RPS2

Remote Proposal Submission 2

User's Manual

K. Peterson & J. W. Younger

Abstract

Remote Proposal Submission 2 (RPS2) is a graphical interface system designed to help an HST observer write and submit an error-free, feasible Phase II proposal that uses its orbit allocation efficiently. This manual describes how RPS2 works, how to install it and how to use it, including tips on alternate ways to configure RPS2 and make it run faster. A separate chapter is devoted to the RPS2 Proposal Editor (PED), which can help an observer write a syntactically correct proposal before he or she submits it to STScI.

This document is for RPS2 Release 8.0, which should be used for preparation of Cycle 8 observing programs.

Revision History:
December 1998 - For RPS2 Version 8.0 release in Jan. 1999
September 1999 - Revised text and layout

Space Telescope Science Institute • Project to Re-Engineer Space Telescope Observing
3700 San Martin Drive Baltimore MD 21218

printed 9/26/99 9:10 PM

1. An Overview of RPS2

1.1 What Can RPS2 Do for You?

The RPS2 system can help you write and submit a feasible Phase II observing program that will schedule on HST and make efficient use of your allotted telescope time.

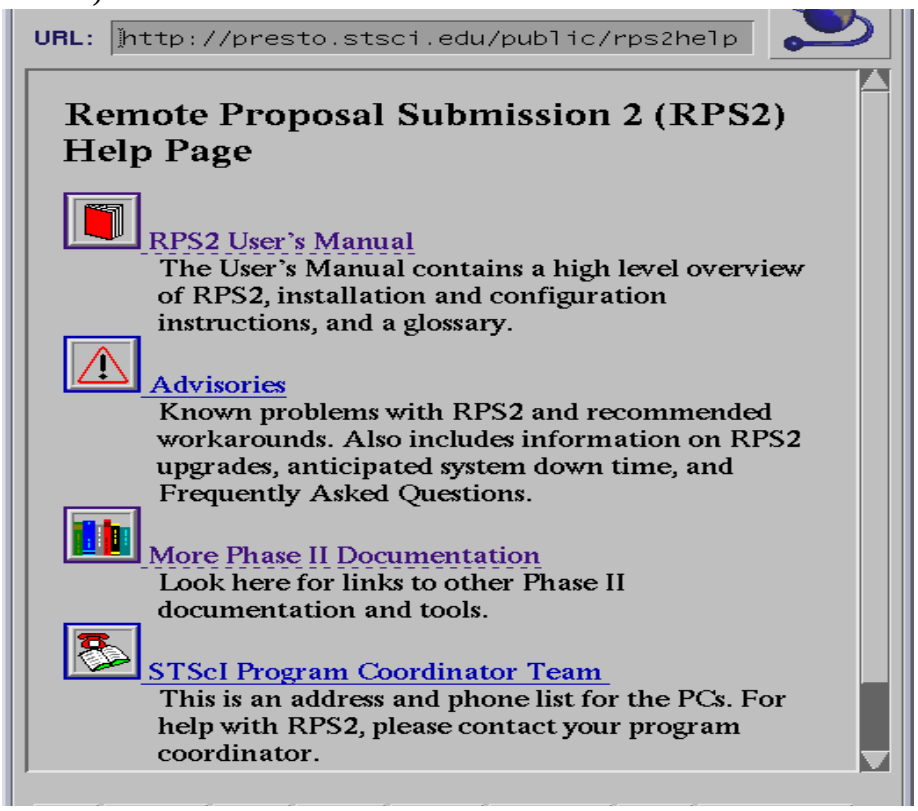
1.2 Who Can Use RPS2?

Anyone with internet access can use RPS2 in some form. Currently, RPS2 is configured to run only in unix environments (see section 2.2). However, it can be run remotely and displayed back to a graphical device that supports XWindows (e.g., VAX Workstations, Macs). For anyone who does not have access to a graphical workstation, see section 6.3.

1.3 Getting Help

If you encounter any problems while installing or running RPS2, please contact your *Program Coordinator (PC)*. The PCs maintain a *list of advisories* describing known problems and recommended workarounds. This list is available on the STScI Web page for Phase II Proposal Development, *http://www.stsci.edu/public/rps2home.html.*

You also can get this list by clicking on "*Help*" on the RPS2 User Interface (see section 4.1).



1.4 What's New?

The look and feel of version 8 is basically the same as version 7. Here is a summary of the changes and improvements:

- The scheduling subsystem behaves more like the operational software at STScI, improving the accuracy of RPS2 scheduling predictions.
- A user can create an easy-to-read formatted exposure listing from the RPS2 file.
- Schedulability cannot be run before feasibility has been run.
- The syntax checker has been overhauled and made more robust.
- Some of the pages in the Proposal Editor have been modified.

1.5 How Accurate is RPS2 and What are Its Limitations?

1.5.1 Orbit Structure

RPS2 can help you use your orbit allocation as efficiently as possible. One way you can do this is through the graphical timeline located in the structure section of the Visit Analysis Report (see section 4.5.2). Although this is a good model of what will happen during your observations, it is not an exact timeline of the observing events. If you need help optimizing your proposal, talk with your PC.

Some of the graphical timeline limitations are due to conceptual simplifications built into RPS2. One important example is "alignment" overhead. Alignments are generally groups of exposures that use the same aperture and pointing, but the detailed rules are much more complex. As a simplification, RPS2 does not display the alignment structure, so it cannot show you where the alignment overheads actually occur. In RPS2 all alignment overhead is distributed evenly among the exposures within the alignment.

Since HST is in low earth orbit, the target viewing time varies over the course of the year. Unless you specify timing special requirements, you will not be guaranteed the precise structure given in the Visit Analysis Report. Generally, this is not a problem, but it can be if timing is critical.

- For instance, after viewing the RPS2 output, you may assume that your exposures will be taken every 96 minutes for ~50 minutes. But your visit might actually be scheduled when the target is in or near the Continuous Viewing Zone (CVZ), in which case the exposures would be taken back-to-back over a shorter period of time.
- Another example is a group of exposures that appear in the same orbit in RPS2 may not end up back-to-back because the exposures may be scheduled in orbits affected by the South Atlantic Anomaly (SAA).

If exposures must be done back-to-back, you should use the SEQuence NON-INTerruptible Special Requirement. Using timing special requirements can increase overhead and decrease the scheduling flexibility of your visit, so you should use them only if scientifically justified. (See the *Phase II Proposal Instructions* for information regarding timing special requirements.)

1.5.2 Scheduling

While RPS2 has, at its heart, the same software that is used at STScI to evaluate and prepare a proposal, not all of the STScI processing capabilities are available. Therefore, there are some situations in which RPS2 will display a visit as schedulable at a particular time when it is actually not schedulable, and other situations where the converse is true.

For example, the current version of RPS2 does not have the capability to test for guide star availability. Therefore, a visit which appears to be schedulable in RPS2 may in fact not be schedulable due to lack of appropriate guide stars for the target. This can be determined only after your Phase II proposal has been submitted and processed at STScI.

2. How RPS2 Works

2.1 The Components of RPS2

The RPS2 system consists of six pieces of software held together by a seventh.

1. The User Interface (UI) is the front door to RPS2. Through it you can edit your proposal, process it and check for errors, display it and see how it schedules on the telescope, and submit it to STScI.
2. The Proposal Editor (PED) is a simple-to-use, graphical editor designed to help you write error-free proposals.
3. The Preprocessor (PP) checks for errors in your proposal and creates files used by other subsystems.
4. Transformation (Trans) takes the information generated by the PP and constructs a detailed plan for executing your proposal on HST.
5. The Constraint Analysis Spike Module (CASM) checks to see if and when your observations can be scheduled on HST.
6. The Description Generator (DG) collates the descriptions and diagnostics from the PP, Trans and CASM subsystems, and can display them as either on-screen or PostScript graphical reports.
7. The Control System (CS) handles communication among the other pieces of RPS2 and determines the order of processing.

To make RPS2 mirror actual scheduling on HST, its subsystems use much of the same software as STScI does to prepare a proposal for flight. When RPS2 processes a proposal, up to three subsystems can be used: PP, Trans, and CASM.

- The PP performs syntax checking and feeds information to Trans and CASM. *PP is called the Syntax subsystem.* (Note: The Proposal Editor uses the same syntax checking as the PP.)
- Trans takes input from the PP and calculates how much time each exposure will take (including spacecraft and instrument overhead times) and how the target visibility periods will be filled. *Trans is called the Feasibility subsystem.*
- CASM takes input from the PP (target information, requested special requirements, and so on) and Trans, and then calculates when each constraint can be met during the upcoming cycle. CASM is the heart of the Spike system, used for long range planning at STScI. *CASM is called the Schedulability subsystem.*

2.2 Possible Configurations

Precompiled binaries of RPS2 are available for several different unix platforms and configurations (see Table 2.2, "Available Configurations," on page 8). For instructions on installing one of these configurations, see section 3.

Table 2.2 Available Configurations

Computer	OS	Full Configuration	Smaller Configuration
Sun	SunOS 4.x	not available	sun4-RPS2bin.tar.Z
Sun	Solaris 2.x	solaris-RPS2bin.tar.Z	available on request
DEC Alpha	Digital Unix	not available	alpha-RPS2bin.tar.Z
SGI	IRIX	not available	available on request
HP	HP/UX	not available	available on request

The Full Configuration (*`solaris-RPS2bin.tar.Z`* for Solaris 2.x):

- This is the recommended configuration. It includes all of RPS2's subsystems and allows you to run all of RPS2 on your machine. It requires more disk space and is more computationally intensive when RPS2 is running (compared to the smaller configuration).
- After unpacking the code, it uses about 66 MB of disk space; after the archive file is deleted, it uses 47 MB. While the Feasibility and Schedulability program images are large, they can run on a Sparc IPX with 16MB of physical memory and 84MB of swap space.

The Smaller Configuration (`sun4-RPS2bin.ta` for SunOS 4.x, `alpha-RPS2bin.ta` for DEC Alpha, otherwise available on request):

- This configuration includes the User Interface (UI), PreProcessor (PP), and Description Generator) DG subsystems, all of which will run locally. It does not include the Feasibility and Schedulability subsystems, which will run remotely on the STScI machines.
- In this configuration the Control System (CS) handles communications between the local and remote subsystems via the Internet. Consequently, there will be times when heavy network traffic will cause slow processing time.
- You need approximately 17 MB of disk space in order to retrieve and unpack this configuration of RPS2. After installation is complete, the archive file can be deleted, leaving only about 713MB on your disk.

In principle you should be able to run the smaller configuration of RPS2 on any unix system to which tcl/tk/tcl-dp/itcl has been ported. The source code for the smaller configuration is available, so if you would like to try compiling it on your own system, please contact your PC, who will put you in touch with the RPS2

development team.

At this time, the RPS2 software is available only as precompiled binaries on the platforms listed in Table 1. If you do not have one of these platforms, you can run RPS2 by one of the alternative methods described in section 6.1.

2.3 Client-server Systems

RPS2 is a client-server system. The user interacts with a "client" program which in turn communicates with one or more "server" programs. A server program can be on a local machine or on any machine accessible via the Internet. In RPS2 there are four separate servers. All four servers (PP, Trans, CASM and DG) are running at STScI and can be accessed through the Internet. In the full configuration of RPS2, all servers can be run locally on your computer. In the smaller configuration, only the PP and DG are run on your machine.

If the servers are not run locally, RPS2 will automatically access the servers at STScI. In addition to slower run times and potential network connection problems, there is a potential security risk. If the local servers are not started, the `xhost` command will need to be issued to allow a server at STScI to display windows to your screen. This allows access to your machine from outside. You should check with your system manager to see if this is acceptable for your institution.

2.3.1 Mix and Match Servers

When RPS2 is started, it attempts to bring up all the servers in your installed configuration. If the servers are already up, nothing happens. The servers will stay up until they are explicitly brought down or the system on which they are running is rebooted. (This should not be a problem; servers consume minimal resources when they are not being used.) Switches are available for controlling whether the servers come up automatically (See section 4.2.2).

Servers started on one workstation can be accessed by people running RPS2 on a different workstation through the Internet. If you are running RPS2 on a different machine from the one on which RPS2 servers are running, you will have to execute the command:

```
xhost  +<servername>
```

where <servername> is the name of the machine on which the RPS2 servers are running. This allows the machine running the servers to display XWindows on your machine. As mentioned earlier, this can pose a security risk.

Using these capabilities you can mix and match servers to optimize RPS2 for your situation. For example, after installing the full configuration, you find that running all the servers on your machine slows it down considerably. You may wish to turn off the Feasibility and Schedulability servers and change the default so they do not come up when you start RPS2.

Instructions on setting these preferences is in section 4.2. If you require any assistance deciding which options are best for your system, contact your PC.

3. Installing RPS2

3.1 Installation Requirements

Installing the RPS2 software should be relatively easy, but it does require a basic understanding of Unix. If you are familiar with Unix, you should be able to install it without any special privileges and with minimal help from the system managers. If you or your system manager has problems installing or running RPS2, please contact your Program Coordinator.

Check with your system manager to determine the best place to install RPS2. This is particularly important at institutions where there are multiple HST observers; in this case the software can be installed once in a place where everyone can use it.

Note: Users at STScI do not have to install this software. It has already been installed on the unix science cluster and is accessible from /usr/local/bin/.

3.2 Retrieving the RPS2 Software

The code can be retrieved from STScI using the World Wide Web or anonymous FTP (for a list of what is available, see section 2.2).

Via the WWW:

1) Open URL: `http://www.stsci.edu/ftp/software/rps2/`
2) Click on the desired file (such as `solaris-RPS2bin.tar.Z`)
3) Type the full path (of the directory in which you want to install RPS2) and the chosen RPS2 file name in the dialog box

Via anonymous FTP:

1) Type `cd <directory in which you want to install RPS2>`
2) Type `ftp ftp.stsci.edu`
3) Type in the name "`anonymous`" as your user name
4) Give your e-mail address as your password
5) Type `cd software/rps2`
6) Type `bin`
7) Type `get solaris-RPS2bin.tar.Z` (or other desired file)
8) Type `quit`

You need to retrieve only one file; all the software elements needed to install and use RPS2 are archived together in this file.

3.3 Unpacking the Software

Once you have downloaded the file into your installation directory, examine the file

name. If the file still has the `.Z` extension, use the following command. (Remember that unix commands are case sensitive.) For the solaris configuration:

```
zcat solaris-RPS2bin.tar.Z | tar -xf -
```

If the file no longer has the `.Z` extension, your FTP or WWW client uncompressed the file for you. Use the command:

```
tar -xf solaris-RPS2bin.tar
```

When this is finished, the installation directory will contain some files (like `INSTALL` and `README`) and subdirectories (like `docs/` and `testprops`/). Read the `INSTALL` and `README` files. The `docs/` directory contains a version of the RPS2 User's Manual as well as a copy of a blank Phase II proposal template.

3.4 Installing the Software

While in the installation directory, execute the following command:

```
make install
```

This command modifies the source code to conform to its current location and sets the necessary file permissions. The "`make install`" command only needs to be run once, unless the files are moved to another location.

3.5 Testing Your Installation

A sample RPS2 proposal is available in a subdirectory of the installation directory called `testprops/`. The file name is: `2.prop` (a WFPC2 example). Copy this proposal into a new directory and follow the instructions starting with section 4.1.

Note that this sample proposal contains minor errors designed to illustrate how RPS2 diagnostics work. The `README` file in the `testprops/` directory outlines what is illustrated in the sample proposal. This proposal is merely for demonstrating how RPS2 works. Sample proposals are available on the WWW from the Phase II proposal development page, *`http://presto.stsci.edu/public/rps2home.html`* and from the Instruments Page, *http://www.stsci.edu/instruments*/

4. How to Use RPS2

4.1 Bringing up the RPS2 Graphical Interface

You should set up a directory (used for both input and output files) just for a single proposal. Since RPS2 generates a large number of intermediate files, the amount of room you will need in this directory will depend on the size of your proposal. Only very large proposals require more that one megabyte for these intermediate files.

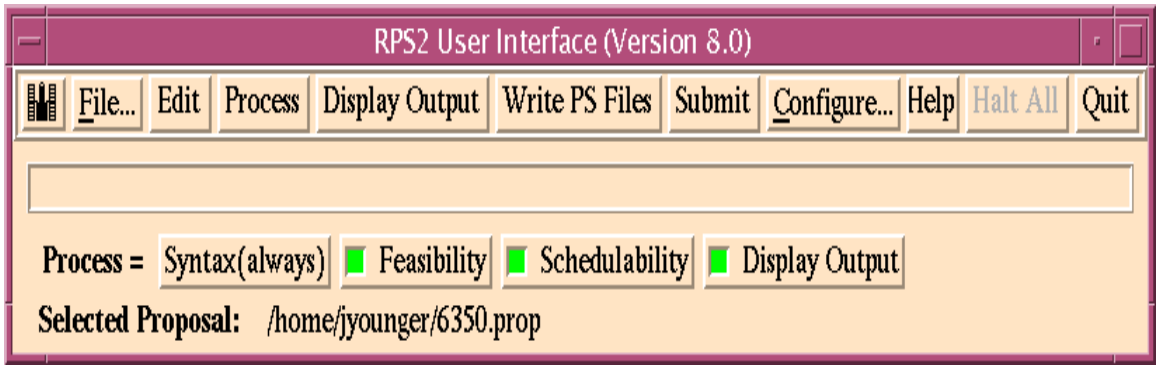
Start RPS2 by typing the following at the Unix prompt:

```
<path of installation>/RPS2 &
```

For example, if RPS2 was installed in directory /user/local/bin/rps2, type:

```
/usr/local/bin/rps2/RPS2 &
```

User Interface (UI)



At first you will see the servers start (described in section 4.2.2); then a small graphical window, shown above, will appear on your screen. This window has buttons along the top row that invoke the tools and options for RPS2. Place the mouse pointer on top of a button to see a short explanation of its function.

4.2 Configuring Your RPS2 Options

The "*Configure*" button on the UI allows you to change the default configurations of RPS2. Clicking on this button brings up a menu of options (described in the following subsections), and when you change the configuration of an option, it goes into effect immediately. When you "*Quit*" RPS2, these defaults are saved in your home directory to a file named `.rps2-init`. The next time you bring up RPS2 these options will still be in effect. Many of the options have to do with the subsystem servers. If you need to know more about the client/server nature of RPS2, see section 2.

4.2.1 Changing Server Routing

If you cannot run some (or all) of the servers on your own computer and servers

exist on a computer which is more convenient to you than the one at STScI, a preference for this computer can be specified using the "*Configure Routing*" option from the "*Configure*" menu. Click on the word "*Machine*," type in the Internet Protocol (IP) address or IP name (e.g., 130.167.107.14 or anarky.stsci.edu) of the computer desired, and click on "*Apply*."

4.2.2 Configuring the Servers and Bringing Them Up and Down

When RPS2 comes up, it will by default attempt to start all the servers in your installed configuration. If they are already up from a previous session, RPS2 will detect this and will not attempt to start new ones. This behavior can be changed using the "*Configure Servers*" option from the "*Configure*" menu. The four servers of RPS2 are listed, and each one can be configured not to start when RPS2 is started. You can also stop and restart the servers in real time from this menu. This can be useful if you want to experiment with which servers you want to run locally and which you want to run at STScI through the Internet. (Note that it takes approximately 5 minutes for the operating system to clear the server connection. So if you bring the servers down and try to bring them back up immediately, they will not come up because they already appear to be running.)

4.2.3 Changing the Default Editor

When you first bring up RPS2 and click the "*Edit*" button, you are asked to select a default editor (a text editor or PED). You can change the default editor later by selecting the "*Configure Editor*" option from the "*Configure*" menu.

4.2.4 Changing Display Default Settings

In order to set the default font size for reports created by the Description Generator, select the "*Configure Description_Generato***r**" option from the "*Configure*" menu. Next to the word "*zoom*" enter an integer between 0 and 2 (0 for 10 point, 1 for 14 point - the default, and 2 for 18 point). After you have entered 0, 1, or 2, click on "*Apply*."

4.2.5 Configuring Proposal Processing

By default RPS2 will only process only the visits in a proposal that have changed since the last time it was processed. Additionally RPS2 will only run the subsystems that have not been run yet on a version of the proposal. (So if you run syntax checking and then decide to finish processing, the syntax checking step will not run the second time.) There are no known problems with these time saving features, but as a safety measure, there is a way to turn this off. Select the "*Configure Processing*" option from the "*Configure*" menu and click on "*Entire Proposal***.**" Then click on "*Apply*" to commit the change.

4.3 Getting Started on a Phase II Proposal

General Observers (GOs) should start with the template supplied by their Program Coordinator. This template has most of the text sections already filled in from your Phase I proposal; the rest of it contains a single visit and single exposure skeleton with all the keywords and special requirements already provided.

- If you have been given a partially completed template by your PC, save the template (*be sure to delete the mail header*) to a file named <#>`.prop` where <#> is the proposal ID number that was supplied by your PC.
- If you do not have a partially completed template, you can obtain a blank template by clicking on the "*File...*" button at the top of the RPS2 window and then selecting the "*Create New File*" option in the pop-down menu.
- Then choose an editor. We recommend RPS2's proposal editor (PED), a graphical proposal writing tool with menus of options and as-you-go syntax checking. HST users will find this tool useful in preparing a syntactically correct drafts of their proposal. For more information about PED, see section 5. If necessary you also can use the regular text editors available on your system.

The appropriate *Instrument Handbooks* provide guidelines for instrument use, and the *Phase II Proposal Instructions (Version 8.0G)* provides the proper syntax to write a proposal. Before writing the entire proposal, you may wish to try out a small subset of the proposal (i.e., one visit with a few exposures) to become accustomed to proposal writing and see how RPS2 works.

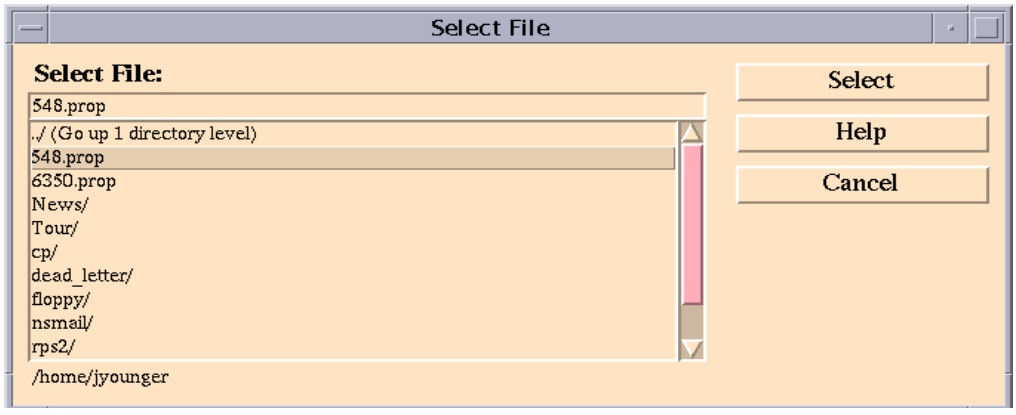
Please contact your PC with any questions about writing your proposal.

4.4 Processing a Phase II Proposal

Once a draft of the proposal has been written, RPS2 can be used to check for syntax problems, evaluate potential implementation problems, and determine if and where it can be scheduled.

If you haven't already done so, choose the "*Select Existing File*" option from the "*File...*" button at the top of the window and then use the resulting dialog box to select your proposal from your proposal directory.

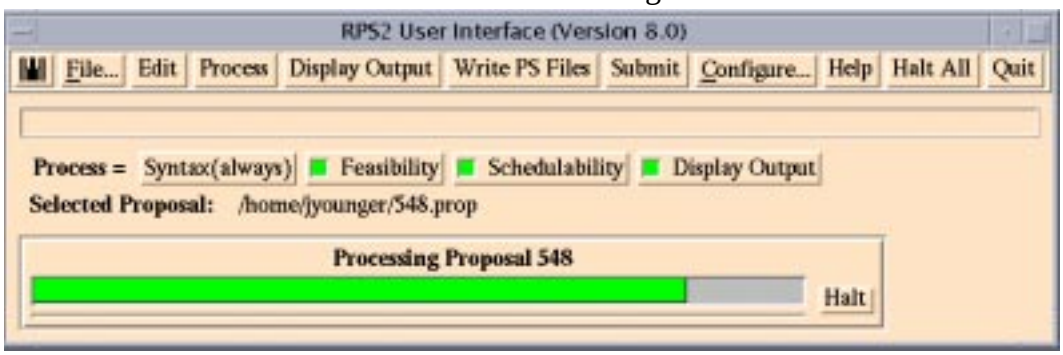
File Selection Box



Choose your .prop file and then click on "*Select.*"

Then click on "*Process*" on the User Interface (UI). In a few seconds, a progress meter will appear, appended to the bottom of the existing UI.

User Interface with the Progress Meter



A proposal with one target and one visit will generally take about 10 minutes to process fully the first time. If you find RPS2 runs too slowly, see sections 4.4.1 and 4.11 for some tips on getting it to run faster.

4.4.1 Using Only Some of the Subsystems during Processing

You can select which subsystems will run during processing by using the switches that appear on the front panel of the RPS2 User Interface. They look like:

Process = Syntax (always) [] Feasibility [] Schedulability [] Display Output

When you bring up RPS2 for the first time, all three subsystems will be selected (the button in front of them will be *green*). Click on the *green* button to turn *off* the selection of a subsystem.

- **Syntax**: This cannot be turned off. Most observers encounter syntax problems during the initial processing of their proposals. During this early stage of proposal development, you should turn Feasibility and Schedulability *off* and run only this subsystem to save time. Syntax checking is always required before Feasibility and Schedulability.
- **Feasibility and Schedulability**: These subsystems can be turned *off*. For more information on the interdependencies of these subsystems, see section 2.1. *Feasibility cannot be turned off if Schedulability is on*. This prevents a user from trying to run Schedulability before Feasibility has been run on a visit. Feasibility will be run on only those visits that have changed, even if the button is *on* (*green*).
- **Display Output**: This option assembles the raw output from the other subsystems into the on-screen reports. You probably want to review the output immediately after processing, but if you prefer to wait (or use the "**Write PS Files**" option instead) you can turn *off* the Display Output button.

4.4.2 Files Created during Processing

Processing a proposal will put many small files in your working directory. In general, you should not delete these files as they store intermediate results and the final output. Many of the files are recalculated each time you process, but if you need to view the current output again later, you can do so without reprocessing the proposal. Note that there is a target file created by the Schedulability subsystem

which is created on the first run and reused in subsequent runs (the file is simply updated as you change targets in the proposal). If you delete all the files, or move the proposal to a new directory, that target file will need to be recreated, which takes time.

4.4.3 RPS2 Description Generator

When processing is complete, a new screen will automatically appear called "RPS2 Description Generator", which is a separate interface used to display the processing results (see section 4.5). This interface also has options displayed at the top, allowing you to select a particular report or change a display format. A short explanation of an option will appear just below the top row if the cursor is placed over its button.

Note that the main progress meter on the RPS2 User Interface remains at the step "Creating Visit Analysis Reports" until you use the Description Generator. Once you click on something in the display, the progress meter will say "Activating Description Generator Buttons."

The option to have the Description Generator come up automatically after processing can be turned off by clicking *off* the *red* button next to the words "**Display Output**" on the main RPS2 panel.

4.4.4 Processing in Batch Mode

If you need to process many proposals in batch, this can be done using the command line version of RPS2 (section 6.3) and a shell script. Contact your PC if you need help. Here is an example shell script:

```
rps2-com -- process $1.prop

lpr $1*.ps
```

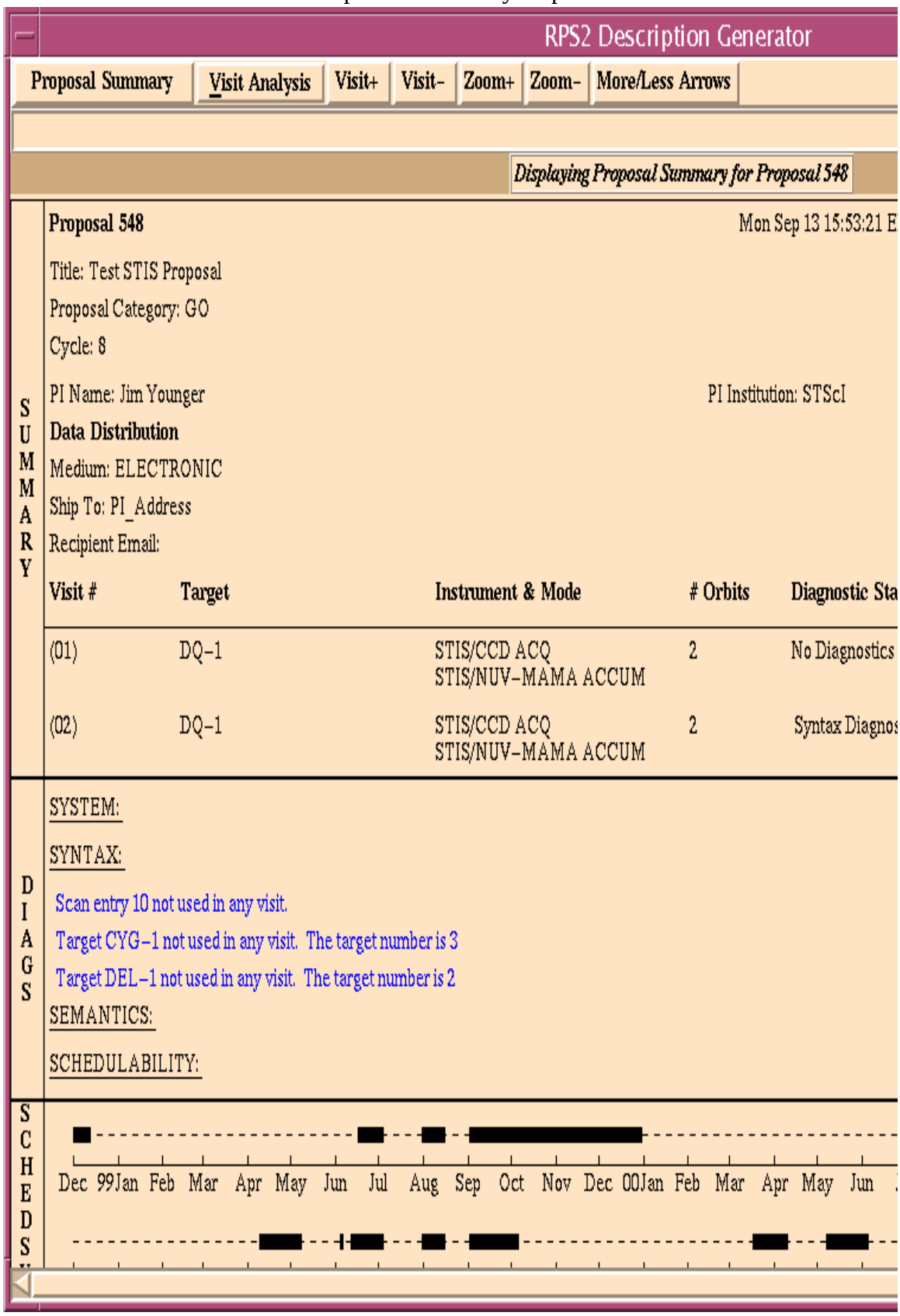
4.4.5 How to Stop Processing a Proposal

Some proposals will be so large that they will take a long time to process through all the subsystems. If a proposal has started processing and you wish to stop it (e.g., you forgot to change something), click on "*Halt*" (next to the progress meter). This aborts all subsequent processing on that proposal. If you are processing and/or displaying multiple proposals and would like to stop everything, use the "*Halt All*" button at the top of the User Interface. If the progress meter fails to go away completely after performing these steps, there probably has been a communications error. Quit RPS2 and start over again.

4.5 Reviewing the RPS2 output

The information generated during RPS2 processing is summarized in two types of graphical reports: the Proposal Summary Report and the Visit Analysis Report. These are created automatically after processing (the default) or by clicking on "*Display Output*" at the top of the window.

Proposal Summary Report



•If you need an explanation of a section of a report, click on the "*HELP*" vertical bar to the right that section in the report. Click on "*Dismiss*" at the bottom of the small window when you are finished reading the help message.

•If the font of the report is too small, click on "*Zoom+*" at the top of the window to increase the font size. The report can also be made smaller by clicking on "*Zoom-*." Zoom is a configurable option. (See section 4.2.4 for more details.)

•If the initial display of the Structure section of the Visit Analysis Report is hard to read because it is crowded with lots of labels and arrows, use the "*More/Less Arrows*" button at the top of the window to toggle on the fewer arrows option.

4.5.1 The Proposal Summary Report

Select this report by clicking on "*Proposal Summary*" at the top of the Description Generator window. It consists of the following sections (picture on page 17):

•A ***summary*** of proposal level information as well as high level information about all the visits in a proposal along with their target(s), science instrument(s) used, number of HST orbits used, and any outstanding diagnostics.

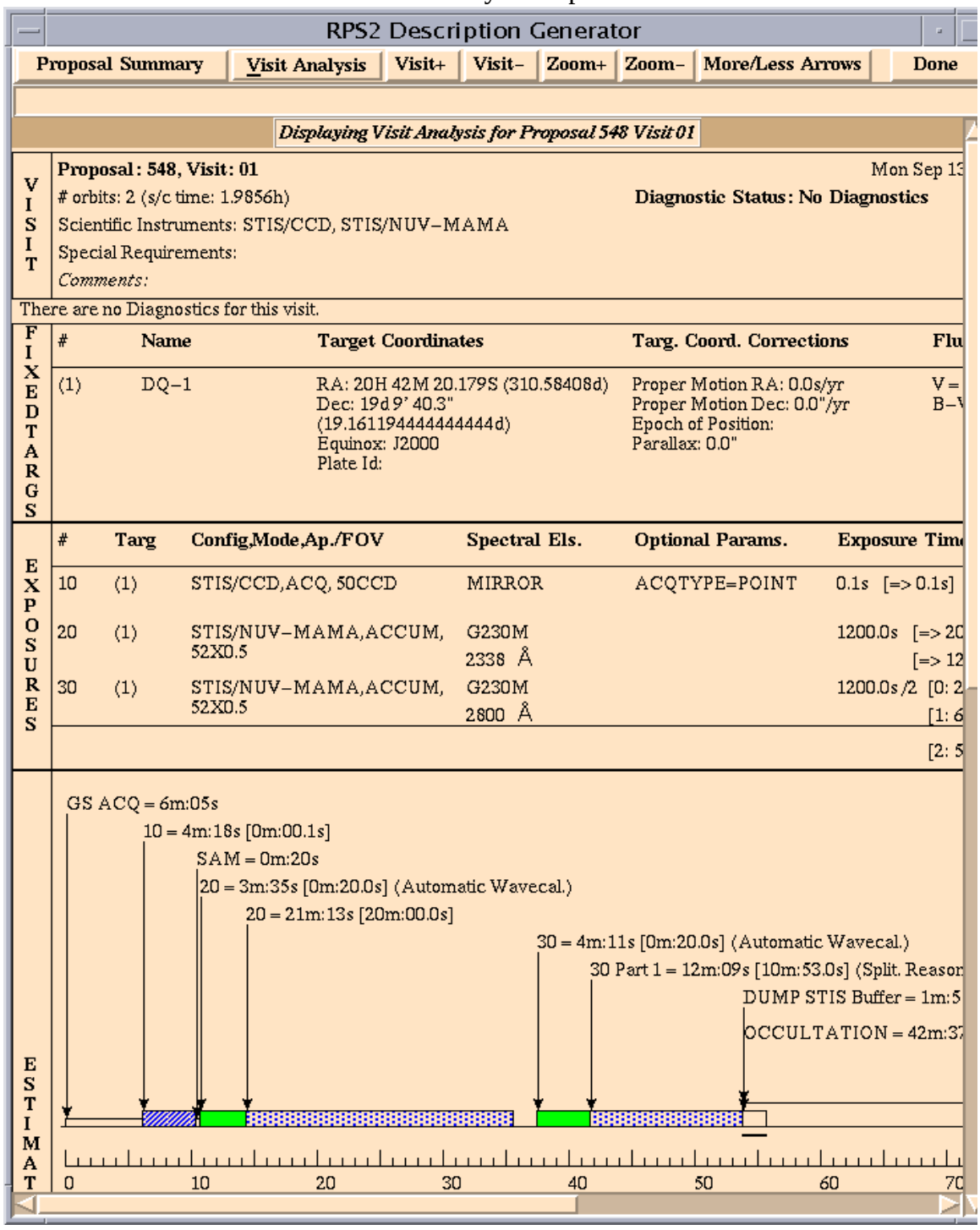
•A list of proposal-level ***diagnostics***, which affect the entire proposal, but not necessarily an individual visit.

•At the bottom of the report is a graphical summary of each visit's ***schedulability*** (identical to the one that appears at the bottom of each visit report described in section 4.5.2).

4.5.2 The Visit Analysis Report

Select a report for a particular visit by clicking on "*Visit Analysis*" at the top of the window, and then select the visit of interest in the pull-down menu. (Alternatively, you can use the "*Visit+*" and "*Visit-*" buttons to select visits in numerical order.)

Visit Analysis Report



Each Visit Analysis Report is divided up into the following sections:

4.5.2.1Visit

This section lists the number of orbits used, science instruments, visit-level special requirements, visit-level comments, and if there are any diagnostics.

4.5.2.2Diagnostics

This section summarizes all outstanding error, warning, or informational messages in the visit. They are broken down into System, Syntax, Feasibility, and Schedulability. These messages are generally short; longer, more detailed explanations are available by clicking on the message of interest. A small window will appear on the screen with both the short and long explanations. Click on "*Dismiss*" at the bottom of the small window when you are finished reading the message.

Since the output from Feasibility and Schedulability depends on correct proposal syntax, you should resolve any Syntax problems before addressing the Feasibility and Schedulability diagnostics.

4.5.2.3*Targets*

This section lists the name and any other important information (coordinates, flux, and so on) for each target used in the visit.

4.5.2.4Exposures

This section lists the exposures in the visit, reflecting what was specified for each exposure in the proposal. It also indicates the allocation of exposures to orbits and whether any exposures have been lengthened or shortened by RPS2 in order to use orbits more efficiently.

To save space on the display, the target for each exposure is referred to only by its number in the target list. If you have more than one target in the target list for that visit, you should check to make sure you are observing the right target in each exposure.

4.5.2.5*Estimated Structure*

This section is generated by the Feasibility subsystem. It is a graphical representation of the exposure section with (hopefully) helpful comments, meant to assist the observer in designing a visit so that it fits into the number of allocated orbits. There are exposure-level special requirements that allow you to specify which exposures can be lengthened or shortened and by how much. If you are iterating on how to best fill your orbits, you probably should turn off Schedulability to reduce processing time.

A key to the color and hatching scheme is available by clicking on the "*HELP*" vertical bar to the right of the section. Each orbit is represented by a 96-minute orbital timeline, and the major events of the orbit are noted separately (e.g., guide star acquisition, target acquisition, exposures, unused target visibility, and

occultation). Note that each exposure is shown with all associated overhead included. The actual on-target exposure time is displayed in brackets next to the total time.

4.5.2.6*Scheduling*

This section is generated by the Schedulability subsystem and has a graphical timeline of when the visit could be scheduled. All calculated constraints are displayed in separate bar graphs above the visit's scheduling timeline. These are physical constraints affecting target visibility (e.g., Sun, Moon) and absolute or relative constraints reflecting special requirements on timing and/or orientation.

All user-imposed constraints are plotted, but only those physical constraints which actually affect scheduling are plotted. For example, if the target is within 50 degrees of the Sun at some point during the year, the Sun constraint graph will appear and will have a dashed line during the time that the target is too close to the Sun for observation with HST. If the target is never within 50 degrees of the Sun, no Sun constraint graph will appear.

The resulting Total Scheduling Windows which are the intersection of the individual constraint plots, are plotted at the bottom. You can click on the name next to each constraint time line to see a report which contains all the schedulable date spans.

4.6 How to Make PostScript Printouts

Hard copy versions of the Proposal Summary and all Visit Analysis reports can be created as PostScript files (extension `.ps`) by clicking on "*Write PS Files*" at the top of the RPS2 User Interface window. If any of the reports contain diagnostics, a separate file, having the long explanation of each diagnostic, will be generated for each report. A file named `help.ps` is also generated; it contains all the help panels concatenated together. You can then print these files to a local PostScript printer.

4.7 How to Modify a Proposal

Changes can be made to the proposal while still in RPS2 by clicking on "*Edit*" at the top of the window. If you have chosen the proposal editor (PED) to be your editor (see section 4.2.3), the proposal will be loaded into PED and initial syntax checking will be performed automatically. Otherwise the editor will be the one selected in your EDITOR environment variable (e.g., emacs). If your EDITOR environment variable has not been set, the internal RPS2 text editor will be used. When editing is complete, just save the file and you can reprocess the proposal. You need not quit the editor.

The following unix command will set the EDITOR environment variable on most unix systems:

```
setenv EDITOR <full path of your preferred editor>
```

You may wish to add this to your shell setup file (for example in the C shell, place this command in your `.cshrc` file). If you are unsure of the editors available on

your system or where they are located, talk to your system manager.

4.8 How to Make Backup Versions

If you would like to save a copy of the current version of a proposal before making changes, you can do this from the "*Backup...*" option available from the "*File...*" menu. When you click on "*Backup...*" you will be given a sub menu with the options: "*Backup Current Version*," "*Restore Backup Version*," and "*Remove Backup.*" All options prompt you for a name for your backup. This name can be any alphanumeric text string (including periods, dashes and underscores) but should not include characters that would be unacceptable in a file name (such as blanks or slashes).

The advantage of using this method for backing up your work, is that all the cached files and other output files that have been created so far are archived with the proposal. This will save processing time when you restore the backup. You also have all the cached files and output files in your current working directory so incremental changes will take less time.

Avoid creating new directories or proposal numbers when trying new things; this will cause unnecessary reprocessing and slow your work.

4.9 Keeping Track of What You Have Done

Each proposal that you work on with RPS2 has a history file. This file is maintained across different RPS2 sessions and contains time and date stamps. To view the history file, select the "*History...*" option from the "*File...*" menu. The two choices are "*View History*" and "*Add To History.*" "*View History*" will bring up a screen with the most recent activity listed at the top. This screen can be left up if desired and will be updated as new steps are done. "*Add To History*" will allow you to add a line of free text to the history log file.

4.10 System Errors

When RPS2 detects a system error, it displays a window to the screen. The text in the box should be self-explanatory. For example if you click on the "*Submit*" option, but have not first "*Select*"ed a proposal, you will get a small window which says "No Proposal Selected." Click on "*Dismiss*" at the bottom of the small window when you are finished reading the error message.

Less straightforward error conditions may yield system messages that are hard to understand. If you encounter an cryptic error message, contact your Program Coordinator; be sure to include the text of the error message in your message to the PC. This is easy to do since each session of RPS2 keeps a log of all system errors. The `rps2-error` file can be found in the directory RPS2 was started in, and can be copied directly into an e-mail message. Along with the error message, please include the version of the proposal that had the problem and the history of your session (this is stored in the file `####.log`).

4.11 Making RPS2 Run Faster

Some large programs can take more than a half hour to process. This can be frustrating if you need to iterate many times to resolve problems. Here are some tips to make RPS2 run faster:

- **Start with a small portion of the proposal:** If you have several similar targets, visits or exposures, write and process one or two first to eliminate or reduce duplicate errors.
- **Run only the subsystems you need:** In the initial writing stages just run the Syntax checking subsystem. When you are iterating to fill your orbits, you can turn off the Schedulability subsystem (see section 4.4.1 for information on how to do this).
- **Do not move the proposal to another directory:** The first run of the Schedulability subsystem creates and saves a file containing information for each target which is reused on subsequent runs. This file takes a long time to generate. If you would like to try an experiment with the proposal, use the **backup** feature to save your current work (see section 4.8). Then you can change the proposal in your working directory and not have to regenerate the target file.
- **Use batch processing:** You can make a small shell script which runs the command line version of RPS2; then print out the PostScript reports. If you want to compare different strategies, save several versions of the proposal with different file names and process them in series overnight (see section 4.4.4).
- **Examine Syntax diagnostics files directly:** Displaying the output for proposals with many targets (like large survey proposals) can take a very long time. If all you want to do is check for the presence of Syntax errors, read the raw *.p*-diag files instead.

4.12 On-line Help Summary

When you place the cursor on one of the buttons at the top of the RPS2 User Interface, a short description of that option appears just below in one-line message box.

The "*Help*" button on the User Interface will bring up a World Wide Web client (such as Netscape) and display the RPS2 Help Page. This page (see section 1.3) contains the RPS2 User's Manual, up-to-date advisories list of known RPS2 problems, links to other Phase II information and a list of Program Coordinators.

The default World Wide Web client which the RPS2 "*Help*" button invokes can be set by typing

```
setenv RPS2_WWW <full path of your preferred WWW client>
```

at the Unix prompt before running RPS2. You may wish to add this to your shell setup file (for example in the C shell, place this command in your `.cshrc` file).

When you click on one of the Description Generator "*Help*" bars at the right margin of a section in a report, a description of that section will be displayed.

Command line users, as well as any user who prefers to view their output in PostScript form, can print out the file `help.ps` in order to have a hard copy of all the Description Generator section help panels.

4.13 Submitting Your Completed Phase II Proposal to STScI

Once you have written your Phase II proposal, processed it, examined the output products -- confirming that it is error-free and within its orbit allocation -- and are confident that the proposal will obtain the data you need, you can submit your proposal to STScI using the "*Submit*" button on the main RPS2 User Interface.

If you haven't done so already, "*Select*" the proposal file that you wish to submit and confirm that the proposal file (`<#>.prop`) uses your assigned proposal ID number.

The diagnostic files in the proposal directory from the most recent processing of the proposal will be checked before the proposal is actually submitted to STScI. If there are outstanding diagnostics, a red warning message will appear to remind you that there are still problems with the proposal. If there is a bug or limitation in RPS2 that makes it impossible for you to submit the proposal without errors, ***you must discuss this problem with your Program Coordinator before you submit the proposal.*** If the problem cannot be resolved, you can submit the proposal, but it will be flagged as having problems and your PC will tell you how to proceed.

If you submit a proposal with errors that have not been reviewed with your PC, it *will not* be accepted for processing and scheduling until these errors have been discussed with the PC.

The submission software returns an automatic electronic acknowledgment to you via e-mail that your proposal has been received. Both your Program Coordinator and your Contact Scientist will be notified of your submission, and you should receive a message from your PC regarding the acceptance of your proposal in a few days.

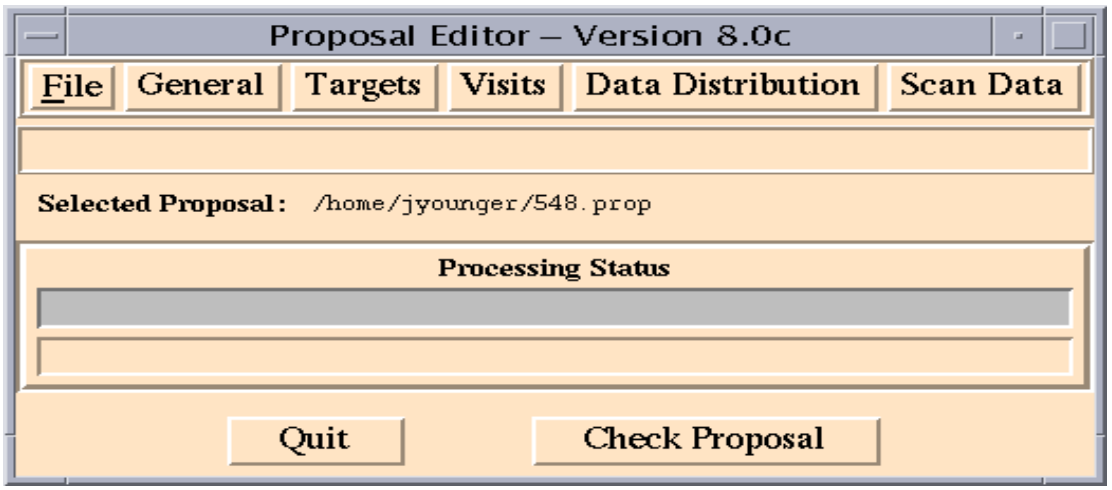
5. The Proposal Editor (PED)

5.1 Why Use PED?

5.1.1 Ease of Proposal Preparation

The **P**roposal **ED**itor ("PED") is a graphical editor meant to be used for writing and modifying Phase II proposals (`.prop` files); it is included in the Cycle 8 RPS2 software installation package. Users, both new and old, will find preparing their proposal much easier when using this editor instead of a regular text editor. Instead of typing into an ASCII text file, PED allows you to enter data quickly and easily with specially designed forms and tables, and you get immediate syntax checking and feedback on illegal entries.

To use PED, you must have a terminal or workstation that supports XWindows.



A new Phase II proposal can be created with PED, and existing proposals can be revised with PED. When proposals are saved in this editor, a new `.prop` file is written to your directory.

5.1.2 Processing Speed

PED uses the same syntax checking software as the RPS2 Preprocessor (PP); it checks for the legality of all entries you make, such as spectral elements, apertures, special requirements, and so on. Because there are so many rules and possible combinations, sometimes PED can take a few seconds to react to an entry.

If you do not use PED to edit your proposal, the PP must generate the necessary intermediate files used by other subsystems when you process your proposal.

RPS2 will run faster if the proposal is written and revised with PED, since it will not have to check the syntax or create these intermediate data files.

5.2 What's New in this Version

There are a few, relatively minor changes in Proposal Editor for this cycle.

•**Syntax Checking**: Compared to earlier versions, the syntax checking and the analysis functions are faster, and the parser has been overhauled to catch subtler errors.

•**Appearance:** The layout of some of the forms has been modified (General Proposal Information, Visit Editor, Exposure Editor, Scan Data, and Data Distribution).

5.3 Features Not Included in This Version of PED

PED does not have the following (though desired) features for Cycle 8:

•You cannot add, delete, or replace a common entry (e.g., a special requirement) to all exposures or visits. However, this can be done easily in a text editor.

• You cannot save the text entered into the .prop file as comments. These are the lines starting with an "!" (but **not** the text you enter after regular comment keywords, such as the Visit_Comments). Next time you save the proposal in PED, your lines starting with "!" will be lost.

• You cannot use the Return or Tab keys to move between entry boxes yet. You have to position the mouse in the entry box and click once.

5.4. Caveats and Advice When Switching Editors

PED works best when all proposal editing is done with it because the editor has much greater control over the entries you make. However, as mentioned in the previous section, there are several instances when it might be useful to switch to another editor temporarily.

As with all editors, caution must be exercised if you have more than one editor open on a file at one time. If changes are made in one editor, they do not automatically appear in the other. Furthermore, saving the wrong version can lead to lost data if a file is overwritten. It is strongly recommended, therefore, that you **save your proposal in PED and exit PED completely before changing to another editor** (It is not necessary to exit RPS2).

After using a regular text editor, you can read the proposal back into PED by choosing "**Edit**" from the RPS2 User Interface, assuming RPS2 is configured to use PED as your default editor (see section 4.2.3), or by typing `rps2-ped` at the unix prompt (include the full path name if necessary). If you have made any typing errors outside of PED, PED may not be able to understand what you have entered:

• If you gave a duplicate or bad *number* to a target, exposure or visit, PED will renumber it. If you gave a duplicate or bad *name* to a target, PED will rename it. PED will provide a diagnostic message, and you can rename or renumber entries in the appropriate editing screens.

• If you misspelled or misplaced any of the template keywords (PI_Name, Visit_Number, Sp_Element, etc.), PED will issue a warning and not open. You will have to go back to your regular text editor and fix the keyword(s).

• If you typed in a value for a keyword that PED cannot understand (i.e., it cannot be parsed), PED will issue a warning. You can open the appropriate PED screen to enter a correct value. However, if you do not enter a new value, PED will keep the bad value when you save the proposal.

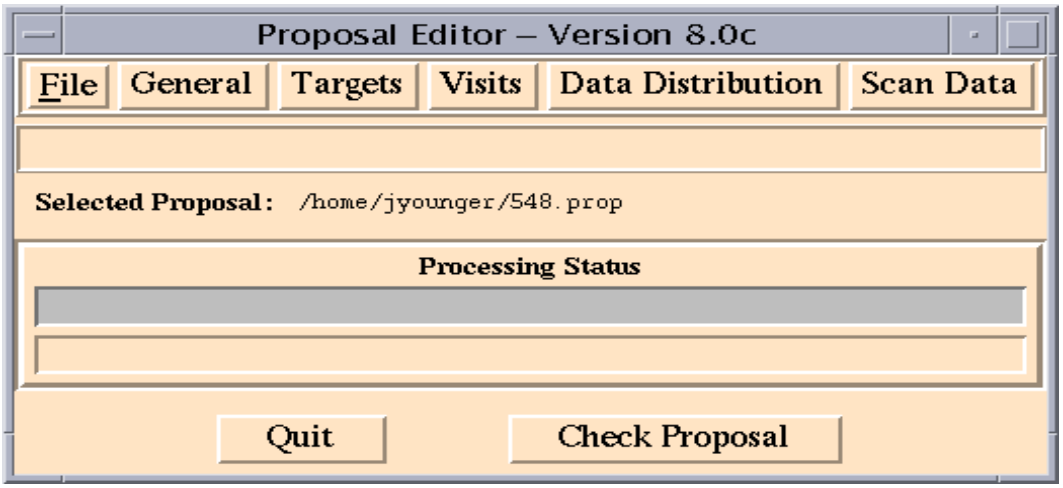
5.5 How to bring up the PED User Interface

To open PED from RPS2, click on the "**Edit**" button on the RPS2 User Interface after selecting a proposal file.

If this is the very first time you are editing a proposal through this version of RPS2, you will be prompted to select an editor: PED or your default editor (e.g., emacs or vi). You can always switch editors later by choosing the "**Configure Editor**" option under the RPS2 "**Configure**" button.

If you do not have RPS2 open, you can still run PED by typing `<path of installation>rps2-ped &` at the unix prompt.

PED User Interface



The PED User Interface has a similar appearance as the RPS2 User Interface. By default, the proposal you select in RPS2 is selected for editing in PED. Therefore, PED will immediately perform syntax checking on that proposal and provide a dialog box of any existing syntax diagnostics. This may take a minute or so to complete.

PED also will create a backup file (`#.prop-ped_backup`) before you can make any changes to the proposal.

5.6 How to Edit a Proposal with PED

The following buttons appear across the top of the PED user interface:

Except for "**File**", each of the top row buttons corresponds to a unique section of the Phase II proposal file. Therefore, clicking on one of those last 5 buttons will bring up a particular editor screen where the user can enter or modify data. A description of each button is given in the following sections: 5.6.2 , 5.6.3 , 5.6.4 , 5.6.5 , 5.6.6 and 5.6.7 .

The following buttons appear at the bottom of the PED user interface:

- **Quit**: exit out of PED
- **Check Proposal**: perform syntax checking on the entire proposal.

5.6.1 Information Regarding All PED screens

The various PED screens are all structured similarly. The following information and tips will be relevant to most of the PED screens.

1. Data can be typed directly into the highlighted **entry boxes** (*blue* on most systems). To enter data, position the mouse in the box and click once. A blinking cursor will appear. At this time, it is not possible to move between entry boxes using the Return or Tab keys; you must reposition the mouse and click once. If an entry box is darkened (*grey* on most systems), it means that it is not legal to enter data there due to other options or configurations you have chosen.
2. Any box that turns slightly darker when you position the mouse pointer over it can be clicked on to make a selection or bring up another list or data entry box.
3. Some entry screens contain "**sliders**." To enter data, you can either type directly into the entry box or move the slider. To move the slider, place the mouse over the slider button and click and drag it in the desired direction.
4. Many screens contain lists of certain items. As an example, the **Visit List** is illustrated below:

To **add** a new <item>, click the "**New <item>**" button below the list.

To **renumber** the <items>, click the "**Renumber <item>**" button below the list.

To **edit** an existing <item>, position the mouse over the appropriate <item>. The line will become highlighted. Click the left mouse button, and a pull-down menu will appear. Choose "Edit <item>". Alternatively, you can just double click on the item to edit it.

To **copy** an <item>, position the mouse over the appropriate <item>. The line will become highlighted. Click the left mouse button and choose "Copy <item> from the pull-down menu.

To **delete** an <item>, position the mouse over the <item> and click the left mouse button. Choose "Delete <item>" from the pull-down menu.

5. Most of the entry screens contain two boxes: "**Done**" and "**Check <item>**", where "**<item>**" may be flux, optional parameter, target, exposure, visit, proposal, etc. Click on "**Done**" if you are finished entering data and want to close the screen without performing an immediate syntax check. Otherwise, choose "**Check <item>**" to perform a syntax check on the data in that screen. In this way, you choose whether to check entries in individual screens as you go (e.g, flux data, exposure) or check a whole target or visit at once. If problems are found, a new window will pop up with diagnostic messages.
6. Just as with RPS2 diagnostics, the messages are generally short, but longer explanations are available by clicking on the message of interest.
7. Much of the exposure-level data (e.g., instrument configuration, operating mode) is entered by selecting items from pop-up menus. These menus list both the available and unavailable options, as detailed in the *Phase II Proposal Instructions*. You can click on any of the available options (on the left side of the menu) to add them to your proposal. If you click on any of the unavailable options (on the *grey*, right side of the menu), text will appear at the bottom of the screen explaining why that option is not available.

5.6.2 File Menu

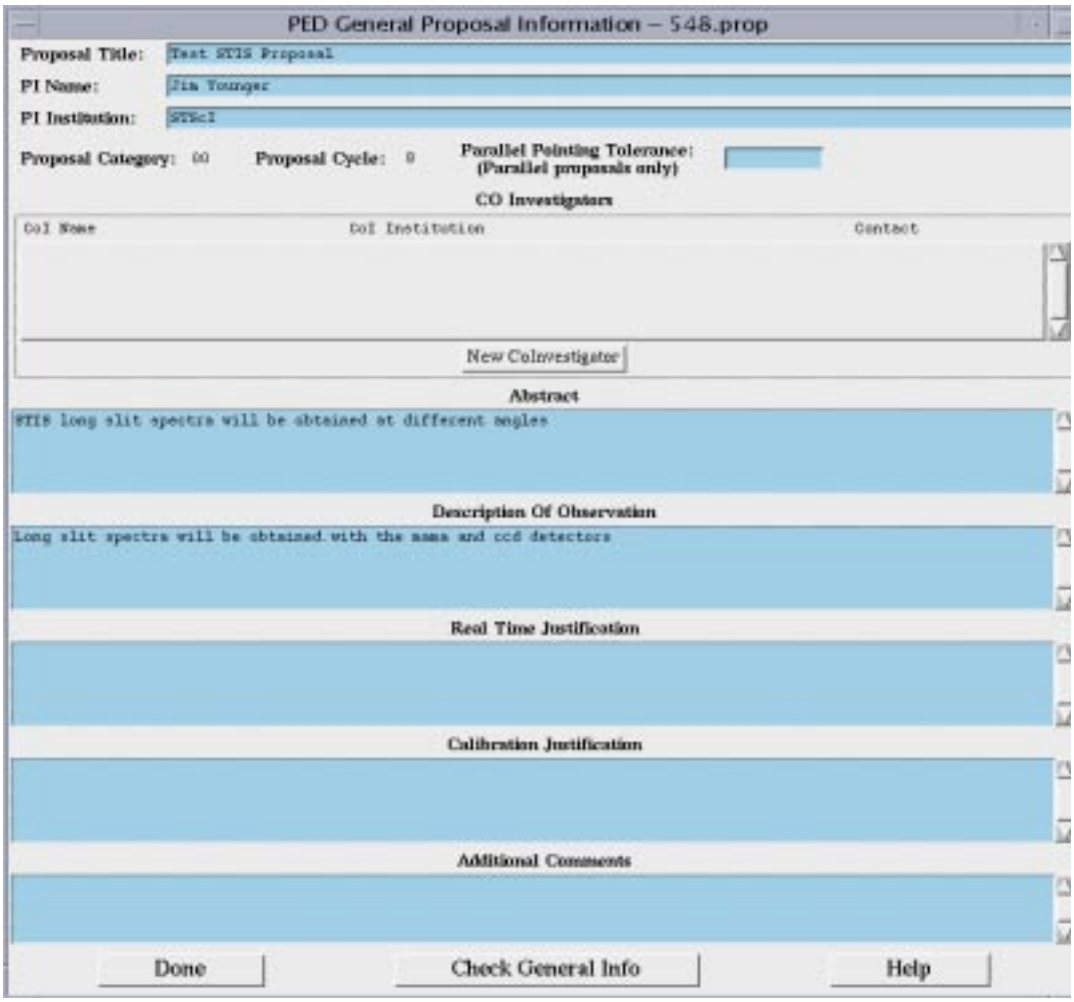
This button contains a pop-down menu (obtained by clicking on the button) of the following options:

- **New** - create a new, blank RPS2 template. You will be prompted to enter the new file name (include the directory path if necessary). The file must be of the form #.prop, where # is your HST proposal ID. (Or, it can be any number if you're creating a template for your own use).
- **Open** - open an existing .prop file
- **Save** - save (write) the selected .prop file. **You must save the proposal before running RPS2**. RPS2 will process the version which is saved on disk, not the version loaded in PED.
- **Save Quickly** - Same as the save option.
- **Qui**t - exit PED. **You do not need to exit PED to run RPS2.**

5.6.3 General Information Page

General This button brings up this PED General Proposal Information page.

Proposal Information



In this screen, you can enter/edit the following portions of your template:

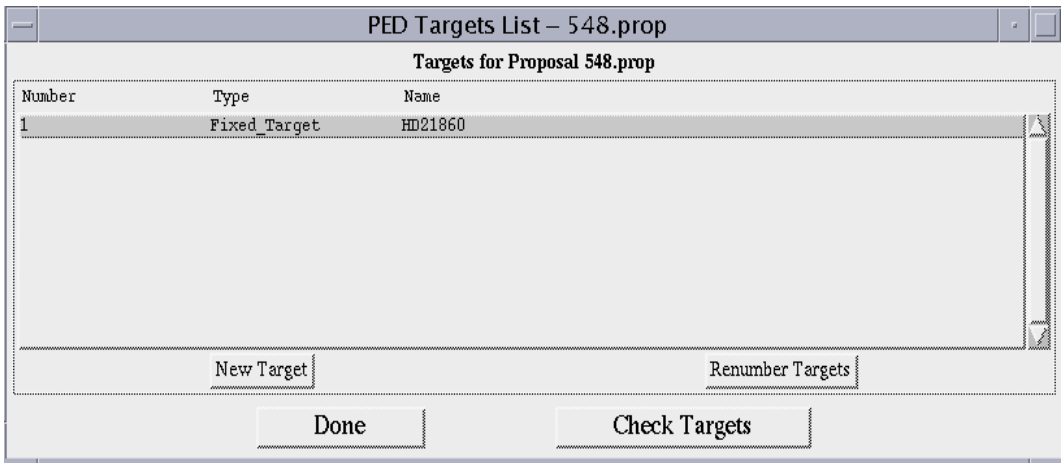
- •Proposal Title
- •Principal Investigator Name and Institution
- •Parallel Pointing Tolerance (for Parallel Proposals)
- •Co-Investigator Names and Institutions
- •Abstract
- •Description of Observation(s)
- •Real Time Justification Calibration Justification
- •Additional Comments

For GOs, this information will be already filled in from the Phase I proposal. However, GOs should review the text sections and update them as necessary.

5.6.4 Targets List

Targets This button brings up the Targets List Page

PED Targets List.



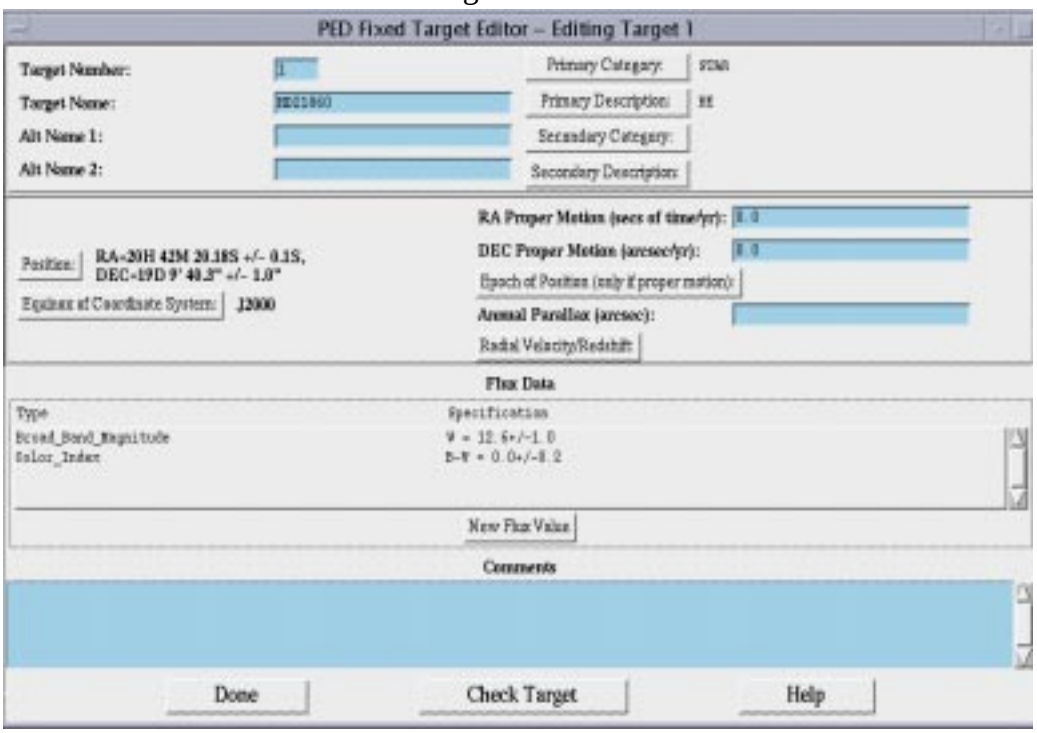
In this screen, you can enter a new target, modify, renumber or delete an existing target(s), and check the syntax for all targets. All your targets are listed here. See item 4 in section 5.6.1 for information on how to edit the Target List.

When adding a "**New Target**", a menu will appear. Chose one of the following:

- •Fixed Target
- •Moving Target (Solar System target)
- •Generic Target (Target of Opportunity or parallel target)

The appropriate Target Editor page (shown below) then will appear. When you add a new target, a default target name will be entered automatically. *Be sure to change that default name to the proper target name.*

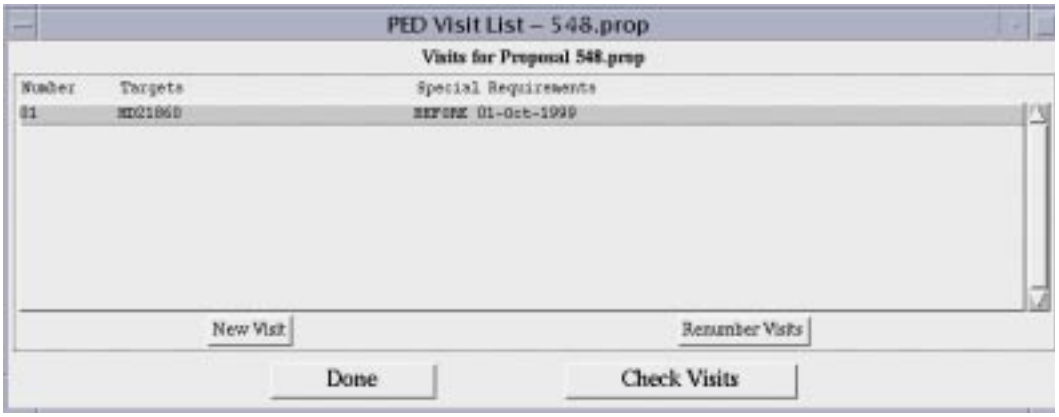
Target Editor



5.6.5 Visits List

Visits This button brings up the PED Visit List.

Visit List

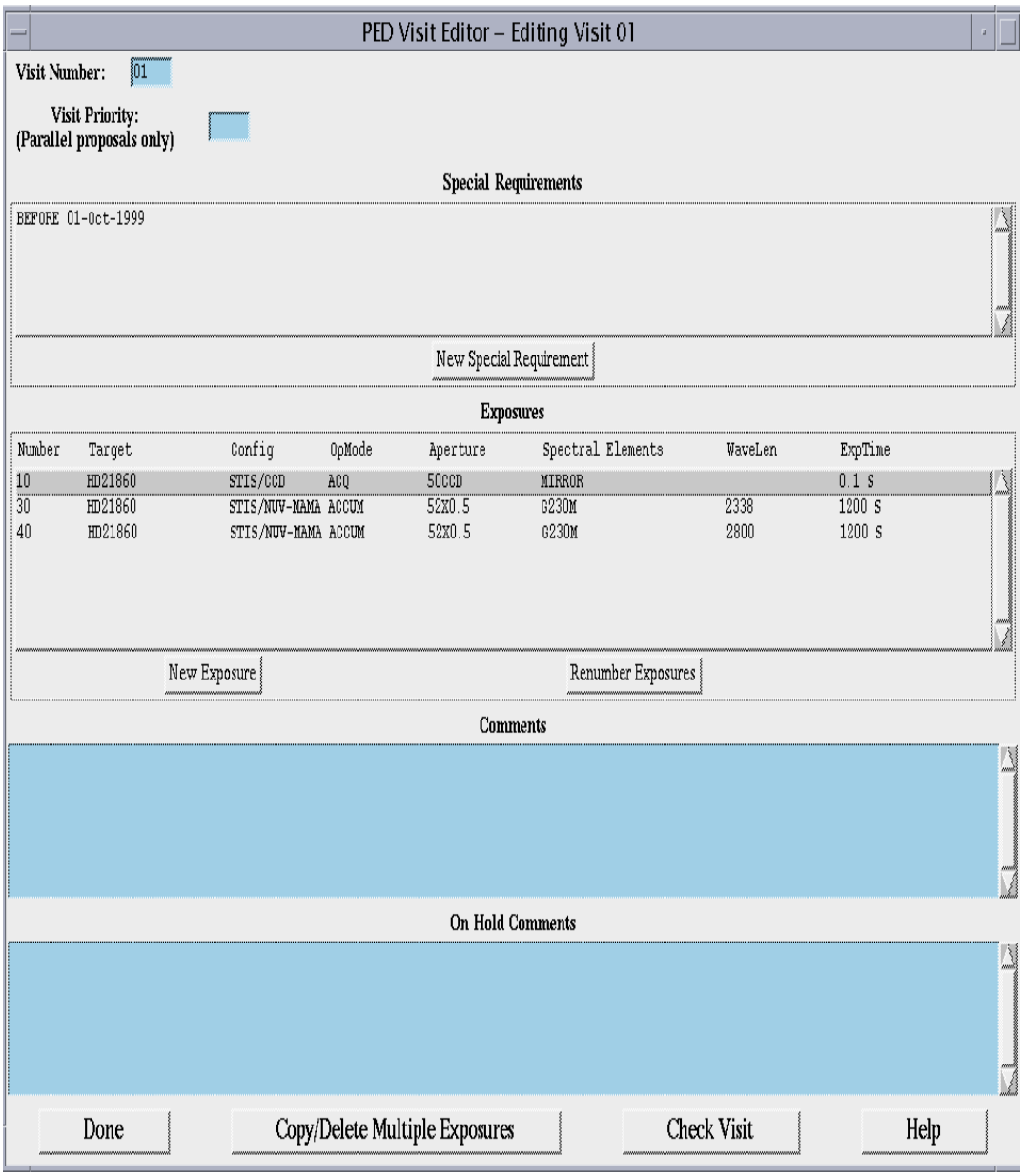


This screen contains the existing visit(s), the target(s) used in each visit, and any visit-level special requirements. From this list, you can create a new visit, modify, renumber or delete existing visits, and check the syntax for all visits. See item 4 in section 5.6.1 for information on how to edit the Visit List.

When you add or modify a visit, the PED Visit Editor is brought up on your screen (see next section).

5.6.5.1Visit Page

Visit Editor

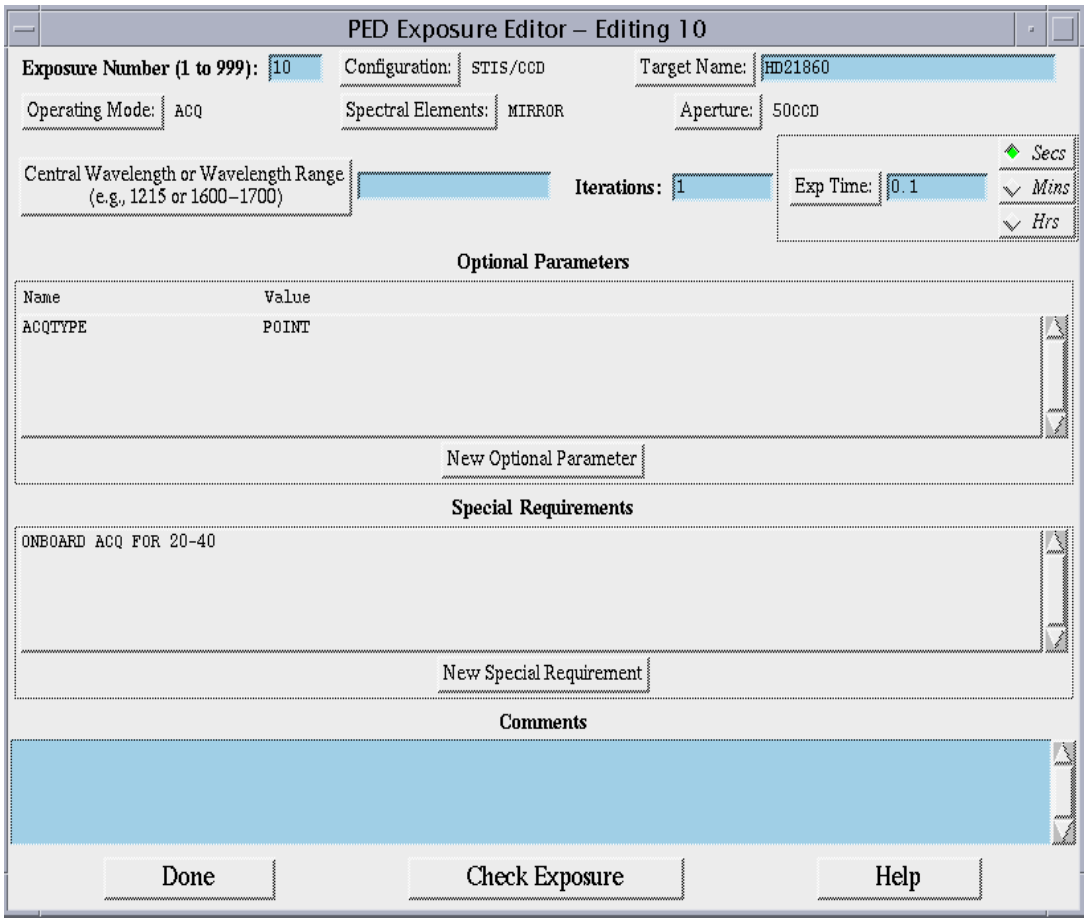


This page lists visit-level special requirements and the individual exposures contained in the visit. You can add, delete or edit the special requirements or exposures. You can also renumber the exposures. Refer to item 4 in section 5.6.1 for information on how to edit these lists. From this page, you can also copy or delete one or more exposures in the current visit. This page also contains areas to enter visit-level comments and "On Hold" comments.

When you add or modify an exposure, the PED Exposure Editor is brought up on your screen (see next section).

5.6.5.2Exposure Page:

Exposure Editor



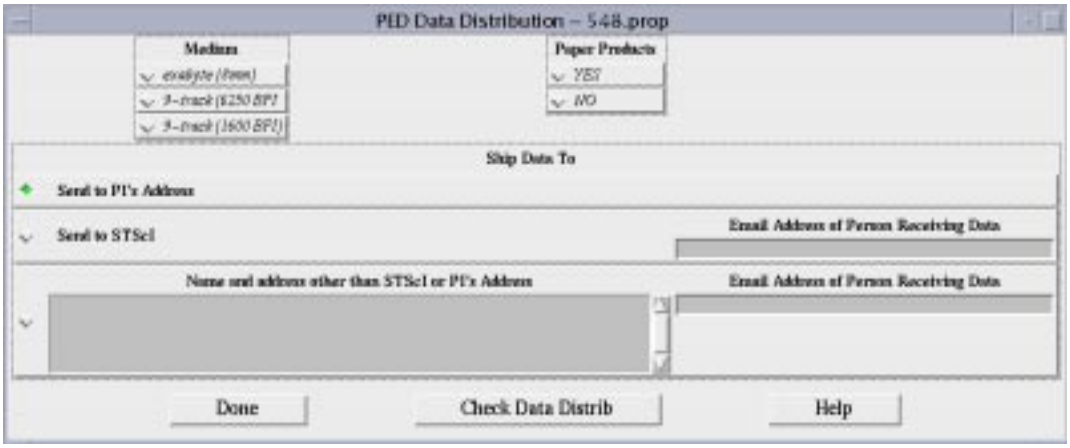
On this page, you enter all the exposure-level data such as Science Instrument, configuration, aperture, etc. This page also lists the exposure-level optional parameters and special requirements. See item 4 in section 5.6.1for information on how to edit these lists. Each exposure has its own Exposure Editor page.

From this page you can also add, modify, or delete an exposure's optional parameters and comments.

5.6.6 Data Distribution Page

This button brings up the PED Data Distribution editor.

Data Distribution



On the data distribution screen, you can enter the following information:

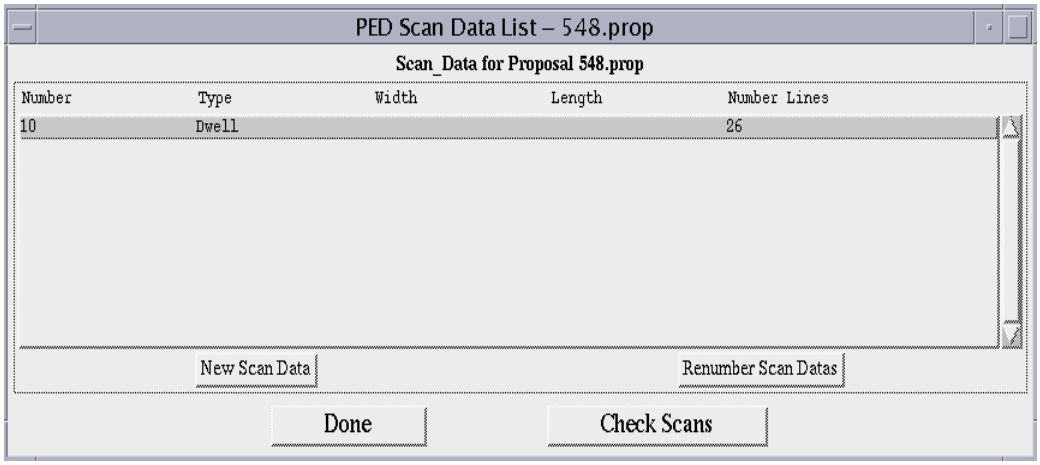
- •Data Medium [exabyte 8mm, 9-track (6250 or 1600 BPI)]
- •Where to ship the data tape, if applicable
- •Recipient's email address, if other than PI

Click on the boxes to make your selections. To check your entries, click on "**Check Data Distribution**." To close the screen, click on "**Done**."

5.6.7 Scan Data Page

Scan Data This button brings up the Scan Data List Page.

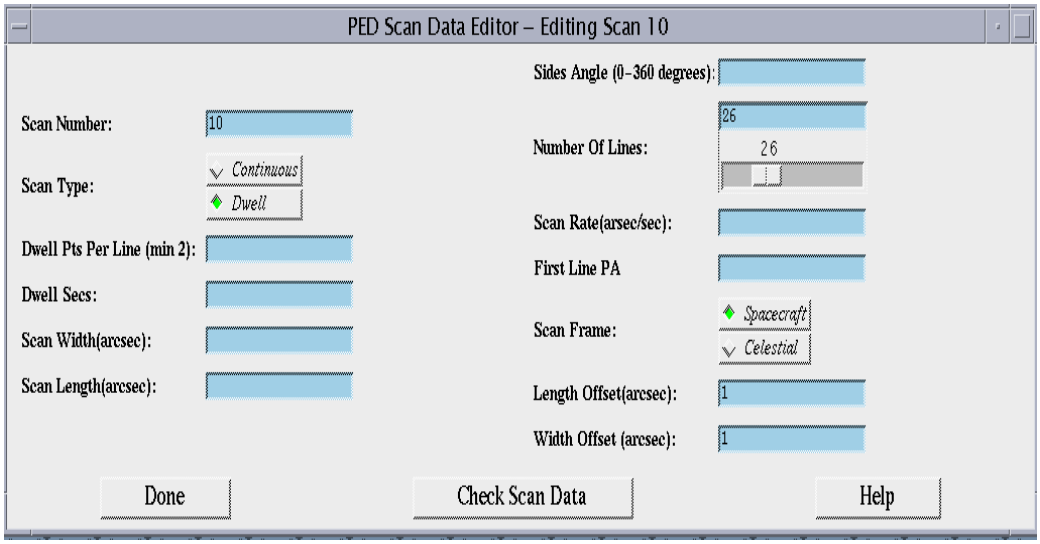
Scan Data List.



From this screen, you can add a new Scan (Dwell or Continuous), modify, renumber or delete existing Scans, and check the syntax for all Scans. Once you have Scans entered, they will all be listed on this screen. See item 4 in section 5.6.1 for information on how to edit the Scan Data List. Note that FGS Scans have been eliminated.

Here is a sample Scan Editor page:

Scan Editor



5.7 Getting Help

On-line help is available via the "**Help**" button on the RPS2 User Interface, and on the Visit Editor, Exposure Editor, and Data Distribution pages. Clicking on this button will open a World Wide Web client which will open the RPS2 Help Page at STScI. From this page, you can find up-to-date information and advisories about PED and RPS2.

If you need additional assistance, contact your *Program Coordinator (see the Program Information page)* at any time.

6. Alternate Ways to Run RPS2

6.1 Summary of Alternatives

The following table presents alternatives for users who do not have one of the supported platforms described in section 2.2.

Table 6.1

		Do you have an account on a supported platform?	
		Yes	No
Do you have an XWindows display?	Yes	Install and run RPS2 on a supported workstation and redirect display to your own graphics terminal.	Obtain a guest account from STScI and run RPS2 there, redirecting the display to your own graphics terminal.
	No	Install RPS2 on the supported workstation and run the command line version of RPS2.	Obtain a guest account from STScI and run the command line version of RPS2.

6.2 How to Run RPS2 Remotely from Your XWindows Display

Any XWindows display device with TCP/IP (the Internet Protocol) may be used to display the interactive version of RPS2. The display device does not need to be a Sun; any XWindows client will work. (Some examples are: non-Sun Unix workstations, XTerminals, personal computers running XWindow emulation software, and VAX workstations.) In all cases follow these four steps:

- Log on remotely using rlogin or telnet to the workstation on which RPS2 has been installed.
- In the local session, enable the workstation with RPS2 to write to your screen. This step is different for each system (VAXes, PCs, etc.). For example, on an XTerminal, you would type xhost +<computer running RPS2>. Your system manager may be able to help.
- In the remote session, set the Unix environment variable DISPLAY to point to your graphics display (e.g., on Unix this might be setenv DISPLAY <your computer name>:0.0).
- In the remote session, run the command <path of installation>/RPS2 &

You can skip the step of setting the DISPLAY variable by using the command line option for setting display at the time you invoke RSP2:

```
<path of installation>/RPS2 -display <host:0.0> &
```

6.3 How to Run the Command Line Version of RPS2

If you have Internet access but do not have access to a suitable graphics terminal which can display XWindows, your PC can assist you in obtaining a special account on an STScI computer. You can then run the command-line version of RPS2 (but not PED) using the following steps.

6.3.1 Transfer Proposal File

Once you have completed the first draft of your proposal, use telnet or rlogin to login remotely to your STScI account. Then use a transfer protocol (such as FTP or Kermit) o copy your proposal from your home computer to your STScI account. Remember to create a separate subdirectory for your proposal and use the file name: `<#>.prop`, where `<#>` is the proposal ID number supplied by your PC.

6.3.2 Process Proposal with RPS2

Type the following command while in the directory which contains your draft proposal:

```
rps2-com -- process <#>.prop
```

(There is a blank, two dashes, and a blank between "rps2-com" and "process.")

This command combines two of the interactive RPS2 commands: "**Process**" and "**Write PS Files**." You will see status information scroll by on your terminal screen. When processing is finished, a complete set of PostScript files will be available in the proposal directory. These files will contain the Proposal Summary Report, Visit Summary Reports for each visit, diagnostic reports for each visit, and the concatenated help file. These are the same files that would have been created had you been running the interactive version of RPS2 and used the "Write PS Files" option.

6.3.3 Retrieve RPS2 output products

In order to look at these PostScript reports, you will need to transfer all the `.ps` files back to your home machine and print them on a PostScript printer.

6.3.4 Submit proposal to STScI

Once you have worked out all the details of your proposal, use the command line version of RPS2 to submit your proposal to STScI. You may want to compose a comments file to include remarks on RPS2 with your submission. Put the completed version of the proposal (and the comments file if you wrote one) together in one directory on your STScI account. Then issue the following command:

```
rps2-com -- submit <#>.prop <comments file name>
```

(Again, there is a blank, two dashes, and a blank between "rps2-com" and "submit.")

As with the "**Submit**" option in the interactive RPS2, the submission software returns an automatic electronic acknowledgement to you via e-mail.

6.4 How to Use RPS2 from Home

If you have RPS2 installed on a system at work but want to work from a non-graphical terminal at home (or in your office), you can use the command line version of RPS2 as outlined in section 6.3. If you do not have a PostScript printer at home, you can at least review the Syntax diagnostics on-line by reading the raw *.p*-diag files. If you have not run the graphical RPS2 interface first (and therefore have not started the servers), you may want to start the servers on your work system to avoid the internet connection to servers at STScI.

6.4.1 Starting the Servers from the Command Line

To start the servers type

```
<path of installation>/start-RPS2-servers
```

This will start all server processes. This will also create log files in your current directory (each with the extension `.log`). As long as the servers are running, the RPS2 client will be able to use the local servers instead of the ones running at STScI.

6.4.2 Shutting Down the Servers

To stop the servers type

```
<path of installation>/stop-RPS2-servers
```

The command is available as a convenience and not a necessity. Normally there is no reason to shut down the servers, since they consume minimal resources when they are not being used. Remember that if the local servers are stopped, RPS2 will run, but it will be accessing the heavily loaded STScI servers generally resulting in slower response.

7. Glossary and Index

CASM (Constraint Analysis Spike Module) - This is the core of STScI's Spike constraint software configured separately to serve both RPS2 and Spike. Input files come from the PP and Trans. Output files contain information on the schedulability of each visit and are passed to the DG.

Controller - This is the brain of the Control System that determines the order in which services are run and starts them at the appropriate time.

CS (Control System) - This is the subsystem that facilitates the communication among the other pieces of the RPS2 software system. It includes the Controller, the Dispatcher, the Router, and the Server wrappers for each service. The user interacts with the Control System via the User Interface.

DG (Description Generator) - This is the subsystem that combines the output (descriptive and diagnostic) of the other RPS2 subsystems (PP, Trans, and CASM) into a set of reports that are easy to read and informative. There are two types of reports: the proposal summary and visit analysis reports.

Diagnostics - This is a generic name for the error, warning, and informational messages that the subsystems produce.

Dispatcher - Thiss is part of the Controller which communicates between the client and a server.

Orbit - This is the 96 minute HST orbit around the Earth. Do not confuse this with the "**visibility period**" which usually is the ~50 minute usable portion of an orbit where the target is not occulted by the Earth.

PC (Program Coordinator) - The primary STScI contact for the PI. The PC will help the PI throughout the observing program preparation process. He or she will also process the program for scheduling on HST after it has been submitted to STScI.

PED (Proposal Editor) - This is the graphical editor for writing and editing HST Phase II proposals.

PP (Preprocessor) - This subsystem checks the syntax (and reports diagnostics to the DG) and then converts the RPS2 file into the files needed for Trans and CASM.

Proposal Summary Report - This is one of the DG product reports that summarizes the proposal visit by visit. It includes the target name, instrument and mode, number of orbits used, and diagnostic status. It also includes proposal-level syntax diagnostics, general proposal-level information, and a scheduling summary.

Router - This is the part of the Control System that chooses which Server to use for a particular process. (If there are two computers running the CASM Server then it will choose which one to use.)

RPS2 file - This is the file that contains a Phase II proposal, hopefully in the proper

"visit based" syntax. The naming convention is `<#>.prop` where `<#>` is the proposal ID number.

RPS2 system - This is the proposal preparation software package that helps an HST observer to write and submit a Phase II proposal, which is not only syntactically correct, but also feasible, schedulable, and makes optimal use of its allocated orbits.

Server - This is another part of the Control System that serves as a "wrapper" to allow the other subsystems (e.g., Trans, and CASM) to be run remotely without modification. There will be servers for each subsystem running at STScI. The server software for a subsystem must be running on a machine in order for that machine to be used for processing by that subsystem.

Service - This is the generic name for any of the subsystems (e.g., Trans or CASM) which is run remotely by the Control System.

Spike - This is the STScI software that determines the schedulability of an observation and helps STScI develop a long range plan of observations on-board HST.

Trans - This is the STScI software (and a subsystem in RPS2) that determines the feasibility of a proposal. The files it generates are used to populate the STScI operational database, which in turn are used when preparing and scheduling visits for execution on HST.

UI (User Interface) - This subsystem allows the user to talk to the Control System. In addition to processing a proposal, the UI can be used to start the DG, start PED, and submit a proposal to STScI.

Visit - This is a group of exposures that should be observed together. If a visit has more than one target, they should be located close together. The observer using the RPS2 determines what exposures go into each visit of their proposal.

Visit Analysis Report - From the output of other RPS2 subsystems, the DG assembles an analysis report for each visit. It contains a summary of the visit -- exposure by exposure -- its Feasibility, Schedulability, and any diagnostics.