FASE : Future Astronomical Software Environment : How to include tools and systems into FASE Environment.

Y. Granet*, L. Paioro§, C. Surace*, B. Garilli§, P. Grosbol€, D. Tody$ and FASE Consortium

*CeSAM[3] - Laboratoire d'Astrophysique de Marseille
§INAF – IASF Milano
€European Southern Observatory
$ National Radio Astronomy Observatory / VAO

Abstract

The new OPTICON Network 9.2 (FP7) is working on a concrete prototype of shared astronomical software environment for scalable and desktop systems (FASE). A prototype of packaging system has been defined that allows to include the new softwares and tools as well as major legacy systems (e.g. AIPS, CASA, IRAF/PyRAF, Starlink and ESO Common Pipeline Library) within FASE environment has been developed and here presented.

FP7 Network

The high level requirements and the main architectural design have already been defined within the OPTICON Network 3.6 in collaboration with NRAO/USVAO, financed by the EU FP6 framework. OPTICON Network 9.2 (FP7) has been set up to make available to the community FASE prototype leading to an eventual reference implementation of the basic core system, as well as a packaging system. Several papers have been published during these years providing information on evolution and activities of this framework. [2]

FASE

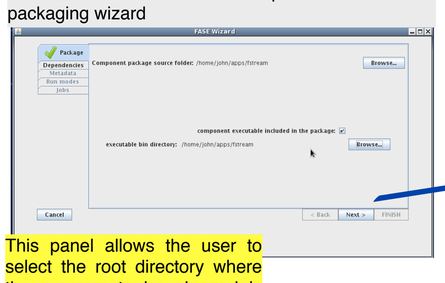
For several years the astronomical community has used different data reduction and analysis softwares.
The Future Astronomical Software Environment (FASE) project[1] aims to create a new astronomical software framework based on Virtual Observatory standards. The main concept targets are :

a) to allow the reuse of the most important legacy software (IRAF, MIDAS, AIPS, GIPSY, ...) within a modern framework;
b) to make easier the support and development of new interoperable and distributed applications or simple computational tasks;
c) to increase software sharing and astronomical software development collaboration;
d) to define stable, controlled and open software interfaces;

To reach these goals, we built up a first prototype (which aims to become a first basic reference implementation) with the purpose of demonstrating the concepts and fesability. In order to facilitate the integration of astronomical softwares within the FASE framework, we provide also a packaging system.
The core implementation is defined in python and ANSI-C. The packaging system has been provided with a Wizard coded in JAVA to be used whatever the Operating System is.

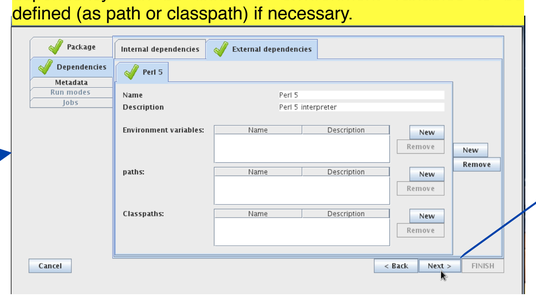
Packaging system

To package a tool, the user has to give some information. This is the initial panel of the FASE packaging wizard



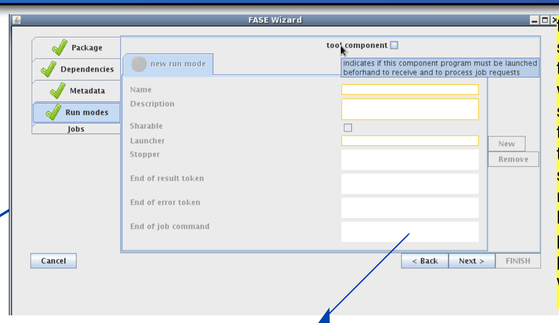
This panel allows the user to select the root directory where the program to be plugged is located

this panel lists the internal (included in the delivery) and external (installed in the target machine) dependencies of the tool. Every dependency can be linked to environment variables to be defined (as path or classpath) if necessary.

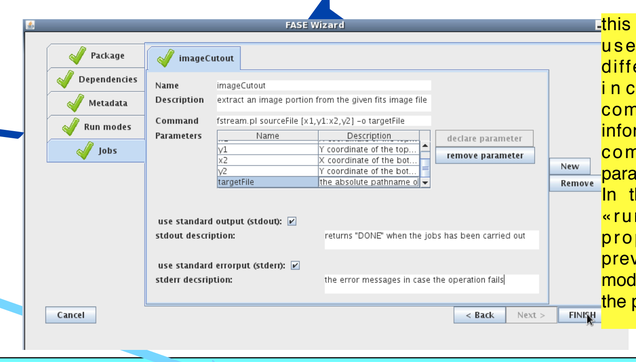


this panel is used to give information about the component itself. The concurrency check box specifies that the component can be executed in parallel.

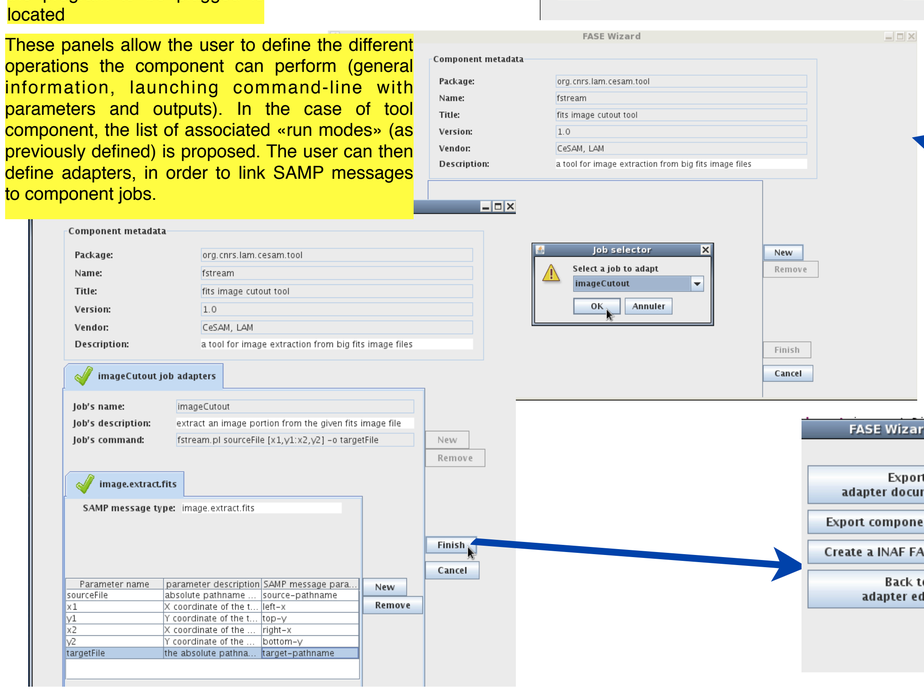
this panel allows the user to specify the nature (tool or task) of the program the user wants to integrate. The specification is described in the white paper of FASE[1]. In the case of tool the user can specify different running modes, namely ways to handle the managed sub-process of the original program as well as to deal with the multisession aspects



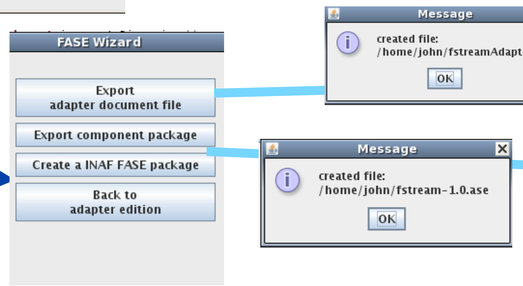
this panel allows the user to define the different operations included in the component (general information, launching command line with parameters and outputs. In the case of tool a «run mode» list is proposed from the previous defined run modes. Time to finish the parametrisation.



These panels allow the user to define the different operations the component can perform (general information, launching command-line with parameters and outputs). In the case of tool component, the list of associated «run modes» (as previously defined) is proposed. The user can then define adapters, in order to link SAMP messages to component jobs.



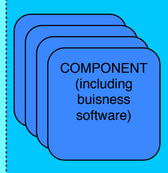
This panel allows to export the component document file (xml file) which describes the component and the way it works. The user can export as well the package precursor (which includes the distributed component) to be completed by the protocol adapter file.



Package to be loaded in the FASE Environment

```
<?xml version="1.0" encoding="UTF-8"?>
<componentAdapter>
    <jobAdapter>
        <jobName>imageCutout</jobName>
        <adapter>
            <oType>image.extract.fits</oType>
            <inputDictionary>
                <inputSynonym>
                    <key>source-pathname</key>
                    <param>sourceFile</param>
                </inputSynonym>
                <inputSynonym>
                    <key>left-x</key>
                    <param>x1</param>
                </inputSynonym>
                <inputSynonym>
                    <key>top-y</key>
                    <param>y1</param>
```

component precursor package




```
<?xml version="1.0" encoding="UTF-8"?>
<component>
    <metadata>
        <name>fstream</name>
        <description>a tool for image extraction
from big fits image files</description>
        <package>org.cnrs.lam.cesam.tool</package>
        <concurrency>true</concurrency>
        <version>1.0</version>
        <title>fits image cutout tool</title>
        <vendor>CeSAM, LAM</vendor>
        <ivoaref>fstream_0.1</ivoaref>
        <doc />
    </metadata>.............
    <env>
        <binDir>.</binDir>
        <internalDependencies />
        <externalDependencies>
            <dependency>
```

Conclusion

- During past years the FASE project has defined the concepts of the architecture for a future shared astronomical software environment, taking into account the previous and still widely used reduction and analysis legacy softwares. A packaging system prototype has been developed in order to demonstrate the concepts defined in the FASE architecture. Such packaging system prototype allows to easily plug legacy programs within the FASE framework. Many tests to plug commonly used legacy software and to develop new software using the framework facilities are undergoing
- This software is freely available on request to the authors.

References

[1] https://www.eso.org/wiki/bin/view/Opticon/WebHome
[2] https://www.eso.org/wiki/bin/view/Opticon/GeneralDocuments
[3] CeSAM : Poster 013 : this meeting

Find me or Scan and find this poster at http://lamwws.oamp.fr/PublicationDAV/ADASSXX/CeSAM_Poster_P014