

efforthye / fast-filesystem-mcp Public

- <> Code
- Issues 6
- Pull requests 2
- Discussions
- Actions
- Projects
- Security

1 Branch
 6 Tags

<b>efforthye</b> docs: update readme.md	471bc5b · 5 months ago
.github	docs: create funding.yml 8 months ago
api	Fix MCP protocol issues for Cla... 8 months ago
public	Link last year
scripts	fix: implement safe MCP logger ... 6 months ago
src	feat: fast_read_multiple_files ne... 6 months ago
.DS_Store	Release v3.5.0: Enhanced Sear... 7 months ago
.gitattributes	Add .gitattributes for correct lan... 7 months ago
.gitignore	Add .gitattributes for correct lan... 7 months ago
.vercelignore	Disable Vercel deployment - swi... 8 months ago
CHANGELOG.md	fix: implement safe MCP logger ... 6 months ago
DEPLOYMENT.md	Add deployment documentation 8 months ago
LICENSE	license: add LICENSE 8 months ago
README.md	docs: update readme.md 5 months ago
glama.json	feat: create glama.json 8 months ago
package.json	fix: implement safe MCP logger ... 6 months ago
pnpm-lock.yaml	fix: add missing @vscode/ripgr... 8 months ago
tsconfig.json	Update tsconfig.json for proper ... 8 months ago
vercel.json	fix: Remove invalid runtime setti... 8 months ago

- README**
- Apache-2.0 license



# fast-filesystem-mcp

Advanced filesystem operations for Claude Desktop with large file handling capabilities and Claude-optimized features.

## Quick Start

Add to your Claude Desktop config.

- **Basic setup**

```
{
  "mcpServers": {
    "fast-filesystem": {
      "command": "npx",
      "args": ["-y", "fast-filesystem-mcp"]
    }
  }
}
```

- **With backup files enabled**

```
{
  "mcpServers": {
    "fast-filesystem": {
      "command": "npx",
      "args": ["-y", "fast-filesystem-mcp"],
      "env": {
        "CREATE_BACKUP_FILES": "true"
      }
    }
  }
}
```

## Backup Configuration

Control backup file creation behavior.

- `CREATE_BACKUP_FILES=false` (default): Disables backup file creation to reduce clutter
- `CREATE_BACKUP_FILES=true` : Creates backup files before modifications

**Note:** Backup files are created with timestamps (e.g., `file.txt.backup.1755485284402` ) to prevent data loss during edits.

## Debug and Logging Configuration

The MCP server uses a safe logging system that prevents JSON-RPC communication errors.

- `DEBUG_MCP=true` or `MCP_DEBUG=true` : Enable debug logging to stderr
- `MCP_LOG_FILE=/path/to/log.txt` : Write logs to file instead of stderr
- `MCP_SILENT_ERRORS=true` or `SILENT_ERRORS=true` : Suppress error messages in responses

**Note:** Debug output is automatically suppressed by default to prevent JSON parsing errors in Claude Desktop.

## New Version Update

---

To update to the latest version, follow these steps.

### 1. Uninstall previous version

```
npm uninstall -g fast-filesystem-mcp
```



### 2. Clean cache and dependencies

```
npm cache clean --force  
pnpm store prune
```



### 3. Install latest version

```
npm install -g fast-filesystem-mcp
```



### 4. Verify installation

```
npm list -g fast-filesystem-mcp  
fast-filesystem-mcp --version
```



## Features

---

### Core File Operations

- Fast File Reading/Writing: Optimized for Claude Desktop with chunking support
- Sequential Reading: Read large files completely with continuation token support
- Large File Handling: Stream-based writing for files of any size
- Directory Operations: Comprehensive directory listing, creation, and management
- File Search: Name and content-based file searching with filtering

### Advanced Capabilities

- Pagination Support: Handle large directories efficiently

- **Chunked Reading:** Read large files in manageable chunks
- **Streaming Writes:** Memory-efficient writing for large files
- **Backup & Recovery:** Automatic backup creation and error recovery
- **Retry Logic:** Built-in retry mechanism for reliable operations

## Performance Optimizations

- **Claude-Optimized:** Response sizes and formats optimized for Claude
- **Memory Efficient:** Streaming operations prevent memory overflow
- **Smart Exclusions:** Automatically excludes system files and directories
- **Progress Tracking:** Real-time progress monitoring for large operations

## Available Tools

### File Operations

Tool	Description
<code>fast_read_file</code>	Read files with chunking support
<code>fast_read_multiple_files</code>	Read multiple files simultaneously with sequential reading support
<code>fast_write_file</code>	Write or modify files
<code>fast_large_write_file</code>	Stream-based writing for large files
<code>fast_get_file_info</code>	Get detailed file information

### Complex File Management

Tool	Description
<code>fast_copy_file</code>	Copy files and directories with advanced options
<code>fast_move_file</code>	Move/rename files and directories safely
<code>fast_delete_file</code>	Delete files and directories with protection
<code>fast_batch_file_operations</code>	Execute multiple file operations in sequence

### Archive Management

Tool	Description
<code>fast_compress_files</code>	Create compressed archives (tar, tar.gz, tar.bz2)
<code>fast_extract_archive</code>	Extract compressed archives with options

## Directory Synchronization

Tool	Description
<code>fast_sync_directories</code>	Advanced directory synchronization with multiple modes

## Advanced Editing Tools

Tool	Description
<code>fast_edit_file</code>	Precise line-based file editing with multiple modes
<code>fast_edit_block</code>	Safe block editing with exact string matching
<code>fast_edit_blocks</code>	Batch block editing for multiple precise changes
<code>fast_edit_multiple_blocks</code>	Edit multiple sections in a single operation
<code>fast_extract_lines</code>	Extract specific lines or ranges from files

## Directory Operations

Tool	Description
<code>fast_list_directory</code>	List directory contents with pagination
<code>fast_create_directory</code>	Create directories recursively
<code>fast_get_directory_tree</code>	Get directory tree structure

## Search Operations

Tool	Description
<code>fast_search_files</code>	Search files by name or content
<code>fast_search_code</code>	Advanced code search with ripgrep integration
<code>fast_find_large_files</code>	Find large files in directories

## System Operations

Tool	Description
<code>fast_get_disk_usage</code>	Check disk usage information
<code>fast_list_allowed_directories</code>	List allowed directories

# Editing Tools

## Precise File Editing

The fast-filesystem MCP now includes powerful editing tools for source code and text files.

### `fast_edit_file` - Single Block Editing

Supports multiple editing modes.

- **replace**: Replace text or entire lines
- **replace\_range**: Replace multiple lines at once
- **insert\_before**: Insert content before specified line
- **insert\_after**: Insert content after specified line
- **delete\_line**: Delete specific lines

```
{
  "tool": "fast_edit_file",
  "arguments": {
    "path": "/path/to/file.js",
    "mode": "replace",
    "line_number": 10,
    "new_text": "const newVariable = 'updated value';",
    "backup": true
  }
}
```



### `fast_edit_multiple_blocks` - Batch Editing

Edit multiple parts of a file in a single operation.

```
{
  "tool": "fast_edit_multiple_blocks",
  "arguments": {
    "path": "/path/to/file.js",
    "edits": [
      {
        "mode": "replace",
        "old_text": "oldFunction()",
        "new_text": "newFunction()"
      },
      {
        "mode": "insert_after",
        "line_number": 5,
        "new_text": "// Added comment"
      }
    ],
    "backup": true
  }
}
```



```
}  
}
```

### fast\_extract\_lines - Line Extraction

Extract specific lines by number, range, or pattern.

```
{  
  "tool": "fast_extract_lines",  
  "arguments": {  
    "path": "/path/to/file.js",  
    "pattern": "function.*",  
    "context_lines": 2  
  }  
}
```



### fast\_search\_and\_replace - Advanced Replace

Powerful search and replace with regex support.

```
{  
  "tool": "fast_search_and_replace",  
  "arguments": {  
    "path": "/path/to/file.js",  
    "search_pattern": "console\\.log\\(\\..*\\)",  
    "replace_text": "logger.info($1)",  
    "use_regex": true,  
    "max_replacements": 10,  
    "backup": true  
  }  
}
```



## Editing Features

- **Automatic Backup:** Creates backups before modifications
- **Error Recovery:** Restores from backup on failure
- **Line-based Operations:** Precise control over specific lines
- **Pattern Matching:** Regular expression support
- **Batch Operations:** Multiple edits in single transaction
- **Context Extraction:** Extract lines with surrounding context

## Large File Writing

- **fast\_large\_write\_file**
  - **Streaming:** Writes files in chunks to prevent memory issues
  - **Backup:** Automatically creates backups before overwriting

- Verification: Verifies file integrity after writing
- Retry Logic: Automatic retry on failure with exponential backoff
- Progress Tracking: Real-time monitoring of write progress

## License

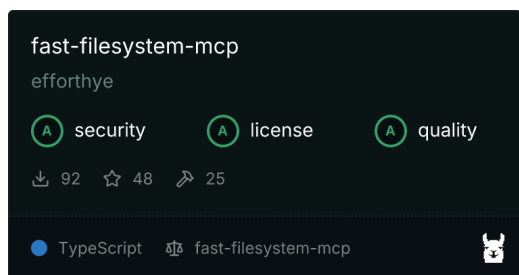
Apache 2.0

Copyright 2025 efforthye

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

## Badge



### Fast Filesystem

by efforthye  
Misc

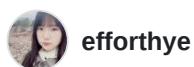
MseeP.ai

Audited

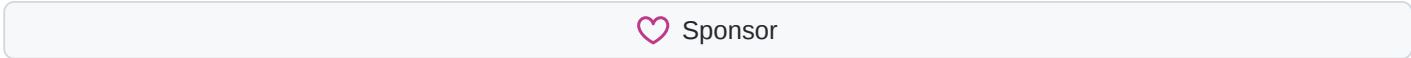
MseeP verified

## Repository

### Sponsor this project



efforthye



[Learn more about GitHub Sponsors](#)

### Contributors 5



### Languages



- TypeScript 84.0%
- JavaScript 14.8%
- Shell 1.1%
- HTML 0.1%