# DVOACAP WRAPPER UPDATE - SETUP GUIDE
========================================

## WHAT WAS FIXED
--------------
The DVOACAP wrapper now uses the EXACT JSON format the API expects:

OLD (BROKEN):
```
{
  "Method": 30,
  "Transmitter": {...},
  "Receiver": {...},
  "Frequencies": [...],
  "Hours": [...]
}
```

NEW (CORRECT):
```
{
  "Arguments": {
    "RxLocations": [{"Lat": 44.90, "Lon": 20.50, "Label": "BELGRADE"}],
    "Hours": {"Start": 1, "Step": 1, "Count": 24},
    "Freqs": [6.07, 7.20, 9.70, ...],
    "IncludeMuf": true
  }
}
```

## FILES UPDATED
-------------
1. dvoacap_wrapper_fixed.py - Corrected wrapper with proper API format
2. generate_propagation_voacap.py - Updated to use the new wrapper

## INSTALLATION STEPS
------------------

Step 1: Replace the old wrapper
   Copy dvoacap_wrapper_fixed.py to your project directory as dvoacap_wrapper.py:

```
   cp dvoacap_wrapper_fixed.py dvoacap_wrapper.py
```

Step 2: Replace the generation script
   Copy the updated generation script:

```
   cp generate_propagation_voacap.py [your_project_directory]/
```

Step 3: Ensure you have dvoa.dll

```
   Download DVOACAP from: https://github.com/VE3NEA/DVOACAP
   Extract dvoa.dll to your project directory
```

Step 4: Test the wrapper

```
   python dvoacap_wrapper.py

   You should see:
   ✓ DVOACAP engine loaded from dvoa.dll
   [TEST 1] Single path/frequency/hour
   ...
```

Step 5: Generate predictions with DVOACAP

```
   python generate_propagation_voacap.py --dvoacap
```

API CHANGES
-----------

OLD METHOD CALLS (Won't work anymore):

```
   result = engine.predict(
       tx_lat, tx_lon,
       rx_lat, rx_lon,
       frequency_mhz,
       hour_utc,
       ssn
   )
```

NEW METHOD CALLS (Use these):

```
   # Simple single prediction
   result = engine.predict_simple(
       tx_lat=44.374,
       tx_lon=-64.300,
       rx_lat=44.90,
       rx_lon=20.50,
       frequency_mhz=14.15,
       hour_utc=18,
       ssn=140,
       rx_label="BELGRADE"
   )

   # Multi-band 24-hour prediction
   result = engine.predict_multi_band(
    tx_lat=44.374,
    tx_lon=-64.300,
    rx_lat=44.90,
    rx_lon=20.50,
    rx_label="BELGRADE",
    ssn=140
)

# Full control with exact API format
result = engine.predict(
    tx_lat=44.374,
    tx_lon=-64.300,
    rx_locations=[
        {"Lat": 44.90, "Lon": 20.50, "Label": "BELGRADE"},
        {"Lat": 51.50, "Lon": -0.10, "Label": "LONDON"}
    ],
    frequencies=[7.1, 14.15, 21.2],
    hours={"Start": 0, "Step": 1, "Count": 24},
    ssn=140
)
```

NEW FEATURES
------------

1. Multi-location support
   Can now predict to multiple RX locations in one call:

```
rx_locations = [
    {"Lat": 44.90, "Lon": 20.50, "Label": "BELGRADE"},
    {"Lat": 51.50, "Lon": -0.10, "Label": "LONDON"},
    {"Lat": 35.68, "Lon": 139.69, "Label": "TOKYO"}
]
```

2. Multi-frequency support
   Can now predict multiple frequencies at once:

```
frequencies = [7.1, 10.13, 14.15, 18.1, 21.2, 24.95, 28.4]
```

3. Flexible hour ranges
   Can specify any hour range:

```
hours = {"Start": 12, "Step": 2, "Count": 12}  # Every 2 hours from noon
```

4. Built-in predict_multi_band()

   Automatically predicts all HF amateur bands for 24 hours

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

Example 1: Quick single prediction

```
   from dvoacap_wrapper import DVOACAPEngine

   engine = DVOACAPEngine("dvoa.dll")
   result = engine.predict_simple(
       tx_lat=44.374, tx_lon=-64.300,  # Halifax
       rx_lat=51.5, rx_lon=-0.1,       # London
       frequency_mhz=14.15,            # 20m
       hour_utc=18,
       ssn=140
   )

   print(f"Quality: {result['quality']}")
   print(f"Reliability: {result['reliability']}%")
   print(f"SNR: {result['snr_db']} dB")
```

Example 2: Full band analysis

```
   result = engine.predict_multi_band(
       tx_lat=44.374, tx_lon=-64.300,
       rx_lat=51.5, rx_lon=-0.1,
       rx_label="LONDON",
       ssn=140
   )
```

Example 3: Generate propagation report

```
   python generate_propagation_voacap.py --dvoacap
```

TROUBLESHOOTING
---------------

Error: "DVOACAP DLL not found"
   → Ensure dvoa.dll is in the same directory as dvoacap_wrapper.py

Error: "DVOACAP prediction error"
   → Check that your JSON format matches the "Arguments" structure
   → Run the test: python dvoacap_wrapper.py

No results returned:
   → Check that SSN value is reasonable (0-300)

  → Verify lat/lon coordinates are valid
  → Check DVOACAP result parsing in predict_simple()

MIGRATION CHECKLIST
-------------------
[ ] Backup old files
[ ] Copy dvoacap_wrapper_fixed.py as dvoacap_wrapper.py
[ ] Copy updated generate_propagation_voacap.py
[ ] Verify dvoa.dll is present
[ ] Test wrapper: python dvoacap_wrapper.py
[ ] Test generation: python generate_propagation_voacap.py --dvoacap
[ ] Update any custom scripts that call the old API

NEXT STEPS
----------
Once everything is working:
1. Generate predictions with: python generate_propagation_voacap.py --dvoacap
2. View results in your dashboard
3. Compare DVOACAP vs ITU-R predictions to see the accuracy improvement
4. Schedule automatic updates with update_predictions.sh

For questions or issues, refer to:
- DVOACAP docs: https://github.com/VE3NEA/DVOACAP
- VOACAP_SETUP_GUIDE.txt in your project

Happy DXing!
73 de VE1ATM