Troubleshooting Guide

Common issues and solutions for UtilityFog-Fractal-TreeOpen.

Installation Issues

Python Version Compatibility

Problem: `ERROR: Python 3.8 is not supported`

Solution:

```
# Check your Python version
python --version

# Install Python 3.9+ using your system package manager
# Ubuntu/Debian:
sudo apt install python3.11

# macOS with Homebrew:
brew install python@3.11

# Windows: Download from python.org
```

Missing Dependencies

Problem: `ModuleNotFoundError: No module named 'plotly'`

Solution:

```
# Install all required dependencies
pip install -r testing_requirements.txt

# Or install specific missing packages
pip install plotly pandas numpy networkx
```

Permission Errors

Problem: `Permission denied` during installation

Solution:

```
# Use user installation
pip install --user utilityfog-fractal-tree

# Or use virtual environment
python -m venv venv
source venv/bin/activate  # On Windows: venv\Scripts\activate
pip install utilityfog-fractal-tree
```

Runtime Issues

CLI Command Not Found

Problem: `ufog-diagnose: command not found`

Solution:

```
# Check if package is installed
pip list | grep utilityfog

# Add pip bin directory to PATH
export PATH="$HOME/.local/bin:$PATH"

# Or run directly with Python
python -m utilityfog_frontend.cli_viz.cli --help
```

Visualization Not Working

Problem: Blank or broken visualization output

Solution:

```
# Check browser compatibility (Chrome/Firefox recommended)
# Ensure all visualization dependencies are installed
pip install plotly>=5.0.0 pandas>=1.3.0

# Try different export format
python -c "
from utilityfog_frontend.cli_viz.cli import VisualizationCLI
cli = VisualizationCLI()
cli.export_svg('test.svg')  # Try SVG instead of HTML
"
```

Memory Issues

Problem: `MemoryError` or system slowdown during large simulations

Solution:

```
# Reduce simulation size
from UtilityFog_Agent_Package.agent.feature_flags import FeatureFlags

# Enable memory optimization
flags = FeatureFlags()
flags.enable_memory_optimization = True
flags.max_tree_depth = 5  # Limit tree depth
flags.max_agents = 100    # Limit number of agents
```

Port Already in Use

Problem: `Address already in use: 8080`

Solution:

```
# Find process using the port
lsof -i :8080  # On Unix systems
netstat -ano | findstr :8080  # On Windows

# Kill the process or use different port
ufog-visualize --demo --port 8081
```

Performance Issues

Slow Startup

Problem: Application takes long time to start

Solution:

```
# Disable unnecessary features during development
from UtilityFog_Agent_Package.agent.feature_flags import FeatureFlags

flags = FeatureFlags()
flags.enable_telemetry = False
flags.enable_detailed_logging = False
flags.enable_visualization = False  # For headless operation
```

High CPU Usage

Problem: Excessive CPU usage during simulation

Solution:

```
# Adjust simulation parameters
import time

# Add delays in message processing loops
def process_with_delay():
    # Your processing code here
    time.sleep(0.01)  # 10ms delay

# Reduce update frequency
flags.heartbeat_interval = 5.0  # Increase from default 1.0s
flags.telemetry_interval = 10.0  # Increase from default 1.0s
```

Development Issues

Import Errors

Problem: `ImportError: cannot import name 'FogletAgent'`

Solution:

```
# Ensure you're in the correct directory
cd UtilityFog-Fractal-TreeOpen

# Install in development mode
pip install -e .

# Check Python path
python -c "import sys; print('\n'.join(sys.path))"
```

Test Failures

Problem: Tests failing with various errors

Solution:

```
# Run tests with verbose output
python -m pytest -v

# Run specific test file
python -m pytest tests/test_specific.py -v

# Check test dependencies
pip install pytest pytest-cov coverage
```

Git Issues

Problem: Git operations failing or slow

Solution:

```
# Clean git cache
git gc --aggressive

# Reset sparse checkout if needed
git sparse-checkout disable
git checkout .

# Check git configuration
git config --list
```

Environment-Specific Issues

Docker Container Issues

Problem: Container fails to start or crashes

Solution:

```
# Check container logs
docker logs <container_id>

# Run with interactive shell for debugging
docker run -it utilityfog-fractal-tree /bin/bash

# Check resource limits
docker stats <container_id>
```

Windows-Specific Issues

Problem: Path or encoding issues on Windows

Solution:

```
# Use UTF-8 encoding
$env:PYTHONIOENCODING="utf-8"

# Use forward slashes in paths
python -c "import os; print(os.path.join('path', 'to', 'file'))"

# Run as administrator if needed
```

macOS-Specific Issues

Problem: SSL certificate or permission issues

Solution:

```
# Update certificates
/Applications/Python\ 3.11/Install\ Certificates.command

# Fix permissions
sudo chown -R $(whoami) /usr/local/lib/python3.11/site-packages/
```

Getting Help

If you're still experiencing issues:

1. **Check the logs**: Look for detailed error messages in console output
2. **Search existing issues**: GitHub Issues (https://github.com/Goldislops/UtilityFog-Fractal-TreeOpen/issues)
3. **Create a minimal reproduction**: Isolate the problem to the smallest possible example
4. **Report the issue**: Include:
   - Operating system and version
   - Python version (`python --version`)
   - Package version (`pip show utilityfog-fractal-tree`)
   - Full error traceback
   - Steps to reproduce

Diagnostic Information

Run this command to gather system information for bug reports:

```
ufog-diagnose --json --verbose > diagnostic_info.json
```

This will create a file with:
- System information
- Python environment details
- Package versions

- Configuration settings
- Recent log entries

Include this file when reporting issues for faster resolution.