HIPE Owner's Guide

1. Preface

This document gives you a general overview of HIPE (Herschel Interactive Processing Environment), the main interface to the Herschel Data Processing (DP) software. HIPE provides a graphical interface plus command-line access to the data processing capabilities of the Herschel Common Science System (HCSS).

Read this document if you want to obtain a first understanding of how HIPE looks and its functionality.

Tip

If you are interested in more advanced features, including scripting and batch processing, please have a look at the *Scripting and Data Mining* guide.

1.1. When things go wrong

In the unfortunate event of a HIPE crash, or a freeze that forces you to kill the application, please submit a Helpdesk ticket so that developers can look into the problem.

In some cases, a *dump file* is created, listing the values of crucial parameters when the crash happened. Please include it in your ticket, since it will greatly help developers in diagnosing what went wrong. Here is how to look for the dump file:

- Sometimes HIPE crashes because the underlying Java Virtual Machine has crashed. In this case, the JVM produces a dump file called `hs_err_pid*.log`, where `*` is the process identification number. This dump file is created in the working directory, that is, the directory HIPE was started from. If you are not sure, please use the search function of your operating system.

- When you kill HIPE manually, a dump file called `hipe_dump_`*`yyyymmdd_hhmmss`*`.txt`, where *`yyyymmdd_hhmmss`* is the date and time of creation.

Depending on how you kill HIPE, the dump file may or may not be created. The following table shows the various cases and the location of the home directory for each supported operating system. Dump files are created in a `.hipe/dumps` folder within your home directory.

Table 1. Where and when HIPE creates a dump file

Operating system	Dump file created	Dump file *not* created	Home directory location
Linux/Unix	**Ctrl**+**C** or **kill**	**kill -9**	`/home/`*`username`*
Mac OS	**Ctrl**+**C**, **kill** or *Force Quit*	**kill -9**	`/Users/`*`username`*
Windows	**taskkill** or *Task Manager* → *Applications*	**taskkill /f** or *Task Manager* → *Processes*	`C:\Documents and Settings\`*`username`* (Windows XP), `C:\Users\`*`username`* (Windows Vista)

2. Getting to know HIPE

HIPE (Herschel Interactive Processing Environment) provides an integrated suite of graphical interfaces that can interact with each other. It allows for interactively choosing your active data in your session, visualizing that data in various ways and selecting tools that can operate on the data. Both command-line and GUI interfaces are available. High-level interactions, which can involve GUIs, are also echoed as commands on the command-line that allow the saving of commands used in a session and the generation of scripts from these interactions.

This section provides an overview of the fundamental elements of HIPE.

2.1. The HIPE philosophy

These are the main objectives that guided the development of HIPE:

- An integrated application, giving access to all data processing functionality in a unified graphical interface.
- A single look-and-feel for window layouts, toolbars, buttons and menus.
- A customizable layout, which allows you to decide which windows are relevant and how these windows are layed-out on screen.
- User guidance, including command-line echoing of main graphical functionality, allowing you to learn the scripting language by interacting with the system.
- An extendible application, which can be enriched with new modules that are automatically integrated.

2.2. Installation and startup of HIPE

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.

Important

In case of any problems during installation please contact the Herschel Helpdesk via the Herschel Science Centre website.

HIPE is part of the Herschel Data Processing system. You can download an installer from the Herschel Science Centre on this page. Installation instructions are provided at the bottom of the page.

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

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

Development builds 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 (details on this page).

HIPE needs the Java SE Runtime Environment version 6. We recommend to install the latest update for best performance. To see which Java version is installed on your machine type the following in a terminal window:

```
java --version
```

You can run HIPE on a server or individual workstation under one of these operating systems, either 32-bit or 64-bit:

- Windows XP and Vista.
- Any Linux distribution compliant with the Linux Standard Base version 3.0 or higher. A list of compliant distributions is available on this page. HIPE may work on non-compliant distributions too, but this is not guaranteed.
- Mac OS X 10.5 Update 2 or later.

The minimum amount of memory needed is two GBytes. Four GBytes are the highly recommended minimum for data processing; more may be needed in some cases (e.g. large PACS maps).

The maximum amount of memory available to HIPE can be set by choosing the *Advanced* installation. To change the available memory afterwards, modify the properties `java.vm.memory.min` and `java.vm.memory.max` in the relevant property file, as shown below:

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

The **Advanced** installation also allows you to specify the name and location of a Versant database. Only advanced users and calibration scientists should need to specify a database: if you are unsure about this option, you probably do not need it.

Recommended browsers to visualize the HIPE Help System are the following:

- Microsoft Internet Explorer 6 or newer.
- Netscape 7 or newer.
- Firefox 1.5 or newer.
- Apple Safari.

Once the software is installed, HIPE can be started by several means. Using Windows, Herschel software can be started under the Start menu after a standard installation. Alternatively, HIPE can be started from a command line:

```
hipe
```

The installer will tell you where the `hipe` executable is located.

Note

Under Mac, starting from the command line only works if you installed a developer build via the Continuous Integration System. If you used the *InstallAnywhere* installer instead, you have to start the application via its icon.

When starting 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.

Upgrading and uninstalling

To uninstall HIPE, either follow the usual uninstallation procedure for your operating system, or execute the **uninstall_hipe** command in the `uninstall_hipe` folder.

It is possible to install different HIPE versions alongside each other. However, the safest way to upgrade to a new version of HIPE is to uninstall the existing version first.

Important

Most files of a HIPE installation are stored into a `.hcss.d` folder within your home folder. This folder is shared among all the HIPE installations on a computer, which means

> that *the uninstaller will not clean it up* to avoid damaging other installations. You can remove this folder manually after you have removed HIPE completely from your system. Any HIPE installation left *will stop working* after removing this folder.

2.3. Obtaining help in HIPE

You can open the HIPE Help System via the Help menu. Help pages will open in a new tab of your default web browser.

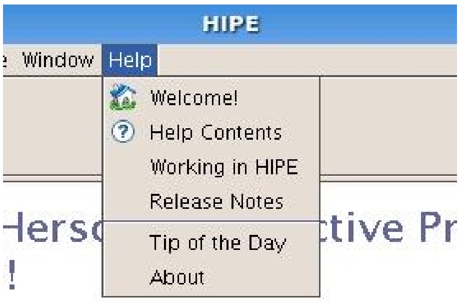


Figure 1. Accessing the HIPE Help System.

You can access the following help pages from the Help menu:

- **Help Contents:** this option will open the main help window. The table of contents on the left displays the available documentation, with more advanced material appearing towards the bottom (see Figure 2, still showing the old documentation structure).
- **Working in HIPE:** this option will open Chapter 1 of the *Hipe Owner's Guide*, the same you are reading now.
- **Release Notes:** this option will open a document with the most recent additions and changes to the system.

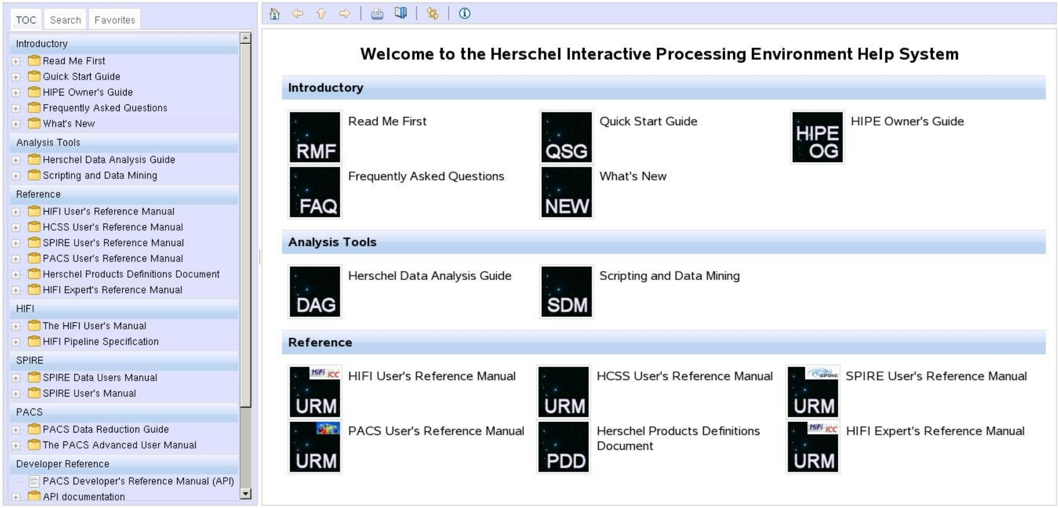
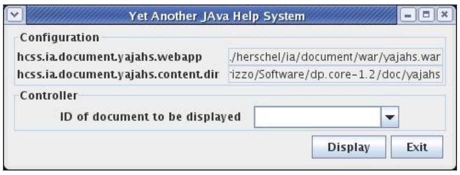

TOC
Search
Favorites
Introductory
Read Me First
Quick Start Guide
HIPE Owner's Guide
Frequently Asked Questions
What's New
Analysis Tools
Herschel Data Analysis Guide
Scripting and Data Mining
Reference
HIFI User's Reference Manual
HCSS User's Reference Manual
SPIRE User's Reference Manual
PACS User's Reference Manual
Herschel Products Definitions Document
HIFI Expert's Reference Manual
HIFI
The HIFI User's Manual
HIFI Pipeline Specification
SPIRE
SPIRE Data Users Manual
SPIRE User's Manual
PACS
PACS Data Reduction Guide
The PACS Advanced User Manual
Developer Reference
PACS Developer's Reference Manual (API)
API documentation
Welcome to the Herschel Interactive Processing Environment Help System
Introductory
RMF
Read Me First
QSG
Quick Start Guide
HIPE OG
HIPE Owner's Guide
FAQ
Frequently Asked Questions
NEW
What's New
Analysis Tools
DAG
Herschel Data Analysis Guide
SDM
Scripting and Data Mining
Reference
URM
HIFI User's Reference Manual
URM
HCSS User's Reference Manual
URM
SPIRE User's Reference Manual
URM
PACS User's Reference Manual
PDD
Herschel Products Definitions Document
URM
HIFI Expert's Reference Manual

Figure 2. The HIPE Help System home page.

2.3.1. Obtaining help outside HIPE

When you quit HIPE, links in any help pages will not work anymore. Even if you restart HIPE, you will have to open the Help System again.

To access help outside HIPE you can use the **show_help**, which resides in the same folder as the **hipe** executable. This will bring up the following window:



hcss.ia.document.yajahs.webapp ./herschel/ia/document/war/yajahs.war

hcss.ia.document.yajahs.content.dir izzo/Software/dp.core-1.2/doc/yajahs

ID of document to be displayed

Figure 3. The standalone Help System.

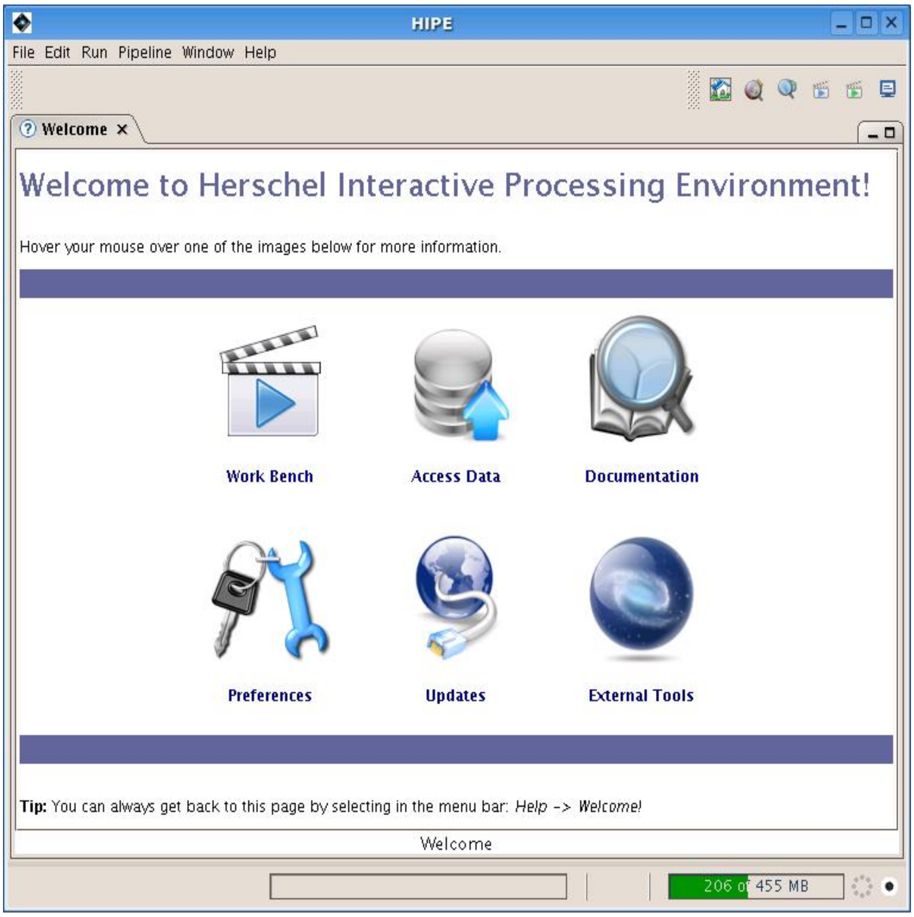
Clicking on the Display button will start the HIPE Help System.

For advanced users: accessing the Javadoc

If you are comfortable with Javadoc documentation, you can access it in the HIPE Help System home page by clicking on *HCSS Developer's Reference Manual (API)* in the table of contents. To obtain the traditional frame-based Javadoc layout, click on the *FRAMES* link on any Javadoc page. To get 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 have both layouts available, you may want to open the Javadoc layout in a new tab or window of your browser, by right-clicking on the *FRAMES* link.

2.4. The HIPE Welcome screen

When you start HIPE you are greeted by a Welcome screen with six icons, which are described in the next sections. Note that the *Updates* icon is not yet functional. Placing the mouse over each of the icons on the screen provides a small description in the status bar at the bottom (see Figure 4).



Welcome to Herschel Interactive Processing Environment!

Hover your mouse over one of the images below for more information.

Tip: You can always get back to this page by selecting in the menu bar: *Help -> Welcome!*

Welcome

Figure 4. Information on 'Welcome' screen icons. See bottom strip of the HIPE screen for the explanation of each icon the mouse is placed over.

You can return to the Welcome screen at any time by selecting Help → Welcome! from the menu bar or by clicking on the icon at the top right of the HIPE window.

2.4.1. Icon: Work Bench

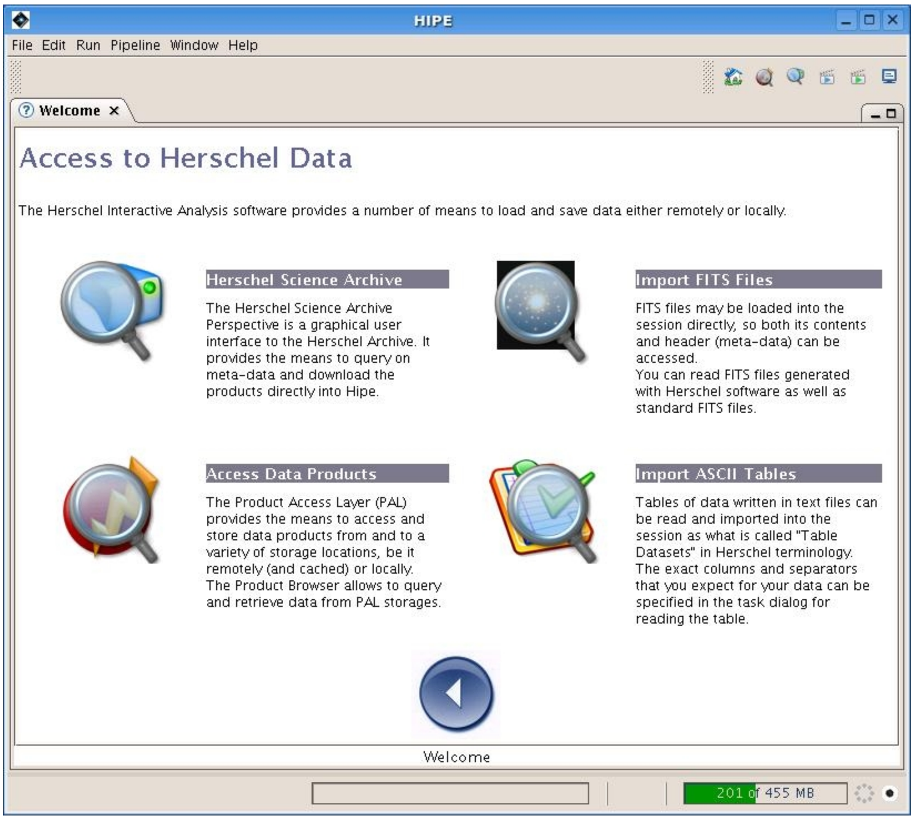
Clicking on the icon takes you to the *Work Bench* perspective (for information on perspectives in HIPE see Section 3.1). The default workbench is shown in Figure 10. This is the main work area for doing data analysis. Here you can look at data values, plot spectra and images, create scripts for batch processing and run analysis tools. The contents of the workbench can be updated with various *views* available under the *Window* menu (see Section 3.3 on available Views).

There is also a *Full Work Bench* perspective, with more windows than the regular workbench. You can open both perspectives via the Window → Show Perspectives menu.

2.4.2. Icon: Access Data

The icon opens a window with four more icons, providing access to data held either locally or at a remote site (for example the Herschel Science Archive). It also allows the import of FITS and ASCII table files (see Figure 5).

For more information on these features see the *Data Analysis Guide*: Chapter 1.



Access to Herschel Data

The Herschel Interactive Analysis software provides a number of means to load and save data either remotely or locally.

Herschel Science Archive

The Herschel Science Archive Perspective is a graphical user interface to the Herschel Archive. It provides the means to query on meta-data and download the products directly into Hipe.

Import FITS Files

FITS files may be loaded into the session directly, so both its contents and header (meta-data) can be accessed.
You can read FITS files generated with Herschel software as well as standard FITS files.

Access Data Products

The Product Access Layer (PAL) provides the means to access and store data products from and to a variety of storage locations, be it remotely (and cached) or locally.
The Product Browser allows to query and retrieve data from PAL storages.

Import ASCII Tables

Tables of data written in text files can be read and imported into the session as what is called "Table Datasets" in Herschel terminology.
The exact columns and separators that you expect for your data can be specified in the task dialog for reading the table.

Figure 5. HIPE Data Access initial page.

Click the arrow at the bottom to go back to the *Welcome* screen.

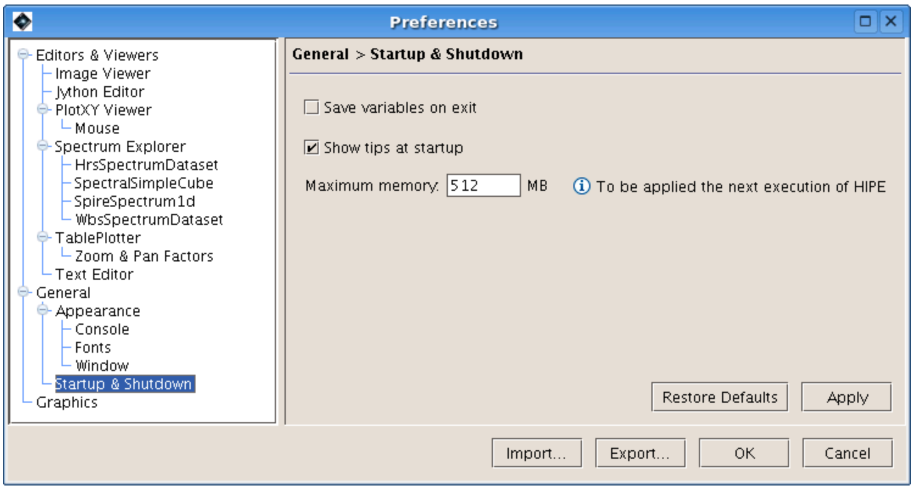
2.4.3. Icon: Documentation

Click on the icon to open HIPE documentation in your default Web browser. For more information see Section 2.3.

See the *Read Me First* document for a brief description of all the available manuals.

2.4.4. Icon: Preferences

The icon allows you to access the *Preferences* dialogue window shown in the following figure:



General > Startup & Shutdown

☐ Save variables on exit

☑ Show tips at startup

Maximum memory: 512 MB

To be applied the next execution of HIPE

Figure 6. The HIPE preferences window.

You can set preferences in the available categories and revert to the default values at any time. You can also export/import the values of the preferences to/from an XML file.

2.4.5. Icon: Updates

The icon functionality is **not implemented yet**. Clicking on this item will allow you to search for software updates available from the Herschel Science Centre.

At the bottom of the screen is a large back-arrow button that allows you to return to the original "Welcome" screen.

2.4.6. Icon: External Tools

The icon takes you to a set of icons linking to Virtual Observatory tools, including VOSpec, VOPlot and Aladin. Included in this listing is the Herschel Science Archive (HSA) browser, as Herschel components are VO-compliant. The HSA also uses a VO-like interface with HIPE. Clicking on any of the icons launches the external VO tool. Help and assistance with these tools are provided separately from within the tools or associated websites, except for the HSA browser interface which is described in the *Data Analysis Guide*.

When you click on the icon of an external application, it will be automatically downloaded and launched. After the download you may see a security dialogue window telling you that the application's digital signature is missing or has an error. For the applications shown in this window you can safely click on the Run button. You can check the *Always trust content from this publisher* checkbox to avoid being prompted again.

Note

An external application is downloaded the first time you click on its icon, and whenever a new version becomes available. Note that the application will be downloaded *even if you have a version already installed* on your hard disk.

At the bottom of the screen is a large back-arrow button that allows you to return to the original "Welcome" screen.

3. Views and perspectives

A *view* is a window embedded in the main HIPE window, like the one shown in Figure 7.

A *perspective* is a collection of related views. You can customise a perspective by adding, deleting and moving views. To save space, you can also rearrange different views as tabs of the same window.

The following sections describe in detail the views and perspectives available in HIPE.

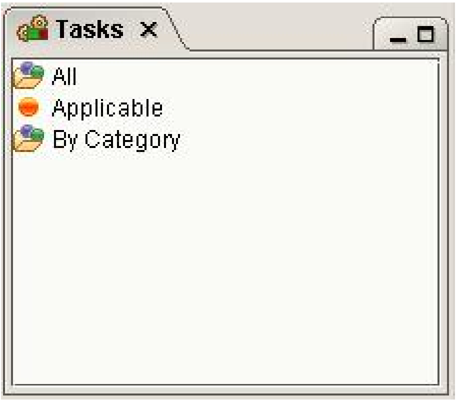


Figure 7. A single element (view) for a HIPE perspective.

3.1. HIPE perspectives

HIPE comes with five perspectives, available in the Window → Show Perspectives menu.

3.1.1. The Product Browser perspective

The **Product Browser** perspective provides a way of getting and briefly viewing data from databases and data stores, both locally and remotely stored. This is illustrated in Figure 8.

For more information see the *Data Analysis Guide*: ????.

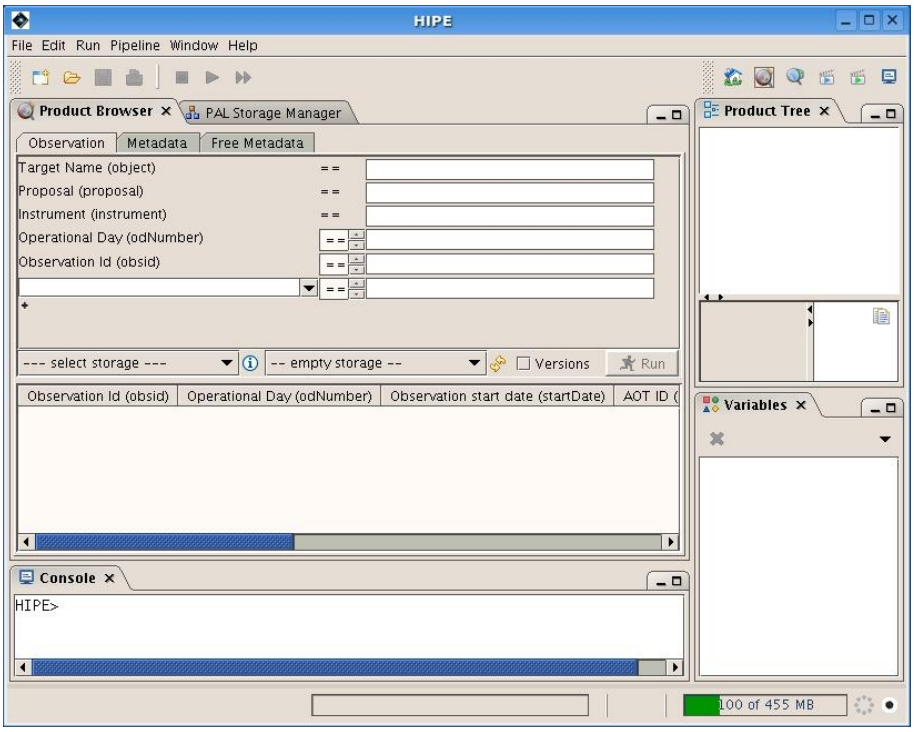


Figure 8. Product Browser perspective. This provides access to data stores both on-line and on your computer.

3.1.2. The Herschel Science Archive perspective

The **Herschel Science Archive** perspective provides a convenient means of querying and obtaining data from the Herschel Science Archive (HSA). For more information see the *Data Analysis Guide*: Chapter 1.

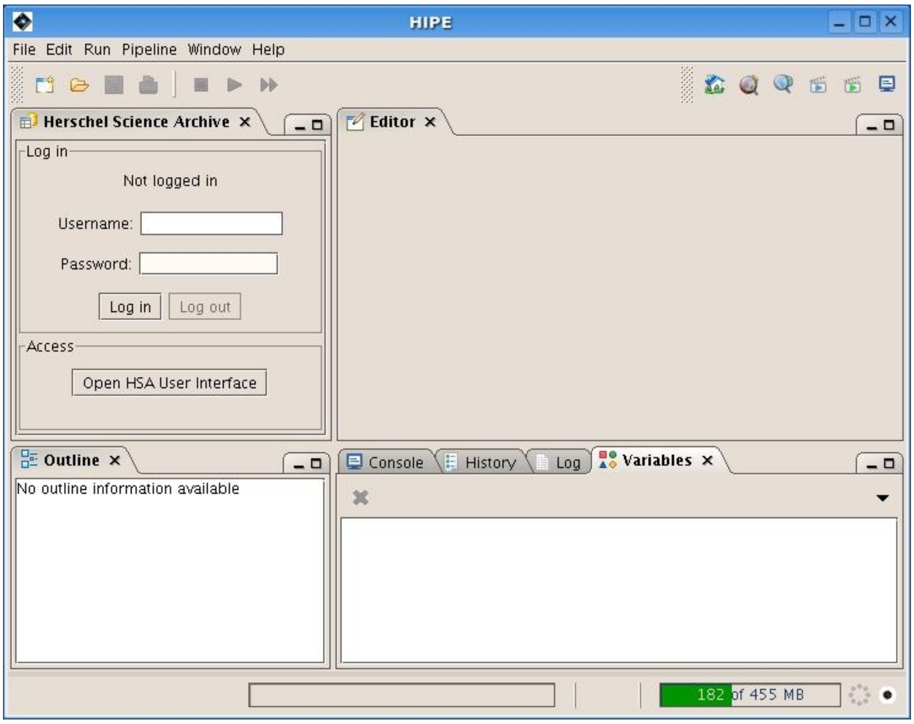
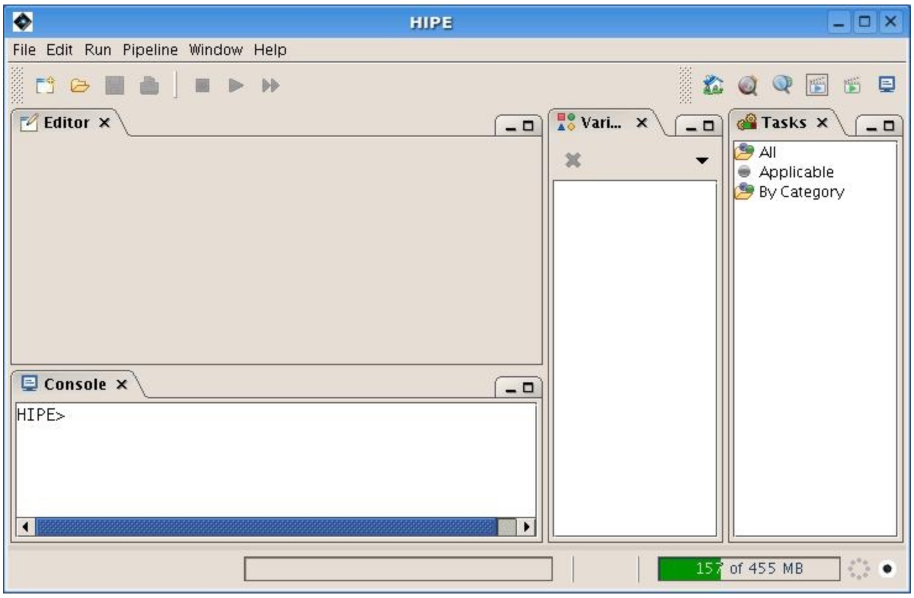


Figure 9. Herschel Science Archive perspective.

3.1.3. The Work Bench perspective

The **Work Bench** perspective provides four views for working with tasks, variables, scripts and command-line inputs. a slimmed-down general environment similar to the work bench but with only with four windows (views). The editor, console, variable list, outline, tasks views are available (see Figure 10).

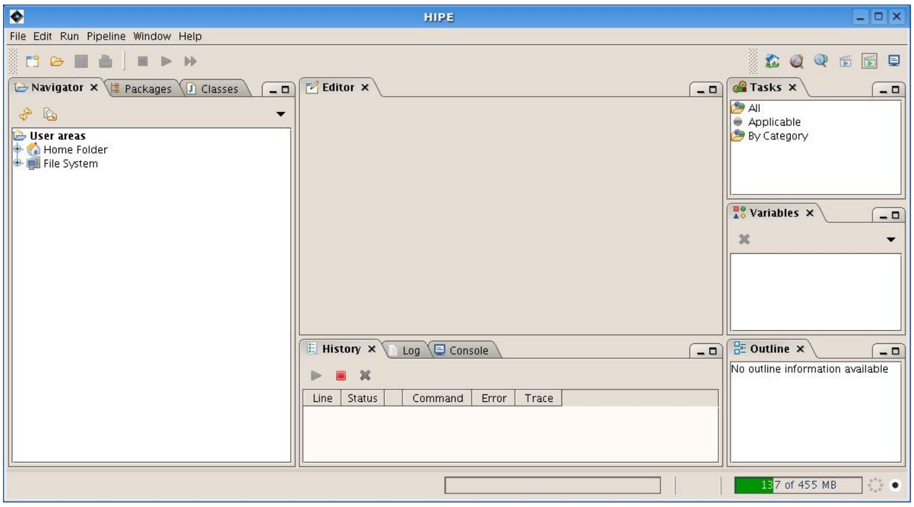


```
HIPE>
```

Figure 10. HIPE default view of the work bench perspective.

3.1.4. The Full Work Bench perspective

The **Full Work Bench** perspective provides more views than the standard work bench. Note that some views are arranged as tabs (see Figure 11).



No outline information available

Figure 11. The Full Work Bench perspective.

3.1.5. The Classic(JIDE) perspective

The **Classic JIDE** perspective provides an environment similar to the legacy JIDE application. Using this perspective is only recommended to ease the transition from JIDE to HIPE.

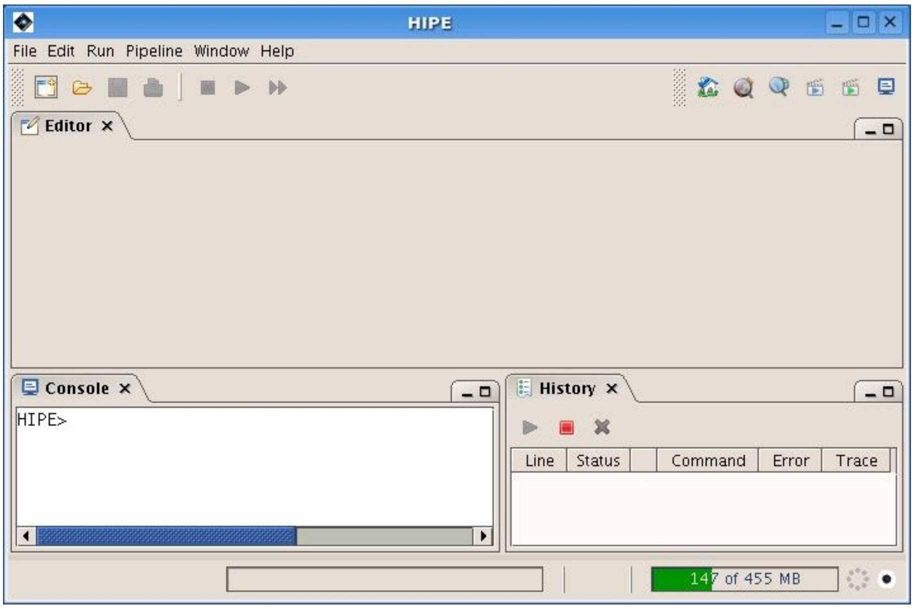


Figure 12. HIPE's 'classic' JIDE perspective.

3.2. Changing HIPE Perspectives

You can modify a perspective by deleting and adding views, and by moving, resizing and rearranging existing views. The following actions are available:

- **Closing, minimising and maximising views**. Each view has the usual icons for closing (next to the view title), minimising and maximising (in the small tab on the right-hand side). Minimised windows appear at the bottom of the workspace (see Figure 13).

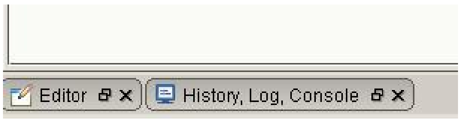


Figure 13. Minimised views at the bottom of the HIPE window.

- **Resizing views.** Just click and drag the borders of a view

- **Customising tabs**. If you right click on the title of a view, a context menu appears. With the *Tab Orientation* and *Tab Direction* submenus you can change the position and orientation of title tabs. Experiment with the various options until you find what works best for you.

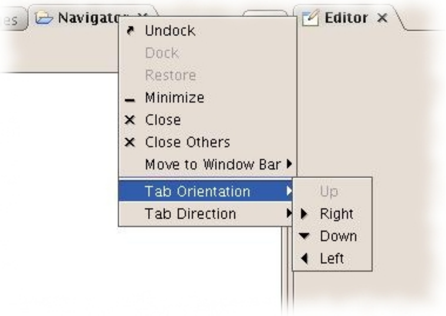


Figure 14. Customising tabs in a HIPE view.

- **Moving views.** To move a view within the main HIPE window, click on the view title and drag it around. A black outline box shows where the view will be placed when you release the mouse button.

- **Undocking views.** Undocking a view means turning it into an independent window, separate from the main HIPE window. To undock a view, click on its title and drag it away. A small rectangle under the mouse pointer (see Figure 15), with *no black outline* anywhere in the HIPE window, means that the view will be undocked when you release the mouse button. To dock an undocked view, right-click on its title and choose *Dock* from the context menu.

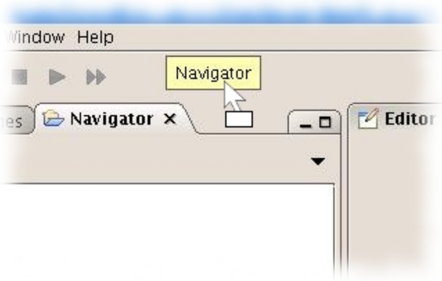


Figure 15. A view being undocked.

- **Moving through tabs**. If a window is too narrow to display all its tabs, three new controls appear: two arrows to cycle through the tabs, and one more arrow to get the list of tabs (see Figure 16).

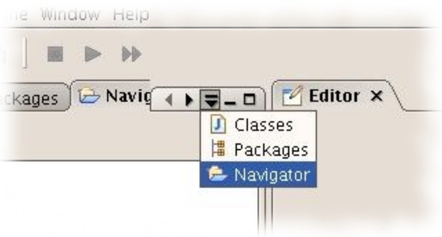


Figure 16. Moving through tabs.

- **Adding views**. To add a view to a perspective, select it from the Window → Show View menu. If a view is already in a perspective, it will *not* be added again.

- **Resetting a perspective**. Any change you make to a perspective will be kept in future HIPE sessions. To restore a perspective to its original settings, choose Window → Reset perspective, or right-click on the perspective icon at the upper right corner of the main HIPE window and choose *Reset perspective* from the context menu (see Figure 17).

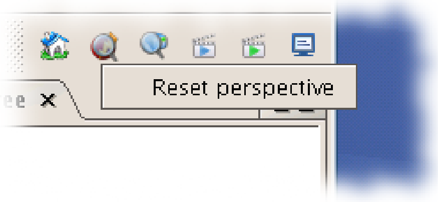


Figure 17. Resetting a perspective.

3.3. HIPE views

3.3.1. Classes

This view lists all the classes (routines) available in the session. These can include scripts you loaded into the system. To obtain help on an item, right-click on it and choose *Help in URM*. This opens the corresponding entry of the *User Reference Manual* in your web browser. If there is no information available in the URM, the menu entry does not appear.

3.3.2. Console

The *Console* view provides a command-line interface to all HIPE commands and functions.

Press the **Up** key to access previous commands. This feature also works *across* sessions: if you have just started HIPE, pressing the **Up** key will show the last command of the previous session.

Most graphical tools within HIPE echo in the *Console* view the commands corresponding to the action being taken. This is useful if you want to automate a certain procedure through a script.

For more information on the scripting language used by HIPE, see Scripting and Data Mining.

3.3.3. Data Access

This view is **deprecated**. Use the *Product Browser* perspective instead (see Section 3.1.1).

3.3.4. Editor

Use the *Editor* view to display and edit scripts and other files. When you open a Jython script, a set of editing commands becomes available under the Edit menu (see Figure 18) and as icons in the toolbar below. Hover your mouse over each icon to obtain a tooltip.

You can customise the Editor window for Jython scripts and other text files by choosing Edit → Preferences. This opens the main HIPE Preferences window. Set the options under Editors & Viewers → Jython Editor and Editors & Viewers → Text Editor to change how the editor behaves.

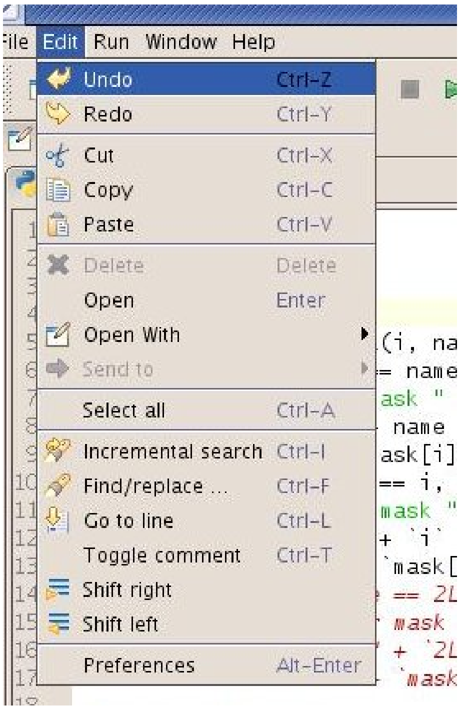
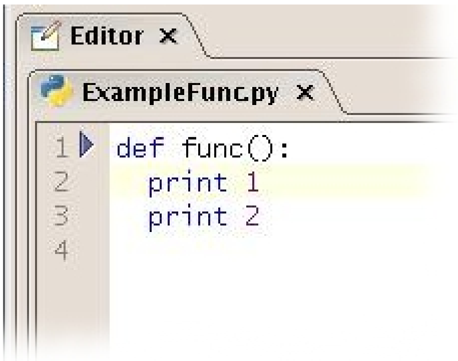


Figure 18. The Edit menu.



```
def func():
    print 1
    print 2

```

Figure 19. Editing a Jython script. The line with the cursor is highlighted. The arrow shows the next line to be executed.

You can run scripts with the following four icons:

- ▶ *Run*: runs either a single line of your script (the one indicated by the arrow) or all the *selected* lines at once.

- ▶▶ *Run all*: runs the entire script.

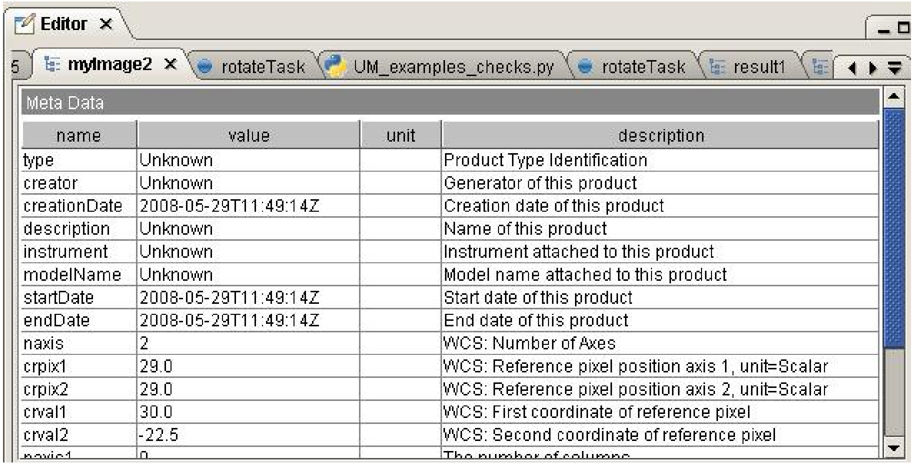
- ■ *Stop*: stops execution of the script. Note that the current line of the script will be completed before the script stops running. This can lead to a delay before the script stops.

  This button also stops other activities within HIPE, like data analysis tasks. Note that if a task has not been programmed to "listen" to the *Stop* button, pressing it will have no effect.

- *Line by line*: when pressed, this button is surrounded by a darker background. This causes a change in the behaviour of the *Run* button, which is best shown by an example.

  Copy the script shown in Figure 19. Ensure that the *Line by line* button is *not* pressed. Select the two lines saying `print 1` and `print 2` and press the *Run* button. You will get an error message. This is due to the fact that these two lines are within a *code block* (a function in this case). You are trying to execute only part of the block, which confuses the interpreter. If you press the *Line by line* button and try again, the code will run with no errors.

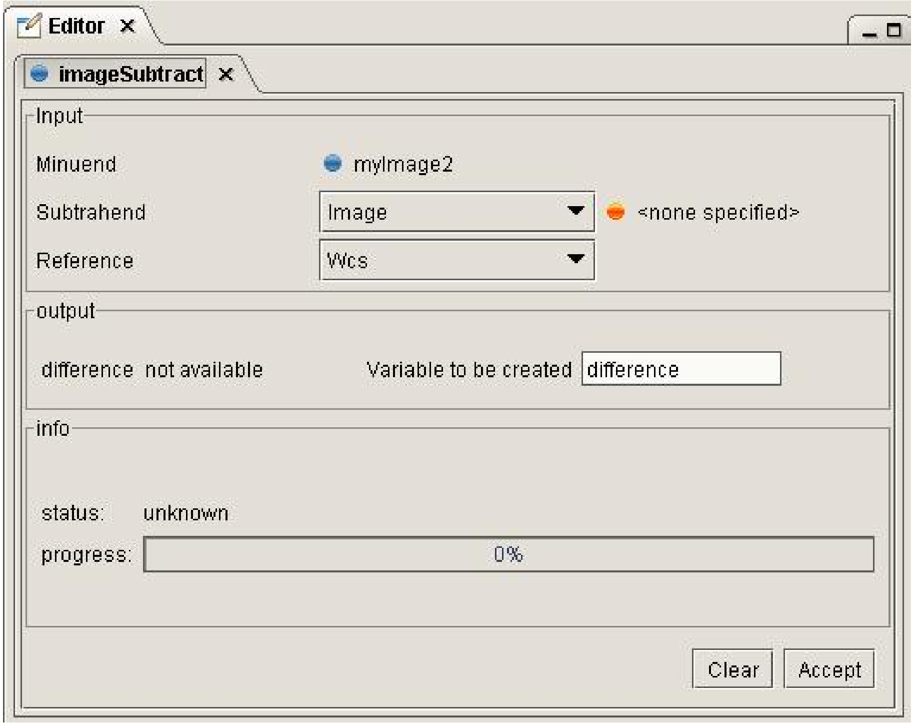
The *Editor* view is used for much more than editing script. For example, it can show detailed information about data products (see Figure 20) and is where the graphical interface of data analysis tasks is shown (see Figure 21 for a typical example).



Meta Data

name	value	unit	description
type	Unknown		Product Type Identification
creator	Unknown		Generator of this product
creationDate	2008-05-29T11:49:14Z		Creation date of this product
description	Unknown		Name of this product
instrument	Unknown		Instrument attached to this product
modelName	Unknown		Model name attached to this product
startDate	2008-05-29T11:49:14Z		Start date of this product
endDate	2008-05-29T11:49:14Z		End date of this product
naxis	2		WCS: Number of Axes
crpix1	29.0		WCS: Reference pixel position axis 1, unit=Scalar
crpix2	29.0		WCS: Reference pixel position axis 2, unit=Scalar
crval1	30.0		WCS: First coordinate of reference pixel
crval2	-22.5		WCS: Second coordinate of reference pixel
naxis1	0		The number of columns

Figure 20. A window shows metadata associated with an image within the Editor view.



Minuend mylmage2
Subtrahend Image <none specified>
Reference Wcs

difference not available
Variable to be created difference

status: unknown
progress: 0%

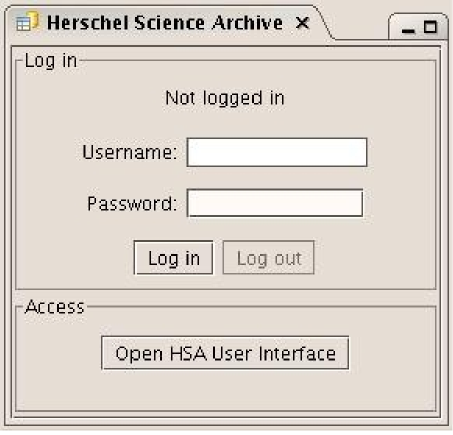
Figure 21. Window showing a task interface within the Editor view.

3.3.5. Export Herschel Data from HIPE

This view allows Herschel data to be exported from the session into a directory structure which is identical to the one found in the tar file of observations received from the Herschel Science Archive.

3.3.6. Herschel Science Archive

In this view you can log into the Herschel Science Archive (HSA) and open its graphical interface. See the *Data Analysis Guide* for more information: Chapter 1.



Not logged in

Figure 22. The Herschel Science Archive interface view.

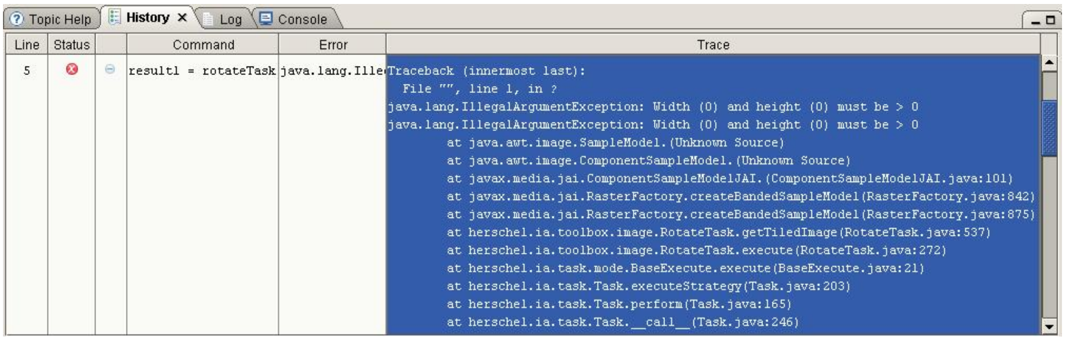
3.3.7. HIFI pipeline

This view reacts to `HifiTimelineProduct` objects only. Selecting a `HifiTimelineProduct` will show the tasks that can be used on that object.

3.3.8. History

The *History* view provides a list of the commands executed at the console and from Jython scripts. This also shows whether the command was successful or not. Pay attention to these icons:

- ☑ The command was successful.
- ⊗ The command caused an error.
- ⊕ This icon appears next to a failed command, and indicates that the row can be expanded to reveal additional information. In particular, the *Trace* column will show a detailed *traceback* of the error (see Figure 23).



Line	Status		Command	Error
5	⊗	⊖	result1 = rotateTask	java.lang.Ille

Trace

```
Traceback (innermost last):
  File "", line 1, in ?
java.lang.IllegalArgumentException: Width (0) and height (0) must be > 0
java.lang.IllegalArgumentException: Width (0) and height (0) must be > 0
        at java.awt.image.SampleModel.(Unknown Source)
        at java.awt.image.ComponentSampleModel.(Unknown Source)
        at javax.media.jai.ComponentSampleModelJAI.(ComponentSampleModelJAI.java:101)
        at javax.media.jai.RasterFactory.createBandedSampleModel(RasterFactory.java:842)
        at javax.media.jai.RasterFactory.createBandedSampleModel(RasterFactory.java:875)
        at herschel.ia.toolbox.image.RotateTask.getTiledImage(RotateTask.java:537)
        at herschel.ia.toolbox.image.RotateTask.execute(RotateTask.java:272)
        at herschel.ia.task.mode.BaseExecute.execute(BaseExecute.java:21)
        at herschel.ia.task.Task.executeStrategy(Task.java:203)
        at herschel.ia.task.Task.perform(Task.java:165)
        at herschel.ia.task.Task.__call__(Task.java:246)
```

Figure 23. Traceback of an error from the History view.

You can also copy the commands to the clipboard and save them to file. This is useful if you want to re-run the commands later as a script. To copy or save the command history, follow these steps:

1. Click on the header of the *Command* column to select the entire column. Alternatively, click and drag the mouse pointer within the column to select only a few cells.

2. Right click anywhere in the table. A context menu appears, with the *Copy* and *Save* options.

   - If you choose *Copy*, you can then paste the information in a script within the *Editor* view.

   - If you choose *Save*, a standard dialogue window for saving to file appears.

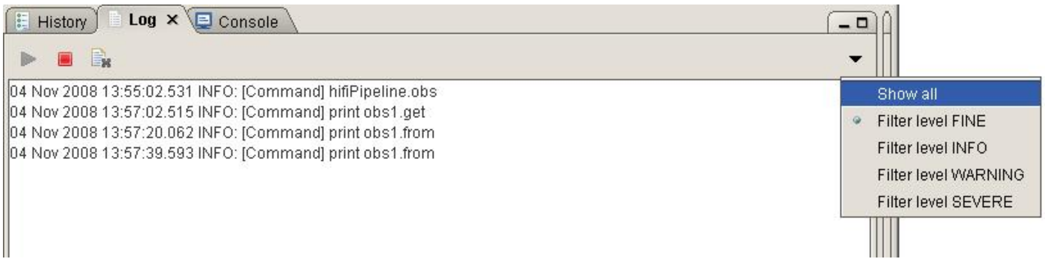
Information in the *Error* and *Trace* columns can be copied and saved in the same way.

3.3.9. Import Herschel data to HIPE

This view allows Herschel data to be imported from the tar file of observations received from the Herschel Science Archive.

3.3.10. Log

This view lists all the commands that have been executed, like the *History* view, but also any output and warning generated by the system. You can filter messages according to their severity level: click the arrow at the upper right corner of the view to reveal the menu shown in Figure 24.



```
04 Nov 2008 13:55:02.531 INFO: [Command] hifiPipeline.obs
04 Nov 2008 13:57:02.515 INFO: [Command] print obs1.get
04 Nov 2008 13:57:20.062 INFO: [Command] print obs1.from
04 Nov 2008 13:57:39.593 INFO: [Command] print obs1.from
```

Figure 24. Filtering messages in the Log view.

3.3.11. Navigator

The *Navigator* view provides access to the files and directories on your disk. Right-click on any item to display a context menu with the available actions (see for instance Figure 25). Double-click on an item to open it in HIPE with the default tool.

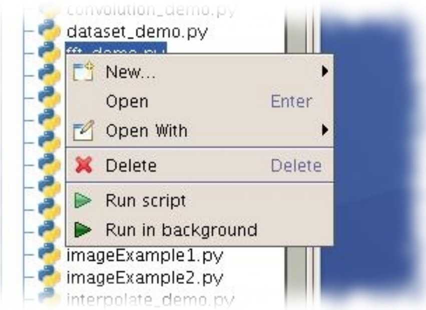
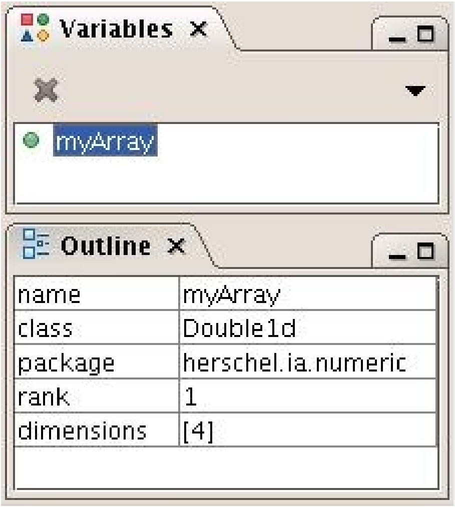


Figure 25. The Navigator view showing the options available for a Jython script.

3.3.12. Outline

This view displays information about the selected variable in the *Variables* view (see Section 3.3.19). If more than one variable is selected, information is displayed about the *first* selected variable. Common information displayed is the variable name, type (*class*) and *package* in which the variable type is defined (see for instance Figure 26).

Note that you cannot edit the contents of this view.



name	myArray
class	Double1d
package	herschel.ia.numeric
rank	1
dimensions	[4]

Figure 26. Outline of a variable in the Outline view.

3.3.13. PAL Storage Manager

In this view you can manage *pools* and *storages*, which you can use to store, query and retrieve your Herschel data. For more information see the *Data Analysis Guide*: Chapter 1.

Note that this view is deprecated. Equivalent functionality will become available in the HIPE *Preferences* window.

3.3.14. Packages

This view shows the *packages* available to the session and the *classes* within each package. (see Figure 27). When you click on an item, some additional information is displayed in the *Outline* view.

Packages and classes are advanced concepts you will only need if you want to develop Jython scripts. For more information, see Scripting and Data Mining.

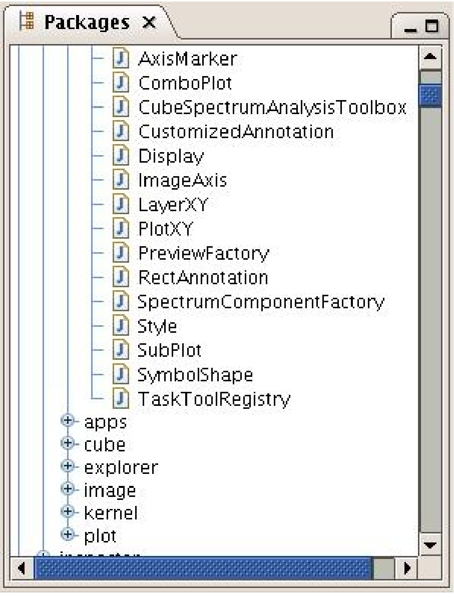


Figure 27. The Package view.

3.3.15. Product Browser

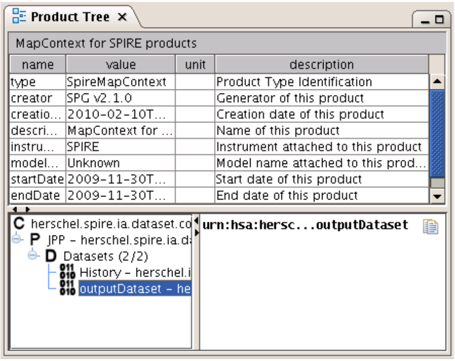
This view is the main component of the *Product Browser* perspective (see Section 3.1.1). It allows you to query and browse products stored locally or remotely.

For more information see the *Data Analysis Guide*: ????.

3.3.16. Product Tree

Like the *Product Browser* view, this view is also part of the *Product Browser* perspective.

When you select a product in the *Variables* view, the *Product Tree* view gives you additional information.



MapContext for SPIRE products

name	value	unit	description
type	SpireMapContext		Product Type Identification
creator	SPG v2.1.0		Generator of this product
creatio...	2010-02-10T...		Creation date of this product
descri...	MapContext for ...		Name of this product
instru...	SPIRE		Instrument attached to this product
model...	Unknown		Model name attached to this prod...
startDate	2009-11-30T...		Start date of this product
endDate	2009-11-30T...		End date of this product

herschel.spire.ia.dataset.co
JPP - herschel.spire.ia.d
Datasets (2/2)
History - herschel.i
outputDataset - he

urn:hsa:hersc...outputDataset

Figure 28. The Product Tree view.

The top pane shows the *metadata* associated with the product. The lower left pane shows a tree structure of all the inner products and datasets contained in the product. This is especially useful for exploring entire observations, which have an "onion" structure made of several layers. The lower right pane shows the unique identifier for any product selected in the lower left pane. This identifier is useful if you want to manage the product via the command line. Click the icon to the right of the identifier to copy the string to the clipboard.

3.3.17. Save Products to Pool

In this view you can save data (Herschel data products) from your session into a *pool*. See the *Data Analysis Guide* for examples: Section 1.3.10.

3.3.18. Tasks

This view lists all the *tasks* (data analysis routines and other tools) available in HIPE. The tasks are listed under three folders, which you can open and close with a double click:

- **All:** lists all the available tasks.
- **Applicable:** this folder is populated only when you select a variable in the *Variables* view. In this case, it lists all the tasks that can be applied to that variable.
- **By category:** this folder has four subfolders (*General*, *Hifi*, *Pacs* and *Spire*) listing general tasks and tasks relevant to a specific instrument.

Double click on a task to open its dialogue window in the *Editor* view (see Figure 21 for an example).

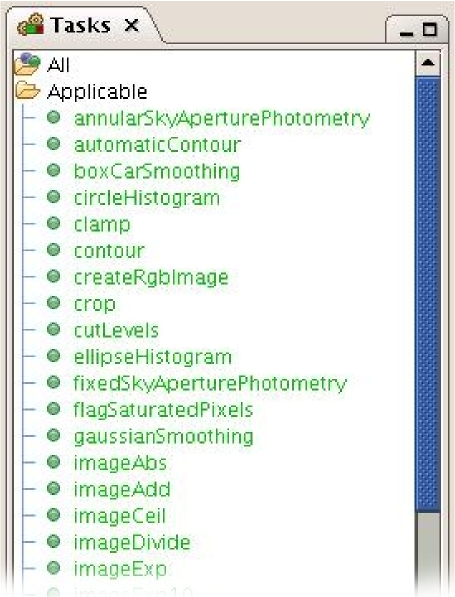


Figure 29. Tasks view with the tasks applicable to an image.

Right-click on a task to display a context menu with the following entries:

- **Open** and **Open With:** opens the dialogue window of the task.
- **View source:** displays the source code of the task. This is useful if you want to inspect the algorithm used or if you plan to modify the task. Note that, for this entry to work, you must have selected the option to include the source code when you installed HIPE.
- **Help in URM:** opens the help page for the task in the *User Reference Manual*. This entry is not available for all tasks.

3.3.19. Variables

This view shows the variables available in your session. Select a variable to obtain more information in the *Outline* view (see Section 3.3.12). Double click on a variable to display its contents with the default tool. Right click on a variable to display a context menu with the following options:

- **Open:** displays the contents of the variable with the default tool.
- **Open With:** displays a submenu with a list of tools that can be used to display the contents of the variable.
- **Send to:** displays a submenu with entries to save the variable to FITS file or to a *local store* (see the *Data Analysis Guide* for more information on local stores).
- **Show methods:** displays in the *Console* view all the *methods* (routines) applicable to the variable. This is useful if you are working on a script or issuing command-line instructions involving the variable.
- **Rename** and **Delete:** self-explanatory.
- **Help in URM:** displays the help page about the variable type from the *User Reference Manual*. This option may not be available for all variables.
- **Help in DRM:** displays the help page about the variable type from the *Developer Reference Manual*, also known as *API Documentation* or *Javadoc*. This option may not be available for all variables. Note that the DRM contains advanced, developer-oriented information. See the *Scripting and Data Mining* guide for information on how to use the DRM.

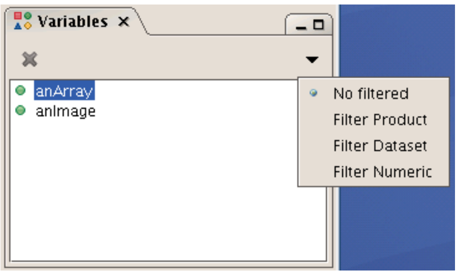


Figure 30. Variables view with filtering menu.

Click the cross below the view title to delete all variables. Click the arrow at the upper right corner of the view to display a menu to filter variables according to their type (see Figure 30).

3.3.20. Welcome

This is the initial *Welcome* window (see Figure 4).

3.4. Viewers in HIPE

A convenient feature of HIPE is that recognises the type of variables held in a session (whether it is a dataset, a spectrum, an image, a scalar constant and so on). Items appearing in the *Variables* or *Outline* views, *with a green dot to their left*, can potentially be opened. See Section 3.3.19 for how to display the contents of a variable with the default tool or with another available tool.

An example is shown for a `SimpleImage`. A right-click on a variable of this type in the *Variables* view shows there are three viewers (see Figure 31). The *Product Viewer* shows associated metadata and array values; the *Wcs explorer* shows WCS settings; the *Standard Image Viewer* displays the image.

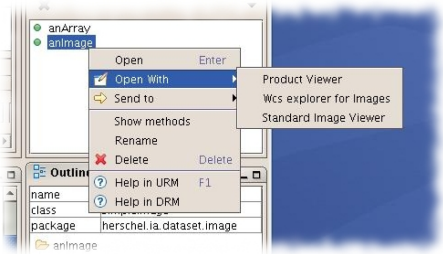


Figure 31. Available viewers are shown with a right-click.

4. IDL to HIPE/Jython command mapping

The following tables contain the HIPE/Jython equivalents of the most common IDL commands and functions.

Table 2. Basic commands

Command	IDL	HIPE/Jython equivalent
Create a variable	a = 5	a = 5
Get info on a variable type	help,a	print a.__class__
Print value of variable	print, a	print a
Create an array	a = [2., 3.]	a = Float1d([2., 3.])
Create a list	-	a = [2., 3.]
Create an automatic array	a=findgen(10)	a=Float1d.range(10)
Get info on array variable	print, a	print a

Command	IDL	HIPE/Jython equivalent
Get one element of array	print,a(1)	print a[1]
Define new 1D array of 10 elements	a = fltarr(10)	a = Float1d(10)
Assign value inside an array	a(4)=219	a[4] = 219
Define new 2D array of 10,5 elements	a = fltarr(10,5)	a = Float2d(10, 5)
First element index number	0	0

Table 3. Plotting

Command	IDL	HIPE/Jython equivalent
Open a plotting window	window,retain=2	p=PlotXY()
Plot two numeric arrays a & b	plot,a,b	p=PlotXY(a,b)
Define axis ranges and styles	plot,a,b,[xy]range=[0.,10.], [xy]title=	PlotXY(a,b,[xy]range=[0.,10.], [xy]title="λ)
Define linestyle	plot,a,b,linestyle=1	p.style.line=2 (later)
Define plotting symbol	plot,a,b,psym=2	p.style.symbol=5 (later)
Define plot title	plot,a,b,title='title'	PlotXY(a,b,title='title')
Overplot	oplot,a,c	p[1]=LayerXY(a,c)
Make Annotations	xyouts,0.2,0.7,Label'	d.addAnnotation("Label", 0.2, 0.7)
Make postscript hardcopy	set_plot,'ps' device,filename="file.ps" device,/close set_plot,'X'	p.saveAsEPS("file.ps")
Make a JPG hardcopy	-	p.saveAsJPG("file.jpg")
Make a PNG hardcopy	-	p.saveAsPNG("file.png")
Make a PDF hardcopy	-	p.saveAsPDF("file.pdf")
Further customizations	-	(right-click on plot and select Properties)

Table 4. Importing/exporting data

Command	IDL	HIPE/Jython equivalent
Reading an ascii table	readcol,'file.dat',a,b,c	t = simpleAsciiTableReader(file = "file.dat")
Plot read data	plot,a,b	p=PlotXY(t["c0"].getData(), t["c1"].getData())
		right-click on "t" and Open with "TablePlotter"
Read a Comma Separated Table (.csv) ascii table	readcol,'file.csv', DELIMITER=','	t = asciiTableReader(file = "file.csv")
Read a image FITS file	im=mrdfits("image.fits")	im = fitsReader(file = "image.fits")

Command	IDL	HIPE/Jython equivalent
Display image	tvcsl,im	right-click on "im" and Open with "ImageViewer"
Read a cube FITS file	cube=mrdfits("cube.fits")	im = fitsReader(file = "cube.fits")
Display cube	-	right-click on "cube" and Open with "CubeAnalysisToolbox"
Read a spectrum FITS file	sp=mrdfits("spec.fits")	sp = fitsReader(file = "spec.fits")
Display spectrum	plot,wave,flux	right-click on "sp" and Open with "SpectrumExplorer"
Writing to FITS	mwrfits,image,'image.fits'	simpleFitsWriter(product = image, file = "image.fits")
Writing an ascii table (csv by default)	get_lun,u openw,u,'file.csv' printf,u,a,b close,u free_lun,u	asciiTableWriter(table = t, file = "file.csv")

Table 5. Arithmetic

Command	IDL	HIPE/Jython equivalent
Adding	3+4	3+4
Multiply	3. * 4.	3. * 4.
Powers	3^4	3**4
Absolute	abs()	absolute(), fabs()
Arc cosine	acos()	arccos()
Natural logarithm	alog()	log()
10th base logarithm	alog10()	log10()
Arc sine	asin()	arcsin()
Arc tangent	atan()	arctan()
Ceil	ceil()	ceil()
Conjugate	conj()	conjutage()
Cosine	cos()	cos()
Hyperbolic cosine	cosh()	cosh()
Exponential	exp()	exp()
Floor	floor()	floor()
Invert (matrix)	invert()	Matrix (module)
Bit shift operations	ishft()	right_shift(),left_shift()
Sine	sin()	sin()
Hyperbolic sine	sinh()	sinh()
Square root	sqrt()	sqrt()
Tangent	tan()	tan()

Command	**IDL**	**HIPE/Jython equivalent**
Hyperbolic tangent	tanh()	tanh()
Random 0-1 generator	randomu()	random()
Reverse array 'a'	reverse(a)	a[::-1]
Collapse of array	total(a)	sum(a)
Number of elements	n_elements()	len(), size()
Number of parameters	n_params()	len(*args)
Extra parameters	_extra	**kwargs
Size	size()	shape(),arrayvar.type()
Pause time	wait	time.sleep()
Execute script	execute()	exec()

Thse are external resources that you may find useful:

- IDL to Python
- Jython homepage
- Python for astronomers

Note that HIPE does not include the latest Jython version, so examples from these external sites might not always work.