

IDV to AUX



Bring your IDV files.
Work in PowerWorld Simulator.

A translator from IDV files to AUX files

Maintained and Authored by:

Saurav Mohapatra, PhD, PE

saurav@powerworld.com

October 5, 2026



PowerWorld
Corporation

support@powerworld.com
<https://www.powerworld.com>

The gap

IDV-based Automation

- Large volume of study changes saved as IDV files
- Staff who know it well
- Established study processes built around it

&

PowerWorld Simulator

- A rich, visual and interactive interface
- Quick for new staff to find their way around
- Many teams would rather do the work here

The gap: your case modifications and workflows are written in IDV

1 A deep IDV archive

Years of network changes, recorded as IDV files.

2 Skills tied to IDV terminologies

The people who wrote them think in IDV commands and activities.

3 Manual rework is costly

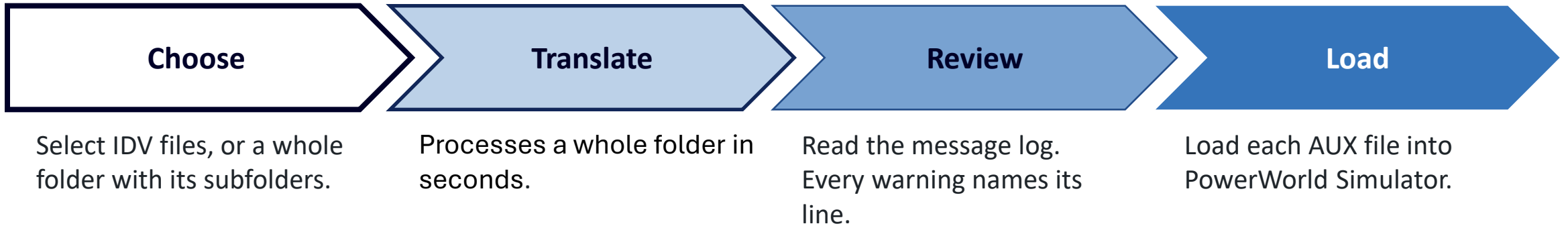
Rebuilding each change into AUX by hand takes precious time. Every step invites possible errors.

So the work stays in another program — even where PowerWorld Simulator would be the easier tool.

What IDV to AUX is

A translator.

IDV files in, AUX files out.



Two ways to run it

Desktop app - Everyday use

Command line - Batch work.

Nothing to pre-configure

Self-contained executables, with the documentation beside them that states the current coverage.

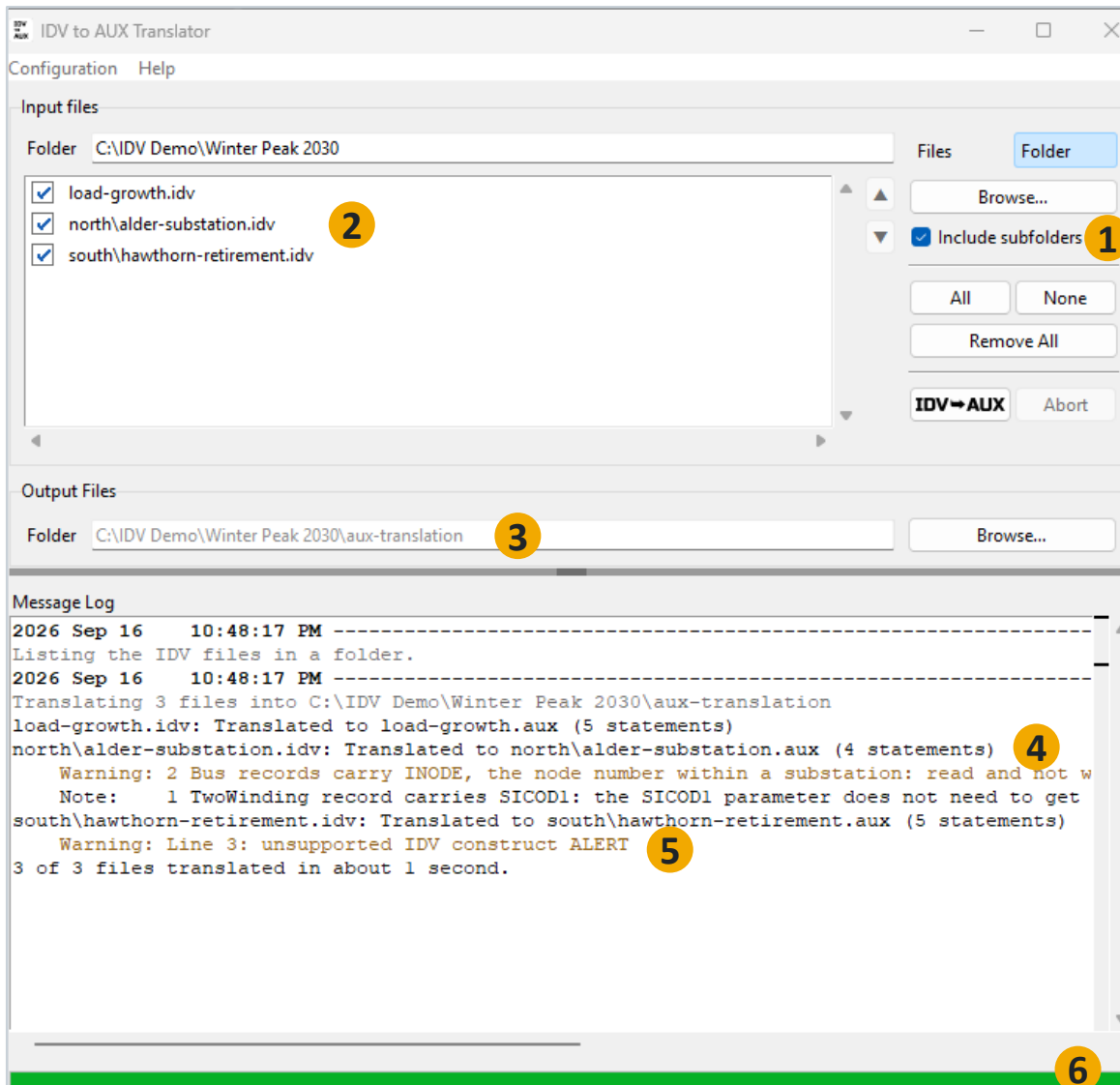
One file in, one file out

Each IDV file becomes an AUX file you load into PowerWorld Simulator. (extension *.idv.to.aux)

Note: Files are standard AUX files and are supported by PowerWorld Simulator 24, build date September 25, 2026 or later.

The tool at work

IDV
→
AUX



1 Whole archives at once

A folder and all its subfolders, translated in one run.

2 Named from the folder

Rows show each file and its subfolder.

3 Your folder structure, kept

In Folder mode, the same folder tree is written.

4 Traceable

Log states each IDV file and the translated AUX file.

5 Transparent about gaps

Information/Warnings about lines not translated.

6 Keeps going

Unreadable lines are skipped and the rest translates.

What it understands: Focus is on case modification constructs

All four kinds of content an IDV file can carry.

112 API routines and growing

BAT_

Batch commands for buses, loads, machines, branches, two- and three-winding transformers, shunts, areas, zones, dc lines and more.

16 LMI activities and growing

RDCH

RDCH, PURG, MOVE, EXTR, SPLT and LTAP change the case; TEXT, MENU, VERSION and Q steer the reading.

12 RAW records and growing

RAW

Buses, branches, transformers and more inside an RDCH block, for RAW versions 33 to 36.

1 input processor command

@

@END, which closes a response file.

Future extensions

The same translation architecture can extend beyond IDV files, to other data inputs:

Subsystem files

Other helper files

Built to be trusted, and what to expect

It never guesses.

When a value has no AUX equivalent, it is left out and reported — never approximated into something that looks right.

- ✓ Reads each file at the RAW version it declares and says which one.
- ✓ Writes only field mappings checked against AUX format's fields.
- ✓ Documents every supported command with a worked example, shipped with the tool. Verifiable by end-users.
- ✓ Developed and tested against real IDV files, not just textbook samples.
- ✓ Files are always locally read/written. Users have to manually send data when requesting coverage expansions.

! Warnings worth reading

Each names the IDV line it came from.

! Some fields may be pending

Where AUX has no field yet, the value is reported.

! Files are read as written

A file or field that contradicts itself is reported, not silently skipped.

Easily check a translation by loading it into a case and confirming the expected changes.

What it is for, and what it is not

It is

- ✓ A translator for IDV files
- ✓ A bridge while your team members adopt PowerWorld Simulator
- ✓ Clear about anything it could not carry across

It is NOT

- ✗ A replacement for PowerWorld Simulator
- ✗ A power flow solver, or a check of engineering results
- ✗ A converter for base cases, dynamics data or Python scripts
- ✗ Two-way: AUX back to IDV is out of scope

Bring what already works. Start working in PowerWorld now!

1

Choose

A folder of IDV files
that has your work

2

Translate

Run it with
subfolders included

3

Read

Go through the log
and review any gaps

4

Compare

Load the AUX into
a PowerWorld case

You judge it on your own files.



IDV archive moves to AUX



Your team does studies in
PowerWorld Simulator



Every gap is visible, never hidden

Next step: Go ahead, and test IDV to AUX on your own IDV files.