Version Control & Backup Guide

Complete guide for managing versions and backups of the Purchase Requisition System.

Table of Contents

Quick Start

View Current Version

```
git describe --tags
```

Create a New Version

```
scripts\create-version.bat v3.0.1 "Bug fixes"
```

Backup Current Version

```
scripts\backup-version.bat v3.0.0
```

List All Versions

```
scripts\list-versions.bat
```

Version Control Setup

Initial Setup (Already Done)

The repository has been initialized with:

- ☑ Git repository initialized
- ☑ `.gitignore` configured
- ☑ Initial commit created (v3.0.0)
- ☑ Version tagging scripts ready

Verify Setup

```
git status
git log --oneline
git tag -l
```

Creating Versions

Using the Script (Recommended)

1. Create a Minor Version (bug fixes, small updates)

```
scripts\create-version.bat v3.0.1 "Fixed HOD approval bug"
```

2. Create a Feature Version

```
scripts\create-version.bat v3.1.0 "Added email notifications"
```

3. Create a Major Version

```
scripts\create-version.bat v4.0.0 "Complete UI redesign"
```

Manual Git Commands

Commit Changes

```
git add .
git commit -m "Description of changes"
```

Create Version Tag

```
git tag -a v3.0.1 -m "Version 3.0.1 - Bug fixes"
```

View Tags

```
git tag -l
git show v3.0.1
```

Backup System

Three Types of Backups

1. Version Backup (Tagged Release)

Backs up a specific git version:

```
scripts\backup-version.bat v3.0.0
```

What it includes:

- Source code from that version
- Current database files
- Current uploads directory
- Current .env configuration

2. Current State Backup

Backs up everything as-is right now:

```
scripts\backup-version.bat
```

What it includes:

- All current files (including uncommitted changes)
- Database files
- Uploads directory
- Environment configuration
- node_modules (full backup)

3. Automated Daily Backup

Set up scheduled backups (see Automated Backups)

Backup Storage

Backups are stored in:

```
C:\Projects\purchase-requisition-backups\
├── prs-v3.0.0-20250106_143022\      # Folder backup
├── prs-v3.0.0-20250106_143022.zip   # Compressed backup
├── prs-backup-20250106_150000\      # Current state backup
└── prs-backup-20250106_150000.zip   # Compressed
```

What Gets Backed Up

☑ Included:

- All source code
- Database files (*.db)
- Uploaded files
- Environment configuration (.env)
- Documentation
- Scripts

✕ Excluded:

- node_modules (too large, can be reinstalled)
- Log files
- Temporary files

Restoring Versions

Restore from Backup

1. List Available Backups

```
scripts\list-versions.bat
```

2. Restore a Backup

```
scripts\restore-version.bat prs-v3.0.0-20250106_143022
```

⚠ **Warning:** This will:

- Stop running servers
- Create a safety backup of current state
- Replace all files with the backup
- Reinstall dependencies
- Reinitialize database

3. After Restore

```
cd backend
npm start

# In another terminal
cd frontend
python -m http.server 3000
```

Restore from Git Tag

1. View Available Tags

```
git tag -l
```

2. Checkout a Specific Version

```
git checkout v3.0.0
```

3. Return to Latest

```
git checkout master
```

Automated Backups

Setup Windows Task Scheduler

1. Open Task Scheduler

- Press `Win + R`
- Type: `taskschd.msc`
- Press Enter

2. Create New Task

- Click "Create Basic Task"
- Name: "PRS Daily Backup"
- Description: "Automated backup of Purchase Requisition System"

3. Configure Trigger

- Select "Daily"
- Set time: 2:00 AM (when system is idle)
- Start date: Today

4. Configure Action

- Action: "Start a program"
- Program/script:

  ```
  C:\projects\purchase-requisition-system\scripts\automated-backup.bat
  ```

5. Additional Settings

- Run whether user is logged on or not
- Run with highest privileges
- Configure for: Windows 10/11

6. Finish and Test

- Right-click task → Run
- Check backup directory:

  ```
  C:\backups\purchase-requisition-system\
  ```

Automated Backup Features

- **Daily Backups**: Every day, kept for 30 days
- **Weekly Backups**: Every Sunday, kept for 90 days
- **Monthly Backups**: 1st of each month, kept permanently
- **Automatic Cleanup**: Old backups are deleted automatically
- **Detailed Logging**: All backup operations logged

View Backup Logs

```
type C:\backups\purchase-requisition-system\backup-log.txt
```

Best Practices

Version Naming Convention

Use semantic versioning: `vMAJOR.MINOR.PATCH`

- **MAJOR** (v4.0.0): Breaking changes, major redesigns
- **MINOR** (v3.1.0): New features, backwards compatible
- **PATCH** (v3.0.1): Bug fixes, small updates

Examples:

```
v3.0.0 - Initial release
v3.0.1 - Fixed login bug
v3.0.2 - Performance improvements
v3.1.0 - Added email notifications
v3.1.1 - Fixed email bug
v4.0.0 - New UI redesign
```

When to Create a Version

☑ Create a version when:

- Releasing to production
- Completing a major feature
- Before making significant changes
- After fixing critical bugs
- Before training users on new features

When to Backup

☑ Backup before:

- Making major changes
- Upgrading dependencies
- Database schema changes
- Deploying to production
- Training sessions (in case rollback needed)

☑ Regular backups:

- Daily automated backups (recommended)
- Weekly manual verification
- Monthly archive backups

Commit Message Best Practices

Good commit messages:

```
git commit -m "Fix: HOD approval not saving comments"
git commit -m "Feature: Add email notification system"
git commit -m "Update: Improve PDF generation performance"
git commit -m "Security: Fix SQL injection vulnerability"
```

Bad commit messages:

```
git commit -m "fixed stuff"
git commit -m "updates"
git commit -m "..."
```

Recovery Strategy

Level 1: Undo Recent Changes

```
git status              # Check what changed
git diff                # See specific changes
git checkout -- file    # Undo changes to a file
git reset HEAD~1        # Undo last commit
```

Level 2: Restore from Git

```
git log --oneline       # Find commit to restore
git checkout <commit>   # Go to that commit
git checkout -b recovery  # Create recovery branch
```

Level 3: Restore from Backup

```
scripts\list-versions.bat  # Find backup
scripts\restore-version.bat prs-v3.0.0-20250106_143022
```

🔍 Useful Commands Reference

Git Commands

```
# View version history
git log --oneline --graph --decorate --all

# View all tags
git tag -l

# View tag details
git show v3.0.0

# View changes
git diff
git diff v3.0.0 v3.0.1

# View file history
git log --follow -- backend/server.js

# Search commits
git log --grep="fix"

# View who changed what
git blame backend/server.js

# Create branch
git checkout -b feature-branch

# Merge branch
git checkout master
git merge feature-branch
```

Backup Commands

```
# List all versions
scripts\list-versions.bat

# Create new version
scripts\create-version.bat v3.1.0 "New features"

# Backup specific version
scripts\backup-version.bat v3.0.0

# Backup current state
scripts\backup-version.bat

# Restore backup
scripts\restore-version.bat prs-v3.0.0-20250106_143022
```

🆘 Emergency Recovery

If Something Goes Wrong

1. Don't Panic!

- Safety backups are created automatically during restores
- Git history preserves everything
- Multiple backup copies exist

2. Check Current State

```
git status
git log --oneline -10
```

3. Find Last Good State

```
scripts\list-versions.bat
```

4. Restore

```
# Option A: From Git
git checkout v3.0.0

# Option B: From Backup
scripts\restore-version.bat prs-v3.0.0-20250106_143022
```

5. Contact Support

- Check TROUBLESHOOTING.md
- Review backup-log.txt
- Check safety backups in backup directory

💾 Backup Verification

Regular Checks (Monthly)

1. Test Restore Process

```
# In a test directory
scripts\restore-version.bat prs-backup-latest
cd backend
npm install
node scripts/hashPasswords.js
npm start
```

2. Verify Backup Contents

- Check database files exist
- Verify uploads directory
- Confirm .env file present
- Test application functionality

3. Check Backup Sizes

```
# Should be 10-50 MB typically
dir C:\backups\purchase-requisition-system\*.zip
```

4. Review Backup Logs

```
type C:\backups\purchase-requisition-system\backup-log.txt
```

Quick Reference Card

Task	Command
Create version	`scripts\create-version.bat v3.0.1 "Description"`
Backup version	`scripts\backup-version.bat v3.0.0`
Backup current	`scripts\backup-version.bat`
List versions	`scripts\list-versions.bat`
Restore backup	`scripts\restore-version.bat backup-name`
View git log	`git log --oneline`
View all tags	`git tag -l`
Check status	`git status`

Need Help?

- 📖 See: TROUBLESHOOTING.md
- 📖 See: README.md
- 📧 Check backup logs for errors
- 🔍 Review Git documentation: https://git-scm.com/doc

Document Version: 1.0
Last Updated: 2025-01-06
System Version: v3.0.0