

CHANGE MANAGEMENT FOR MIRTH CONNECT

MirthCommander

Build on TEST. Prove it. Ship it to PROD. MirthCommander shows what differs between your TEST and PROD Mirth Connect servers, promotes a channel only after every blocking check passes, proves the new version is running, and rolls back on its own if a step fails.

Up to 13 checks before anything moves

Verified in PROD, not assumed

Automatic and one-click rollback

Database changes included

Claude builds on TEST only

WHY INTEGRATION TEAMS NEED IT

Moving a channel shouldn't be a gamble.

Most teams still release by exporting a channel from TEST and importing it into PROD by hand. TEST and PROD drift apart one hotfix at a time, dependencies go live in the wrong order, and the way back is an old export file. These are the questions nobody can answer quickly today:

"What is actually different between TEST and PROD right now?"

"If this deploy fails tonight, how fast are we back, and how do we know?"

"Does PROD have every setting and database column this channel needs?"

"Who changed PROD last week, and what was there before?"

See the Drift

Every channel and shared code library on **TEST** and **PROD**, compared.

Differences in plain English first, then line by line, with recognised passwords hidden. Looking changes nothing.

Promote with Checks

Up to 13 checks on both live servers: running cleanly in TEST; settings, shared code, channels and database objects it references present in PROD; ports and versions. One failure blocks it, with the fix named.

Proof, Not Hope

PROD is backed up, the channel written and read back, deployed, and proven to be the version running. If a step after the write fails, PROD is restored from the backup and verified automatically, or flagged for a person.

Claude, Safely on TEST

Ask in plain English to review, build or change a channel. Claude prepares a proposal; a person reviews and applies it. It is given no access to PROD.

HOW IT WORKS

Every promotion checked, proven and reversible.

MirthCommander runs on a workstation or server inside your network and talks to Mirth through its standard REST API. Nothing is installed on your Mirth servers.

1 See what differs

TEST and PROD side by side, in the same channel-group tree your team uses in the Mirth Administrator.

3 Promote with proof

Back up PROD, write, read back, deploy, prove the new revision is running, smoke-check. Each step is recorded.

2 Review the checks

An administrator sees every check, the change in plain English and line by line, and approves.

4 Roll back in one click

Automatically after a failed step, or later from the history, where the rollback takes a fresh backup first and proves the result.

THE CHECKS

Nothing moves until PROD is ready for it.

The checks run against both live servers when you review, and again when the promotion starts; PROD is re-read just before the write. One blocking failure stops the promotion and names the fix.

BEFORE A CHANNEL MOVES	WHAT MIRTHCOMMANDER CONFIRMS ON THE LIVE SERVERS
It works in TEST	The saved version is the one running in TEST, with no errors
PROD is what it says	PROD is running what it has saved, so the backup is faithful
Its dependencies exist	The configuration settings, shared code, downstream channels, and database tables and columns it references are present in PROD
It will deploy cleanly	No other PROD channel uses its port, its name is unique, and both servers run the same Mirth version
Nothing is unresolved	No earlier promotion of the channel is waiting for attention

BEYOND CHANNELS

Database changes and AI, held to the same standard.

Database changes (MySQL)

Versioned scripts, each with an undo script.

- ✓ Compares scripts, tables, columns and lookup data
- ✓ Blocks a channel needing a column PROD lacks
- ✓ Transactional rows never copied to backups
- ✓ Verified against TEST; automatic undo

Claude assistant, TEST only

Review, build or change a channel from a plain request.

- ✓ Given access to TEST only, never PROD
- ✓ Only proposes; a person reviews and applies
- ✓ Flags new JavaScript Writers and web calls
- ✓ Backed up and verified on Apply; one-click undo

Safeguards and record

- ✓ Administrator password for PROD changes
- ✓ Passwords and the AI key encrypted by Windows
- ✓ Each Mirth server's certificate pinned
- ✓ Recognised credentials in channels masked
- ✓ Every release kept with its encrypted backup
- ✓ Hash-chained, tamper-evident activity log

REQUIREMENT	DETAIL
Mirth Connect	4.5.2 on TEST and PROD
Runs on	Windows 10/11 or Server; Node.js
Databases	Optional: MySQL
AI assistant	Optional: your Anthropic key
Coming soon	Two-AI second opinion, as advice

Your production, under your control.

Runs on your premises and connects directly to your servers. The optional assistant sends masked TEST channel content to Anthropic with your own key. AI never decides what reaches production: the checks and a person do.

- No MirthCommander cloud, no account with us
- Standard Mirth REST API; nothing installed on Mirth
- Tested against Mirth Connect 4.5.2 servers