



# Importing CDF files in CDPP/AMDA via SAMP

## Introduction

The new version of CDPP/AMDA, that will be publicly available this summer, will include an implementation of SAMP to allow interconnection with other clients.

The semantics of messages exchanged between SAMP clients are defined by MTypes (Message Type), outside of the recommendations of the protocol. A list of existing MTypes can be loaded in the IVOA wiki : <http://wiki.ivoa.net/twiki/bin/view/IVOA/SampMTypes>.

Most of existing MTypes are defined in astrophysical context, and are not usable in the case of plasma science.

This is why we need to define our own MTypes, and especially to give the possibility to work with CDF format via SAMP protocol.

## Definition of MTypes to load a CDF file : table.load.cdf

We defined a MType very similar to existing ones (like 'table.load.votable').

### **table.load.cdf**

Loads a table in CDF format.

- Arguments:
  - url (string) : URL of the CDF document to load
  - table-id (string) *optional* : identifier which may be used to refer to the loaded table in subsequent messages
  - name (string) *optional* : name which may be used to label the loaded table in the application GUI
- Return Values : none

For the moment, we didn't discuss within the IVOA for approval of this MType as a part of the SAMP standard.

This message has been introduced only for test in our side, and it can be redefined differently later, if necessary.

## Import CDF files in CDPP/AMDA via SAMP

AMDA is a web application written in JavaScript by using the library Sencha Ext JS. So, we used the 'SAMP Web profile' described in the version 1.3 of recommendations ( [http://www.ivoa.net/Documents/SAMP/20120411/REC-SAMP-1.3-20120411.html#tth\\_sEc5](http://www.ivoa.net/Documents/SAMP/20120411/REC-SAMP-1.3-20120411.html#tth_sEc5)).

The JavaScript library 'sampjs' (<http://astrojs.github.com/sampjs/>), developed by Mark Taylor, has been used in AMDA to implement this feature.

## 1. CDDP/AMDA connection to a SAMP hub

AMDA can't have an embedded SAMP hub, due to the fact that it's a web application. So, it's necessary to first run a tool that includes a SAMP hub.

In our case, we generally use Topcat (<http://www.star.bris.ac.uk/~mbt/topcat/>).

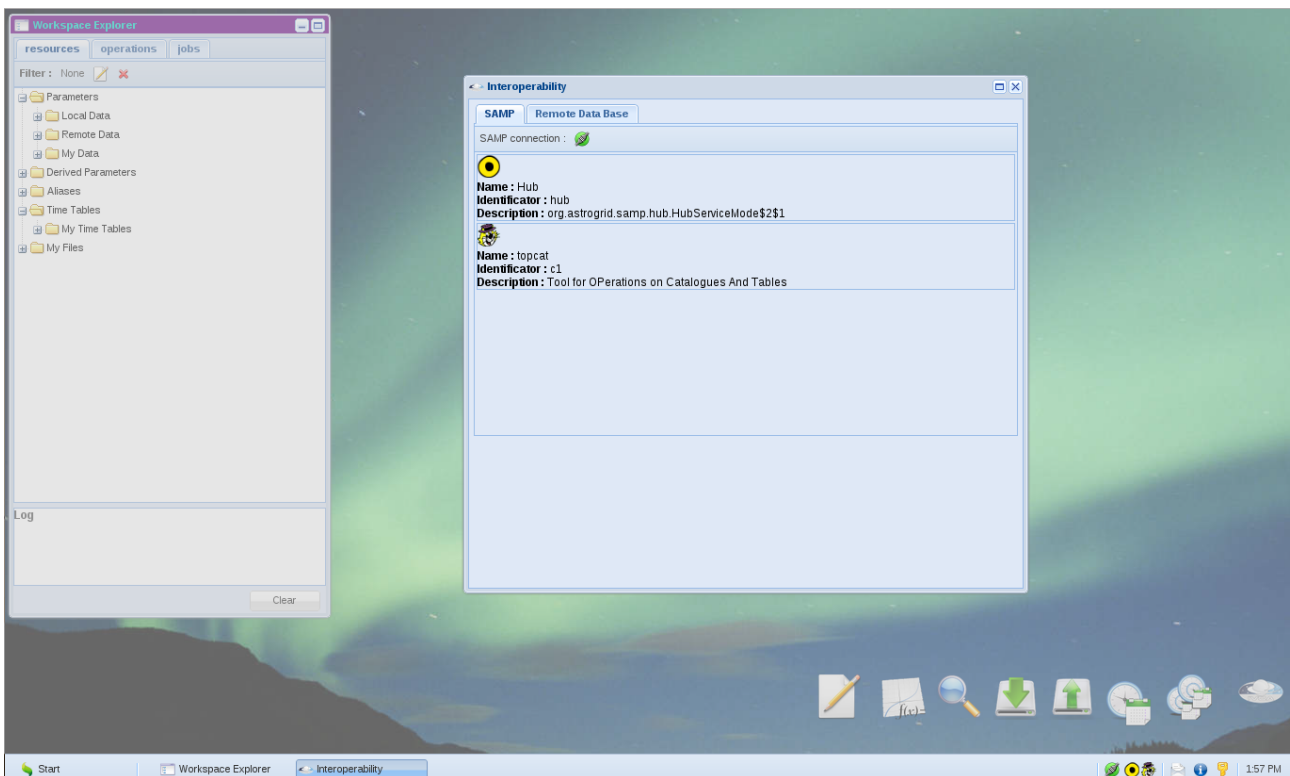


Figure 1 : CDDP/AMDA connection to a SAMP hub

## 2. Client to send CDF files

For the moment, we don't know any tool able to send a CDF file via SAMP. This is why we implement a test tool to do this work (<http://manunja.cesr.fr/~budnik/SAMPTEST/>).



Figure 2 : Test client to send CDF files via SAMP

The CDF file included in this tool comes from CAA (Cluster Active Archive, <http://caa.estec.esa.int/caa/home.xml>).

### 3. CDF File importation in CDPP/AMDA via SAMP

In CDPP/AMDA, a user has a specific assigned workspace, where he can upload her/his own data.

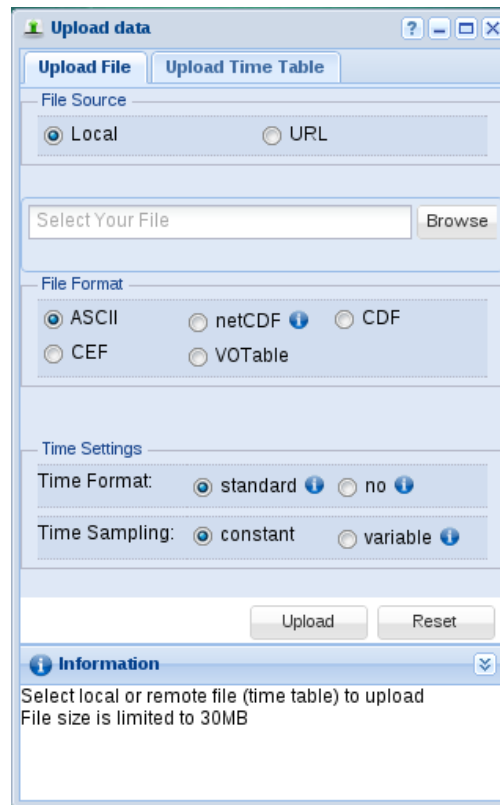


Figure 3 : Upload module in CDPP/AMDA.

When CDPP/AMDA receives a message 'table.load.cdf', the upload module is automatically opened, filled and submitted to start the file upload in the user workspace (hidden steps for the user).

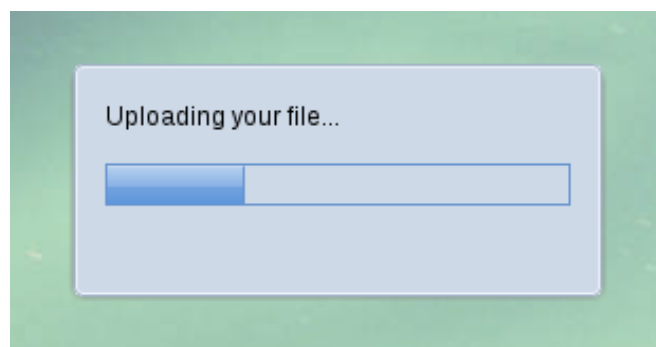


Figure 4 : CDF file is uploaded in the user workspace

The user can then select parameters with which he wants to work in CDPP/AMDA :

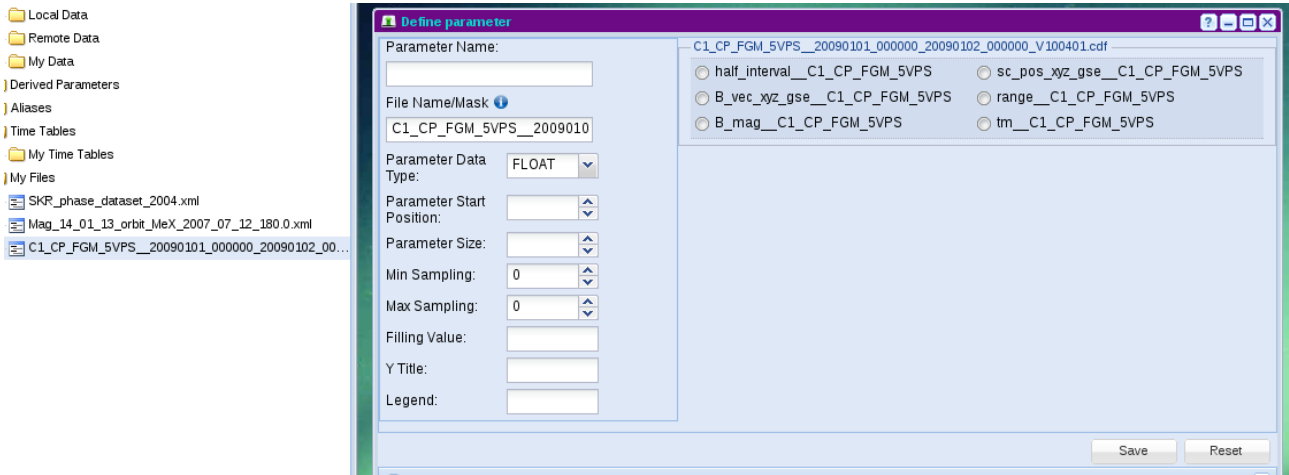


Figure 5 : Parameter selection

Now, he can use this parameter like any other parameters in AMDA, for example to do a plot:

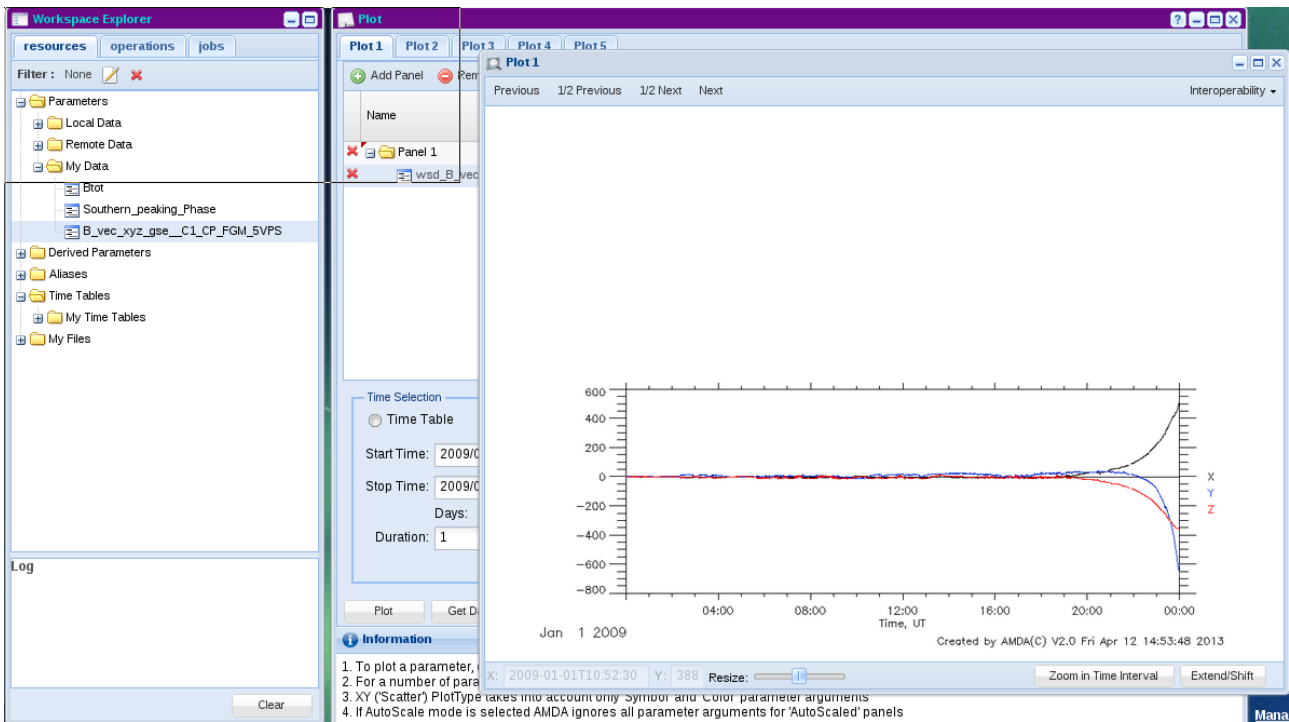


Figure 6 : Plot of the imported parameter

## Perspectives & Conclusion

Thanks to SAMP, a user can send a CDF file in his CDDP/AMDA workspace, in which he can select parameters to use.

This was made possible thanks to the extensibility of the SAMP vocabulary

For the moment, we don't know any tool that implements the possibility to send a CDF file through the SAMP protocol. But we expect that it happens.

Later, we shall also define MTypes for netCDF and CEF formats.

Finally, CDPP/AMDA may be proposed as a visualization tool for some data providers that implement SAMP.