DevOps and Version Control Reference

Track changes, version solutions, and collaborate with Git.

Reference **Solution** Dev Ops | Concept | Reference

DevOps & Version Control (Reference): Modern development practices for industrial automation solutions, combining audit trails, change tracking, and Git integration while maintaining centralized database reliability.

Overview
Architecture
Core Components
Workflow Integration
Security & Compliance
Configuration
Best Practices
Limitations
Troubleshooting
Integration Points

Overview

FrameworX provides comprehensive DevOps capabilities through two integrated systems:

System	Purpose	Access
Track Changes Tools	Internal audit and impact analysis	Designer UI
Git Integration	External version control and collaboration	Automatic export

This hybrid approach maintains industrial-grade reliability while enabling modern CI/CD workflows, code reviews, and multi-environment deployments.

Architecture

Change Management Stack

Git Repository	External VCS
JSON Export Layer	Git Integration
Track Changes Database	Internal Tracking
Solution Configuration DB	Source of Truth

Data Flow

Operation	Internal DB	Track Changes	Git Export
Create Object	Stored	Logged	Exported
Modify Object	Updated	Tracked	Diff created
Delete Object	Removed	Recorded	File removed
Build Solution	Versioned	Milestone	Tagged

Core Components

Track Changes Tools

Internal change tracking and analysis:

Tool	Function	Use Case
Recent Changes	Change log with persistence	Audit trail
Version Control	Table version tracking	Online sync

Cross-Reference	Object usage mapping	Impact analysis
Use Count	Usage frequency	Optimization
Unused Objects	Cleanup candidates	Maintenance

Git Integration

External version control and collaboration:

Feature	Function	Use Case
Auto Export	Continuous sync to Git	Version tracking
JSON Format	Diff-friendly structure	Code review
Selective Export	Module-based control	Performance
Environment Variables	Configuration management	Multi-environment
CI/CD Support	Automation hooks	DevOps pipeline

Workflow Integration

Development Lifecycle

Phase	Track Changes	Git Integration
Development	Recent Changes tracking	Feature branches
Review	Cross-Reference analysis	Pull requests
Testing	Version Control sync	CI validation
Deployment	Build history	Release tags
Maintenance	Unused Objects cleanup	Hotfix branches

Team Collaboration

Designer-Based Workflow:

1. Multiple users edit solution
2. Track Changes logs modifications
3. Version Control maintains consistency
4. Runtime diagnostics verify sync

Git-Based Workflow:

1. Changes exported to JSON
2. Developers review diffs
3. Pull requests for approval
4. Merge to main branch
5. Deploy from Git tags

Security & Compliance

Audit Features

Requirement	Track Changes	Git Export
Who	User tracking	Commit author
What	Object changes	File diffs
When	Timestamps	Commit history
Why	Change notes	Commit messages

Compliance	21 CFR Part 11	SOC 2

For runtime-side audit posture — MCP Custom Tool execution audit, MCP scoping, the three-sink model, and 21 CFR Part 11 / IEC 62443 SR 6.1 / NERC CIP-007 R4 / ISO 27001 A.8.15 hooks — see the FrameworX Enterprise IT Readiness Reference.

For IT-team-facing audit and compliance posture (install lifecycle, SIEM integration patterns, regulatory framework hooks for FDA 21 CFR Part 11 / IEC 62443 / NERC-CIP), see the parallel FrameworX Enterprise IT Readiness Reference. The DevOps reference covers engineering author-time concerns; the IT Readiness reference covers operations-time concerns; the two cross-link at the audit boundary.

Data Protection

Data Type	Track Changes	Git Export
Configuration	Full tracking	Exported
Passwords	Tracked changes	Never exported
Runtime Values	Not tracked	Not exported
Binary Assets	Reference only	Checksum verified

Configuration

Track Changes Settings

Location: Track Changes Recent Changes

Setting	Options	Default
Auto-Delete	Various triggers	Previous Sessions
Tracking	Enabled/Disabled	Enabled
Persistence	Days to keep	30

Git Export Settings

Location: Solution Settings Version Control

Setting	Options	Default
Enable Export	On/Off	Off
Auto Export	On/Off	Off
Export Path	Directory	Solution-Data\Git
Export Delay	Seconds	5

Best Practices

Change Management

Track Changes:

- Configure appropriate persistence
- Review Recent Changes daily
- Monitor Version Control before deployment
- Use Cross-Reference before deletions
- Clean Unused Objects regularly

Git Integration:

- Enable auto-export for tracking
- Use meaningful commit messages
- Implement branching strategy
- Review changes through PRs
- Tag releases consistently

Performance Optimization

System	Optimization	Impact
Track Changes	Set auto-delete policy	Database size
Cross-Reference	Regular refresh	Query speed
Git Export	Selective modules	Export time
Version Control	Batch changes	Network traffic

Limitations

Track Changes Limitations

Not Detected:

- Indirect string references
- External API access
- Dynamic object names
- Runtime-only changes

Git Export Limitations

Not Exported:

- Runtime tag values
- Historical data
- User passwords
- Temporary objects
- Binary executables

Troubleshooting

Problem	Possible Cause	Solution
Changes not tracked	Tracking disabled	Enable in settings
Git export missing	Permissions issue	Check directory access
Version mismatch	Unsaved changes	Save and build
Cross-Reference incomplete	Indirect references	Manual verification
Large export files	Too many objects	Selective export

Integration Points

Internal Integration

Module	Integration Point	Purpose
Runtime	Version sync	Online configuration
Designer	Find Results panel	Navigation
Security	User tracking	Audit trail
Solution	Build/Publish	Milestones

External Integration

System	Method	Purpose

GitHub/GitLab	JSON export	Version control
Jenkins/Azure DevOps	Webhooks	CI/CD pipeline
Jira/Azure Boards	Commit links	Issue tracking
Teams/Slack	Notifications	Change alerts

In this section...
