




Markqant (.mq) Format Specification

"Markqant isn't a format. It's AI sheet music for memory fields." 

"Same meaning. Fewer notes. Faster recall."

Overview

Markqant is a quantum-compressed markdown format designed specifically for AI consumption. It applies Smart Tree's revolutionary quantum compression techniques to markdown documents, achieving 70-90% size reduction while maintaining perfect semantic fidelity.

Format Structure

Line 1: MARKQANT_V1 2025-07-16T10:30:00Z 45678 4567 [flags]
Line 2: <token_dictionary>
Line 3+: <quantized_content>

Header Format

- MARKQANT_V1 : Version identifier
- ISO8601 timestamp: Last modification time
- Original size in bytes
- Compressed size in bytes
- Optional flags: -zlib , -streamed , -delta , -readonly , -semantic

Token Dictionary

Common markdown patterns are tokenized:

Token	Pattern	Description
T00	#	H1 header
T01	##	H2 header
T02	###	H3 header

Token	Pattern	Description
T03	####	H4 header
T04	```language	Code block start
T05	```\n	Code block end
T06	-	Unordered list item
T07	*	Alt unordered list
T08	1.	Ordered list start
T09	[text](url)	Link pattern
T0A	**text**	Bold text
T0B	*text*	Italic text
T0C	>	Blockquote
T0D	---	Horizontal rule
T0E	\n\n	Paragraph break
T0F	`	`

Dynamic Token Assignment

Patterns that appear 3+ times get dynamic tokens (T10-TFF):

- Common phrases like "Usage:", "Example:", "Note:"
- Repeated code snippets
- Common URLs or paths
- Technical terms

Section Tagging (Optional)

Content can include semantic section tags for easy extraction:

```
::section:Installation::
npm install smart-tree
::section:Usage::
st --mode quantum
```

This enables:

- Direct section extraction without parsing
- Cross-memory references in MEM|8
- Semantic navigation for AI agents

Compression Algorithm

1. **First Pass:** Scan for repeated patterns
2. **Token Assignment:** Assign tokens to patterns by frequency
3. **Encoding:** Replace patterns with tokens
4. **Dictionary:** Prepend token->pattern mapping
5. **Optional zlib:** Further compress if beneficial

Example

Original markdown (1,234 bytes):

```
# Smart Tree Documentation
```

```
## Installation
```

To install Smart Tree, run:

```
```bash
cargo install smart-tree
```

## Usage

Basic usage:

```
st --mode quantum /path/to/dir
```

Advanced usage:

```
st --mode summary-ai --depth 5 /path/to/dir
```

# Features

- **Fast:** 10x faster than tree
- **AI-friendly:** Optimized for LLMs
- **Compressed:** 99% size reduction

# Contributing

See [CONTRIBUTING.md](#) for details.

Compressed markqant (234 bytes):

```
MARKQANT_V1 2025-07-16T10:30:00Z 1234 234
T10=Smart Tree
T11="`bash
T12=st --mode
T13=/path/to/dir
T00T10 Documentation
T01Installation
To install T10, run:
T11
cargo install smart-tree
T05
T01Usage
Basic usage:
T11
T12 quantum T13
T05
Advanced usage:
T11
T12 summary-ai --depth 5 T13
T05
T01Features
T06T0AFast**: 10x faster than tree
T06T0AAI-friendly**: Optimized for LLMs
T06T0ACompressed**: 99% size reduction
```

## T01Contributing

See [CONTRIBUTING.md](#) for details.

### ## Benefits {#benefits }

1. **Token Efficiency**: 80%+ reduction in AI token usage
2. **Network Friendly**: Smaller payloads for API calls
3. **Cache Friendly**: More docs fit in context windows
4. **Pattern Recognition**: AI can learn document structure faster
5. **Version Tracking**: Built-in timestamp for freshness

### ## Implementation Notes {#implementation-notes }

- Tokens are case-sensitive
- UTF-8 encoding throughout
- Backward compatible with markdown parsers (via conversion)
- Streaming decompression supported
- Can be piped through `st --decode-mq` for human reading

### ### Streaming Mode {#streaming-mode }

A markqant decoder can begin parsing Line 3+ before dictionary is complete, if tokens a

### ## CLI Tools {#cli-tools }

### ### Visual Diagnostics {#visual-diagnostics }

```
```bash
```

```
st --inspect-mq file.mq
```

Output:

```
📄 File: file.mq
📅 Modified: 2025-07-16
📦 Compression: 81%
📖 Dictionary Size: 17 entries
🧠 High-frequency token: T10 ("Smart Tree")
🚦 Sections: Installation, Usage, Features
```

Compression/Decompression

```
mq compress input.md -o output.mq  
mq decompress input.mq -o output.md  
mq stats input.md
```

Future Extensions

1. **Semantic tokens:** TAPI for API docs, TEXAMPLE for examples
2. **Language-specific tokens:** Python, Rust, JS pattern tokens
3. **Cross-reference tokens:** Link related sections
4. **Delta compression:** Store only changes between versions
5. **Neural compression:** Learn project-specific patterns

The Cheet Says...

"Why send a whole markdown symphony when you can send the greatest hits?

Markqant is like MP3 for docs - same tune, fraction of the size!

Rock on with your compressed docs!" 🤘