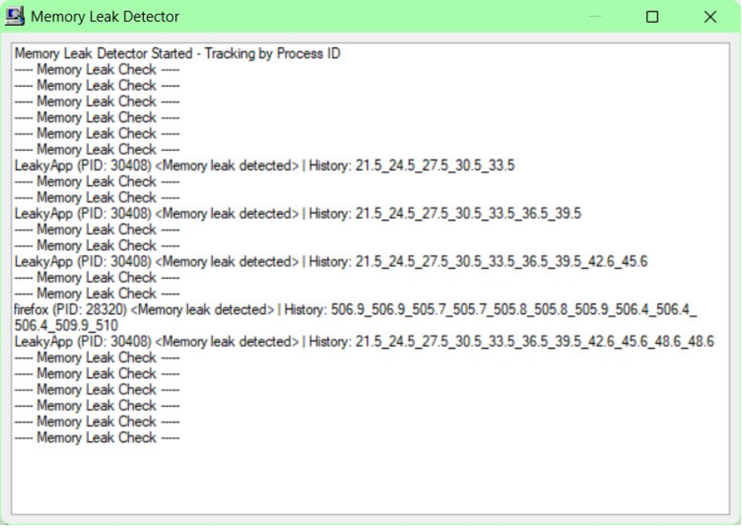
Memory Leak Detector for Windows
Version: 1.3
**Last Updated: February,12 2026
Language: PureBasic 5.46x64 LTS or above

A real-time Windows memory leak detection tool built with PureBasic. Monitors all running processes and alerts you when sustained memory growth patterns indicate potential memory leaks.



```
Memory Leak Detector Started - Tracking by Process ID
----- Memory Leak Check -----
----- Memory Leak Check -----
----- Memory Leak Check -----
----- Memory Leak Check -----
----- Memory Leak Check -----
----- Memory Leak Check -----
LeakyApp (PID: 30408) <Memory leak detected> | History: 21.5_24.5_27.5_30.5_33.5
----- Memory Leak Check -----
----- Memory Leak Check -----
LeakyApp (PID: 30408) <Memory leak detected> | History: 21.5_24.5_27.5_30.5_33.5_36.5_39.5
----- Memory Leak Check -----
----- Memory Leak Check -----
LeakyApp (PID: 30408) <Memory leak detected> | History: 21.5_24.5_27.5_30.5_33.5_36.5_39.5_42.6_45.6
----- Memory Leak Check -----
----- Memory Leak Check -----
firefox (PID: 28320) <Memory leak detected> | History: 506.9_506.9_505.7_505.7_505.8_505.8_505.9_506.4_506.4_
506.4_509.9_510
LeakyApp (PID: 30408) <Memory leak detected> | History: 21.5_24.5_27.5_30.5_33.5_36.5_39.5_42.6_45.6_48.6_48.6
----- Memory Leak Check -----
----- Memory Leak Check -----
----- Memory Leak Check -----
----- Memory Leak Check -----
----- Memory Leak Check -----
----- Memory Leak Check -----
```

Features

- **Process ID Tracking**: Monitors each process instance individually by ProcID
- **Real-time Monitoring**: Continuous memory snapshots with configurable intervals
- **Smart Detection**: Distinguishes between legitimate memory usage and actual leaks using consecutive growth patterns
- **Configurable Thresholds**: Tune sensitivity to match your monitoring needs
- **System Process Filtering**: Automatically excludes common Windows system processes
- **Detailed Logging**: Timestamps and memory history for each detected leak
- **Live Display**: Shows detected leaks in real-time with process name and PID
- **Grace Period**: Ignores normal startup memory allocation

Requirements

- **PureBasic 5.46 LTS or later** (Windows x64) to Compile
- **Windows PowerShell** (included with Windows 7+)
- **Windows Operating System** (tested on Windows 11)

Quick Start

1. Download LeakDetector OR Compile `Leakdetector3.pb` with PureBasic
2. Run the executable
3. Configure settings (or use defaults)
4. Click **Apply** to start monitoring
5. Watch the output window for detected leaks
6. Check `MemoryLeakLog.txt` for detailed history any time or after system crash.

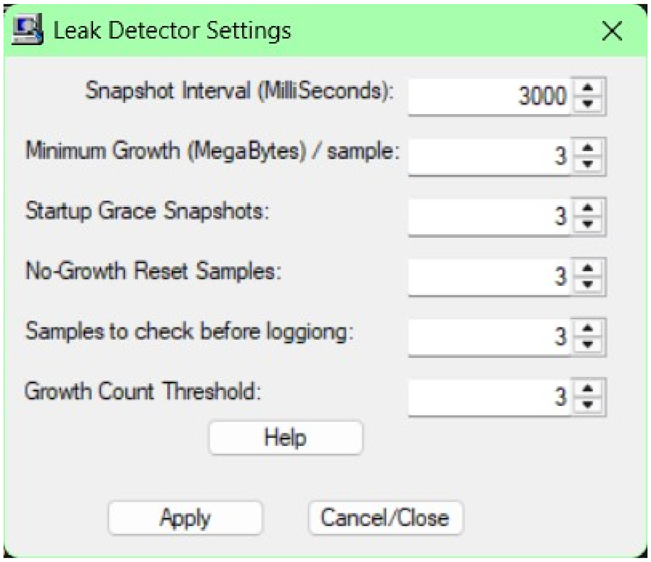
How It Works

The detector uses PowerShell to capture memory snapshots of all running processes at regular intervals. It analyzes memory growth patterns over time and flags processes that show sustained growth exceeding your configured thresholds.

Detection Logic:

1. Takes memory snapshots at configured intervals
2. Tracks memory history for each process (by ProcID)
3. Examines recent snapshots for consecutive growth
4. Counts growth hits when increase exceeds minimum threshold
5. Logs leak when hit count reaches threshold
6. Resets counter if no growth detected for specified samples

Configuration Settings (showing defaults)



Leak Detector Settings

Snapshot Interval (MilliSeconds): 3000

Minimum Growth (MegaBytes) / sample: 3

Startup Grace Snapshots: 3

No-Growth Reset Samples: 3

Samples to check before loggiong: 3

Growth Count Threshold: 3

Default Snapshot Interval: 3000 ms
Default Minimum Growth: 3 MB
Default Growth Count Threshold: 3

Log File (MemoryLeakLog.txt)

```
Sample entry from LeakyApp designed to leak:

2026-02-12 13:45:25 LeakyApp (PID: 23932) <Memory leak detected> | History:
20_23_26_29_32
```

Memory values are in megabytes (MB), separated by underscores showing the progression over time.

System Process Filtering

The following process types are automatically excluded from monitoring:
- Windows system processes (System, csrss, services, etc.)

- Desktop Window Manager (dwm)
- Windows services (svchost, RuntimeBroker, etc.)
- Explorer and shell processes

You can modify the exclusion list in the `GetOmitProcesses()` procedure in the source code.

Known Limitations

- **Windows Only**: Requires PowerShell (Windows-specific)
- **Snapshot Timing**: Very short-lived processes may be missed between snapshots
- **High Memory Apps**: Programs legitimately using large amounts of memory may trigger false positives (adjust thresholds accordingly)
- **Process Restarts**: Each process restart creates a new ProcID with fresh history

Testing

A test application (`LeakyApp.pb`) is available that intentionally leaks 3 MB every 3 seconds. Use this to verify your configuration and understand detection behavior.

To test:
1. Download or Compile `LeakyApp.exe`
2. Start the Memory Leak Detector
3. Click "Apply" with default settings
4. start the leakyApp
5. LeakyApp should be detected within 15-20 seconds

Development Credits

Original Concept & Initial Implementation: Randy Walker

Core construction by ChatGPT
- Core detection algorithm
- Settings configuration interface
- PowerShell integration

Major Revisions, Enhancements & Documentation by Claude (Anthropic AI):
- Refactored to track by Process ID instead of process name
- Fixed PowerShell output parsing and format string handling
- Redesigned system process filtering (omit list)
- Added ProcNames map for ProcID-to-name tracking
- Enhanced output to display both process name and PID
- Rewrote help window documentation for clarity
- Added comprehensive debugging output
- Created README and user documentation

Contributing

Contributions welcome! Please feel free to submit pull requests or open issues for bugs and feature requests.

License

Not a lawyer – It’s free for any use.

Forum & Support

Discuss this tool on the [PureBasic Forum](https://www.purebasic.fr/english