

Phase 2B Rollback Template

Overview

This template provides exact rollback commands for Phase 2B consolidation operations. Each consolidation day generates a specific rollback stub using this template.

Template Variables

- `{DAY_NUMBER}` : Day number (1, 2, or 3)
- `{CONSOLIDATION_BRANCH}` : Target consolidation branch name
- `{SOURCE_BRANCHES}` : Space-separated list of source branches
- `{PRE_MERGE_COMMIT}` : Commit hash before consolidation
- `{TIMESTAMP}` : Execution timestamp
- `{BASELINE_ID}` : Baseline capture identifier

Rollback Script Template

```
#!/bin/bash
# Phase 2B Day {DAY_NUMBER} Rollback Script
# Generated: {TIMESTAMP}
# Baseline ID: {BASELINE_ID}

set -euo pipefail

echo "=== Phase 2B Day {DAY_NUMBER} Rollback Initiated ==="
echo "Target: {CONSOLIDATION_BRANCH}"
echo "Pre-merge commit: {PRE_MERGE_COMMIT}"
echo "Timestamp: {TIMESTAMP}"

# Safety confirmation
read -p "⚠️ CONFIRM ROLLBACK for Day {DAY_NUMBER} consolidation? (yes/NO): " confirm
if [[ "$confirm" != "yes" ]]; then
    echo "❌ Rollback cancelled by user"
    exit 1
fi

# Step 1: Checkout target branch
echo "🔄 Checking out {CONSOLIDATION_BRANCH}..."
git checkout {CONSOLIDATION_BRANCH}

# Step 2: Reset to pre-merge state
echo "⏮️ Resetting to pre-merge commit: {PRE_MERGE_COMMIT}"
git reset --hard {PRE_MERGE_COMMIT}

# Step 3: Force push to remote (with safety check)
echo "🚀 Force pushing reset to remote..."
read -p "⚠️ CONFIRM FORCE PUSH to origin/{CONSOLIDATION_BRANCH}? (yes/NO): " force_confirm
if [[ "$force_confirm" == "yes" ]]; then
    git push --force-with-lease origin {CONSOLIDATION_BRANCH}
    echo "✅ Remote branch reset successfully"
else
    echo "⚠️ Remote push skipped - local reset completed only"
fi

# Step 4: Restore source branches (if archived)
echo "🔄 Checking source branch availability..."
SOURCE_BRANCHES=({SOURCE_BRANCHES})
for branch in "${SOURCE_BRANCHES[@]"; do
    if git show-ref --verify --quiet refs/heads/$branch; then
        echo "✅ Source branch '$branch' still available locally"
    elif git show-ref --verify --quiet refs/remotes/origin/$branch; then
        echo "🔄 Restoring source branch '$branch' from remote..."
        git checkout -b $branch origin/$branch
    else
        echo "⚠️ Source branch '$branch' not found - may need manual restoration"
    fi
done

# Step 5: Verification
echo "🔍 Verifying rollback state..."
echo "Current branch: $(git branch --show-current)"
echo "Current commit: $(git rev-parse HEAD)"
echo "Expected commit: {PRE_MERGE_COMMIT}"

if [[ "$(git rev-parse HEAD)" == "{PRE_MERGE_COMMIT}" ]]; then
    echo "✅ Rollback verification PASSED"
else
    echo "❌ Rollback verification FAILED"
fi
```

```

    exit 1
fi

echo "=== Phase 2B Day {DAY_NUMBER} Rollback Completed Successfully ==="
echo "📋 Next steps:"
echo "    1. Verify all source branches are available"
echo "    2. Check baseline drift report: docs/phase-2B/day{DAY_NUMBER}_drift.txt"
echo "    3. Review consolidation logs if re-attempting"

```

Manual Rollback Commands

If the automated script fails, use these manual commands:

Emergency Reset Commands

```

# 1. Checkout target branch
git checkout {CONSOLIDATION_BRANCH}

# 2. Hard reset to pre-merge state
git reset --hard {PRE_MERGE_COMMIT}

# 3. Force push (DANGEROUS - use with caution)
git push --force-with-lease origin {CONSOLIDATION_BRANCH}

```

Source Branch Recovery

```

# Check if source branches exist
git branch -a | grep -E "({SOURCE_BRANCHES})"

# Restore from remote if needed
git checkout -b <branch_name> origin/<branch_name>

# Or restore from archive (if using archive workflow)
# See .github/workflows/cleanup-archive-branches.yml

```

Safety Checklist

Before executing rollback:

- [] Confirm correct DAY_NUMBER and CONSOLIDATION_BRANCH
- [] Verify PRE_MERGE_COMMIT hash is correct
- [] Ensure no critical work exists on target branch post-merge
- [] Backup current state if needed
- [] Notify team of rollback operation
- [] Check that source branches are recoverable


Post-Rollback Actions

After successful rollback:

1. Update project status documentation
2. Analyze failure cause from drift reports
3. Plan remediation if re-attempting consolidation
4. Update baseline captures if needed
5. Communicate rollback completion to stakeholders

Emergency Contacts

- Engineering Team: [Contact Info]
 - Repository Maintainers: [Contact Info]
 - Escalation Path: [Contact Info]
-

 **WARNING:** Force push operations are destructive. Always verify commit hashes and coordinate with team before execution.