

Deployment Workflow Guide

Deployment Workflow Guide

Last Updated: 2025-12-17

Current Version: `{{api:version|version.pmtnm_version}}` (Git: `{{api:version|version.git_commit_hash}}`)

Overview

This document describes the deployment workflow for the PMTNM platform, including how code changes are deployed to test and production environments, and how version information is tracked and displayed.

Deployment Methods

1. GitHub Actions (Automatic Deployment)

When to use: Normal workflow for deploying to `test.pmtnm.org` or `pmtnm.org`

How it works:

1. Make your code changes
2. Commit your changes
3. Push to GitHub (`test` or `main` branch)
4. GitHub Actions automatically:
 - Updates git commit files (hash, message, info)
 - Deploys changed files via FTP
 - `test.pmtnm.org` gets the latest version

Workflow:

```
# Make changes
git add .
git commit -m "Your commit message"
```

```
git push origin test # or 'main' for production
```

What happens:

- GitHub Actions detects the push
- Runs `scripts/create_version_cache.sh` to create version cache files server-side
- Deploys all changed files including the git commit files
- test.pmtnm.org shows the correct commit message automatically

2. Manual Upload (`upload_only_changes.sh`)

When to use: Quick fixes or when you need immediate deployment

How it works:

1. Make your code changes
2. Commit your changes
3. Run the upload script:

```
./scripts/upload_only_changes.sh
```

4. Script automatically:
 - Updates git commit files to match your new commit
 - Uploads changed files via FTP
 - test.pmtnm.org gets the latest version

Workflow:

```
# Make changes
git add .
git commit -m "Your commit message"
./scripts/upload_only_changes.sh
```

What happens:

- Script detects your new commit
- Updates all git commit files automatically
- Uploads everything to test.pmtnm.org

- No need to manually update git files

Version Tracking

Current Version Information

The platform tracks version information in multiple ways:

1. **Git Commit Hash** - Short hash (7 characters) of the current commit
2. **Commit Message** - The message from your git commit
3. **Site Version** - Based on file modification times (auto-generated)
4. **PMTNM Version** - Manual version number (updated for significant releases)

Viewing Version Information

API Endpoint:

- <http://localhost:2025/api/version.php> (localhost)
- <https://test.pmtnm.org/api/version.php> (test environment)

Response includes:

```
{
  "version": {
    "git_commit_hash": "01c9845",
    "site_version": "2025-12-17-cd504936",
    "pmtnm_version": "2025.12.17.01"
  },
  "git": {
    "commit_message": "committed5",
    "branch": "test",
    "commit_date": "2025-12-16 18:29:28 -0700"
  }
}
```

Dedicated Page:

- http://localhost:2025/software_version.php - Full version display with comparison

Version Sync Monitoring

Monitor Version Sync

To monitor when localhost and test.pmtnm.org are in sync:

```
php scripts/monitor_version_sync.php
```

What it does:

- Gets target version from localhost
- Monitors test.pmtnm.org every 5 seconds
- Reports when versions sync
- Shows time to sync

Example output:

```
=== Version Sync Monitor ===  
Target: 01c9845 - 'committed5'  
test.pmtnm.org: 01c9845 - 'committed4'  
  
=== SYNC ACHIEVED! ===  
Time to sync: 45 seconds
```

Git Commit Files

The platform uses three files to track commit information:

1. `data/.git_commit_hash.txt` - Short commit hash (7 characters)
2. `data/.git_commit_message.txt` - Commit message
3. `data/.git_commit_info.json` - Full commit info (hash, message, date, branch)

Automatic Updates

These files are automatically updated by:

- **GitHub Actions:** During deployment (before file upload)
- **upload_only_changes.sh:** Before manual upload

Manual Update

To manually update these files:

This updates all three files to match the current HEAD commit.

Deployment Environments

Test Environment (`test.pmtnm.org`)

- **Branch:** `test`
- **Deployment:** Automatic via GitHub Actions
- **Purpose:** Staging and testing before production
- **URL:** `https://test.pmtnm.org`

Production Environment (`pmtnm.org`)

- **Branch:** `main`
 - **Deployment:** Automatic via GitHub Actions
 - **Purpose:** Live production site
 - **URL:** `https://pmtnm.org`
-

Best Practices

1. Commit Messages

Use clear, descriptive commit messages:

- Good: "Add financial records click logging"
- Bad: "fix" or "update"

2. Testing Before Production

Always test on `test.pmtnm.org` before deploying to production:

1. Push to `test` branch
2. Test on `test.pmtnm.org`
3. Verify everything works
4. Merge to `main` for production

3. Version Monitoring

Use the monitoring script to verify deployments:

```
php scripts/monitor_version_sync.php
```

4. Git Commit Files

- Don't manually edit git commit files
- Let scripts handle updates automatically
- Files are included in deployment automatically

Troubleshooting

Version Not Syncing

Problem: test.pmtnm.org shows old commit message

Solution:

1. Check if git commit files were deployed:
 - Verify `data/.git_commit_info.json` exists on server (created by `create_version_cache.sh`)
 - Check file contents match your commit
2. For GitHub Actions:
 - Check GitHub Actions logs
 - Verify files were uploaded
3. For manual upload:
 - Run `./scripts/upload_only_changes.sh` again
 - Verify files are included in upload

API Not Showing Commit Message

Problem: `api/version.php` shows "N/A" for commit message

Solution:

- On localhost: Should use git command (has `.git` directory)
 - On test.pmtnm.org: Needs git commit files deployed
 - Check `data/.git_commit_info.json` exists and has correct data
-

Related Documentation

- [Version Comparison API](#)
 - [Git Hash Deployment](#)
 - [Site Version Caching](#)
-

Current Version

Note: The version information below is fetched live from the API using the `{{api:...}}` syntax:

Git Commit Hash

```
{{api:version.php|version.git_commit_hash}}
```

Commit Message

```
{{api:version.php|git.commit_message}}
```

Branch

```
{{api:version.php|git.branch}}
```

Site Version

```
{{api:version.php|version.site_version}}
```

PMTNM Version

```
{{api:version.php|version.pmtnm_version}}
```

Full Version JSON

```
<pre>{{api:version.php}}</pre>
```

Summary

- **GitHub Actions:** Automatic deployment when you push (recommended)
- **upload_only_changes.sh:** Manual upload for quick fixes
- **Version Tracking:** Automatic via git commit files
- **Monitoring:** Use `monitor_version_sync.php` to track deployments

Both deployment methods automatically handle git commit files, ensuring test.pmtnm.org always shows the correct version information.