

5.0 GUI Components

This section describes the graphical user interface components in IVP Batch Builder.

5.1 The Command List

The **Command List** is the text area on the right side of the main window shown in Figure 1. It is the current list of batch input file lines being edited. The types of lines displayed in the **Command List** include the following:

- o blank lines
- o comments
- o commands
- o actions

A blank line is a line with nothing in it. Each of the other types is described in Chapter 6 of the IVP Batch Program User's Manual for Verification.

The **Command List** is a selectable list of items, each of which is one batch input line. To select a line from the **Command List**, click on that line. To select multiple lines, use the <shift> and <ctrl> keys, as is standard for most window based applications.

The label above the **Command List** is the name of the batch input file followed by the string "<modified>" if the file has been altered in anyway since its initial opening or latest save.

5.2 The Command Panels

The command panels are displayed on the left side of the main window shown in Figure 1. Each command panel provides control over a subset of the commands and actions understood by the IVP Batch Program. To select a command panel simply click on the tab with the title of the panel you desire. The panels are as follows:

1. **Location:** This panel provides control over the Fcst_Cat, Obs_Cat and Def_Loc commands.
2. **Timing:** This panel provides control over the Start_Time, End_Time, Analysis_Interval, Leadtime_Start, Leadtime_End, and Leadtime_Step commands.
3. **Group Info:** This panel provides control over the Fcst_TS, PE, RiverResponse, and Pairs_File commands.
4. **Actions:** This panel provides control over the ClearGroups, Def_Grp, Calcstats, and Natlstats actions.
5. **Pairs/Output:** This panel provides control over the Location, Pairing_Window, Observed_Table, Persistence and Build_Pairs commands for pairing and the Persistence command and Output_File action for verification in the IVP Batch Program.

Each command or action above is associated with a sub-command panel, which is a panel within the command panel that provides for the editing of a command or action. Instructions for how to use a sub-command panel to add commands or actions to the batch input file are provided in Section 6.4.

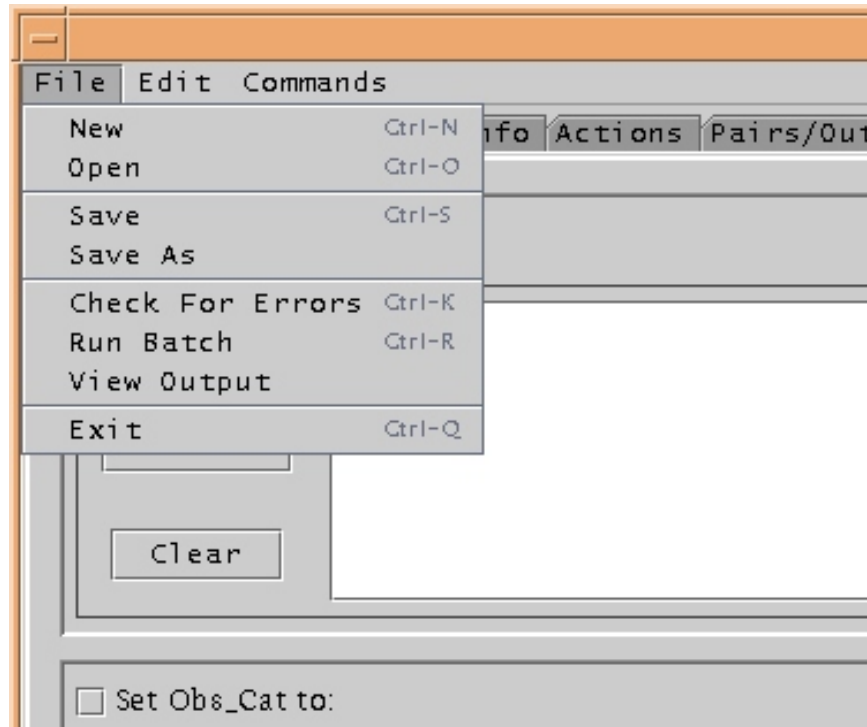


Figure 2. File Menu.

5.3 The File Menu

The **File** menu, displayed in Figure 2, contains the following menu items:

- **New**
Clicking on **New** will open an unsaved, empty **Command List** with which to work.
- **Open**
Clicking on **Open** will open a dialog which will allow the user to select an existing file to open. For instructions on how to open a file, see Section 6.1.
- **Save and Save As**
Clicking on either **Save** or **Save As** will save the currently displayed **Command List**. For instructions on how to save a file, see Section 6.2.
- **Check For Errors**
Clicking on **Check For Errors** will have the IVP Batch Builder search the **Command List** for errors. Any errors found will be stated in a text window with the title "Error Check Results". Read the errors and, when you are done, click "Okay" to close the window. If no errors are found, then a window stating "No errors or warnings were found" will pop-up. Click "OK" to close it.

NOTE: The only errors checked for are basic syntax errors, such as placing a character where a number should be. A successful error check does not imply the batch file will run error free.

- **Run Batch and View Output**

Clicking on **Run Batch** will run the current **Command List** through the IVP Batch Program. Clicking on **View Output** will open a dialog that allows opening of output files generated by the most recent run. For instructions on how to run the batch file and view its output, see Section 6.3.

- **Exit**

Clicking on **Exit** stops the IVP Batch Builder.

*NOTE: Some menu items have key combinations that can be used to select it, just as if it had been clicked. For example, to trigger an **Open**, the user can press <ctrl-o>. If such a key combination is available for a menu item, it will be displayed after the menu item title on the GUI.*

5.4 The Edit Menu

The **Edit** menu, displayed in Figure 3, contains the following menu items:

- **Undo and Redo**

Clicking on **Undo** undoes the last change made to the **Command List** that is on the "undo" stack. **Redo** redoes the last change that was undone that is on the "redo" stack. The undo and redo stacks are described below.

- **Edit**

Clicking on **Edit** will open up an edit window for the currently selected batch input line in the **Command List**. For instructions on how to edit a command, see Section 6.6.

- **Delete**

Clicking on **Delete** will remove the currently selected batch input line(s) from the **Command List**.

- **Delete ALL**

Clicking on **Delete ALL** will remove all of the batch input line(s) from the **Command List**.

- **Copy**

Clicking on **Copy** will to copy the currently selected batch input line(s) from the **Command List** for use with a future **Paste**.

- **Cut**

Clicking on **Cut** will result in a **Copy** and **Delete** for the currently selected batch input line(s) from the **Command List**. The deleted lines will then be available for use with a future **Paste**.

- **Paste**

Clicking on **Paste** will insert the most recently copied batch input line(s) into the **Command List** just before the position of the currently selected line or append the line(s) at the end, if no line is currently selected.

- **Select All and Unselect All**

Clicking on **Select All** will select all of the lines in the **Command List**, which clicking on **Unselect All** will unselect all of the lines.

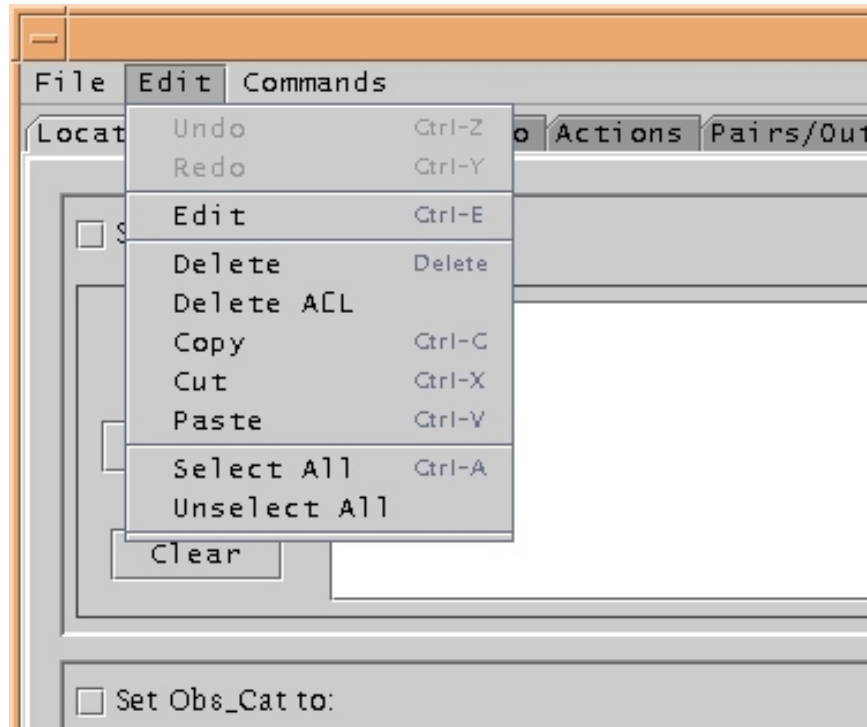


Figure 3. Edit Menu.

UNDO and REDO Stacks

Each change to the **Command List** is stored in an "undo" stack of changes. When **Undo** is clicked, it takes the change that was most recently made, which is on top of the undo stack and undoes it. That change is then pushed on to a "redo" stack. When **Redo** is clicked, it takes the most recently undone change, which is on top of the redo stack and redoes that change. That change is then pushed on to the top of the "undo" stack. Both stacks are cleared whenever a file is saved or opened.

5.5 The Commands Menu

The **Commands** menu, displayed in Figure 4, contains the following menu items:

- **Insert Commands**
Clicking on **Insert Commands** will insert the currently specified command(s) or action(s) into the **Command List** just before the currently selected line or append the command(s) or action(s) at the end if no line is selected. For instructions on how to insert commands or actions, see Section 6.4.
- **Append Commands**
Clicking on **Append Commands** will append the currently specified command(s) or action(s) at the end of the **Command List** if no line is selected. For instructions on how to append commands or

actions, see Section 6.4.

- **Insert Blank Line**
Clicking on **Insert Blank Line** will insert a blank line into the **Command List** just before the currently selected line or append it at the end if no line is selected.
- **Insert Comment**
Clicking on **Insert Comment** will insert a comment into the **Command List** just before the currently selected line or append it at the end if no line is selected. For instructions on how to insert a comment, see Section 6.6.
- **Insert User/Date**
Clicking on **Insert User/Date** will insert a comment into the **Command List** just before the currently selected line or append it at the end if no line is selected. The comment will contain information about when and who last edited the file.
- **Insert @FILE:** Clicking on **Insert @FILE** will spawn an open file dialog for the user to select a file to reference and then either insert or append an appropriately formatted line using the @FILE special token, described in Section 9 of the IVP Batch Program User's Manual for Verification and the IVP Batch Program User's Manual for Pairing.
- **Comment Out Command**
Clicking on **Comment Command** will turn the currently selected line(s) in the **Command List** into comments, adding a '#' to the beginning.
- **Uncomment Command**
Clicking on **Uncomment Command** will remove any '#' from the beginning of the currently selected line(s) in the **Command List**. However, it will not uncomment a line which, without the '#', would not be a valid (i.e., error free) blank line, command or action.

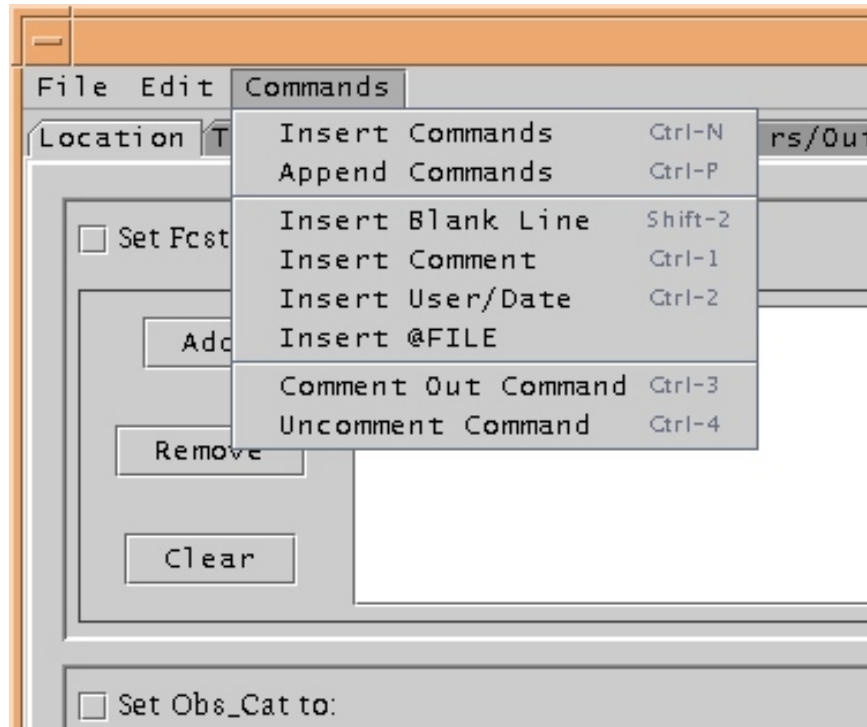


Figure 4. Commands Menu

5.6 The Buttons

Each of the buttons located in the middle of the main GUI, displayed in Figure 1, corresponds to one of the menu items described in the preceding sections. Clicking on the button will result in the exact same action as clicking on the corresponding menu item. This correspondence is as follows:

- **Insert** = **Commands >> Insert Commands**
- **Append** = **Commands >> Append Commands**
- **Delete** = **Edit >> Delete**
- **Edit** = **Edit >> Edit**
- **User/Date** = **Commands >> Insert User/Date**
- **Blank** = **Commands >> Insert Blank Line**
- **@FILE** = **Commands >> Insert @FILE**
- **Comment** = **Commands >> Insert Comment**

5.7 The Edit Window

The **Edit Window** is a window that will open whenever the **Edit >> Edit** menu item or **Edit** button is clicked and a single valid batch input line is selected from the **Command List**. An example is given in Figure 5. For instructions on how to edit a batch input line, see Section 6.6.

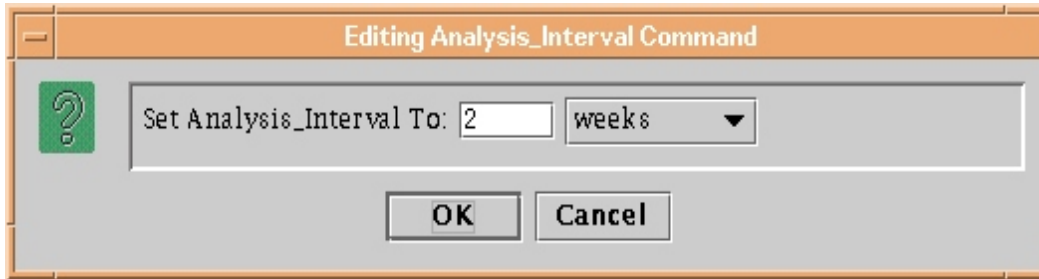


Figure 5. Example Edit Window.

5.8 The Batch Output Viewer Window

The **Batch Output Viewer** is a window that allows users to view output files generated by running the current **Command List** through IVP in batch mode. An example is given in Figure 6. For instructions on how to view the output, see Section 6.3.

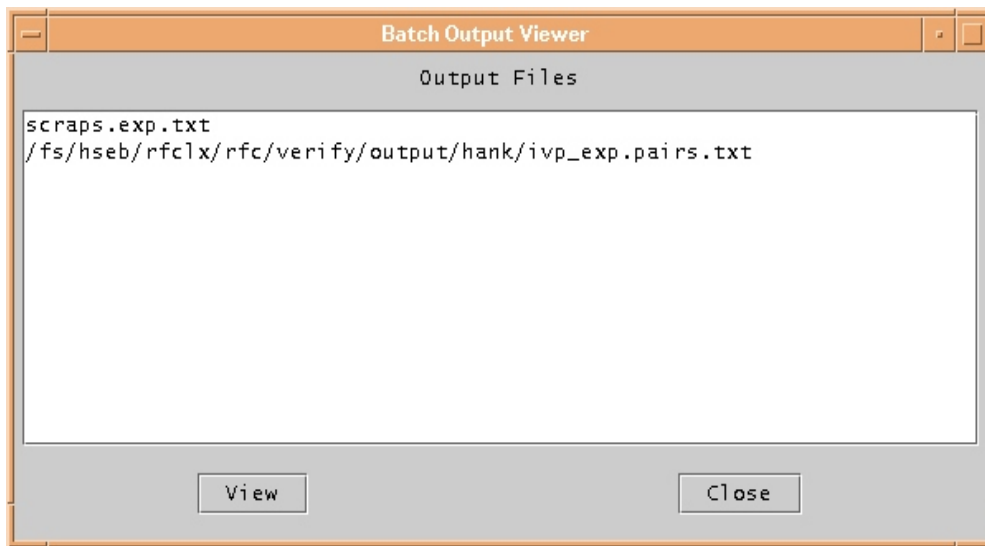


Figure 6. Batch Output Viewer Window.

5.9 The Error Window

The **Error Window** is a window that displays some error message generated by ESPADP Batch Builder. When it pops up, whatever action was taken that generated the error is aborted and the window will close once "OK" is clicked. Please read the error message before closing so that you know what caused the error and can correct it.

*NOTE: Not all error messages will appear in an **Error Window**. Some messages, typically those that will not prevent an action from completing, will be sent to stdout. However, any error that causes*

*the program to stop will be displayed in an **Error Window**.*