REPTILE User Manual (DRAFT)

Dave Dimock

February 27, 2016

Contents

1 Introduction

REPTILE is a general purpose command line utility for extracting internal loads and other data from NASTRAN output files (XDB files). The goal of REPTILE is to make extracting and formatting this data efficient and convenient– especially in situations where data extraction may otherwise be repetitive or labor-intensive.

REPTILE is executed from the command line, which means that it does not have a graphical user interface. It is also not interactive (i.e. it doesn't prompt the user for information), but rather it takes input in the form of text files.

These were intentional design decisions aimed at making it as simple as possible to automate the action of extracting the data– typically a user would create a batch file (.bat) containing a list of commands to perform. Thus, if the NASTRAN model is updated, pulling new loads data is as simple as double-clicking the batch file.

1.1 Installation

There is currently no automatic installer for REPTILE, so the following process is recommended:

1. Create a "bin" folder in your user directory (e.g. C:\Users\username\bin) and place a copy of "reptile.exe" in that folder.
2. Open the Control Panel, type "environment variable" in the search box, and click on "Edit environment variables for your account."
3. Find the "PATH" variable in the "User Variables" box and add the full path of the "bin" folder to the end of the PATH variable (separated with a semi-colon) (e.g. `%PATH%;C:\Users\username\bin`).

Alternatively (though not recommended) one could copy the XDB TOOL executable to the folder where loads extractions are to be performed.

1.2 Command Line

To run REPTILE from the command line:

1. Open a command window (Start / Run / cmd)

2. Type “reptile” at the prompt and press Enter.

With no options specified REPTILE will display help information. The program modes and options are described elsewhere in this document.

1.3 Batch File

To create a batch file to simplify and automate the execution of REPTILE:

1. Open Notepad or another text editor.
2. Write one XDB TOOL command line per line to perform each task.
3. Save the text file with a “.bat” extension.

Now you can double-click the batch file to run all the XDB TOOL commands. It is optional (but recommended) to add a line with “pause” at the very end of the batch file, so that if any error messages appear, the user will be able to read them before the window disappears.

2 Usage

This section describes how to invoke each of REPTILE's modes, as well as the inputs and options that may be supplied. The modes can be roughly grouped into four categories which are described in their corresponding sub-sections.

2.1 Element Loads

This category of REPTILE modes extracts element force data on an element-by-element basis. The forces are reported without any modification from their original values, except in the case of SHELL_FORCES, which can optionally perform a tensor-rotation of the shell element loads.

2.1.1 BAR_FORCES

2.1.2 BUSH_FORCES

2.1.3 CFAST_FORCES

2.1.4 GAP_FORCES

2.1.5 SHELL_FORCES

2.2 Processed Loads

This category of REPTILE modes extracts loads data and performs some kind of manipulation before reporting the data at user-defined locations (rather than element-by-element or node-by-node).

2.2.1 NODE_FORCES

2.2.2 PANELIZER

2.2.3 PANELIZER_GPF

2.2.4 SHELL_AVG

2.2.5 VMT

Not yet implemented.

2.3 Other Data

This category of REPTILE modes extracts FEM data that is not loads.

2.3.1 EIGENVALUE

2.3.2 DEROTATILATOR

Not yet implemented.

2.3.3 DISP

2.4 Utilities

This category of REPTILE modes do not directly operate on XDB files. They provide other useful functions related to the compiling of internal loads data.

2.4.1 CSV2SDIO

2.4.2 COMBINE

2.4.3 POTATO

Not yet implemented.

2.4.4 VODKA

Not yet implemented.

3 Examples and Guidance

This section provides examples and guidance for different scenarios one might encounter while extracting NASTRAN results data. Each example is meant to illustrate a particular loads extraction philosophy, or to highlight common pitfalls and misunderstandings.

Finite element models necessarily contain compromises in their idealization. In order to make appropriate use of the output of a FEM, it is necessary to understand the impact of those compromises. The examples in this section place an emphasis on extracting loads in ways that are consistent with the underlying idealization and, in some cases, compensate for the shortcomings of the model.

3.1 Element-Type Loads

This is a Subsection.

3.1.1 TEST

This is a Sub-Subsection.

3.1.2 TEST

This is a Sub-Subsection.

3.1.3 TEST

This is a Sub-Subsection.

3.2 Cut-Type Loads

This is a Subsection.

3.2.1 TEST

This is a Sub-Subsection.

3.2.2 TEST

This is a Sub-Subsection.

3.2.3 TEST

This is a Sub-Subsection.

3.3 Joint-Type Loads

This is a Subsection.

3.3.1 TEST

This is a Sub-Subsection.

3.3.2 TEST

This is a Sub-Subsection.

3.3.3 TEST

This is a Sub-Subsection.

3.4 Panel-Type Loads

This is a Subsection.

3.4.1 TEST

This is a Sub-Subsection.

3.4.2 TEST

This is a Sub-Subsection.

3.4.3 TEST

This is a Sub-Subsection.

3.4.4 TEST

This is a Sub-Subsection.

3.4.5 TEST

This is a Sub-Subsection.

3.4.6 TEST

This is a Sub-Subsection.

3.4.7 TEST

This is a Sub-Subsection.

4 Technical Details

coordinate systems vector/tensor rotation derotation (quaternions?) panelizer methods

4.1 Coordinate Transformation

This is a Subsection.

4.1.1 Vector Rotation

This is a Sub-Subsection.

4.1.2 Tensor Rotation

This is a Sub-Subsection.

4.1.3 Transformation Matrices

This is a Sub-Subsection.

4.2 Panelizer Methods

This is a Subsection.

4.2.1 Element-Based

Mention Heteroskedastic Bilinear Regression. Explain what you get. Explain how to turn dNxx into Mzx or whatever.

4.2.2 Node-Based

Explain filtering rules, and what you get