INTREPID batch processing (G19)

The guided tours up to now have concentrated on the interactive use of INTREPID.

INTREPID also provides powerful batch capabilities through the use of **task specification files.** Historically, these are written using the INTREPID "Parms Block Structured" language, and are identifed with the extension **`*.job`**.

Protobuf language

From version 5.0, a new language is available for batch processing. This is based on the *Google* Protobuf technology. For version 5.0, both languages are available with complete functionality.

We identify task specification files written using this language by using **`*.task`** for the file extension, rather than **`*.job`** for a Parms file.

This new language provides new capabilities to you not available with the Parms system.

1 The language is defined formally using Protobuf **`*.proto`** files. We provide these as part of the distribution in the **`api`** directory, so you can easily see which keywords are available for which process and the values they can take.

2 The technology provides better error detection, so if there is an error in a **`*.task`** file, the problem can be identified more easily.

3 For advanced users there is high-level tooling support that makes integration easier.

Both Job and Task files are simple structured text files that define the inputs, outputs, parameters and operations required for INTREPID tasks. You can edit them using any text editor. Colloquially, and throughout the documentation, we refer to both formats as "Job files".

Most INTREPID tools can save the current working parameters (Save Options) to a **`*.job`** file. Most tools can also load a set of working parameters (Load Options) from an existing task specification file. In addition, you can execute task specification files independently of the interactive tool.

In this guided tour you will use a text editor to make a copy of an existing job file, then rapidly perform the new task as a batch process in INTREPID.

Overview

- The INTREPID batch mode and task specification file system can greatly streamline your data processing work.
- You can load job files into interactive tools instead of using menus and dialog boxes to specify the task parameters.
- You can execute tasks in INTREPID batch mode using job files.
- Job files are text files and use a simple task specification language.

Creating job files

You can create a job file by

- Using the Save Options command which is available in most INTREPID tools
- Using a text editor to modify an existing job file.
- Examine the **`History`** file to recover the 'job'.
- Examining Job files provided in examples and cookbooks and altering as required.

Duplicating and editing job files

In many cases you will wish to carry out similar but not identical tasks. In this case you can duplicate a job file by

- Making a copy and editing the copy using any text editor (e.g., *Windows NotePad* or *vi* under *Linux*).
- Alternatively you can load an existing job file into an INTREPID interactive tool and use the menus and dialog boxes to adjust only the parameters you wish to change, and using Save Options.

Using job files

You can use INTREPID's powerful task specification file system in two ways:

- You can execute the task specified by one of these files in INTREPID batch mode. You can do this using Project Manager or the command line to run the Batch Process Manager.
- You can load a job file into an INTREPID interactive tool instead of manually specifying the files, parameters and operations.

Job files are particularly useful when you have repetitive 'production line' tasks. You can rerun a job file repeatedly without needing to use menus and dialog boxes.

You can further coordinate job files using scripting provided by your operating system.

Job files support some complex features which must be handled using a top-level tool provided by INTREPID. If your job files include these features, including looping and parameter substitution, use Batch Process Manager (**`fmanager.exe`**) to execute them. For GeoModeller scripting, two further tools, **`invbatch`** and **`gmbatch`**, should be used as appropriate.

Context of this guided tour

In the context of your data processing cycle, this tour describes creating a subsection of a grid dataset based on a subsectioning process performed on another grid earlier in the cycle. The new subsection will cover the same geographic region because it uses the same polygon dataset, but will use different input and output grid specifications.

Sample data

Location of sample data for Guided Tours

We provide two complete sets of sample datasets, one in INTREPID format and one in *Geosoft* format. INTREPID works equally well with both formats. When you want to open a dataset, navigate to the directory containing the required data format.

Where ***install_path*** is the path of your INTREPID installation, the project directories for the *Guided Tours* sample data are
***install_path*\sample_data\guided_tours\intrepid_datasets** and
***install_path*\sample_data\guided_tours\geosoft_datasets**.

For example, if INTREPID is installed in
C:\Intrepid\Intrepid5.nnn,
then you can find the INTREPID format sample data at
C:\Intrepid\Intrepid5.nnn\sample_data\guided_tours\intrepid_datasets

For more information about installing sample data, see "Sample datasets—installing, locating, naming" in INTREPID Guided Tours Introduction (G01)

For a more detailed description of INTREPID datasets, see Introduction to the INTREPID database (G20). For even more detail, see INTREPID database, file and data structures (R05).

Location of sample data for Cookbooks

In the adjacent folder to Guided Tours data is a collection of more exotic geophysics datasets and grids already prepared for the cookbook training sessions. You may also gain some insights into the capabilities of the software by testing the Project Manager's ability to preview and describe the attributes of each of the cookbook datasets.

For an introduction to the Cookbooks, see Using INTREPID Cookbooks (R19).

For a list of available Cookbooks, see INTREPID Cookbook.

Sample data for batch processing

Yet more sample datasets are available in ***install_path*\sample_data\datasets**. We have provided these for users to learn about batch processing with **.task** and **.job** files. For more information, see:

- INTREPID batch processing (G19) (Guided Tour)
- INTREPID task specification, HISTORY and log files (R06) (Reference)

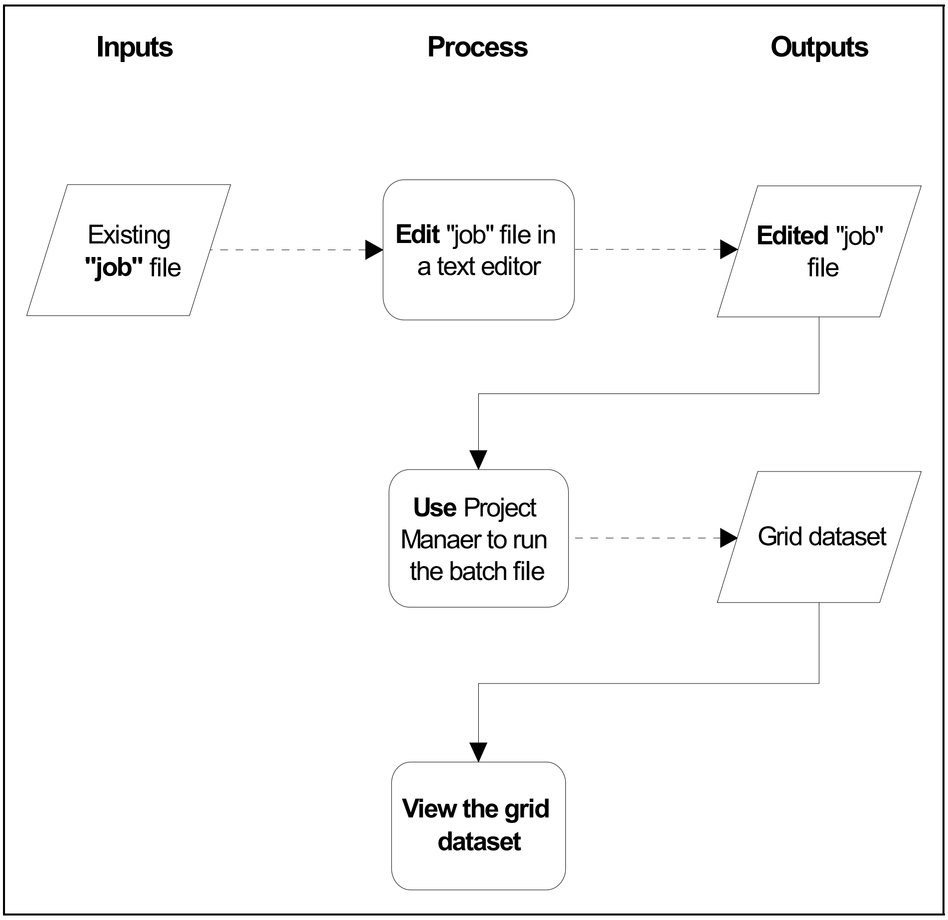
Should you complete this guided tour?

This guided tour is intended for intermediate level users. It describes an INTREPID process that requires precise editing of a text file and understanding of the concept of batch processing. If you are a beginner or wish only to have a brief overview of

INTREPID's capabilities, you can omit this guided tour. Batch processing is, however, a very important and timesaving INTREPID process. You should not omit it from a thorough evaluation of INTREPID's capabilities.

What you will do

Flowchart Summary



Steps to follow

In Managing INTREPID datasets (G06), you created a job file called **sub_anomL1.job**. This job file would perform the subsectioning task described in that guided tour.

In this guided tour you will edit the job file to modify the task, then have INTREPID carry out the new task in its batch operation mode.

The existing job file uses the polygon dataset **polyA** to create a subsection of the grid dataset **smooth_grid**. The new grid dataset created by this process is called **anomL_grid**.

The new job file creates a subsection of the same size and location using the same polygon dataset (**polyA**). It creates the subsection from the microlevelled grid dataset **mlevel_grid**. The new grid dataset created by this process is called **L_mlev_grid**.

You will use **sub_anomL.job** (a solution file we have provided which is identical to **sub_anomL1.job**).

The following table summarises the changes you will make to the existing job file

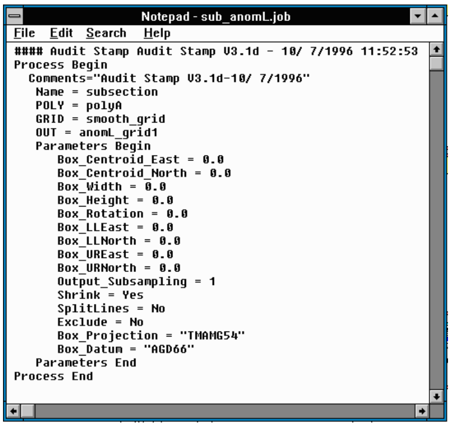
	Existing job file	New job file
Name of job file	sub_anomL.job	sub_anomLM1.job
Source grid dataset	smooth_grid	mlevel_grid
Polygon dataset	polyA	polyA
Target grid dataset	anomL_grid	L_mlev_grid1

Tip: *Job file names always have the extension* **.job**.

Load the job file into a text editor

1 *Load the* **file sub_anomL.job** *into a text editor.*

Launch the text editor of your choice, e.g., *Notepad* for *Windows* or *vi* under *Linux*. Load the file **sub_anomL.job** from directory ***install_path*\sample_data\guided_tours\intrepid_datasets** into the text editor (or, preferably, make a copy of this file called **sub_anomLM1.job** before launching the text editor and edit the copy).



```
#### Audit Stamp Audit Stamp V3.1d - 10/ 7/1996 11:52:53
Process Begin
  Comments="Audit Stamp V3.1d-10/ 7/1996"
   Name = subsection
   POLY = polyA
   GRID = smooth_grid
   OUT = anomL_grid1
   Parameters Begin
      Box_Centroid_East = 0.0
      Box_Centroid_North = 0.0
      Box_Width = 0.0
      Box_Height = 0.0
      Box_Rotation = 0.0
      Box_LLEast = 0.0
      Box_LLNorth = 0.0
      Box_UREast = 0.0
      Box_URNorth = 0.0
      Output_Subsampling = 1
      Shrink = Yes
      SplitLines = No
      Exclude = No
      Box_Projection = "TMAMG54"
      Box_Datum = "AGD66"
   Parameters End
Process End
```

Edit the job file

2 *Edit the file so that it inputs the grid* **`mlevel_grid`** *and produces the grid* **`L_mlev_grid1`**.

Modify the **`Grid =`** and **`Out =`** lines of the file so they read as shown below.

```
Intrepid_Task {
  Name = subsection
  POLY = polyA
  GRID = mlevel_grid
  OUT = L_mlev_grid1
  Parameters Begin
   Box_Centroid_East = 0.00
   ...
   Box_Projection = "TMAMG54"
   Box_Datum = "AGD66"
  Parameters End
Process End
```

If you load a job file that you have saved yourself it will contain relative paths for the input/output dataset names. For example:

```
...
POLY = .\sample_data\guided_tours\intrepid_datasets\polyA
GRID = .\sample_data\guided_tours\intrepid_datasets\mlevel_grid
OUT = .\sample_data\guided_tours\intrepid_datasets\L_mlev_grid1
```

Save the edited job file

3 *Save the edited job file as* **`sub_anomLM1.job`**.

Save the job file with the new name **`sub_anomLM1.job`** in directory ***`install_path`*****`\sample_data\guided_tours\intrepid_datasets\tutorials`**. **Note:** Do not save the modifications back to the original file. Leave it in its original state for the next person who uses this guided tour.

Launch a command prompt

4 *Launch a command prompt window.*

Launch a command prompt. You can use *Project Manager* to do this. **Choose New Terminal Window**, **New Shell**, or **New Command Window** (depending on your operating system) from the **File** menu.

Execute a job file in batch mode

5 *Execute the job file.*

At the command line prompt, type the command **`fmanager -batch sub_anomLM1.job`**.
This command instructs the Batch Process Manager to run **`sub_anomLM1.job`** as a batch process. All INTREPID job files can be launched in this way.

INTREPID will execute the task in batch mode, creating the grid dataset **`L_mlev_grid1`**.

Every batch process will write a report which can be inspected either while the process is running or after it has completed. See “Displaying diagnostic information from Log files” in INTREPID Project Manager (T02).

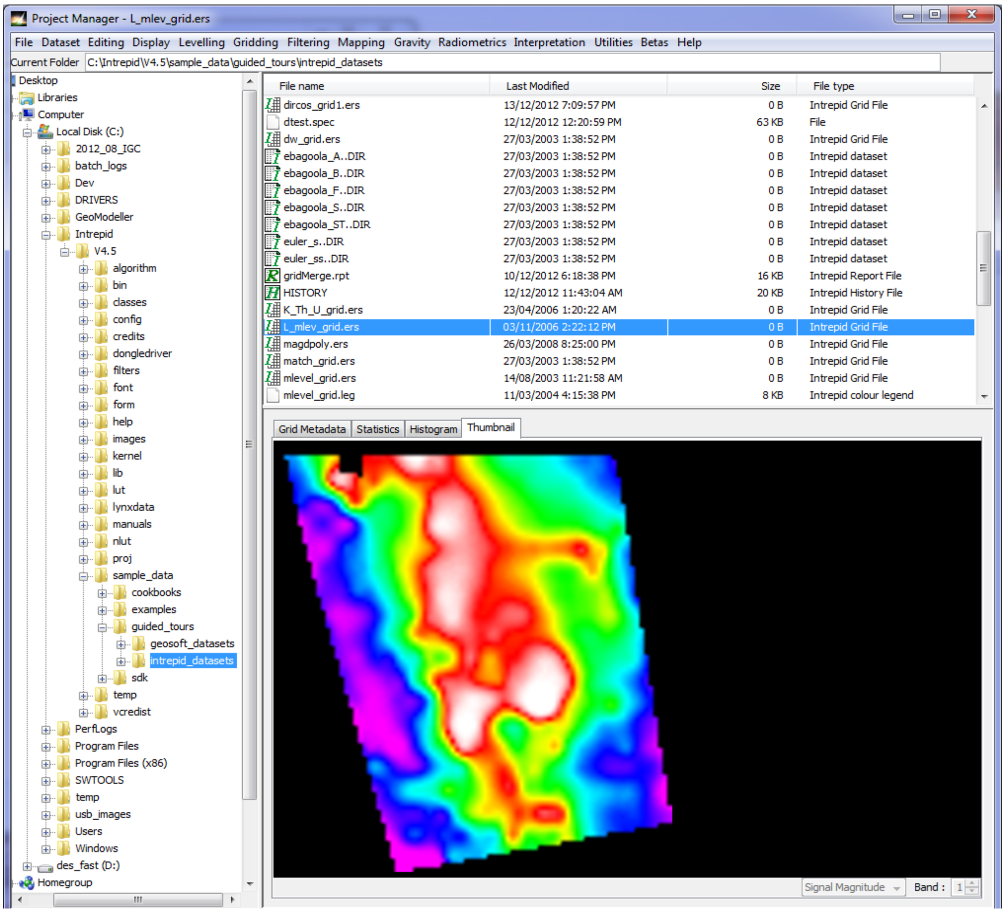
Note: Alternatively, to run the task, you can locate the **`.job`** file in the Project Manager, select it and choose **Run job file** from the shortcut (right click) menu. The Project Manager launches the Batch Process Manager to run the task.

6 *Inspect the log file created by the batch process (optional).*

Use Project Manager to display this logging information. See "Displaying diagnostic information from Log files" in INTREPID Project Manager (T02).

View the resulting dataset

7 *Use a thumbnail tool to view the dataset* **`L_mlev_grid`** *(a solution dataset provided by us, which is identical to* **`L_mlev_grid1.ers`**



Current Folder C:\Intrepid\V4.5\sample_data\guided_tours\intrepid_datasets

File name	Last Modified	Size	File type
dircos_grid1.ers	13/12/2012 7:09:57 PM	0 B	Intrepid Grid File
dtest.spec	12/12/2012 12:20:59 PM	63 KB	File
dw_grid.ers	27/03/2003 1:38:52 PM	0 B	Intrepid Grid File
ebagoola_A..DIR	27/03/2003 1:38:52 PM	0 B	Intrepid dataset
ebagoola_B..DIR	27/03/2003 1:38:52 PM	0 B	Intrepid dataset
ebagoola_F..DIR	27/03/2003 1:38:52 PM	0 B	Intrepid dataset
ebagoola_S..DIR	27/03/2003 1:38:52 PM	0 B	Intrepid dataset
ebagoola_ST..DIR	27/03/2003 1:38:52 PM	0 B	Intrepid dataset
euler_s..DIR	27/03/2003 1:38:52 PM	0 B	Intrepid dataset
euler_ss..DIR	27/03/2003 1:38:52 PM	0 B	Intrepid dataset
gridMerge.rpt	10/12/2012 6:18:38 PM	16 KB	Intrepid Report File
HISTORY	12/12/2012 11:43:04 AM	20 KB	Intrepid History File
K_Th_U_grid.ers	23/04/2006 1:20:22 AM	0 B	Intrepid Grid File
L_mlev_grid.ers	03/11/2006 2:22:12 PM	0 B	Intrepid Grid File
magdpoly.ers	26/03/2008 8:25:00 PM	0 B	Intrepid Grid File
match_grid.ers	27/03/2003 1:38:52 PM	0 B	Intrepid Grid File
mlevel_grid.ers	14/08/2003 11:21:58 AM	0 B	Intrepid Grid File
mlevel_grid.leg	11/03/2004 4:15:38 PM	8 KB	Intrepid colour legend

Extra Examples

From version 4.5, INTREPID is delivering many more examples of more powerful processes involved with supporting tensors, vectors etc. The cookbook section of **`sample_data`** has a new **`tensors\ProcessSurveyJOBS`** directory with many examples of ways to do terrain corrections, grid, form the tensor, handle Falcon datsets, and so forth. Whilst all of this is not comprehensively documented in these tours, the syntax, especially from version 5.0, is somewhat self documenting.

Location of sample .task and .job files and the data for them

With INTREPID we provide collections of sample **`.task`** and **`.job`** files and corresponding sample databases for them to process.

Where ***`install_path`*** is the path of your INTREPID installation, the folder containing these collections is
`install_path``\sample_data\examples\task` and
`install_path``\sample_data\examples\job`

The sample datasets that these **`.task`** and **`.job`** files access and process are in this folder:
`install_path``\sample_data\job\datasets`

For example, if INTREPID is installed in
`C:\Intrepid\Intrepid5.123`,
then you can find the sample datasets in
`C:\Intrepid\Intrepid5.123\sample_data\job\datasets`

Automating data access for sample .task files

For convenient access to the sample datasets while you are using the sample **`.task`** files, if it is not already present there, copy the folder
`install_path``\sample_data\job\datasets` into the
`install_path``\sample_data\examples\task` folder. This will enable the **`.task`** files in that folder to automatically access the sample data when you run them. If sample data you require for this is not present please contact our technical support service.

More information

For more information about installing sample data, see "Sample datasets—installing, locating, naming" in INTREPID Guided Tours Introduction (G01)

For a more detailed description of INTREPID datasets, see Introduction to the INTREPID database (G20). For even more detail, see INTREPID database, file and data structures (R05).

ProtoBuf Examples

The Aurizonia tensor grid integration example from the Spectral Filter guided tour is repeated here as a V5.0 task file.

```
#
# Example task file - gfilt
#
# Perform integration on a FTG tensor grid to give best estimate
of Vertical component of gravity.
#  nb Falcon requires different options
# Usage: fmanager -batch auriz_Tensor_integrate.task
#
IntrepidTask {
  GridFilter {  # FTG data set from Brazil
    InputGridName:       "../datasets/t_integrate.ers";
    FilteredGridName:    "../datasets/t1_gz.ers";
    Band:                0;
    FftGridName:         "../datasets/t1_fft";
    OutputFftGridName:   "../datasets/t1_fft_out";
    #WindowedGridName:   "../datasets/t1_wind";
    #ExpandedGridName:   "../datasets/t1_exp";
    # silly to detrend so don’t.
    DetrendDegree:       0;
    #  Bell data, so left hand coords
    InputGridCoordSysType: END;
    OutputPrecision:       IEEE4ByteReal;
    FillType:              ARTHUR;
    RolloffType:           Cosine_RollOff;
    WindowType:            NO_Window;
    ReApplyTrendAfterReverseFft: false;
    ApplyMaskAfterReverseFft:    true;
    CalculateTensorIntegration:  true;
    #  change the units from Eotvos to mGals
    ScaleTensorIntegration:      true;
    #  here is the specification of 3 inputs to produce the one
output
    InputTensorComponents:       INT_XZ_YZ_ZZ;
    }
}
```

The formal specification of the syntax for this GridFilter tool, is contained in the section **`GridFourierFiltering_INT`** of the “**`intrepid_tasks.proto`**” file found in the **`api`** directory in the distribution.

Key points of this guided tour

- In a previous guided tour you saved a job file from an interactive session.
- In this guided tour you have used the INTREPID batch system to streamline a repetitive task by
  - Editing the existing job file to specify a new task then
  - Executing the job in batch mode.
- You could also reload the edited job file in an interactive session to execute the new task or further reconfigure the job file.
- You can run job files from the command line without launching the interactive tool.
- You can write and edit your own job files using any text editor.
- You can manage repetitive ‘production line’ activities in conjunction with operating system scripting languages.

Frequently Asked Questions

Q : Does the batch system support argument substitution?

A : Yes.The Euler Deconvolution reference manual shows such an example.

Q : Does the batch system support repeat loops?

A : Yes. The Euler Deconvolution reference manual shows such an example.

Q : Can I monitor the progress of the batch task while it is running?

A : Yes. See “Displaying diagnostic information from Log files” in INTREPID Project Manager (T02).

Q : How can I check the syntax of my task file?

A : Try loading the job file into the tool and examine the log file for diagnostic information. See “Displaying diagnostic information from Log files” in INTREPID Project Manager (T02).