

The HEADAS Object Oriented Parameter System (HOOPS)

James Peachey, HEASARC

HOOPS Overview

- HOOPS is a library for prompting users and returning their input to the client software (i.e. a science tool).
- Original concept was an object-oriented wrapper for ISDC's PIL (Parameter Interface Library).
- Final design is a more general interface for IRAF-style parameters which uses PIL. PIL usage is encapsulated in such a way that it could be replaced.

Main HOOPS Class Families

- IPar: Abstraction of a single IRAF-style parameter. Includes all standard parameter fields (name, type, mode, value, min, max, prompt).
- IParGroup: Abstraction of a group (an ordered container) of parameters with generic iterative access.
- IParFile: Abstraction of a group of parameters associated with a file. Includes standard file access.
- IParPrompt: Abstraction of a group of parameters associated with a particular prompting mechanism.

Advantages of HOOPS

- Separate abstractions for parameter groups, files, and prompting mechanisms makes it more flexible.
- Use of abstract interface allows other file formats and/or prompting mechanisms to be added (e.g. a GUI).
- Includes support for multiple parameter files, allowing separation of parameters which are common to a group of tools.

Sample HOOPS Pseudocode for a Simple, Ballistic Tool

```
int main(int argc, char * argv[]) {
    // Create a parameter file object.
    IParFile * pf = PILParFileFactory().Create(argv[0]);

    // Load parameters from disk into parameter file object.
    pf->Load();

    // Create an object to do the prompting.
    IParPrompt * prompter =
        PILPromptFactory().Create(pf, argc - 1, argv + 1);

    // Cause prompts to be generated as appropriate.
    prompter->Prompt();

    // Get group of parameters.
    IParGroup & pg = prompter->Group();

    // Some parameter conversions to/from primitives.
    pg["string_par"] = "This is a string parameter";
    double d = pg["double_par"];
}
```

Future Work

- Correct compilation problems on Windows.
- Provide standard code template to give science tools a uniform look and feel.
- GUIs? Using Python widgets?
- A PIL-free implementation, time permitting.