Frequently Asked Questions

Herschel Data Processing

Note

For the most up-to-date information, please check the known issues page.

1. General information

1.1. What is the relationship between the HCSS, HIPE, JIDE and DP?

The HCSS stands for the *Herschel Common Science System*. It consists of software for the Herschel Science Ground Segment (Commanding, Proposal handling, Mission Planning, etc), in addition to the Data Processing (DP) system. Thus, the Herschel DP is *part* of the HCSS. DP itself deals with the software systems of Standard Product Generation (SPG), Quality Control Pipeline (QCP), Quick Look Analysis (QLA) and Interactive Analysis (IA).

HIPE stands for *Herschel Interactive Processing Environment* and is the main gateway to the DP system. It is a graphical application which includes Jython scripting, data analysis, plotting, communication with the Herschel Science Archive and much more.

JIDE is a legacy graphical interface that was developed before HIPE. It is still present as standalone program and as part of HIPE.

1.2. How do I get hold of HIPE?

HIPE is available for free download at this page.

Development builds (*untested and potentially unstable!*) are available via the Continuous Integration system. These builds are meant only for Herschel software developers. If you are not a software developer, but you still want to get access to these developer's builds for any particular reason, please contact our Helpdesk system at this page.

If you are an **internal user** of the Herschel project, you can also download the software from these locations:

- More installers at this page.
- Installers for release candidates at this page.

1.3. How do I start HIPE? Where is the executable?

The installer should have told you the location of the `hipe` executable file. In case you forgot, it is in the `bin` subfolder of the HIPE installation folder.

1.4. What are the two black windows that appear when I start HIPE?

When you start HIPE under Windows, you might see two command prompt windows called `hipe` and `ia_hipe.exe` appear alongside the HIPE main window. Both can be safely closed once HIPE has started.

1.5. I get startup messages about missing HIFI/PACS/SPIRE modules. Is that serious?

You are probably referring to messages such as the following (for HIFI; messages for PACS and SPIRE are the same):

```
Import -"from herschel.hifi.all import *" returns:
```

```
Import Error --> No module named hifi
```

Such messages are harmless and just reflect the fact that you have not installed the software related to the specified instruments.

2. Installation

Warning

IMPORTANT: *before* installing HIPE, please make sure that your default file compression program can unzip FITS files without corrupting them. Otherwise calibration files shipped with HIPE may be corrupted during installation. For more information see the *Data Analysis Guide*: Section 1.4.6.2.

2.1. The installer for Mac complains about write permissions. Why?

You need write permissions for the directory where you want to install HIPE. For example, you need Administrator privileges to install in `/Applications`.

If you have installed HIPE successfully in the past, and are now experiencing write permission problems, it may be that some other application changed the permissions in the meantime (there have been cases of scientific applications such as *SciSoft* doing this).

Once you revert write permissions to their correct values, HIPE installation should succeed.

2.2. Why does the uninstaller leave some icons and files/folders behind?

This incomplete uninstallation is a known issue and may be caused by a corrupted *InstallAnywhere* global registry. As a workaround, try renaming the *InstallAnywhere* registry file. See point 6 in the InstallAnywhere issues page for detailed information.

If you add files (such as Jython scripts) into HIPE folders after installation, these will not be removed by the uninstaller.

2.3. The installer for Windows does not start and issues the error message *Not a valid Win32 application*. Why?

This problem was encountered under Windows Vista because the installer had not been downloaded correctly, resulting in an empty file. The exact cause is not yet known. As a workaround, trying downloading the installer in a different way (for instance, via command line FTP rather than a web browser) or from another computer.

3. Help and documentation

3.1. Is there a way to open the HIPE Help System without starting HIPE?

Yes, via the **show_help** command, which resides in the `apps` folder in your HIPE installation. The **show_help** command will start a standalone Help System in a window called *Yet Another JAva Help System*. Press the Display button to open the Help initial page in your default browser.

3.2. Why is the HIPE Help System suddenly unable to connect to any help page?

The Help System relies on a local web server to display help pages. If you close the application that started the web server, the server will shut down as well:

- If you started the Help System from HIPE, quitting HIPE will stop the Help System.
- If you started the Help System with the **show_help** command, closing the *Yet Another JAva Help System* window will stop the Help System.

Note that when the Help System stops, any browser window or tab showing help pages *will remain open*, but links on them will not work anymore.

3.3. Why are several important packages missing from the Javadoc?

You are probably looking at the *static* Javadoc located in the `doc/api/` folder of your HIPE installation. The complete Javadoc can now be accessed from the HIPE Help System by clicking on *HCSS Developer's Reference Manual (API)* in the *Developer Reference* section.

The plan is to remove the static version of the Javadoc completely.

3.4. Is it possible to have the traditional, frame-based Javadoc layout?

Javadoc pages appear within the HIPE Help System frame structure. If you click on the *FRAME* link on any Javadoc page, you will obtain the traditional Javadoc layout. To go back to the HIPE Help System layout you will have to use the Back button of your browser (clicking on the *NO FRAMES* link will not work). To avoid losing the HIPE Help System layout, you may want to right-click on the *FRAMES* link to open the traditional Javadoc layout into a new tab of your web browser.

3.5. How can I provide feedback/requests regarding DP documentation?

If you are an external user, please raise a ticket with the Helpdesk of the Herschel Science Centre. If you are an internal user with access to JIRA (the HCSS tracking system) please raise SPR/SCRs on the relevant manual:

- `ia_manuals_readmefirst` for the *Read Me First* document.
- `ia_manuals_quickstart` for the *Quick Start Guide*.
- `ia_manuals_hipeowner` for the *HIPE Owner's Guide*.
- `ia_manuals_howtos` for the *Data Analysis Guide*, formerly known as *HowTo Documents*.
- `ia_manuals_um` for the *Scripting and Data Mining* guide, formerly known as *Advanced User's Manual*.
- `ia_manuals_urm` for the *User's Reference Manuals*.
- `ia_manuals_faq` for the *Frequently Asked Questions* (this document).
- `ia_manuals_wnew` for the *What's New document*.
- `ia_manuals_shared` for issues not related to a single manual.

3.6. I try to open the HIPE Help System with Safari, but all I see is a blank window. Why?

This is a known bug in Safari 4.0, not limited to the HIPE Help System. If the problem persists, try opening the help page with another web browser: just copy and paste the address (something like *http://127.0.0.1:8082/index.jsp?mark=null*) to the address bar of the other browser.

4. Jython scripting

4.1. What is Jython?

Jython is the scripting language used for Interactive Analysis within the DP environment. It is an implementation of the high-level, dynamic, object-oriented language Python, which is already heavily used for scientific purposes (see for example http://www.scipy.org).

Jython is written entirely in Java, and seamlessly integrated with the Java platform. It thus allows you to combine the power of both Java and Python. More information about Jython can be found at the Jython website.

4.2. I am familiar with IDL. Will Jython be hard to learn?

Not at all. The *HIPE Owner's Guide* includes some handy tables with the Jython and HIPE equivalents of the most common IDL commands: Section 4.

4.3. DP Quirks: when the unexpected happens

The DP environment (as it happens in all computer languages) has a number of quirks that can be baffling for users coming from other environments. A section in the *Scripting and Data Mining* guide lists some of such quirks: Section 1.18.

4.4. Why don't these examples from standard Jython textbooks work in HIPE?

The Herschel DP includes the core Jython engine only; additional libraries that are part of the full jython installation are excluded. Occasionaly these extra libraries are mentioned in textbook examples.

The best way to fix this is to download the full Jython installation from the Jython website, and set the property `hcss.jython.user.path` (using for instance the *property generator* tool) to include the `Lib` subdirectory of the full Jython installation.

4.5. What do these `import` and `from ... import` statements do?

The DP software consists of a number of modules and classes, covering different areas of functionality. For example, the `herschel.ia.numeric` modules contains functions and data structure definitions that can be used for numerical analysis. The `import` statements allow you to make such modules available for processing in HIPE.

Most import statements are not needed, since the packages are loaded by HIPE on startup.

4.6. Why does Jython complain when I use the backslash (\) in a file name ?

The backslash is the *escape character* in Java/Jython. For instance \a is interpreted as Control-A while \t is interpreted as the tab character. If you want to use backslashes you need to either 'escape' them, i.e. double them up to "\\", or simply replace each with a forward slash (/).

For example, if you have in mind the following statement:

```
table=ascii.load("\MyDocuments\ascii_demo_data.txt")
```

change it to either:

```
table=ascii.load("\\MyDocuments\\ascii_demo_data.txt")
```

or:

```
table=ascii.load("/MyDocuments/ascii_demo_data.txt")
```

4.7. Is Jython case sensitive?

The short answer: **very**!

```
x = 1
print x # OK
print X # Name error!
```

5. Numeric Library Usage

5.1. Why are some function names **UPPERCASE** and others are not?

In general array functions take the form:

```
output = f(input)
```

For performance reasons as well as serving the Java developers (those that write the DP software), we wrote the numeric library in Java. Many (simple) functions such as `SQUARE` correspond internally to Java constructs known as *static function objects*:

```
f = SQUARE
```

Normally one can have one or many objects of the same type (or class) but with possibly different contents, as shown in the following:

```
# lo and hi are instances of class Pixel
lo = Pixel(0,0)
hi = Pixel(1024,512)
```

You can imagine that we need only one function object that can do the square. If in Java you want to avoid multiple instances, we create a single/static instance of such a class, and Java developers stick to uppercase notation to reflect this.

Now consider the following: you want to solve a linear set of equations using matrix operations: `A x = b`. You can do this by using function objects:

```
f = MatrixSolve(b)
x = f(A)
```

One thing to note from the above example is that you can have many different instances of the MatrixSolve class (for every different b).

Note

The library provides a procedural wrapper for this specific function: `x = SOLVE(A,b)`.

You can find more information on functions in the *Scripting and Data Mining* guide: Section 3.2.

5.2. What are rectangular arrays?

Rectangular arrays are arrays which always have a rectangular shape. This may sound trivial, but Jython as well as Java provide jagged arrays only! (For jagged arrays, see below)

For more information about multidimensional arrays see the *Scripting and Data Mining* guide: Section 2.4.

5.3. What are jagged arrays?

Jagged arrays are in essence arrays of arrays. Taking a two-dimensional array as an example, every row can have a different number of elements:

```
x=[ [1,2],
[3,4,5],
None
```

```
-]
print x[0][2] # error: 2nd index is out of bounds!
print x[1][2] # 5
print x[2][2] # error: 2nd index does not exist!
```

For more information about multidimensional arrays see the *Scripting and Data Mining* guide: Section 2.4.

5.4. How do I remove infinite and NaN numbers from my array?

The basic toolbox predicate functions allow you to filter out infinite and NaN numbers (IS_FINITE, IS_INFINITE, IS_NAN).

Example 1:

```
# ---- Example 1
# set up an array with non-finite elements
x=Double1d.range(5)+1
x[0]=Double.NaN
x[3]=Double.POSITIVE_INFINITY
print x     # [NaN,2.0,3.0,Infinity,5.0]

# using a predicate function directly:
q=x.where(IS_FINITE)
print q     # [1,2,4]
print x[q]  # [2.0,3.0,5.0]
```

Example 2:

```
# ---- Example 2
# using predicate function to create a mask:
mask=IS_FINITE(x)
print mask  # [false,true,true,false,true]

q=x.where(mask)
print q     # [1,2,4]        (as above)
print x[q]  # [2.0,3.0,5.0]  (as above)

# -... asking to filter the non-finite numbers:
q=x.where(~mask)
print q     # [0,3]
print x[q]  # [NaN,Infinity]

# -...replace selected values
x[q]=0
print x     # [0.0,2.0,3.0,0.0,5.0]
```

6. Plotting

6.1. Why does my plot become huge after I add a layer?

You might have set the `width` and `height` parameters without setting the `autoAdjustWindowSize` property to `0`, for example by issuing

```
plot = PlotXY(width=600, height=400)
```

instead of

```
plot = PlotXY(width=600, height=400, autoAdjustWindowSize=0)
```

Another solution is to change the plot size only *after* having added all the layers.

For more information see the section entitled *Resizing a plot* in Chapter 3 of the *Data Analysis Guide*.

7. Herschel Science Archive

7.1. When I try to open the HSA User Interface (HUI) from HIPE, I get a message about *Java WebStart* (`javaws`) not being present. Why?

Java WebStart is a piece of software needed to fetch from the Internet the HSA User Interface. If HIPE cannot find it, it probably means that you are using a 64-bit version of Java prior to 1.6 update 12 (1.6u12). To find out which version of Java you have installed, issue this command from a terminal window:

```
java --version
```

To solve this problem, either switch to a 32-bit version of Java or (recommended) update to Java 1.6u12 or newer.

7.2. I have downloaded FITS files in a TAR archive from the HSA, but they are corrupted. How can I fix them?

You probably used WinZip, or some equivalent compression software, in Windows. WinZip has an option called *TAR file smart CR/LF conversion*, in the *Miscellaneous* tab of the *Configuration* dialogue window (at least for version 12.0), that is enabled by default and causes the corruption. Just disable the option to solve this problem. For more information, please see the *Data Analysis Guide*: Section 1.4.6.2.

7.3. In the HSA user interface I can query and select observations, but I do not get the option in the HSA to pass de data to HIPE, or in HIPE to load the selected products. Why?

If you are using Windows, the problem might be that you instructed your browser to auto-detect proxy settings. In Windows this is a system-wide setting which affects any application connecting to the Internet, including HIPE. Try to change the setting to *Direct connection to the Internet* or equivalent (see your browser documentation for details).

Note however that, if your site enforces proxy connection, the above setting will prevent your computer from connecting to the Internet.

7.4. When I click on *Send to external Application* in the HSA interface, nothing seems to happen. What is going on?

This action should bring to the front the main HIPE window, which will tell you that data is indeed being transferred. In some cases however the HIPE window does not appear. Please bring it to the front manually to follow the progress of your action.

7.5. When I try to retrieve data from the HSA I get an error message in the console about an *invalid magic number for FITS file*. What is going on?

The reason is that the "data" being downloaded is in reality a small file containing an error message. Two issues have been shown to cause this problem:

- Your HSA password has non-alphanumeric characters in it. Please change it to a password containing only letters and numbers.
- Your HSA username has uppercase characters. Please write it using only lowercase letters.

8. Other problems

8.1. HIPE crashed/froze! What do I do?

If you cannot find a solution in this document, you may have found a bug in HIPE. Please report it by opening a Helpdesk ticket. If the system produced a *dump file* when the crash happened, make sure to include it in your ticket. For more information about dump files and where to look for them, see the *HIPE Owner's Guide*: Section 1.1.

8.2. How do I set the maximum heap memory size available to HIPE?

You can set this value from the installer, but only if you choose the *advanced* installation type.

Afterwards, you can change the value in the *Startup & Shutdown* section of the *Preferences* dialogue window, which you can open by choosing Edit → Preferences.

You can also change the minimum and maximum memory sizes in the `installed.properties` file within your HIPE installation directory. Modify the following two properties:

```
java.vm.memory.min=64m
java.vm.memory.max=512m
```

8.3. I have enough memory on my HIPE session but I still get on error on lack of PermSpace. What can I do?

Add or edit the following property in the `installed.properties` file within your HIPE installation directory:

```
java.vm.options = --XX:PermSize=512m --Xincgc
```

8.4. I removed a product/observation via the Product Browser perspective, but it still shows up in queries. Why?

The removal of products and observations via the Product Browser perspective is still undergoing testing and improvement. This issue has been reported by some users and will be corrected in a future version of HIPE.

8.5. How can I specify the location of my local store?

Add or edit the following property in the `installed.properties` file within your HIPE installation directory:

```
hcss.ia.pal.pool.lstore.dir=alternative_path
```

Replace *`alternative_path`* with the path to the new local store location.

8.6. How can I to change the default temporary directory in HIPE?

Add or edit the following property in the `installed.properties` file within your HIPE installation directory:

```
java.vm.options = --Djava.io.tmpdir=new_temp_dir
```

Replace *`new_temp_dir`* with the path to the new temporary directory.

8.7. Why are coordinates switched when loading/saving an image from/to a FITS file?

When creating an image from, say, a `Double2d(50, 100)` and saving it to a FITS file, you might expect that `NAXIS1` will be 50 and `NAXIS2` will be 100. The opposite happens instead. This is because two-dimensional arrays in DP are created and accessed by indicating first the *row* coordinate (i.e. the *y* coordinate) and then the *column* coordinate. Coordinates are not *switched* as such, but are specified in a different order with respect to other applications and languages. To learn more about the reason for this behaviour, see the *Scripting and Data Mining* guide: Section 2.4.1.

8.8. I got a `LockObtainFailedException`. Why?

This happens if you have pools on NFS-mounted system (if in doubt, check with your system administrator). Set this property to solve the problem:

```
hcss.ia.pal.pool.lstore.lock = simple
```

8.9. There are PACS files I cannot delete! Why?

It might not be possible to delete the following files on Windows because their full path exceeds the 260 characters limit.

- `PCalSpectrometer_RelCalSourceFlux_FM_v1.fits_2009...80Z.fits`
- `PCalSpectrometer_RelCalSourceFlux_FM_v2.fits_2009...55Z.fits`
- `PCalSpectrometer_RelCalSourceFlux_FM_v2.fits_2009...14Z.fits`
- `PCalSpectrometer_RelCalSourceFlux_FM_v2.fits_2009...21Z.fits`
- `PCalSpectrometer_RelCalSourceFlux_FM_v2.fits_2009...58Z.fits`

This problem can be avoided by installing HIPE with a short initial path, such as `C:\hipe`, so that full paths remain below the limit.

8.10. HIPE does not accept keyboard input anymore! What's wrong?

If you are running Linux, this could be due to a compatibility problem between Java and SCIM (Smart Common Input Method). A workaround is to stop SCIM or uninstall it. Please refer to the documentation of your distribution for how to do so.

The problem is described in more detail on these pages:

- http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6506617
- http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6299259

8.11. Why do I get a `ClassCastException` when trying to save an observation with the `saveObservation` command?

You probably retrieved the observation without having the corresponding instrument software installed (for instance, you retrieved a PACS observation without having the PACS modules in HIPE).

To solve this problem follow these steps:

1. Reinstall HIPE, including the software of the desired instrument.
2. Delete the following directories (*`home`* being your home directory):
   - *`home`*`/.hcss/lstore/hsa_cache`
   - *`home`*`/.hcss/pal_cache/hsa`
3. Retrieve the observation again.

9. Advanced Information

9.1. What is a *namespace*?

A namespace is a bit in memory put aside by software to map variable names and values. Each variable is mapped to a single value.

In the case of Jython, there is a namespace in which variable names (such as Jython variables, complex objects, and Java class definitions) as associated with values. One problem with this is that if two `import` statements run in the same Jython session happen to import different values for the same class name. For example, the second import statement will overwrite the value for that class provided by the first import statement. This may lead to unexpected behaviour.

9.2. What is the *global* namespace?

The global namespace is the default namespace available from the HIPE prompt. It includes the most common Herschel libraries, such as numeric, plot, display, etc.

It also includes everything you have defined during your session: variables, functions, class definitions and so on.

9.3. Why does my script work from the HIPE Console window but fail when imported?

Commands executed from the Console window of HIPE make use of the global namespace of Jython. In your DP namespace a number of variables have already been defined for you. However, when such commands are saved into a script, and made available through the jython `import` statement, the global namespace is not available. This leads to the failure.

Here is an example. From the HIPE Console:

```
a = Double1d.range(10) # Fine! Double1d is available by default in the
global namespace
```

When putting the same in a script:

```
# In my script.py
a = Double1d.range(10)
```

Then importing the script from the HIPE Console:

```
from script.py import * # Name error! Double1d is not available in script.py
```

Here is the corrected script:

```
# In my script.py
from herschel.ia.numeric import Double1d
a = Double1d.range(10)# Fine! Double1d is now available in script.py
```

9.4. How can I save or restore an object or group of objects?

To save one or more objects in a HIPE session to a local file, use the `save` command:

```
save("foo.sav") # Save all variable names and values
save("foo.sav","x,y,z") # Save specified variable names and values
```

To restore an object from a local file, use the `restore` command:

```
restore("foo.sav") # Restores variables that were stored with -"save".
```

Note

There are limitations to what types of object can be reliably saved and restored. Simple variable values (ints, doubles, strings), objects based on the Numeric library, and standard Dataset objects can be reliably saved and restored.

Saving other objects is not always reliable. Problems have been found when saving/restoring objects defined using Jython constructs, and for objects defined using Java classes, those Java classes need to be carefully designed. Please contact your DP representative for further information.

9.5. Why do I keep getting `IndexError` or `IllegalArgumentException: <query> could not be evaluated correctly` messages when I run my query on my PAL Product Storage?

You could get these message for one of the following reasons:

1. Your query string (the third argument of a query, eg 'p.creator==..') is simply not consistent with the Jython syntax and could not be correctly interpreted. Check your query string by evaluating it on the Jython command line. If your query uses a *handle* to a product (eg the `p` in a query `p.meta[..]` is a handle), then create a dummy product of the type you want to query on the comand line to test the query against.

2. It could be possible that the query references some data that does not exist in *any* of the products that match the product type you have passed in that query. If you see in the details of the error message something along the lines of '<something> does not exist', then this may be the case.

   For example, consider the following MetaQuery:

   ```
   query =MetaQuery(Product, -'p', -'p.meta["temperature"].value==10)
   resultset=storage.select(query)
   ```

   The query first starts creating a shortlist of all products in the storage matching type 'Product'. It then runs the query string on each product in that shortlist. If any of those products don't contain the information referenced in the query string, an error is raised.

   There are two ways to avoid this:

   a. Be as specific as you can when it comes to specifying the product type in a query. If you know the product type you want to query is of type 'CalHrsQDCFull', then specify that. Running queries using the most general product type of 'Product' is not recommended, unless the products you have saved are of this type only.

   b. Run a two-stage query, using the containsKey() operator to check whether a component exists first, eg:

      # Get a sub-set of products that contain the metadata 'temperature'

      ```
      queryOne= MetaQuery(Product, -'p', -'p.meta.containsKey("temperature")')
           resultsetOne = storage.select(queryOne)
      ```

      # Run the original query on this subset

      ```
       queryTwo =MetaQuery(Product, -'p', -'p.meta["temperature"].value==10)
           resultsetTwo = storage.select(queryTwo, resultSetOne)
      ```

9.6. Why is my PAL query so slow?

One of the possible reasons is that you are executing a FullQuery, and full queries by their very nature are the most intense of queries and are therefore the slowest.

FullQuery executions should be run as the last stage of a multi-stage query operation. Below is an example of how to search a storage for products of type 'MyProduct' that are created by a developer called 'timo', but contain a specific value in the product data itself.

Stage one: Find all products of type MyProduct with creator 'timo'

```
attquery = AttQuery(MyProduct, -'p', p.creator=='timo')
resultset = storage.select(attquery)
```

Final stage: Find all products in selection generated from previous queries, # that has a value 10 in the column 'mycolumn' in dataset 'mydataset'

```
fullquery = FullQuery(Product, -'p', -'p["mydataset"]
["mycolumn"].data[5]==10')
storage.select(fullquery, resultset)
```

There can be as many intermediate queries between the first stage and final stage involving AttribQuery or MetaQuery, but FullQuerys should be left to last.