

# User Guide: Open Source HPC Software for Redatuning 2D WAS Field Data

Author: Clara Estela Jiménez Tejero. Developed at BCSI group at ICM-CSIC (Barcelona, Spain).

## Abstract

We introduce a user friendly open-source HPC software designed for efficiently redatuning 2D Wide-Angle Seismic (WAS) field data to the seafloor. The software employs the acoustic wave equation in reverse time, enabling redatuning of OBS gathers.

The input consists of each OBS data in SU format file and the bathymetry of the shotgathers (common for all the OBSs) in a single ASCII file. Furthermore, users can specify various parameters via an input file provided during execution.

- The software was used in [1] and is available on GitHub: [https://github.com/ejimeneztejero/DC\\_WAS](https://github.com/ejimeneztejero/DC_WAS).
- A dataset for testing and validation is available on Zenodo [2] (DOI: <https://doi.org/10.5281/zenodo.16031283>).

## 1 Software Requirements

This open-source software, developed in Fortran 90 for High-Performance Computing (HPC) architectures, relies on Open MPI for parallel processing.

- **MPI Environment:** Parallel compilation and execution require an MPI environment. Ensure that ‘mpif90’ (for compilation) and ‘mpirun’ (for execution) commands are installed and accessible.
- **Seismic Unix (SU):** The Seismic Unix tool [3] is required for handling, converting, and visualizing seismic binary data in SU format.

## 2 Installation

Follow these steps to install the software:

### 1. Automatic Installation:

- Open your terminal and navigate to the 'src/' directory.
- Compile the software by running:

```
make
```

### 2. Manual Installation:

If the automatic installation fails or you prefer a manual approach:

- First, compile the necessary modules:

```
mpif90 -c modules.f90
```

- Next, compile all source files and generate the executable named 'DC\_WAS\_run':

```
mpif90 *.f90 -o DC_WAS_run
```

### 3. Setting up the PATH:

To make the 'DC\_WAS\_run' executable accessible from any directory, add its path to your shell's configuration file (e.g., '.bashrc').

- Open your '.bashrc' file (or equivalent) in a text editor.
- Add the following line, replacing "/home/user/path/to/src" with the actual path to your 'src/' directory:

```
export PATH="/home/user/path/to/src:$PATH"
```

- Save the file and then apply the changes by running 'source ~/.bashrc' or by opening a new terminal session.

## 3 Quick Start

To quickly begin using the software, follow these steps:

1. **Configure Input Parameter File:** Set all required parameters in the input parameter file according to the specifications in [Section 4.1](#).
2. **Prepare Input Data Folder:** Populate the designated input folder with the following files:
  - **WAS Field Data:** Provide a separate SU file for each recorded OBS gather (e.g., 'OBS\_1.su', 'OBS\_2.su', 'OBS\_3.su' for three OBSs). Ensure all data files are converted to SU format (see Seismic Unix tutorial [\[3\]](#)).

- **Bathymetry and Navigation Information:** Create a single ASCII file containing the bathymetry information for the shots. If navigation data is not specified in the SU headers, include it in this file as well. See Section 4.2 for further details on the format.
3. **Run the Program:** Execute the ‘DC\_WAS\_run’ program with your specified parameter file (‘parfile’) in the terminal using the following command:

```
mpirun -np numtasks DC_WAS_run parfile
```

Replace ‘numtasks’ with the desired number of CPU cores for parallel computation. For optimal performance, set ‘numtasks’ equal to the number of OBSs being processed.

## 4 Input data

### 4.1 Input Parameter File

The input parameter file is an ASCII file with a specific structure. Each line contains a parameter name followed by a colon and at least one space, and then the corresponding parameter value. The list of parameters and their descriptions are as follows:

- `obs_file_list` (string):
  - Name of ascii file containing the name of the different OBS data files, one file per OBS. For example, for 4 OBS gathers. In this case, ‘obs\_file\_list: data.txt’, where the file data.txt is an ascii file where each line corresponds to the name of each SU file containing each OBS gather:
 

```
OBS_1.su
OBS_2.su
OBS_3.su
OBS_4.su
```
- `input_folder` (string):
 

Path to the folder containing the input files.  
Example: /home/user/DCtest/data/input
- `output_folder` (string):
 

Path where the output files will be stored. The folder is automatically created at the given path if it does not exist yet.  
Default: `output`.

- **nav\_file** (string):  
Name of the ASCII file containing bathymetry and optionally navigation information.  
Must be located inside input folder.  
See Section 4.2 for details.
- **NumOBS** (Integer):  
Number of OBSs.
- **byte\_shotnumber** (integer):  
Byte position in the SU header where the shot ID is stored.  
Example: 5 (for **tracr** header).  
Default: 5.
- **sx\_sy\_header** (integer):  
Set to 1 to read shot positions from SU headers (**sx**, **sy**); in this case, **scalco** is also read.  
If set to 0, shot positions must be provided in the **nav\_file** (see Section 4.2).  
Default: 0.
- **nt** (integer):  
Number of time samples.
- **dt** (real):  
Time sampling interval of the OBS gathers (in seconds).
- **shot\_init** (integer):  
Shot ID of the first shot from SU file.
- **shot\_fin** (integer):  
Shot ID of the last shot from SU file.
- **shot\_depth** (real):  
Depth of the sources (in meters).
- **dmodel** (real):  
Spatial sampling interval of the velocity model (in meters).  
Choose half distance between shots or finer if needed.

- **water\_velocity** (real):  
Constant water velocity (in meters/second).  
Default: 1500.
- **save\_gmt** (integer):  
Set to 1 to save each OBS in GMT-compatible ASCII format:  
`X(1:NumShots) Y(1:nt) Z=OBS(nt, NumShots)`.  
Default: 0 (not activated).
- **save\_matlab** (integer):  
Set to 1 to save each OBS in MATLAB-compatible ASCII format:  
`OBS(nt, NumShots)`.  
Default: 0 (not activated).

## 4.2 Bathymetry and Navigation Information

The bathymetry information for the shots must be provided in an ASCII file. Depending on the 'sx\_sy\_header' parameter's activation in the input parameter file:

- If 'sx\_sy\_header: 0', the navigation (in UTM coordinates) must be included in the same ASCII file.
- If 'sx\_sy\_header: 1', the navigation (in UTM coordinates) is automatically extracted from the SU headers and doesn't need to be included in this file.

There are two options to structure the bathymetry/navigation file:

- If 'sx\_sy\_header: 0', the file must have a 4-column structure:  
`shotID1 X1 (UTM, meters) Y1 (UTM, meters) Z1 (meters)`  
`shotID2 X2 (UTM, meters) Y2 (UTM, meters) Z2(meters)`  
`... ..`  
`shotIDn Xn (UTM, meters) Yn (UTM, meters) Zn (meters)`
- If 'sx\_sy\_header: 1', the file must have a 2-column structure:  
`shotID1 Z1 (meters)`  
`shotID2 Z2 (meters)`  
`... ..`  
`shotIDn Zn (meters)`

Notes:

- The units for each shot position (X and Y in UTM) are in meters, using the 'Universal Transverse Mercator' coordinates.

- The 'shotID<sub>i</sub>' parameter is the shot number, which must be found in some SU header (probably `tracr`, `fldr` ...). Remember to fix the parameter `byte_shotnumber` in the `parfile` to indicate where to read the shotID. By default, if not specified, shotID will be read from `tracr` (byte 5).
- The depth of the seafloor at each shot position ( $Z_i$ ) can be expressed as positive or negative numbers. The software uses the absolute value,  $|Z_i|$ , for computations.

## 5 Output Data

After executing the program, the following files will be located in the output folder:

- **DC\_OBS.i (SU format):** OBS gather results after DC, where "i" is the OBS number, ranging from 1 to n. If there is only one OBS, it will be named "DC\_OBS".
- **bathymetry\_meters.txt (ASCII file):**  
This file contains the bathymetry interpolated to the grid of the model. It has 2 columns: x-axis (grid model in meters) and y-axis (bathymetry).

## References

- [1] C E Jimenez-Tejero, M Prada, L Gomez de la Peña, C R Ranero, V Sallares, Downward continuation of wide-angle seismic data: implications for traveltime tomography uncertainty, *Geophysical Journal International*, Volume 241, Issue 3, June 2025, Pages 1868–1880, <https://doi.org/10.1093/gji/ggaf137>.
- [2] Jimenez Tejero, C. E. (2025). Data sample for testing the code: Downward Continuation for WAS data (DC\_WAS) [Data set]. In *Geophysical Journal International* (Vol. 241, Number 3, pp. 1868–1880). Zenodo. <https://doi.org/10.5281/zenodo.16031283>.
- [3] [SU]Murillo, Alejandro E. and J. Bell. "Distributed Seismic Unix: a tool for seismic data processing." *Concurrency and Computation: Practice and Experience* 11 (1999): 169-187. Seismic Unix tool: <https://wiki.seismic-unix.org/doku.php>.