the appearing viewer.
- Double-click the “Applicable” folder in the Task View.
- Find and double-click the task “automaticContour”.
- The task GUI will appear in the Editor View.
- Enter some appropriate values and hit the “Accept” button.
- This will run the task.
- Note the script output in the Console View.
- Go back to the Level 2 Viewer and display the image with the Standard Image Viewer (right click).
- Drag the variable “contours” over the image.

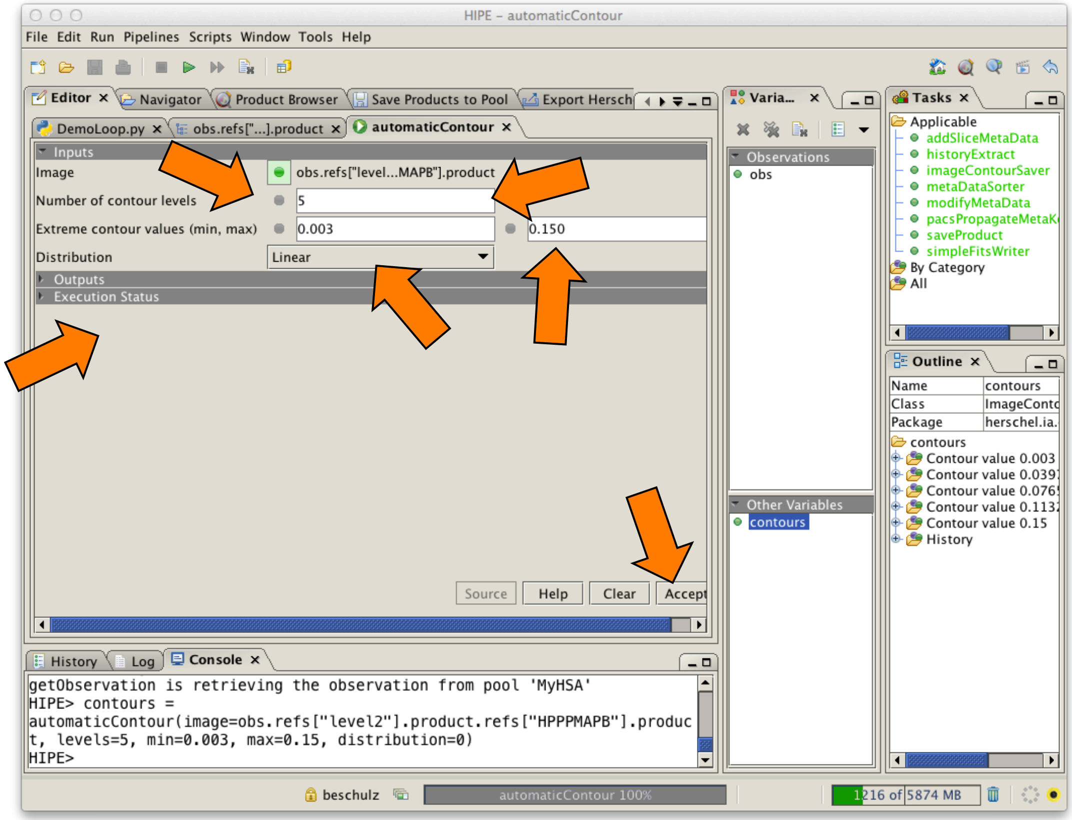


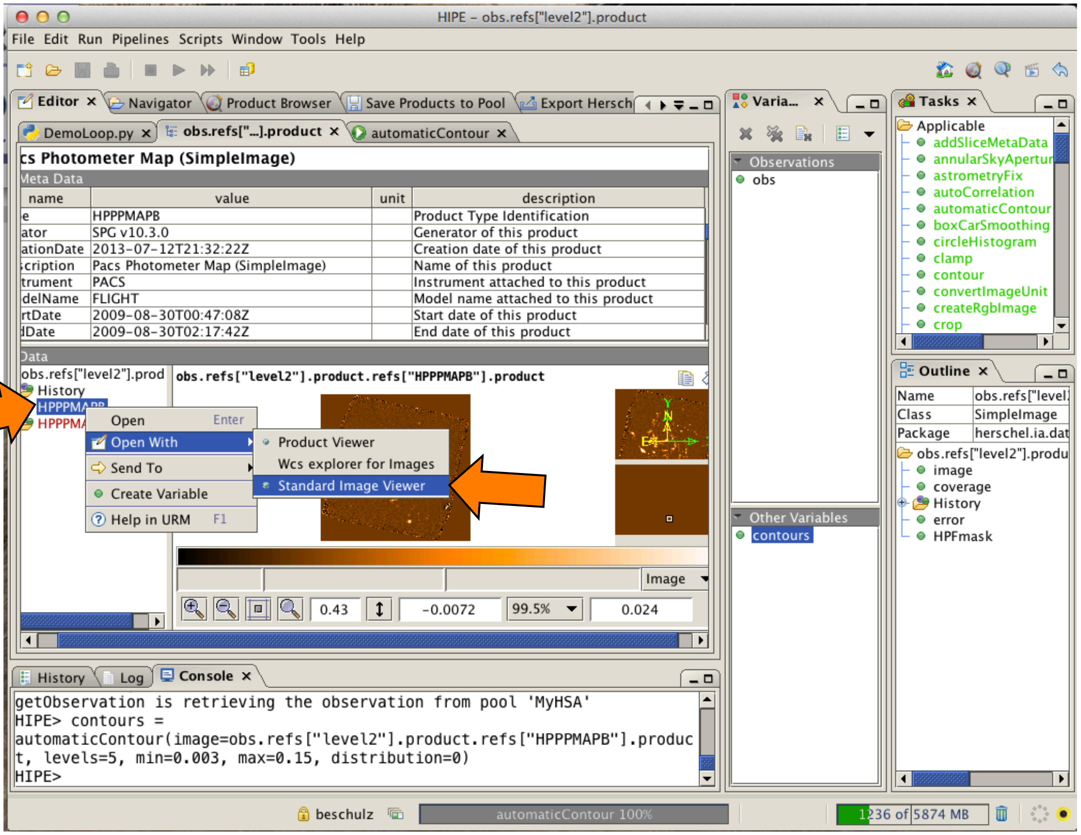
Image: obs.refs["level...MAPB"].product
Number of contour levels: 5
Extreme contour values (min, max): 0.003 0.150
Distribution: Linear

Name	contours
Class	ImageConto
Package	herschel.ia.

```
getObservation is retrieving the observation from pool 'MyHSA'
HIPE> contours =
automaticContour(image=obs.refs["level2"].product.refs["HPPPMAPB"].produc
t, levels=5, min=0.003, max=0.15, distribution=0)
HIPE>
```

Task Example

- Tasks appear in the Tasks View on the upper right in this perspective.
- Load a PACS observation, click "Level 2' in the Outline View and select HPPBMAPB in the appearing viewer.
- Double-click the "Applicable" folder in the Task View.
- Find and double-click the task "automaticContour".
- The task GUI will appear in the Editor View.
- Enter some appropriate values and hit the "Accept" button.
- This will run the task.
- Note the script output in the Console View.
- Go back to the Level 2 Viewer and display the image with the Standard Image Viewer (right click).
- Drag the variable "contours" over the image.



cs Photometer Map (SimpleImage)

Meta Data

name	value	unit	description
e	HPPPMAPB		Product Type Identification
ator	SPG v10.3.0		Generator of this product
ationDate	2013-07-12T21:32:22Z		Creation date of this product
cription	Pacs Photometer Map (SimpleImage)		Name of this product
trument	PACS		Instrument attached to this product
delName	FLIGHT		Model name attached to this product
rtDate	2009-08-30T00:47:08Z		Start date of this product
dDate	2009-08-30T02:17:42Z		End date of this product

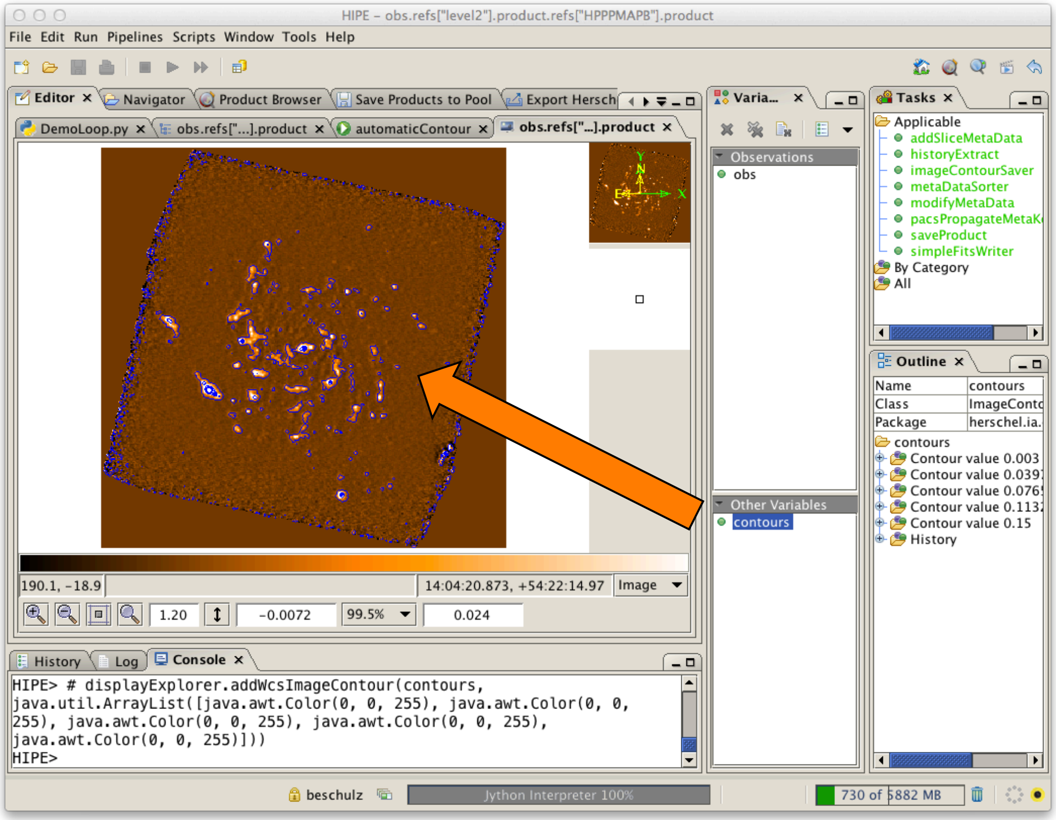
obs.refs["level2"].product.refs["HPPPMAPB"].product

Name	obs.refs["level2
Class	SimpleImage
Package	herschel.ia.dat

```
getObservation is retrieving the observation from pool 'MyHSA'
HIPE> contours =
automaticContour(image=obs.refs["level2"].product.refs["HPPPMAPB"].produc
t, levels=5, min=0.003, max=0.15, distribution=0)
HIPE>
```

Task Example

- Tasks appear in the Tasks View on the upper right in this perspective.
- Load a PACS observation, click "Level 2' in the Outline View and select HPPBMAPB in the appearing viewer.
- Double-click the "Applicable" folder in the Task View.
- Find and double-click the task "automaticContour".
- The task GUI will appear in the Editor View.
- Enter some appropriate values and hit the "Accept" button.
- This will run the task.
- Note the script output in the Console View.
- Go back to the Level 2 Viewer and display the image with the Standard Image Viewer (right click).
- Drag the variable "contours" over the image.



190.1, -18.9 | 14:04:20.873, +54:22:14.97 | Image

1.20 | -0.0072 | 99.5% | 0.024

Name	contours
Class	ImageConto
Package	herschel.ia.

contours
- Contour value 0.003
- Contour value 0.039
- Contour value 0.076
- Contour value 0.113
- Contour value 0.15
- History

```
HIPE> # displayExplorer.addWcsImageContour(contours,
java.util.ArrayList([java.awt.Color(0, 0, 255), java.awt.Color(0, 0,
255), java.awt.Color(0, 0, 255), java.awt.Color(0, 0, 255),
java.awt.Color(0, 0, 255)]))
HIPE>
```

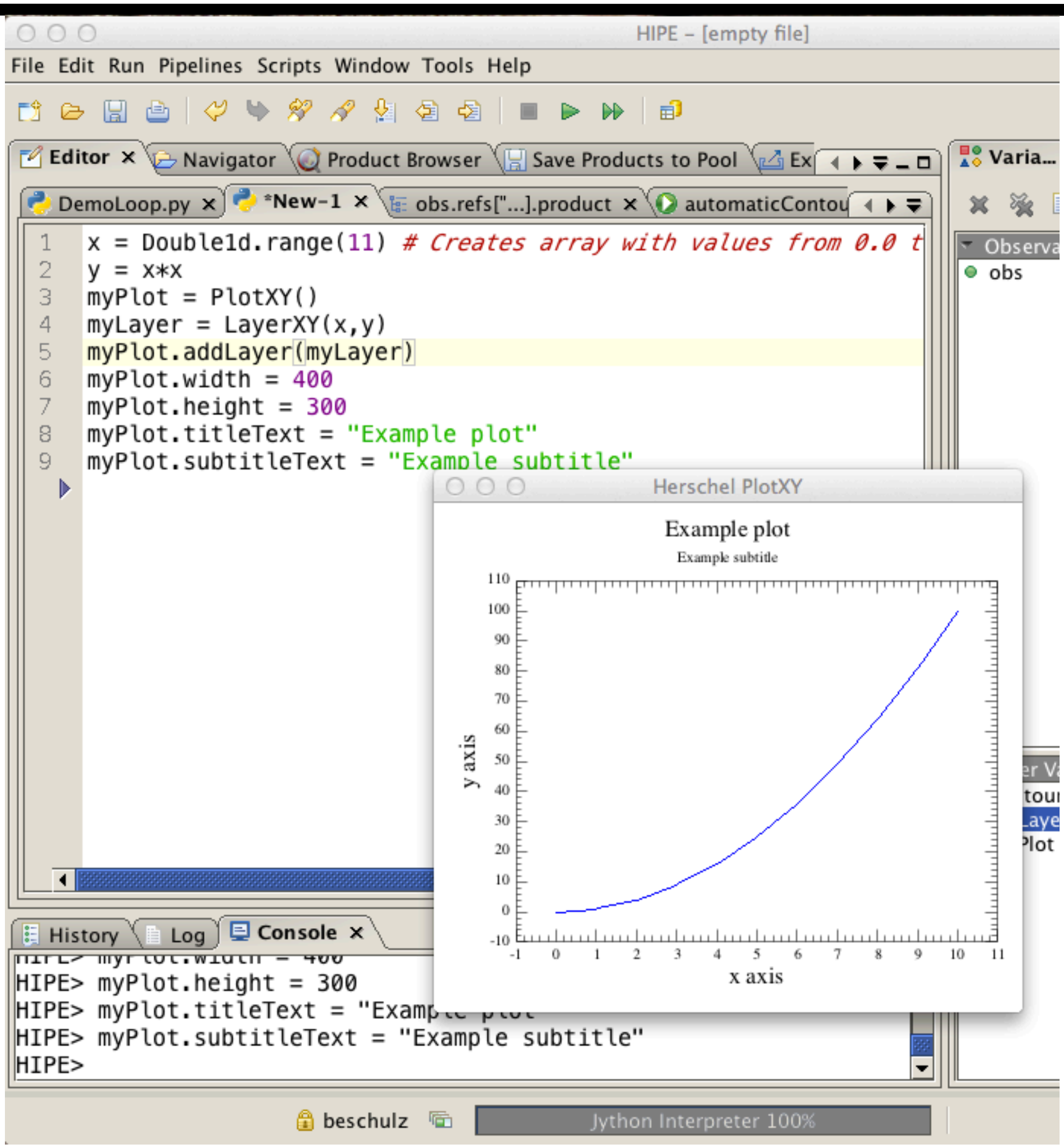
Many more tasks are available, in particular for map arithmetic, photometry, line fitting etc.

Libraries/Packages

- Hipe contains a number of software Libraries that are available to scripting.
- Examples are:
  - Numerics Library
  - Plot Library
  - Image toolbox
  - Spectral toolbox
  - Product Access Layer
  - etc…
- A good overview over available functionality is in the "Categorized view of commands" in the "HCSS User's Reference Manual"

Plot Library

- HIPE contains a powerful plot library with many ways to create high quality publication ready diagrams.
- A simple example code is shown to the right and the resulting plot.
- For more information refer to the documentation in the Data Analysis Guide.
- Several scripted examples for publication-ready plots are available.

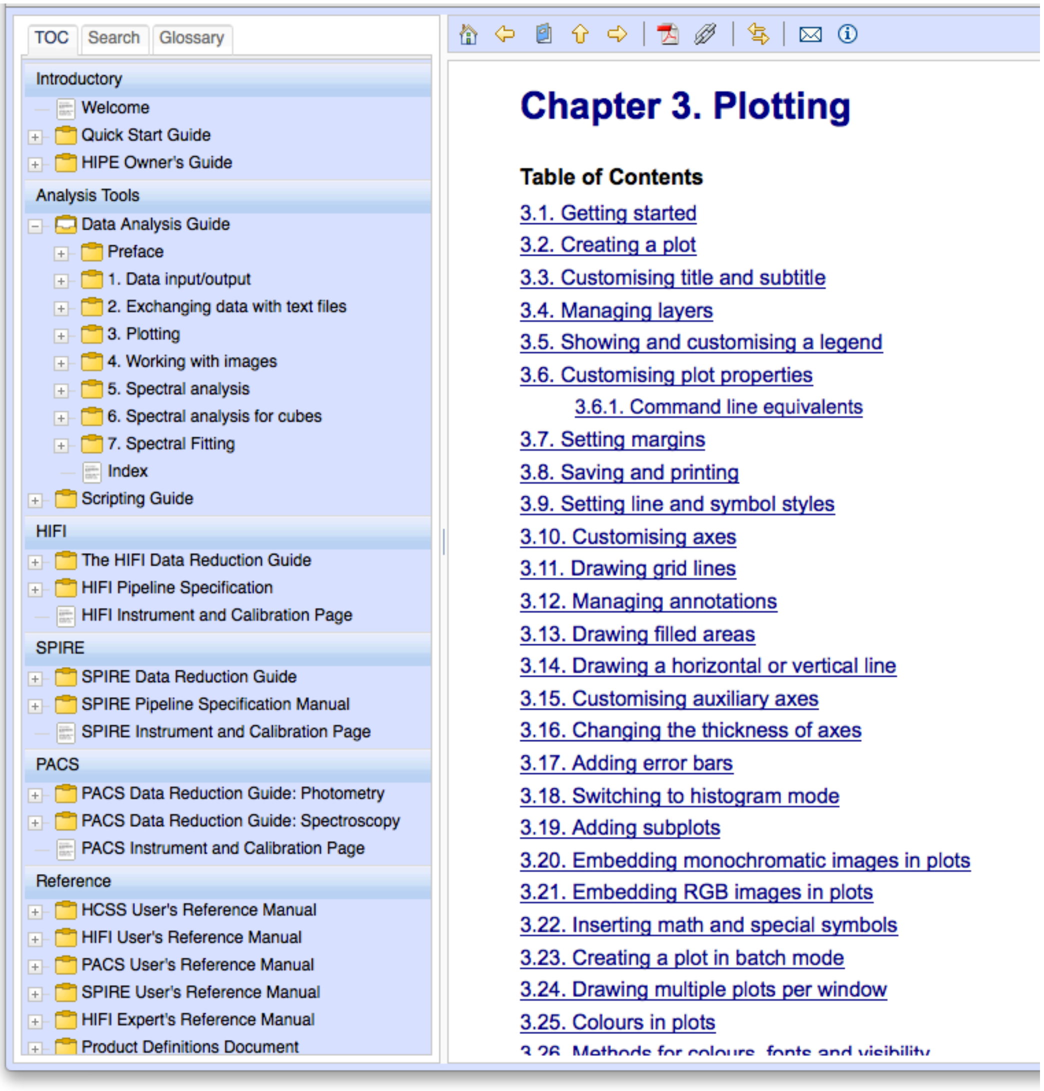


```
x = Double1d.range(11) # Creates array with values from 0.0 t
y = x*x
myPlot = PlotXY()
myLayer = LayerXY(x,y)
myPlot.addLayer(myLayer)
myPlot.width = 400
myPlot.height = 300
myPlot.titleText = "Example plot"
myPlot.subtitleText = "Example subtitle"
```

```
HIPE> myPlot.width = 400
HIPE> myPlot.height = 300
HIPE> myPlot.titleText = "Examp
HIPE> myPlot.subtitleText = "Example subtitle"
HIPE>
```

Plot Package

- HIPE contains a powerful plot library with many ways to create high quality publication ready diagrams.
- A simple example code is shown to the right and the resulting plot.
- For more information refer to the documentation in the Data Analysis Guide.
- Several scripted examples for publication-ready plots are available.



Introductory
- Welcome
- Quick Start Guide
- HIPE Owner's Guide

Analysis Tools
- Data Analysis Guide
  - Preface
  - 1. Data input/output
  - 2. Exchanging data with text files
  - 3. Plotting
  - 4. Working with images
  - 5. Spectral analysis
  - 6. Spectral analysis for cubes
  - 7. Spectral Fitting
  - Index
- Scripting Guide

HIFI
- The HIFI Data Reduction Guide
- HIFI Pipeline Specification
- HIFI Instrument and Calibration Page

SPIRE
- SPIRE Data Reduction Guide
- SPIRE Pipeline Specification Manual
- SPIRE Instrument and Calibration Page

PACS
- PACS Data Reduction Guide: Photometry
- PACS Data Reduction Guide: Spectroscopy
- PACS Instrument and Calibration Page

Reference
- HCSS User's Reference Manual
- HIFI User's Reference Manual
- PACS User's Reference Manual
- SPIRE User's Reference Manual
- HIFI Expert's Reference Manual
- Product Definitions Document

Chapter 3. Plotting

Table of Contents

Some Practical Points at the End

- This was only the tip of the Iceberg. HIPE is a powerful data processing and analysis system with many more aspects to explore.

- Configuring HIPE is important at the start.
- Set the places for the local pools (localStore) and for the HSA cache pool (myHsa) to a disk with enough space.
- Select to compress pools with the line: `hcss.ia.pal.pool.lstore.compress=true` in user.props
- The amount of RAM to reserve for HIPE should be adjusted to the RAM available and should leave some room for the operating system.
- See the Preferences menu for other adjustments, like the appearance of the console. In many cases you will have to re-start the session, so configure first.
- Instruments will have additional configuration requirements.

- There is more documentation available than you might wish for but we are here to give you the necessary guidance.