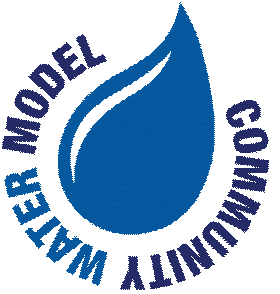
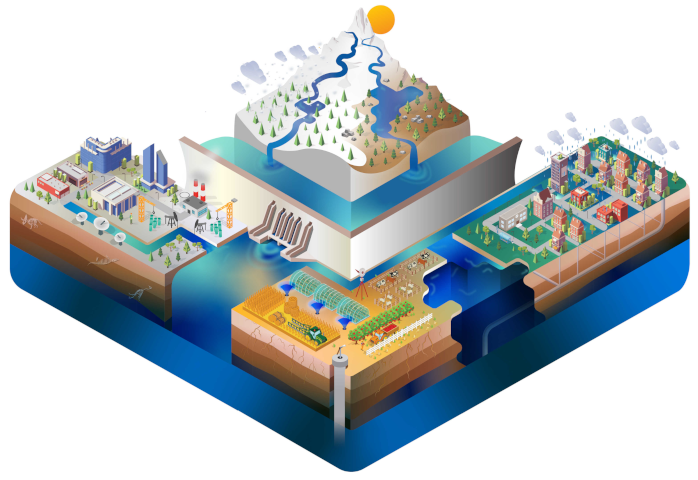
**CWatM GUI Application**



**Documentation and User Manual**

Community Water Model (CWatM) by IIASA

*Version 1.10 -September 2025*



**Table of Contents**

1. Introduction

2. Installation and Requirements

3. Getting Started

4. User Interface Overview

5. Step-by-Step User Manual

6. Advanced Features

7. Data Visualization Tools

8. Technical Documentation

9. Troubleshooting

10. Frequently Asked Questions

11. Contact and Support

**1. Introduction**

The CWatM GUI (Graphical User Interface) is a comprehensive desktop application designed for the Community Water Model (CWatM) developed by the International Institute for Applied Systems Analysis (IIASA). This application provides an intuitive interface for loading, parsing, editing, and managing CWatM configuration files, as well as running the model directly from the GUI.

**Key Benefits:**

• User-friendly interface eliminates command-line complexity

• Real-time configuration file editing with syntax highlighting

• Integrated model execution with progress tracking

• Advanced data visualization tools for masks and basins

• Comprehensive error handling and logging

**Target Audience:**

This application is designed for researchers, hydrologists, water resource managers, and students working with the CWatM hydrological model.

**2. Installation and Requirements**

The application includes an executable version (CWatM\_GUI.exe) for Windows.

No Python or any other installation required.

The GUI is using the CWatM library:



**3. Getting Started**

**Quick Start Guide:**

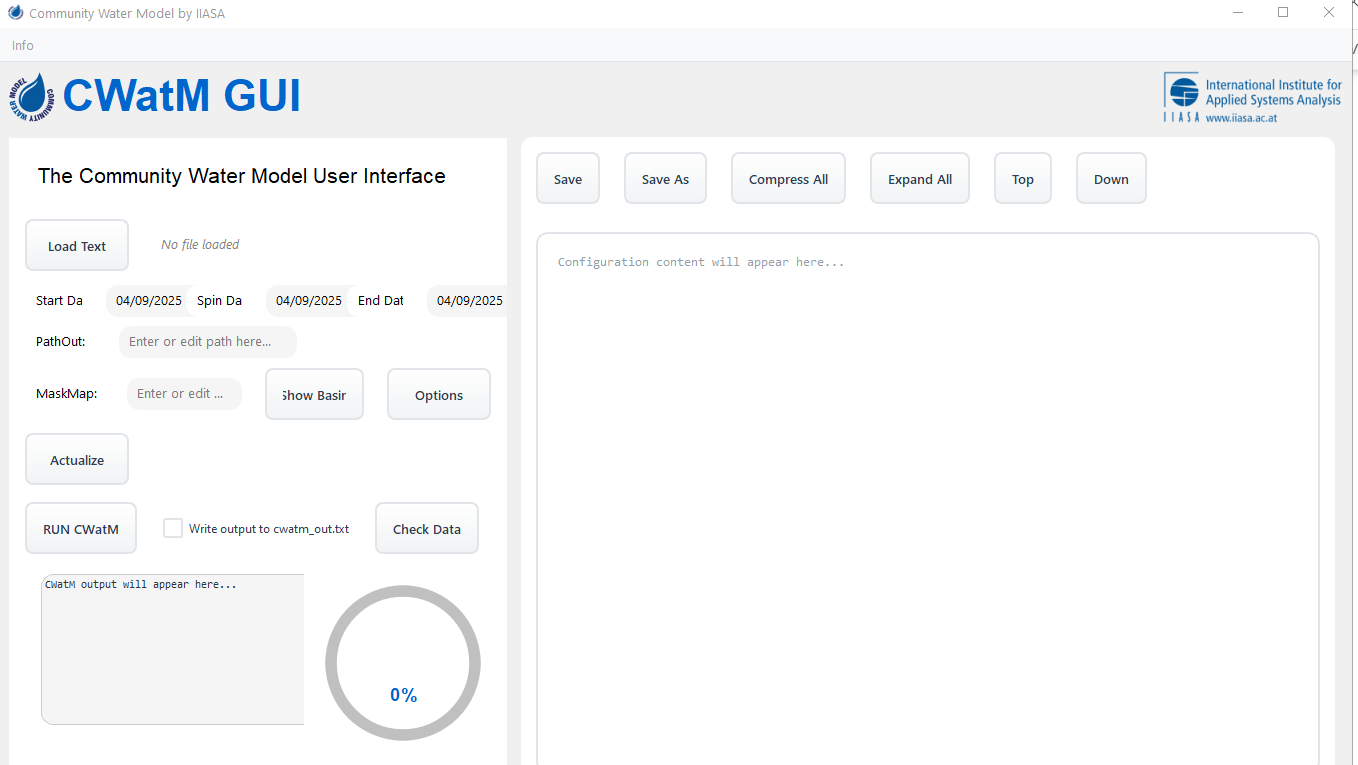


Fig 1: Application after opening

**Step 1: Launch the Application**

* Double-click CWatM\_GUI.exe (Windows)
* the application shows a startup picture – it takes a minute to start
* The application opens in maximized window mode
* You'll see the main interface with left control panel and right display area

**Step 2: Load a Configuration File**

* Click the "Load Text" button (light blue)
* Select a CWatM settings file (.ini format)
* The file automatically parses with syntax highlighting
* Configuration sections appear with expand/collapse controls

**Step 3: Modify Settings (Optional)**

* Adjust dates in the Start Date, Spin Date, and End Date fields
* Modify PathOut and MaskMap paths as needed
* Click "Options" to manage boolean configuration settings
* The "Actualize" button becomes light blue when changes are detected

**Step 4: Run CWatM**

* Click "Actualize" if you made changes to save them
* The "RUN CWatM" button becomes blue after successful parsing
* Optional: Check "Write output also to cwatm\_out.txt" to save output to file
* Click "RUN CWatM" to start model execution
* Monitor progress with the circular progress clock

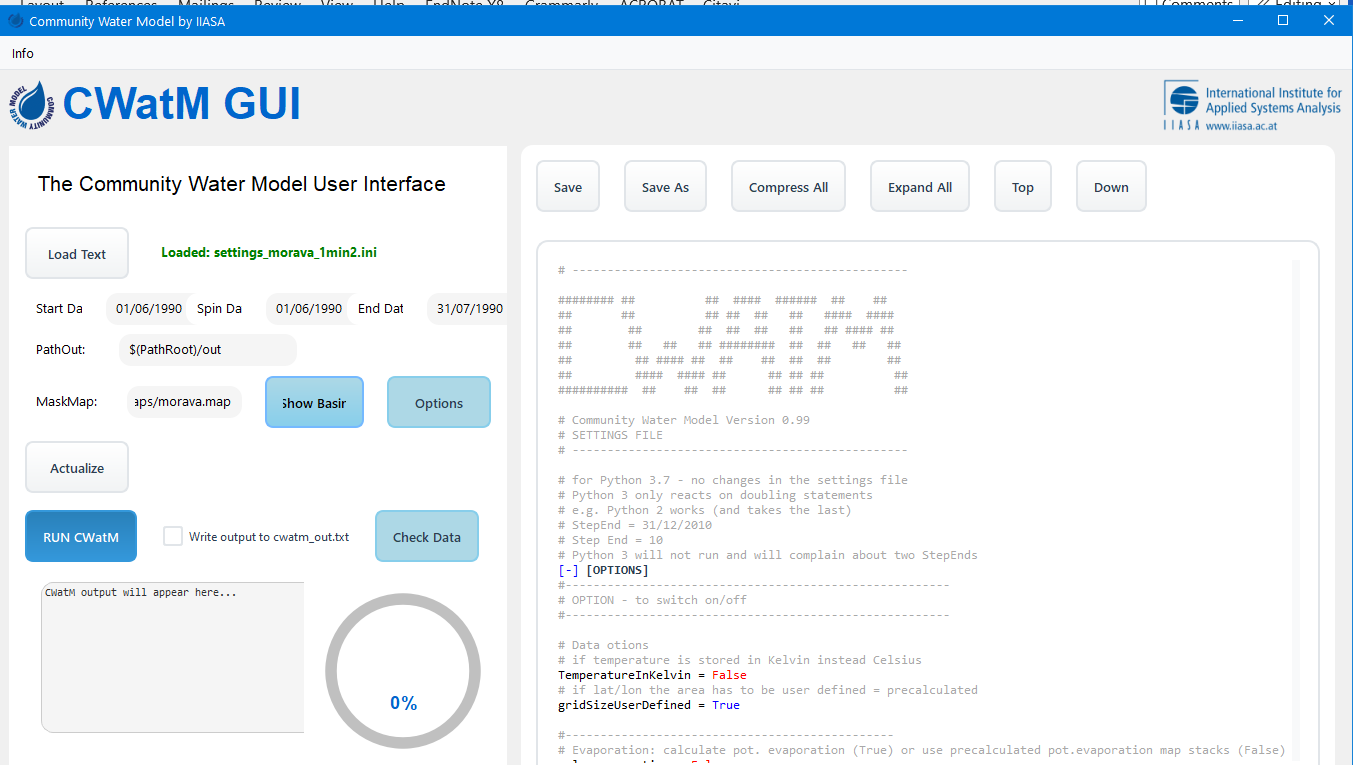


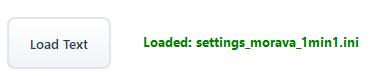
Fig 2: Application with loaded settings file

**4. User Interface Overview**

**Left Control Panel Components:**

**File Loading Area:**

* "Load Text" button for loading configuration files
* Filename display showing currently loaded file
* Automatic parsing indicator



**Date Management Fields:**

* Start Date (StepStart): Model simulation start date
* Spin Date (SpinUp): Model spin-up completion date
* End Date (StepEnd): Model simulation end date
* Real-time validation ensures chronological order

**Path Configuration:**

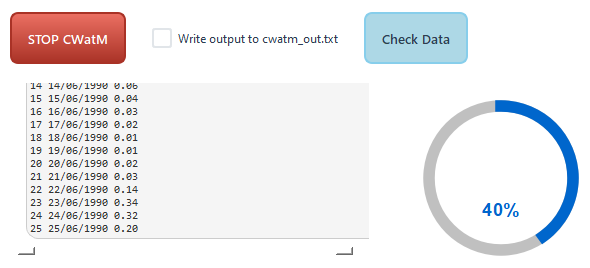
* PathOut: Output directory path
* MaskMap: Mask file path for the model domain
* "View Mask" and "Show Basin" buttons for data visualization

**Action Buttons:**

* "Actualize": Save changes to configuration file (light blue when changes detected)
* "Options": Open boolean configuration management window
* "RUN CWatM": Execute the model (blue when ready, red during execution)
* "Write output also to cwatm\_out.txt": Optional file logging checkbox

**Output Display Area:**

* CWatM Output: Real-time model execution output (225-450px height)
* Progress Clock: Circular indicator showing simulation progress (0-100%)
* Color-coded messages (black for normal, dark red for errors)



**Right Display Panel Components:**

**Toolbar Buttons:**

* Save: Save current file with clean formatting
* Save As: Save to new file location
* Compress All: Collapse all configuration sections
* Expand All: Expand all configuration sections
* Top: Jump to beginning of file
* Down: Jump to end of file

**Text Display Area:**

* Syntax-highlighted configuration content
* Interactive section headers with [-]/[+] controls
* Click-to-toggle expand/collapse functionality
* Preserved original file formatting and whitespace



Fig 3: Right window with settingsfile display

**5. Step-by-Step User Manual**

**5.1 Loading and Parsing Configuration Files**

**Procedure:**

1. Launch CWatM GUI application

2. Click "Load Text" button (always displayed in light blue)

3. Navigate to your CWatM settings file (.ini extension)

4. Select the file and click "Open"

5. Wait for automatic parsing to complete

6. Observe syntax highlighting and section formatting

**Expected Results:**

- Filename appears next to Load Text button

- Configuration content displays with color formatting:

• Comments (# lines) in dark grey

• True values in blue

• False values in red

• Section headers in bold with [-] collapse controls

- Date fields automatically populate from file content

- PathOut and MaskMap fields populate if present

- "RUN CWatM" button becomes blue indicating readiness

**5.2 Managing Dates and Validation**

**Date Field Usage:**

1. **Start Date (StepStart):** Enter model simulation start date

2. **Spin Date (SpinUp):** Enter spin-up completion date

3. **End Date (StepEnd):** Enter model simulation end date

**Supported Date Formats:**

- DD/MM/YYYY (e.g., 01/01/2020)

- D/M/YYYY (e.g., 1/1/2020)

- DD-MM-YYYY (e.g., 01-01-2020)

- YYYY-MM-DD (e.g., 2020-01-01)

**Automatic Validation:**

- System ensures chronological order: Start â‰¤ Spin â‰¤ End

- Invalid dates highlighted with error messages

- "Actualize" button remains disabled until dates are valid

- Real-time validation as you type

**5.3 Path Configuration Management**

**PathOut Configuration:**

1. Enter or modify the output directory path

2. Use absolute paths for reliability

3. Ensure directory exists or will be created by model

4. Path changes trigger "Actualize" button activation

**MaskMap Configuration:**

1. Specify path to domain mask file

2. Supports various raster formats (NetCDF, GeoTIFF, etc.)

3. Use "View Mask" button to visualize mask data

4. Path changes trigger "Actualize" button activation

**5.4 File Operations and Management**

**Actualizing Changes:**

1. Make changes to dates, paths, or options

2. "Actualize" button becomes light blue

3. Click "Actualize" to save changes

4. File automatically re-parses with clean formatting

5. Status message confirms successful save

6. Cursor jumps to StepStart parameter

**Save Operations:**

- **Save:** Saves to current file location

- **Save As:** Prompts for new file location

- Both operations save clean content without visual indicators

- All sections automatically expand before saving

- Original file formatting and spacing preserved

**Section Management:**

- **Compress All:** Collapses all sections for easier navigation

- **Expand All:** Expands all sections for full content view

- Individual section toggle by clicking [-] or [+] controls

- Manual edits preserved during compression/expansion

- Scroll position maintained during operations

**5.5 Running CWatM Model**

**Pre-execution Checklist:**

1. Configuration file successfully loaded and parsed

2. Dates validated and in chronological order

3. PathOut and MaskMap paths configured correctly

4. Any changes actualized and saved

5. "RUN CWatM" button displays in blue color

**Execution Process:**

1. Optional: Check "Write output also to cwatm\_out.txt" for file logging

2. Click "RUN CWatM" button to start execution

3. Button changes to "STOP CWatM" and becomes light red

4. Progress clock begins showing simulation advancement

5. Real-time output appears in CWatM Output area

6. Monitor progress percentage and execution messages

**Sometimes the Run CWATM will show and error. Please restart again. The program sometimes has a hick-up and will run correctly if you restart again**

**During Execution:**

- Model runs in separate thread (GUI remains responsive)

- Output updates immediately in cwatminfo area

- Progress clock shows percentage completion (0-100%)

- Error messages display in dark red color

- Normal messages display in black color

- Auto-scroll keeps latest output visible

**Stopping Execution:**

1. Click "STOP CWatM" button (light red during execution)

2. Model execution stops gracefully within 3 seconds

3. Open file handles and resources automatically cleaned up

4. Button returns to "RUN CWatM" state

5. Progress clock maintains last progress value

**Output File Logging:**

- When checkbox checked, creates "cwatm\_out.txt" in settings file directory

- File overwritten on each new CWatM execution

- Contains all output that appears in GUI cwatminfo area

- Includes timestamp and settings file information header

- Real-time writing with immediate flush for reliability

**6. Advanced Features**

**6.1 Using the Options Window**

**Opening Options Window:**

1. Click "Options" button (150x50px)

2. Dedicated window opens for boolean configuration management

3. Window displays all boolean options from [Options] section

**Managing Boolean Options:**

1. View all True/False configuration parameters

2. Click checkboxes to toggle values (True/False)

3. Changes apply immediately to configuration

4. "Actualize" button colors light blue when options change

5. Close options window when finished

**Options Window Features:**

- Clean, modern interface with CWatM branding

- Automatic detection of boolean parameters

- Real-time configuration updates

- Professional styling with hover effects

- Auto-expansion of relevant configuration sections

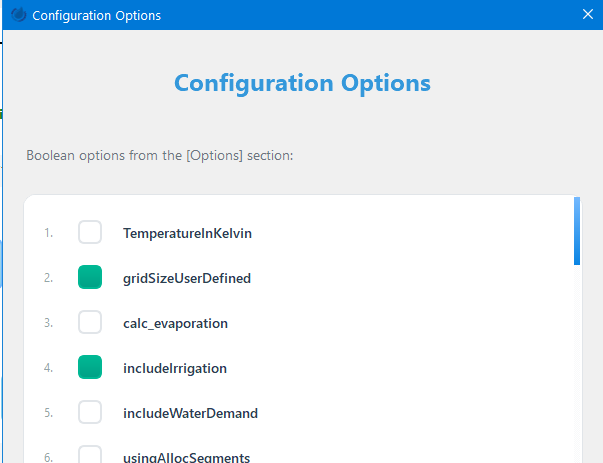


Fig 4: Option window

**6.2 Basin Viewer Utility**

**Advanced NetCDF Visualization:**

- Comprehensive basin data display with native Qt rendering

- No matplotlib dependency for improved performance

- Real-world coordinate system integration

- Multi-layer data visualization capabilities

**Interactive Features:**

1. **Mouse Wheel Zoom:** Precise zoom control at cursor position

2. **Button Zoom:** Dedicated zoom in/out controls

3. **Click and Drag Pan:** Smooth panning with coordinate tracking

4. **Coordinate Display:** Real-time lat/lon coordinates on click

5. **Value Extraction:** Basin and mask values at clicked locations

**Visualization Capabilities:**

- **UPS Data Visualization:** Viridis-like colormap for data values

- **Semi-transparent Overlays:** Green mask overlay with adjustable transparency

- **Coordinate Systems:** Automatic coordinate transformation

- **Data Integration:** Seamless NetCDF and raster data combination

- **Performance:** Native Qt painting for fast rendering

**Using Basin Viewer:**

1. Load configuration file with basin/NetCDF data references

2. Click "Show Basin" button in control panel

3. Select NetCDF file when prompted (if not auto-detected)

4. Interactive basin visualization opens

5. Use mouse controls for navigation and data exploration

6. Click on locations to view coordinates and data values

7. Close window when analysis is complete

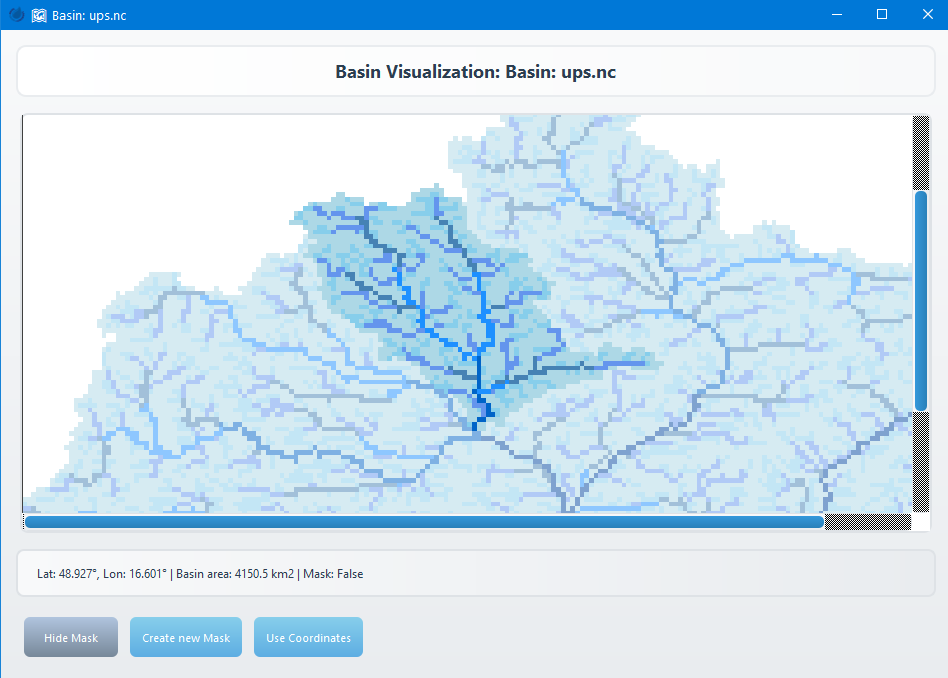


Fig 5: Basin visualization window

**6.3 Data Validation and Configuration Checking**

**Check Data Window:**

- Dedicated window for validating CWatM configuration files

- Runs CWatM in check mode (-c flag) to analyze data without full execution

- Optional comparison against existing discharge NetCDF files

- Results saved to CSV format for further analysis

- Real-time results display with interactive table showing sortable columns

- Error detection for configuration issues, missing files, and data inconsistencies

**Settings Restoration from NetCDF:**

- **Restore Settings Button:** "Restore settings from discharge map" located below  
 NetCDF file selection

- **Conditional Activation:** Button only enabled when a discharge NetCDF file is selected

- **Automatic Extraction:** Reads 'version\_settingsfile' global attribute from NetCDF files

- **Output Format:** Saves as "settings\_restore\_dischargenc.ini" in ASCII UTF-8 format

- **NetCDF4 Integration:** Requires NetCDF4 library for reading global attributes

**Data Validation Workflow:**

1. Open Check Data Window from main interface

2. Select output file for CSV results

3. Optional: Select discharge NetCDF file for comparison analysis

4. Optional: Use "Restore settings from discharge map" to extract configuration

5. Run check to execute CWatM in validation mode

6. Review detailed results table with file paths, parameters, and validation status

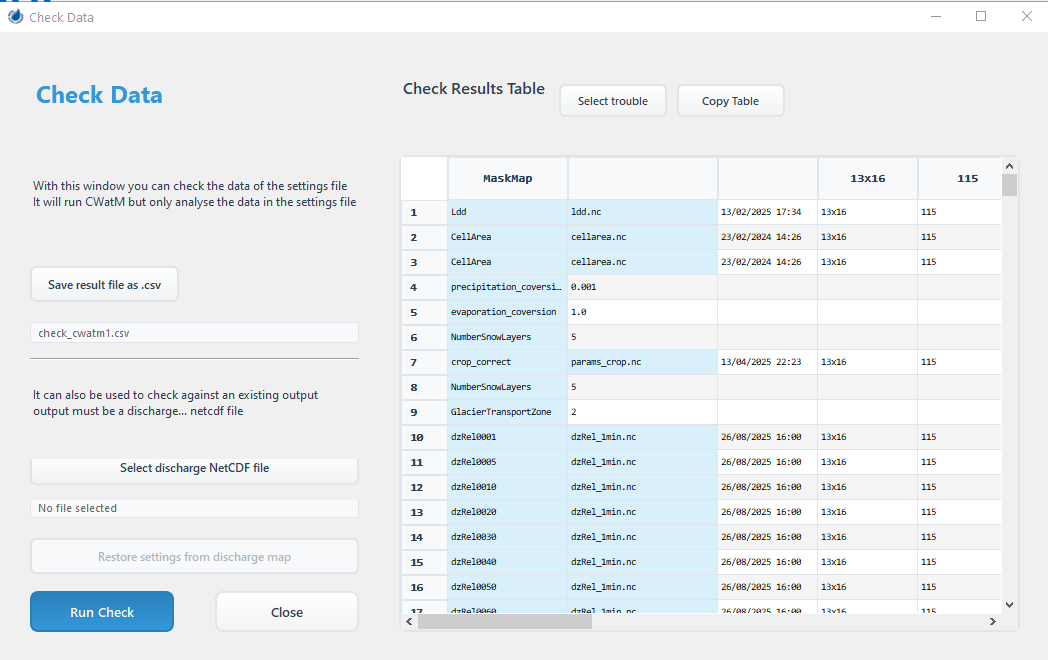


Fig 6: Check Data Window

**6.4 Workflow Guidance System**

**Dynamic Button Coloring:**

The GUI provides intelligent visual guidance through workflow progression:

1. **Load Text:** Always light blue (starting point)

2. **Actualize:** Becomes light blue when changes detected

3. **RUN CWatM:** Becomes blue after successful parsing

4. **STOP CWatM:** Becomes light red during execution

**Change Detection:**

- Monitors Start Date, Spin Date, End Date fields

- Tracks PathOut and MaskMap field modifications

- Detects boolean option changes in Options window

- Smart reset after successful operations

- Real-time color updates for immediate feedback

**7. Technical Documentation**

**7.1 Application Architecture**

**Core Components:**

- **main\_window.py:** Main application window orchestrating all components

- **config\_parser.py:** INI file parsing, validation, and formatting

- **date\_manager.py:** Date input widgets and validation logic

- **file\_manager.py:** File I/O operations and management

- **text\_display.py:** Text area operations and cursor management

**Specialized Widgets:**

- **options\_window.py:** Boolean configuration management interface

- **progress\_clock.py:** Circular progress indicator for model execution

- **cwatm\_worker.py:** Threaded CWatM execution worker

- **mask\_viewer.py:** Mask data visualization utility

- **basin\_viewer.py:** Advanced NetCDF basin data visualization

- **check\_data\_window.py:** Data validation window for CWatM configuration checking with NetCDF comparison and settings restoration

**7.2 Threading and Concurrency**

**Thread Architecture:**

- **Main Thread:** GUI operations and user interaction handling

- **Worker Thread:** CWatM model execution (CWatMWorker class)

- **Signal-Slot Communication:** Qt signals for thread-safe communication

- **Resource Management:** Automatic cleanup and garbage collection

**Thread Safety Measures:**

- All GUI updates performed on main thread via signals

- Worker thread isolation prevents GUI freezing

- Cooperative stop mechanism with timeout fallback

- Exception handling within threads

- Safe resource cleanup on thread termination

**7.3 File Format Support**

**Configuration Files:**

- **Primary Format:** INI configuration files (.ini)

- **Compatibility:** Standard Python configparser format

- **Encoding:** UTF-8 with fallback encoding detection

- **Preservation:** Original formatting and whitespace maintained

**Data Files:**

- **NetCDF:** Full NetCDF4 support for basin and meteorological data

- **Raster Formats:** GeoTIFF, ESRI Grid, HDF5 via rasterio

- **Text Files:** Various text formats for configuration and logs

- **Binary Formats:** Support for CWatM-specific binary data files

**7.4 Print Redirection System**

**Implementation Details:**

- Separate redirectors for stdout and stderr

- Signal-based communication for thread safety

- Real-time text emission without buffering

- Color-coded error message handling

- Integration with file logging system

**7.5 Configuration Management**

**INI File Processing:**

- Automatic section detection and parsing

- Boolean option recognition and management

- Date field extraction and validation

- Path parameter detection and verification

- Comment preservation and syntax highlighting

**Data Validation:**

- Real-time date format validation

- Chronological order enforcement

- Path existence verification

- Parameter type checking

- Configuration consistency validation

**7.6 Memory and Performance**

**Resource Optimization:**

- Output buffer size limitation (100 lines maximum)

- Automatic memory cleanup during operations

- Efficient HTML rendering for syntax highlighting

- Smart garbage collection for large data objects

- NetCDF dataset closure and resource cleanup

**Display Performance:**

- Immediate output updates without artificial delays

- Smart scrolling maintains user viewing context

- Optimized rich text formatting

- Responsive interface during model execution

- Minimal CPU usage during idle states

**7.7 Error Handling and Recovery**

**Global Exception Handling:**

- Comprehensive error handling prevents application crashes

- SystemExit calls intercepted to prevent unexpected termination

- All exceptions logged to cwatminfo area in dark red

- Application continues running after errors

- User-friendly error messages with context

**File Operation Safety:**

- Automatic backup of original file formatting

- Graceful handling of file access permissions

- Recovery from temporary file system issues

- Validation of file paths and existence

- Clean rollback on operation failures

**Model Execution Safety:**

- Thread-safe model execution with proper cleanup

- Automatic closure of NetCDF datasets and file handles

- Cooperative stop mechanism with timeout fallback

- Resource leak prevention through garbage collection

- Safe handling of model errors and exceptions

**7.8 Performance Optimization**

**Memory Management:**

- Output buffer limited to last 100 lines

- Automatic memory cleanup during operations

- Efficient HTML rendering for syntax highlighting

- Optimized scroll position tracking

- Resource-conscious background processing

**Display Optimization:**

- Immediate output updates without buffering delays

- Smart scrolling maintains user viewing position

- Rich text formatting with minimal performance impact

- Responsive UI during model execution

- Efficient syntax highlighting rendering

**8. Troubleshooting**

**8.1 Common Issues and Solutions**

**Problem:**

Configuration file won't load

**Solution:**

1. Verify file is valid INI format

2. Check file permissions and accessibility

3. Ensure file encoding is UTF-8 or compatible

4. Look for syntax errors in configuration file

5. Try loading a different known-good configuration file

**Problem:**

Date validation errors

**Solution:**

1. Use supported date formats: DD/MM/YYYY, DD-MM-YYYY, YYYY-MM-DD

2. Ensure chronological order: Start â‰¤ Spin â‰¤ End

3. Check for typos in date fields

4. Verify dates are within reasonable ranges

5. Clear fields and re-enter dates if validation seems stuck

**Problem:**

CWatM execution fails

**Solution:**

1. Verify all file paths in configuration exist

2. Check PathOut directory is writable

3. Ensure MaskMap file is accessible

4. Review error messages in cwatminfo area

5. Validate configuration file syntax

6. Check system resources (RAM, disk space)

**Problem:**

GUI becomes unresponsive

**Solution:**

1. Wait for current operation to complete

2. Use "STOP CWatM" button if model is running

3. Close and restart application if necessary

4. Check system resource usage

5. Reduce output buffer size if dealing with verbose model output

**8.2 Advanced Troubleshooting**

**Log File Analysis:**

1. Enable "Write output also to cwatm\_out.txt" checkbox

2. Review cwatm\_out.txt file in settings directory

3. Look for error patterns and specific failure points

4. Compare successful runs with failed attempts

5. Check timestamps for performance analysis

**Configuration Validation:**

1. Use CWatM command line tools to validate settings independently

2. Check all referenced files exist and are accessible

3. Verify parameter ranges and types

4. Test with minimal configuration first

5. Gradually add complexity to isolate issues

**8.3 Performance Issues**

**Large Dataset Handling:**

- Increase system virtual memory if needed

- Use data compression where possible

- Consider chunking large NetCDF files

- Monitor memory usage during visualization

- Close visualization windows when not needed

**8.4 Getting Additional Help**

**Error Reporting:**

1. Capture exact error messages from GUI

2. Note system specifications and versions

3. Document steps to reproduce the issue

4. Save problematic configuration files

5. Include relevant log file excerpts

**Community Resources:**

- CWatM documentation and user guides

- IIASA CWatM support forums

- GitHub repository for bug reports and features

- Academic papers and technical documentation

- User community discussions and solutions

9**. Frequently Asked Questions**

**General Usage Questions**

**Q: Can I run multiple CWatM simulations simultaneously?**

**A:** No, the GUI supports one CWatM execution at a time. This prevents resource conflicts and ensures proper output management.

**Q: What happens to my manual edits when I use Compress/Expand All?**

**A:** All manual edits are preserved. The application maintains a separate buffer for user modifications that persist through formatting operations.

**Q: Can I edit configuration files while CWatM is running?**

**A:** Yes, you can edit and save configuration files during model execution. However, changes won't affect the currently running simulation.

**Q: How do I know if my configuration file is valid?**

**A:** The GUI performs automatic validation during parsing. Invalid configurations will show error messages, and the "RUN CWatM" button will remain disabled.

**Technical Questions**

**Q: What file formats are supported for mask and basin data?**

**A:** The application supports NetCDF, GeoTIFF, ESRI Grid, HDF5, and other formats supported by rasterio and xarray libraries.

**Q: How much memory does the application require?**

**A:** Minimum 4GB RAM recommended. Memory usage depends on dataset sizes and model complexity. Large NetCDF files may require 8GB+ RAM.

**Q: Can I use the GUI on Linux or macOS?**

**A:** Yes, the application is cross-platform compatible. Install Python 3.8+ and required packages, then run from source code.

**Q: Where are temporary files stored?**

**A:** Temporary files are created in the same directory as your configuration file. Log files (cwatm\_out.txt) are also stored there.

**Advanced Usage Questions**

**Q: How can I automate repetitive tasks?**

**A:** While the GUI focuses on interactive use, you can prepare template configuration files and use the command-line CWatM for batch processing.

**Q: Can I customize the GUI appearance or add features?**

**A:** The application is open-source with modular architecture. Advanced users can modify the source code to add custom features.

**Q: How do I handle very large datasets?**

**A:** Use data compression, ensure adequate system resources, and consider chunking large NetCDF files. Monitor memory usage during processing.

**Q: What should I do if the progress clock seems stuck?**

**A:** The progress clock updates based on CWatM's internal timestep reporting. Some model phases may take longer without progress updates.

**10. Contact and Support**

**IIASA Contact Information**

**International Institute for Applied Systems Analysis (IIASA)**

Schlossplatz 1

A-2361 Laxenburg

Austria

Phone: +43 2236 807 0

Email: info@iiasa.ac.at

Website: www.iiasa.ac.at

**CWatM Project Support**

**Technical Support:**

- CWatM Documentation: Available through IIASA website

<https://cwatm.iiasa.ac.at/>

- User Guides: Comprehensive documentation for model usage

- GitHub Repository: Source code and issue tracking

<https://github.com/iiasa/CWatM>

- Academic Publications: Peer-reviewed papers on methodology

Burek, P., Satoh, Y., Kahil, T., Tang, T., Greve, P., Smilovic, M., Guillaumot, L., Zhao, F., and Wada, Y.: Development of the Community Water Model (CWatM v1.04) - a high-resolution hydrological model for global and regional assessment of integrated water resources management, Geosci. Model Dev., 13, 3267–3298, https://doi.org/10.5194/gmd-13-3267-2020.

And on: <https://cwatm.iiasa.ac.at/publication.html>

**Community Resources:**

- User Forums: Community discussions and problem-solving

<https://github.com/iiasa/CWatM/discussions>

- Training Workshops: Periodic training sessions for new users

- Academic Collaborations: Research partnerships and applications

- Case Studies: Real-world applications and examples

**Software Development and Contributions**

**Development Team:**

- CWatM GUI Application developed by IIASA research team

<https://cwatm.iiasa.ac.at/publication.html#developer>

- Contributions welcome from the scientific community

- Open-source development model with peer review

- Regular updates and feature enhancements

**Reporting Issues:**

1. Use GitHub repository for bug reports and feature requests

2. Provide detailed error messages and system information

3. Include steps to reproduce problems

4. Attach relevant configuration files (remove sensitive data)

5. Specify software versions and system specifications

**Citations and Academic Use**

**Citing the CWatM GUI:**

When using the CWatM GUI in academic work, please cite:

"CWatM GUI Application, International Institute for Applied Systems Analysis (IIASA), Laxenburg, Austria. Available at: [https://github.com/iiasa/CWatM\_GUI]"

**Related Publications:**

- Refer to IIASA website for current list of CWatM-related publications

- Include relevant methodology papers in your references

- Acknowledge IIASA and the CWatM development team

- Follow journal-specific citation guidelines