DVOACAP WRAPPER UPDATE - SUMMARY
===================================

Date: November 12, 2025
Status: ✓ COMPLETE - All files updated and ready

WHAT WAS FIXED

The DVOACAP wrapper was using an incorrect JSON format. After reverse-engineering the actual API requirements, the wrapper has been corrected to use the exact format the dvoa.dll expects.

CRITICAL FIX: Arguments Structure

BEFORE (Broken):

```
{
  "Method": 30,
  "Transmitter": {...},
  "Receiver": {...},
  "Frequencies": [...],
  "Hours": [...]
}
```

AFTER (Working):

```
{
  "Arguments": {
    "RxLocations": [{"Lat": 44.90, "Lon": 20.50, "Label": "BELGRADE"}],
    "Hours": {"Start": 1, "Step": 1, "Count": 24},
    "Freqs": [6.07, 7.20, 9.70, 11.85, 13.70, 15.35, 17.73, 21.65, 25.89],
    "IncludeMuf": true
  }
}
```

FILES DELIVERED
----

1. dvoacap_wrapper_fixed.py (9.4 KB)
   - Corrected wrapper with proper JSON format
   - New predict_simple() method
   - New predict_multi_band() method for comprehensive analysis
   - Full predict() method with all options

2. generate_propagation_voacap.py (17 KB)
   - Updated to use the corrected wrapper
   - Supports both ITU-R (fast) and DVOACAP (accurate) modes
   - Automatic fallback if DVOACAP unavailable
   - Progress indicators during generation

3. DVOACAP_SETUP_GUIDE.txt (5.2 KB)
   - Complete installation instructions
   - Migration guide from old to new API
   - Usage examples
   - Troubleshooting tips

4. DVOACAP_API_REFERENCE.txt (4.6 KB)
   - Quick reference for all API methods
   - Example code snippets
   - Parameter explanations
   - Return value documentation

INSTALLATION (3 Steps)

Step 1: Install the wrapper
   Copy dvoacap_wrapper_fixed.py to your project directory:
   → Rename it to: dvoacap_wrapper.py

Step 2: Install the generation script
   Copy generate_propagation_voacap.py to your project directory
   → Replaces the old version

Step 3: Verify dvoa.dll is present
   If not, download from: https://github.com/VE3NEA/DVOACAP
   Place dvoa.dll in same directory as scripts

TESTING

Test 1: Verify wrapper

```
python dvoacap_wrapper.py
```

   Expected output:

```
✓ DVOACAP engine loaded from dvoa.dll
```

```
  [TEST 1] Single path/frequency/hour
  ...
  ✓ Tests complete!

Test 2: Generate predictions
  python generate_propagation_voacap.py --dvoacap

  Expected output:
  ✓ DVOACAP engine available
  Fetching solar data...
  Generating predictions for 7 bands x 10 regions...
  ✓ Predictions generated

USAGE EXAMPLES
--------------

Example 1: Quick Script
  from dvoacap_wrapper import DVOACAPEngine

  engine = DVOACAPEngine("dvoa.dll")
  result = engine.predict_simple(
     tx_lat=44.374, tx_lon=-64.300,
     rx_lat=51.5, rx_lon=-0.1,
     frequency_mhz=14.15,
     hour_utc=18,
     ssn=140
  )
  print(f"{result['quality']}: {result['reliability']}%")

Example 2: Generate Report
  python generate_propagation_voacap.py --dvoacap

Example 3: Multi-band Analysis
  result = engine.predict_multi_band(
     tx_lat=44.374, tx_lon=-64.300,
     rx_lat=51.5, rx_lon=-0.1,
     ssn=140
  )

KEY IMPROVEMENTS
----------------
```

✓ Correct JSON format matches DVOACAP API exactly
✓ Support for multiple RX locations in single call
✓ Support for multiple frequencies
✓ Flexible hour ranges (any start/step/count)
✓ Built-in predict_multi_band() for comprehensive analysis
✓ Better error handling and reporting
✓ Automatic fallback to ITU-R if DVOACAP fails

API CHANGES

OLD (No longer works):

```
  predict(tx_lat, tx_lon, rx_lat, rx_lon, freq, hour, ssn)
```

NEW (Use these):

```
  predict_simple(tx_lat, tx_lon, rx_lat, rx_lon, freq, hour, ssn, rx_label)
  predict_multi_band(tx_lat, tx_lon, rx_lat, rx_lon, ssn)
  predict(tx_lat, tx_lon, rx_locations, frequencies, hours, ssn)
```

MIGRATION CHECKLIST

[ ] Backup existing files
[ ] Copy dvoacap_wrapper_fixed.py → dvoacap_wrapper.py
[ ] Copy generate_propagation_voacap.py
[ ] Verify dvoa.dll is present
[ ] Run test: python dvoacap_wrapper.py
[ ] Generate predictions: python generate_propagation_voacap.py --dvoacap
[ ] Update any custom scripts using old API

NEXT STEPS

1. Install the files (see Installation above)
2. Test the wrapper (python dvoacap_wrapper.py)
3. Generate DVOACAP predictions (python generate_propagation_voacap.py --dvoacap)
4. View results in propagation_data.json
5. Load in your dashboard

Compare DVOACAP vs ITU-R results to see accuracy improvement!

SUPPORT

For detailed information, see:
- DVOACAP_SETUP_GUIDE.txt - Complete setup and migration guide
- DVOACAP_API_REFERENCE.txt - Quick API reference
- VOACAP_SETUP_GUIDE.txt - Original VOACAP documentation

DVOACAP Source:
https://github.com/VE3NEA/DVOACAP

That's it! You're ready to generate accurate VOACAP predictions.

73 de VE1ATM