Version History Plugin for BridgeLink

User Guide | Version 3.0.0

1. Introduction

What Is This Plugin?

The **Version History Plugin** brings Git version control directly into BridgeLink, allowing your team to automatically track, commit, and push changes to your integration channels, code templates and global scripts— without leaving the BridgeLink administrator tool

Why Does This Matter?

As your team grows and more clients come on board, tracking development changes across your BridgeLink engine becomes increasingly difficult. Without version control, it is easy to lose track of who changed what — and hard to recover when something goes wrong.

The Version History Plugin solves this by automatically committing changes and pushing them to your remote repository. You get a full history of every change, viewable directly inside the BridgeLink administrator tool.

> **In short:** You will never lose track of your code — and you can restore a previous version quickly whenever you need to.

Key Features

The table below shows which packages come with the plugin:

Feature	Description
History & Rollback	View the full history of changes to any channel, code template or global script. If a new change introduces a bug, roll back to a stable version in seconds
Remote Repository Backup	Channels and code templates are automatically pushed to GitHub, providing a secure off-site backup. If local hardware fails, your configurations are safe

Who Should Use This Plugin?

This plugin is intended for BridgeLink administrators who need to track and review changes made to integration channels, code templates and global scripts over time. No advanced technical knowledge is required — this guide will walk you through each step.

2. Getting Started

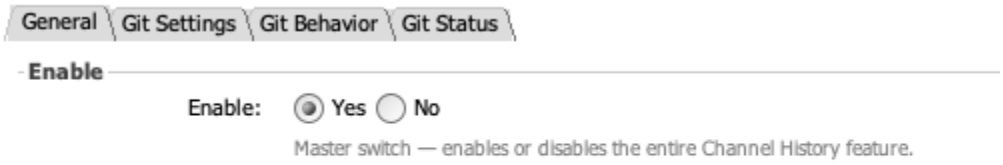
Before proceeding with this guide, verify compatibility with your existing BridgeLink setup. The sections that follow will walk you through configuration, usage, and best practices for the Version History Plugin.

3. Availability

The Version History Plugin is built into the core of BridgeLink and comes pre-installed across all editions. No separate installation is required.

4. Configuration

4.1 General Tab



Field	Description
Enable	Master switch — enables or disables the entire Channel History feature. Select **Yes** to activate or **No** to deactivate.

> **Note:** You must set Enable to **Yes** before any other settings in this plugin will take effect.

4.2 Git Settings Tab

Field	Description
Repository URL	**URL**The SSH URL of your Git repository where channel history will be stored.
Branch Name	The branch to which channel history commits will be pushed (e.g., main).
SSH Private Key	Choose **Paste key** to enter the key directly, or **File path** to specify the key file location on the server (e.g., appdata/myKey.pem).

> **Note:** The SSH private key must be stored on the server. Enter the file path to its location on the server (e.g., appdata/myKey.pem). Only the file path is saved in BridgeLink settings.

Validate Connection — Click to confirm BridgeLink can successfully reach your Git repository with the provided credentials

4.3 Git Behavior Tab

Auto Commit

Field	Description
Enable	Automatically commit channel changes to Git. Select **Yes** to enable or **No** to disable.
Prompt	When set to **Yes**, users will be prompted to enter a commit message before each commit. When set to **No**, the default message will be used automatically.
Default Message	The commit message used when Prompt is set to **No** (e.g., This is default commit from Innovar).

Sync Delete

Field	Description
Sync Delete	When set to **Yes**, deleting a channel or code template will automatically remove it from Git and push the change to remote. Defaults to **No**.

History

Field	Description
Max Commits	The maximum number of commits loaded in the History tab (e.g., 1000).

4.4 Git Settings Tab

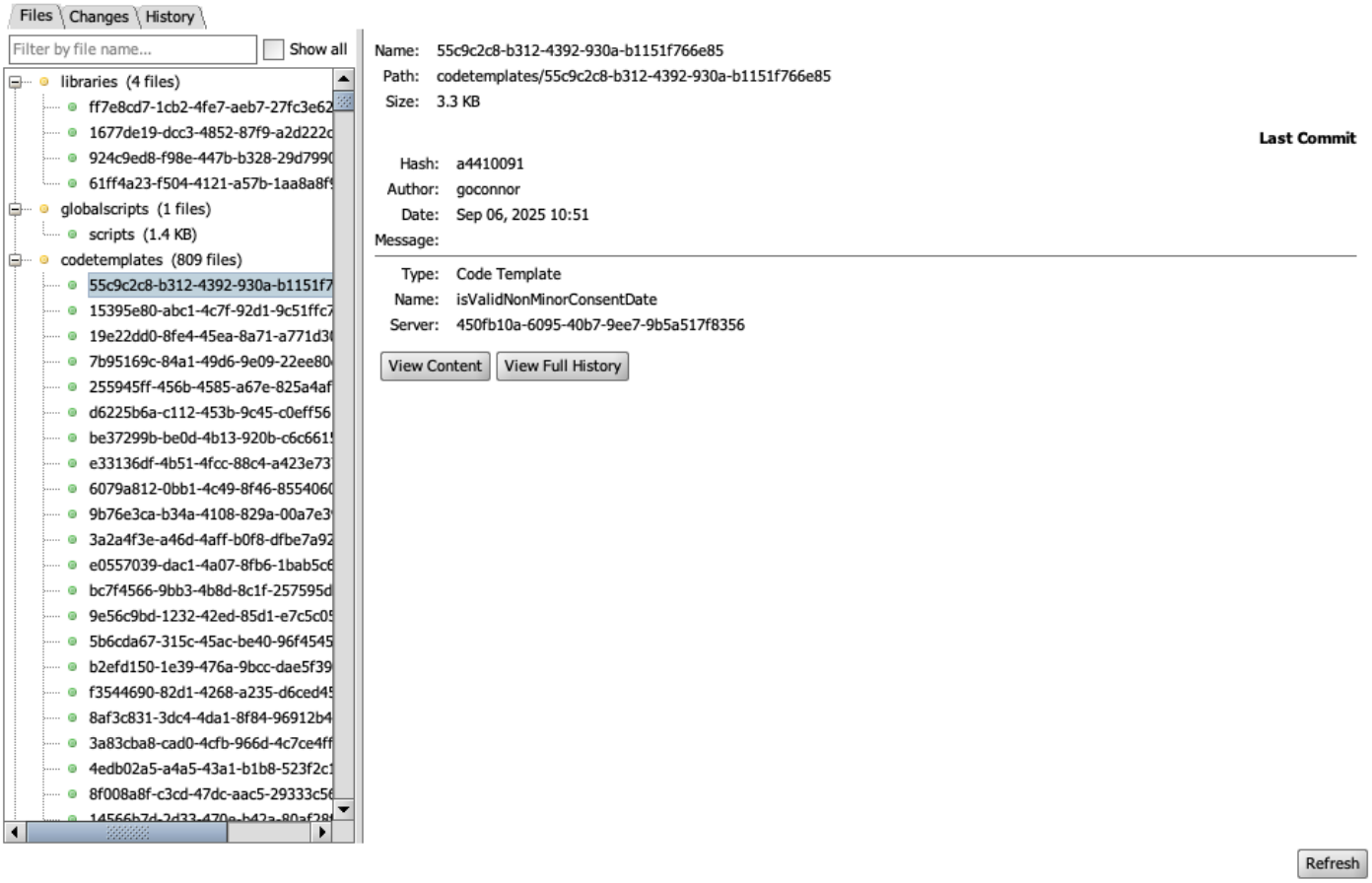
This tab provides an overview of the current state of your Git repository as seen by BridgeLink, including tracked files, pending changes, and full commit history

Repository Summary Bar

At the top of the tab, the following information is displayed:

Field	Description
Local Path	The local directory where BridgeLink stores the Git repository on the server.
Remote	The remote Git repository URL the local copy is synced with.
Branch	The currently active branch (e.g., main).
Size	The total size of the local repository on disk

4.4.1 Files Tab



Lists all tracked files organized by category (e.g., libraries, globalscripts, codetemplates). Each category shows the number of files it contains.

- Use the **Filter by file name** field to search for a specific file.
- Check **Show all** to display all files regardless of category.

When a file is selected, the right panel displays its details:

Field	Description
Name	**URL**The SSH URL of your Git repository where channel history will be stored.
Path	The relative path of the file within the repository.
Size	The file size on disk.

Last Commit

Field	Description
Hash	The short commit hash of the last commit for this file.

Author	The user who made the last commit
Date	The date and time of the last commit
Message	The commit message associated with the last commit.
Type	The BridgeLink resource type (e.g., Library).
Name	The display name(s) of the resource in BridgeLink.
Server	The BridgeLink server instance the resource belongs to.

Button	Description
View Content	Opens the full content of the selected file as stored in Git.
View Full History	Opens the complete commit history for the selected file

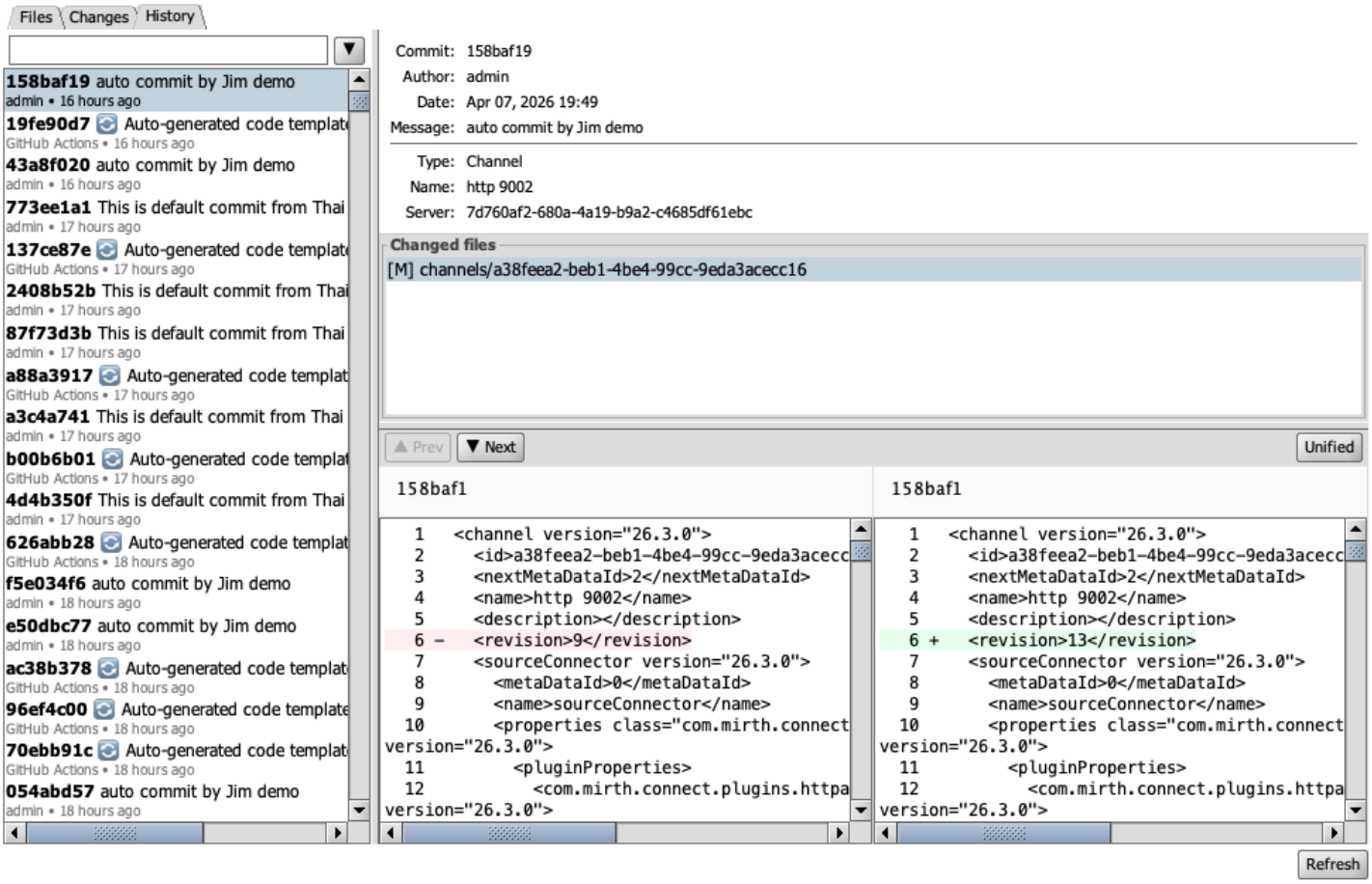
4.4.2 Changes Tab

This tab shows any local modifications that have not yet been committed to Git.

- The **left panel** lists all files that have been modified locally. If there are no pending changes, it will display *"No changes in working tree"*.
- The **right panel** displays a diff of the selected file's local modifications. Select a file from the left panel to view the changes.

Button	Description
Select All	Selects all modified files in the list.
Deselect All	Deselects all currently selected files.
Commit & Push	Commits the selected files and pushes the changes to the remote Git repository.

4.4.3 History Tab



This tab displays the full commit history of the Git repository, allowing administrators to browse, review, and compare past changes.

Left Panel – Commit List

Lists all commits up to the **Max Commits** limit configured in the Git Behavior tab. Each entry shows the commit hash, message, author, and time.

- Use the **Search by message** field to filter commits by their commit message.

Right Panel – Commit Detail

When a commit is selected, the right panel displays its full details:

Field	Description
Commit	The short commit hash.
Author	The user who made the commit.

Date	The date and time of the commit.
Message	The commit message.
Type	The BridgeLink resource type (e.g., Channel).
Name	The display name of the resource in BridgeLink.
Server	The BridgeLink server instance the resource belongs to.

Changed Files

Lists all files modified in the selected commit. Each entry is prefixed with a status indicator (e.g., [M] for modified).

Diff Viewer

Displays a side-by-side or unified diff of the changes made in the selected commit.

Field	Description
Prev / Next	Navigate between changed files in the commit.
Unified	Toggle between side-by-side and unified diff view.
Refresh	Reload the commit history from the repository.

5. Usage

5.1 Channel

5.1.1 Import Channel from Repository

The **Import** feature enables administrators to bring channels directly from the connected Git repository into BridgeLink.

Steps:

1. In the left sidebar, find the **Version History** section.
2. Click **Import**.
3. The **Import Channel From Repo** dialog appears, listing all available channels from the repository with their **Channel Id** and **Channel Name**.
4. Use the **Search** field to filter channels by name or ID.
5. Select the desired channel from the list.
6. Click **Import** to add the channel to BridgeLink, or **Cancel** to close the dialog without changes.

5.1.2 Channel History

The **Innovar** Channel History tab within each channel's edit view provides a complete audit trail of all commits made to that channel.

To access Channel History:

1. Open the desired channel for editing.
2. Click the **Innovar** Channel History tab, located with Summary, Source, Destinations, and Scripts tabs.

The History table displays the following for each commit:

Column	Description
Commit Id	The unique identifier of the commit.
Message	The commit message for the change.
Committer	The user who made the commit.
Date	The relative time of the commit (e.g., 18 hours ago).
Server Id	The unique ID of the BridgeLink server instance.
Server Name	The display name of the BridgeLink server instance.

5.1.3 Diff

Clicking **Diff** opens the **Channel Diff** window, letting administrators compare a previously committed channel version against the current one.

The diff viewer includes two tabs:

XML Diff Tab

Shows a side-by-side comparison of the raw XML between the selected commit (left) and the current (right). Red highlights mark removed content, green marks added content.

Control	Description
Prev / Next	Navigate differences in the file.
Unified	Switch between side-by-side and unified diff view.

Channel Tab

Offers a readable, component-level comparison. The left panel lists all channel components with a status indicator (e.g., [M] for modified):

- **Channel Info**
- **Source** (e.g., HTTP Listener)
- **Destinations** (e.g., Destination 1)

Choosing a component from the left panel shows a simplified diff of its properties on the right for easier identification of changes without reading XML.
Both views display header information:

Field	Description
Left header	The commit hash and author of the previous version, plus its date.
Right header	Marked as CURRENT, shows the author and timestamp of the current unsaved/uncommitted state.

5.1.4 Commit & Push

Click **Commit & Push** to open the **Commit Message** dialog, allowing administrators to manually commit the channel's current state and push it to the remote Git repository.
Steps:

1. In the **Commit Message** field, enter a descriptive commit message.
2. Click **Commit & Push** to save and push the changes, or **Cancel** to exit without committing.

5.1.5 Pull

Click **Pull** to fetch and load the latest channel version from the remote Git repository, refreshing the local copy in BridgeLink.

5.1.6 Revert

Click **Revert** to roll back the channel to a previous committed version.
Steps:

1. In the **Innovar** Channel History tab, select the commit to revert to.
2. Click **Revert** in the Version History sidebar.
3. A confirmation dialog asks *"Would you like to revert channel to this revision?"*
4. Click **Yes** to proceed. The selected version loads in the channel editor.
5. Click **Save** to finalize the revert, or **Cancel** to discard the changes.

Note: The revert is not completed until you save the channel after loading the selected version in the editor.

5.2 Code Template

5.2.1 History

To view the commit history for a code template:

1. In the **Code Templates** view, select the template.
2. Click **History** from the Version History sidebar.
3. The **Code Template History** window displays the template's commit history.

The History table shows for each commit:

Field	Description
Commit Id	The unique identifier of the commit.
Message	The commit message for the change.
Committer	The user who made the commit.
Date	The relative time of the commit (e.g., 18 hours ago).
Server Id	The unique BridgeLink server ID.
Server Name	The BridgeLink server's display name.

Note: From the **Version History** sidebar, actions such as **Diff**, **Commit & Push**, **Pull**, and **Revert** are available and operate as described in Section 5.1.3 – 5.1.6.

5.2.2 Import

Click **Import** to open the **Import Code Template From Repo** dialog, enabling administrators to import code templates from the connected Git repository.

Steps:

1. Click **Import** from the Version History sidebar.
2. The **Import Code Template From Repo** dialog appears, listing all libraries and their code templates in a tree structure.
3. Use the **Search** field to filter by name or ID.
4. Select templates individually, or use **Select All** / **Deselect All** to manage selections.
5. The dialog shows the number of currently selected items (e.g., *0 selected*).
6. Click **Import** to add templates to BridgeLink, or **Cancel** to close without changes.

Note: Libraries with zero templates (e.g., *HIE - TEST (0)*) are visible but cannot be selected.

5.2.3 Save Libraries

Clicking **Save Libraries** opens the **Commit Message** dialog, allowing administrators to commit and push the current state of all code template libraries to the remote Git repository.

Steps:

1. In the **Commit Message** field, enter a descriptive message.
2. Click **Commit & Push** to save and push changes, or **Cancel** to close without committing.

5.3 Global Scripts

The Version History plugin also works with the **Global Scripts** view, offering two actions in the **Version History** section:

Action	Description
Commit & Push	Commit and push the current state of all global scripts to the remote repository.
History	View the full commit history for global scripts.

5.3.1 History

Click **History** to open the **Global Scripts History** window, which shows the full commit history for global scripts. The History table displays for each commit:

Column	Description
Commit Id	The unique commit identifier.
Message	The commit message.
Committer	The user who made the commit.
Date	The date and time of the commit.
Server Id	The BridgeLink server's unique ID.
Server Name	The display name of the BridgeLink server instance.

From the **Version History** sidebar, **Diff** and **Revert** actions are available, functioning as described in Section 5.1.3 and Section 5.1.6.

5.3.2 Commit & Push

This works the same as in Section 5.1.4: opens the **Commit Message** dialog to manually commit and push the current state of all global scripts to the remote Git repository.