

# Testing Guide for Transcribe RO

---

This document provides instructions for testing the Transcribe RO tool to ensure it's working correctly.

## Pre-Installation Tests












---

### 1. Verify File Structure

Check that all required files are present:

```
ls -la
```

Expected files:

-  `transcribe_ro.py` - Main script
-  `requirements.txt` - Dependencies
-  `README.md` - Documentation
-  `QUICKSTART.md` - Quick start guide
-  `LICENSE` - License file
-  `.gitignore` - Git ignore rules
-  `setup_portable.bat` - Windows setup script
-  `setup_portable.sh` - Linux/macOS setup script
-  `run_transcribe.bat` - Windows run script
-  `run_transcribe.sh` - Linux/macOS run script
-  `examples.sh` - Example commands

### 2. Verify Python Syntax

```
python3 -m py_compile transcribe_ro.py
```

Expected: No errors

### 3. Check Python Version

```
python3 --version
```

Expected: Python 3.8 or higher

### 4. Check FFmpeg Installation

```
ffmpeg -version
```

Expected: FFmpeg version information (if not found, refer to README.md for installation)

## Installation Tests

---

### 1. Run Setup Script

**Windows:**

```
setup_portable.bat
```

**Linux/macOS:**

```
./setup_portable.sh
```

Expected:

- ☒ Virtual environment created
- ☒ Dependencies installed
- ☒ No error messages

## 2. Verify Virtual Environment

**Windows:**

```
venv\Scripts\python --version
```

**Linux/macOS:**

```
venv/bin/python --version
```

Expected: Python version information

## 3. Verify Whisper Installation

**Windows:**

```
venv\Scripts\activate  
python -c "import whisper; print('Whisper installed:', whisper.__version__)"
```

**Linux/macOS:**

```
source venv/bin/activate  
python -c "import whisper; print('Whisper installed:', whisper.__version__)"
```

Expected: Whisper version information

## Functional Tests

### 1. Test Help Command

```
python transcribe_ro.py --help
```

Expected: Help message with all available options

### 2. Test Version Command

```
python transcribe_ro.py --version
```

Expected: Transcribe R0 v1.0.0

### 3. Test with Sample Audio (If Available)

Create or download a sample audio file, then:

```
python transcribe_ro.py sample_audio.mp3 --model tiny
```

Expected:

- ☒ Model downloads (first run only)
- ☒ Transcription completes
- ☒ Output file created
- ☒ No errors

### 4. Test Different Output Formats

```
# Test TXT format (default)
python transcribe_ro.py sample.mp3 --model tiny

# Test JSON format
python transcribe_ro.py sample.mp3 --model tiny --format json

# Test SRT format
python transcribe_ro.py sample.mp3 --model tiny --format srt

# Test VTT format
python transcribe_ro.py sample.mp3 --model tiny --format vtt
```

Expected: Different output files created with appropriate formats

### 5. Test No Translation Option

```
python transcribe_ro.py sample.mp3 --model tiny --no-translate
```

Expected: Output contains only transcription, no translation section

### 6. Test Error Handling

```
# Test with non-existent file
python transcribe_ro.py nonexistent.mp3
```

Expected: Clear error message about file not found

## Performance Tests

---

### 1. Test Different Model Sizes

```
# Tiny model (fastest)
time python transcribe_ro.py sample.mp3 --model tiny

# Base model (balanced)
time python transcribe_ro.py sample.mp3 --model base

# Small model (more accurate)
time python transcribe_ro.py sample.mp3 --model small
```

Record processing times for comparison.

### 2. Test with Different Audio Lengths

- Short audio (< 1 minute)
- Medium audio (1-5 minutes)
- Long audio (> 5 minutes)

Expected: Tool handles all lengths without crashing

## Portable Installation Tests

---

### 1. Copy to Flash Drive

1. Copy entire `transcribe_ro` folder to USB drive
2. Navigate to the folder on the USB drive
3. Run setup script
4. Test transcription

Expected: Works identically to local installation

### 2. Test on Different Computer

1. Plug USB drive into different computer
2. Open command prompt/terminal in the folder
3. Run: `run_transcribe.bat` (Windows) or `./run_transcribe.sh` (Linux/macOS)
4. Test with sample audio

Expected: Tool works without reinstallation (models may need to be downloaded)

## Edge Cases and Error Handling

---

### 1. Test Unsupported File Format

```
python transcribe_ro.py document.pdf
```

Expected: Warning message about unsupported format

## 2. Test Very Short Audio

```
python transcribe_ro.py short_clip.mp3 # < 5 seconds
```

Expected: Tool completes successfully

## 3. Test Corrupted Audio File

Create a text file and rename it to `.mp3` , then:

```
python transcribe_ro.py fake_audio.mp3
```

Expected: Clear error message about invalid audio

## 4. Test Without Internet (After Models Downloaded)

1. Disconnect from internet
2. Run transcription with `--no-translate`

```
python transcribe_ro.py sample.mp3 --no-translate --model tiny
```

Expected: Transcription works offline

## 5. Test Translation Without Internet

1. Disconnect from internet
2. Run transcription with translation enabled

```
python transcribe_ro.py sample.mp3 --model tiny
```

Expected: Warning about translation failure, but transcription completes

## Checklist Summary

---

- ☐ All files present
- ☐ Python syntax valid
- ☐ Python version  $\geq 3.8$
- ☐ FFmpeg installed
- ☐ Setup script runs successfully
- ☐ Virtual environment created
- ☐ Dependencies installed
- ☐ Help command works
- ☐ Version command works
- ☐ Basic transcription works
- ☐ All output formats work
- ☐ No-translate option works
- ☐ Error handling works correctly
- ☐ Different models work
- ☐ Portable installation works

- [ ] Edge cases handled gracefully

## Reporting Issues

If any test fails, please report the issue with:

1. Your operating system and version
2. Python version ( `python --version` )
3. FFmpeg version ( `ffmpeg -version` )
4. The exact command you ran
5. The complete error message
6. Any other relevant information

## Automated Test Script

For quick verification, you can run this automated test:

```
# Save this as test_installation.sh (Linux/macOS)
#!/bin/bash
echo "Testing Transcribe R0 Installation..."
echo "===== "

echo -n "1. Python syntax check... "
python3 -m py_compile transcribe_ro.py && echo "✓" || echo "✗"

echo -n "2. Help command... "
python transcribe_ro.py --help > /dev/null 2>&1 && echo "✓" || echo "✗"

echo -n "3. Version command... "
python transcribe_ro.py --version > /dev/null 2>&1 && echo "✓" || echo "✗"

echo "===== "
echo "Basic tests completed!"
```

Run with: `chmod +x test_installation.sh && ./test_installation.sh`